

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**PROTÓTIPO PARA ANÁLISE DA MATURAÇÃO DE  
MELANCIAS**

**WESLEY RAMIREZ VOLKMANN**

**BLUMENAU**  
**2017**

**WESLEY RAMIREZ VOLKMANN**

**PROTÓTIPO PARA A ANÁLISE DA MATURAÇÃO DE  
MELANCIAS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Aurélio Faustino Hoppe, Mestre – Orientador

Profa. Andreza Sartori, Doutora - Coorientadora

**BLUMENAU  
2017**

# **PROTÓTIPO PARA A ANÁLISE DA MATURAÇÃO DE MELANCIAS**

Por

**WESLEY RAMIREZ VOLKMANN**

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Roberto Heinzle, Doutor – FURB

Blumenau, 12 de dezembro de 2017

Dedico este trabalho a minha família, que sempre me apoiou a conclusão do meu curso de graduação.

## **AGRADECIMENTOS**

A minha família pelo apoio incondicional e motivação na realização e conclusão deste trabalho.

Aos meus amigos e professores que sempre me apoiaram durante todo o período de graduação.

A minha orientadora, Andreza Sartori, pela ajuda na escolha do tema e auxílio durante as etapas iniciais de desenvolvimento do trabalho.

Ao meu orientador, Aurélio Faustino Hoppe, pela ajuda e apoio imprescindíveis neste período de conclusão.

“Muitas das falhas da vida acontecem quando as pessoas não percebem o quão perto estão quando desistem.”

Thomas Edison

## RESUMO

Este trabalho apresenta o desenvolvimento de um protótipo para análise do nível de maturação de melancias a partir de uma imagem informada pelo usuário. A análise do nível de maturação é feita pela cor da casca e pela mancha de encosto, utilizando uma árvore de decisão em conjunto com o algoritmo K-Nearest Neighbor (KNN) para realizar a classificação. A extração de características ocorre por meio de técnicas de processamento digital de imagens pelas quais a imagem é redimensionada, convertida para uma escala de cinzas, tendo sua forma obtida através da extração do contorno dos objetos, sendo segmentada pelo algoritmo K-Means e filtros morfológicos. O protótipo foi desenvolvido na linguagem C++ utilizando a IDE Visual Studio 2017 e a biblioteca OpenCV. A assertividade do protótipo foi de 34,69%, sendo um resultado abaixo do esperado. Quando analisados individualmente, o método de classificação pela mancha de encosto se mostrou mais eficiente, atingindo uma taxa 42,85% de acerto. Enquanto, a classificação por KNN teve uma taxa de 28,57%.

Palavras-chave: Melancia. Análise de maturação. K-nearest neighbor. Árvore de decisão.

## **ABSTRACT**

This work presents a development prototype to analyze the ripeness level of watermelons using an user-given image. The analysis of its ripeness level is done through the color of the rind and through the yellow spot, using a decision tree along with the K- Nearest Neighbor (KNN) algorithm to make the classification. The extraction of the characteristics occurs through an image digital processing method through which the image gets resized, converted to a grey scale, have its form acquired through extracting the contours of the object, being segmented by the Kmeans algorithm and morphological filters. The prototype was developed in the C ++ language using the Visual Studio 2017 IDE and the OpenCV library. The general assertiveness of the prototype reached 34,69%, being a below expected result. When analyzed individually, the classification method through the yellow spot shows itself more effective reaching an assertiveness rate of 42,85% when the classification by the KNN method kept itself at an assertiveness rate of only 28,57%.

**Key-words:** Watermelon. Ripeness analysis. K-nearest neighbor. Decision tree.

## LISTA DE FIGURAS

Figura 1 - Produção de Melancia por Estado .....	18
Figura 2 - Comparativo entre mancha de encosto .....	19
Figura 3 - Gavinha ressecada.....	19
Figura 4 - Etapas realizadas pelo algoritmo K-Means .....	20
Figura 5 - Análise realizada pelo KNN com diferentes valores de K .....	21
Figura 6 - Árvore de decisão .....	22
Figura 7 - Estrutura para aquisição de imagens.....	25
Figura 8 - Processamento de imagem realizado sob a imagem da melancia.....	25
Figura 9 - Método do disco.....	26
Figura 10 - Padrões de cores para classificação de laranjas.....	26
Figura 11 - Etapas da abordagem proposta. ....	27
Figura 12 - Mangas do padrão Embrapa. ....	28
Figura 13 - Histograma de representação das cores da manga.....	29
Figura 14 - Diagrama de casos de uso .....	31
Figura 15 - Diagrama de classes.....	32
Figura 16 - Diagrama de atividades do processo de análise da melancia.....	33
Figura 17 - Comparação entre a imagem original com a imagem redimensionada .....	36
Figura 18 - Transformação para escala de cinza .....	36
Figura 19 - Detecção de contornos .....	38
Figura 20 - Preenchimento de contornos .....	38
Figura 21 - Repetição do processo de detecção e preenchimento de contornos.....	39
Figura 22 - Aplicação do método <code>copyTo</code> .....	40
Figura 23 - Segmentação realizada pelo K-Means.....	41
Figura 24 - Imagem erodida .....	42
Figura 25 - Imagem dilatada.....	42
Figura 26 - Resultado do método <code>extrairManchaAmarela</code> .....	44
Figura 27 - Regiões a serem analisadas individualmente.....	45
Figura 28 - Análise individual das regiões que formam a mancha de encosto .....	45
Figura 29 - Resultado do método <code>analisarCorMancha</code> .....	46
Figura 30 - Representação gráfica da base de conhecimento .....	49
Figura 31 - Exemplo de melancia sem mancha de encosto.....	51

Figura 32 - Tela inicial do protótipo.....	51
Figura 33 - Tela de resultados do protótipo.....	52
Figura 34 - Ambiente para aquisição de imagens.....	53
Figura 35 - Imagens laterais externas e caule.....	54
Figura 36 - Imagens internas .....	54
Figura 37 - Exemplo de mancha de encosto não selecionada .....	58

## LISTA DE QUADROS

Quadro 1 - Requisitos funcionais .....	30
Quadro 2 - Requisitos não funcionais .....	30
Quadro 3 - Código responsável pelo redimensionamento da imagem .....	35
Quadro 4 - Código responsável pela transformação para escala de cinza.....	36
Quadro 5 - Código responsável pela detecção de contornos .....	37
Quadro 6 – Função <code>floodFill</code> .....	39
Quadro 7 - Método <code>copyTo</code> .....	39
Quadro 8 - Código responsável por executar o K-Means .....	40
Quadro 9 - Método <code>erodir</code> .....	41
Quadro 10 - Método <code>dilatar</code> .....	42
Quadro 11 - Método <code>extrairManchaAmarela</code> .....	43
Quadro 12 - Código responsável por criar as imagens em cores de cada região .....	44
Quadro 13 - Método <code>media_canal_sem_fundo</code> .....	46
Quadro 14 - Método <code>validarMancha</code> .....	47
Quadro 15 - Método <code>classificação</code> .....	48
Quadro 16 - Estrutura do arquivo <code>.CSV</code> .....	49
Quadro 17 - Métodos <code>treinarKNN</code> e <code>executarKNN</code> .....	50
Quadro 18 - Comparativo entre o protótipo desenvolvido e os trabalhos correlatos .....	58
Quadro 19 - Caso de uso UC01 Analisar imagem .....	65
Quadro 20 - Caso de uso UC02 Carregar imagem .....	65

## **LISTA DE TABELAS**

Tabela 1 - Resultados dos testes da classificação do KNN .....	55
Tabela 2 - Resultados dos testes de classificação da árvore de decisão .....	56

## **LISTA DE ABREVIATURAS E SIGLAS**

2D – Duas Dimensões

CSV – Comma-Separated Values

EA – Enterprise Architect

EMBRAPA – Empresa Brasileira de Pesquisa Agropecuária

HSL – Hue, Saturation and Lightness

IDE – Integrated Development Environment

KNN – K-Nearest Neighbor

LED – Light Emitting Diode

PDI – Processamento Digital de Imagens

RF – Requisito Funcional

RGB – Red, Green and Blue

RNA – Rede Neural Artificial

RNF – Requisito Não Funcional

ROI – Region of Interest

UC – Use Case

UML – Unified Modeling Language

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
1.1 OBJETIVOS.....	16
1.2 ESTRUTURA.....	16
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>17</b>
2.1 MELANCIA .....	17
2.2 SEGMENTAÇÃO DE IMAGENS .....	20
2.3 KNN .....	21
2.4 ÁRVORES DE DECISÃO.....	22
2.5 TRABALHOS CORRELATOS .....	24
2.5.1 Determination of watermelon volume using ellipsoid approximation and image processing.....	24
2.5.2 Classificação de laranjas baseada em padrões visuais .....	26
2.5.3 Classificação automática de frutas por análise de imagem - o caso da manga Tommy Atkins .....	28
<b>3 DESENVOLVIMENTO DO PROTÓTIPO.....</b>	<b>30</b>
3.1 REQUISITOS.....	30
3.2 ESPECIFICAÇÃO .....	30
3.2.1 Diagrama de casos de uso .....	30
3.2.2 Diagrama de classes .....	31
3.2.3 Diagrama de atividades .....	33
3.3 IMPLEMENTAÇÃO .....	34
3.3.1 Técnicas e ferramentas utilizadas.....	34
3.3.2 Operacionalidade da implementação .....	51
3.4 ANÁLISE DOS RESULTADOS .....	52
3.4.1 Montagem do ambiente, captura das imagens e classificação por degustação das melancias.....	52
3.4.2 Análise de resultados.....	54
3.4.3 Comparação com trabalhos correlatos .....	58
<b>4 CONCLUSÕES.....</b>	<b>60</b>
4.1 EXTENSÕES .....	61
<b>REFERÊNCIAS .....</b>	<b>62</b>

<b>APÊNDICE A – DETALHAMENTO DOS CASOS DE USO .....</b>	<b>65</b>
---	-----------

## 1 INTRODUÇÃO

Nos últimos anos, a fruticultura tem ganhado maior atenção com respeito aos setores do agronegócio brasileiro, ocupando a terceira posição entre os maiores produtores de frutas do mundo (CONFEDERAÇÃO DA AGRICULTURA E PECUÁRIA DO BRASIL, 2016). O setor de fruticultura está entre os principais geradores de renda, emprego e de desenvolvimento rural do agronegócio nacional (BUAINAIN; BATALHA, 2007, p. 13).

A atividade frutícola possui elevado efeito multiplicador de renda e, portanto, com força suficiente para dinamizar economias locais estagnadas e com poucas alternativas de desenvolvimento (BUAINAIN; BATALHA, 2007, p. 13). Mesmo depois da automação de muitos setores industriais, a inspeção de frutas ainda realizada por humanos, devido a inexistência de ferramentas computacionais e modelos consolidados para analisar frutas (SIMÕES; COSTA, 2003). Com isso, este grande setor produtivo, é incapacitado de exercer a sua produtividade máxima por não conseguir integrar novas tecnologias a um custo acessível em seus métodos de produção, onde a melhor solução encontrada para atender os altos níveis produção é o aumento de mão de obra (CARVALHO et al., 2014).

Grande parte das abordagens automáticas para esta tarefa concentra-se na análise de algum tipo de imagem digital do espectro visível, isto é, a parte do espectro a qual o olho humano é sensível, ou não visível, como por exemplo imagens de Raio-X (SIMÕES; COSTA, 2003). A análise de imagens juntamente com a visão computacional tem-se mostrado uma solução eficiente para resolver o problema da classificação de frutas, obtendo resultados semelhantes aos de uma pessoa. Os métodos em sua maioria utilizam o reconhecimento e análise de cores e formas para encontrar falhas como manchas, fungos e imperfeições no formato das frutas, bem como utilizam algoritmos de redes para criar um sistema de aprendizagem para aprimorar o desempenho da ferramenta (LEEMANS et al., 1998, 1999; RODRIGUES; LAVOIER FILHO, 2013).

Segundo Syazwan et al. (2012), a aparência externa é um dos principais fatores que influenciam o valor das frutas. No caso das melancias, por exemplo, os consumidores são atraídos pelo tom de cor da casca e pela sua distribuição na superfície. No entanto, tais características são distintas entre frutos, dificultando a criação de parâmetros de classificação por similaridade.

Diante deste contexto, este trabalho apresenta a criação de um protótipo para a análise da maturação de melancias através de um método não invasivo, utilizando técnicas de processamento de imagens para a classificação do fruto.

## 1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um protótipo para análise da maturação da melancia de forma não-invasivo por meio do uso de técnicas de processamento de imagens.

Os objetivos específicos são:

- a) identificar e extrair as características que definem se a melancia está madura ou não;
- b) classificar a melancia utilizando algoritmos de aprendizagem de máquina;
- c) criar e disponibilizar uma base de dados para a comunidade científica.

## 1.2 ESTRUTURA

O presente trabalho encontra-se subdividido em quatro capítulos: introdução, fundamentação teórica, desenvolvimento do protótipo e conclusões. No capítulo 1 são apresentados os objetivos e motivação para realização deste trabalho. O capítulo 2 contém a fundamentação teórica, responsável por abordar os principais assuntos retratados neste trabalho como o detalhamento das características do objeto de estudo, técnicas de processamento de imagem e aprendizado de máquinas. Na sequência, o capítulo 3 relata o processo de desenvolvimento do protótipo abordando os requisitos aos quais o protótipo pretende atender, os diagramas de casos de uso, classes e atividade. As ferramentas utilizadas, detalhes da implementação, operacionalidade e resultados também são descritos nesta seção. Por fim, no capítulo 4 são redigidas as conclusões finais alcançadas durante a produção deste trabalho juntamente com as sugestões para futuras extensões.

## 2 FUNDAMENTAÇÃO TEÓRICA

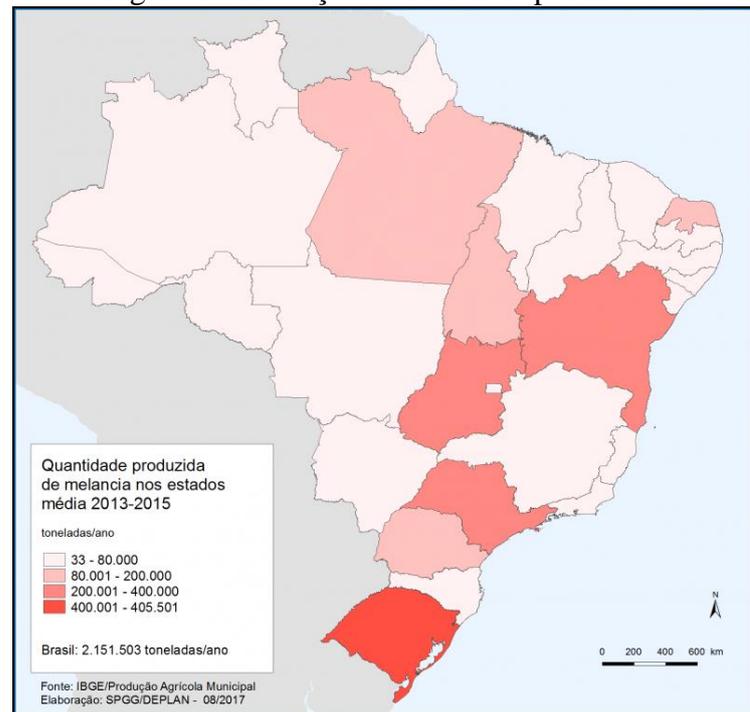
O objetivo deste capítulo é proporcionar uma explanação sobre os assuntos mais relevantes para o desenvolvimento do trabalho. Ele está subdividido em cinco partes, onde na seção 2.1 descreve-se sobre as principais características da melancia (*Citrullus Lanatus*) e sua produção nacional. Na seção 2.2 tem-se o detalhamento das técnicas de segmentação. A seção 2.3 aborda o algoritmo de aprendizagem de máquina KNN e na seção 2.4, árvores de decisão. Por fim, na seção 2.5 são descritos os trabalhos correlatos.

### 2.1 MELANCIA

A melancia (*Citrullus Lanatus*) pertence à família das cucurbitáceas (hortaliças) (MEDEIROS; ALVES, 2016), sendo originária da África Tropical, onde em sua forma selvagem é encontrada em muitos locais de clima tropical e subtropical sendo um fruto redondo pequeno com um diâmetro médio de 12cm (MORI, 1996). Os frutos grandes e doces foram obtidos no antigo Egito, difundindo-se posteriormente para Europa e Ásia (RESENDE; COSTA, 2003).

No Brasil, o cultivo da melancia é predominantemente exercido pela pequena propriedade familiar devido a sua rusticidade e baixo investimento de capital (ATLAS SÓCIOECONÔMICO, 2015). Os cultivares mais plantados são os de origem americana ou japonesa, os quais se adaptaram bem as nossas condições climáticas (COSTA; LEITE, 2007). Na Figura 1, pode-se observar os estados com maior produção de melancia no período de 2013-2015, onde o Rio Grande do Sul é consolidado como o maior produtor nacional, responsável por 19% da produção no período (ATLAS SÓCIOECONÔMICO, 2015).

Figura 1 - Produção de Melancia por Estado



Fonte: Atlas Sócio Econômico (2015).

Conforme Almeida (2003), a melancia é uma planta herbácea de ciclo vegetal anual que pode possuir forma redonda, oblonga ou alongada chegando a 60cm de comprimento. Sua coloração é verde podendo ser claro ou escuro, com uma tonalidade única, listrado ou manchado e sua polpa em geral é vermelha, mas também pode ser amarela, laranja, branca ou verde (ALMEIDA, 2003). A colheita da melancia é iniciada entre 35 e 45 dias após a abertura das flores femininas, o que corresponde ao período de 65 a 75 dias após o plantio (COSTA; LEITE, 2007). O fruto necessita ser colhido maduro, pois sua qualidade não melhora após a colheita (ALMEIDA, 2003).

Segundo Mathias (2016), quando não se há um controle do início do plantio, pode-se determinar o ponto de colheita da melancia a partir de certas características fisiológicas. Costa e Leite (2007) relatam 4 indicadores pelos quais o ponto de colheita pode ser determinado:

- a) mudança de coloração da parte do fruto em contato com o solo (mancha de encosto), que passa de branco para amarelo ou creme, onde a melancia a esquerda da Figura 2 apresenta um maior nível de maturação por ter a coloração de sua mancha amarelo/creme;

Figura 2 - Comparativo entre mancha de encosto



Fonte: adaptado de Incrível (2016).

- b) secamento da gavinha existente no mesmo nó do pedúnculo do fruto;

Figura 3 - Gavinha ressecada



Fonte: Projeto de horta (2014).

- c) mudança da cor da casca que de um verde brilhante passa para um tom mais opaco;  
 d) produz um som oco ao ser batida com o nó do dedo.

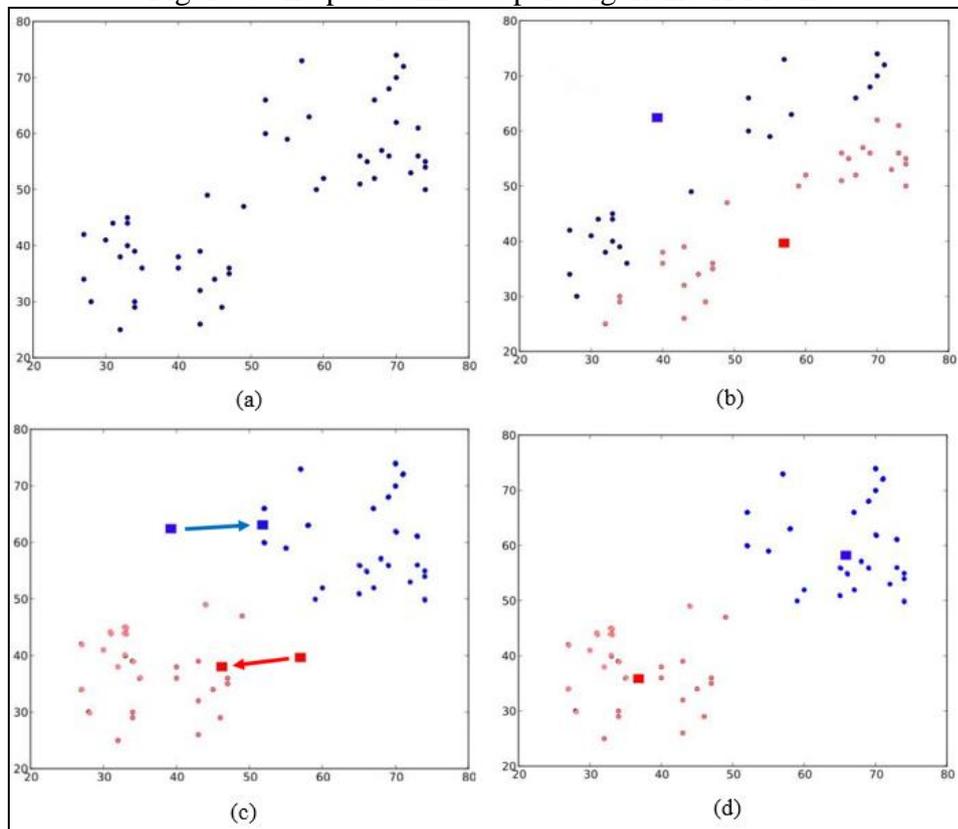
Para Incrível (2016) tais características apenas indicam o nível de maturação da fruta. Elas não garantem necessariamente que a melancia tenha um sabor adocicado. Ainda segundo o autor, as marcas de polinização e o seu formato, alongada ou redonda, são os melhores meios para se prever o sabor da melancia. Neste caso, quanto mais marcas de polinização, mais doce será a melancia. Em relação à forma, o autor destaca que se ela for arredondada será doce. Do contrário, alongada, a melancia terá gosto aguado.

## 2.2 SEGMENTAÇÃO DE IMAGENS

Conforme Facon (1993, p. 87), o alvo da segmentação é obter, a partir de uma imagem digitalizada pré-processada, um conjunto de “primitivas ou segmentos significativos” que contém a informação semântica relativa à imagem de origem. Gonzalez e Woods (2008, p 711) ainda descrevem a etapa de segmentação como uma etapa de complexidade variável, sendo dependente do problema a ser resolvido. No contexto deste trabalho, como a principal característica de análise é a cor, optou-se pelos algoritmos de segmentação por cor, no caso, o K-Means.

Conforme Dhanachandra et al. (2015), o algoritmo K-Means é um método de clusterização para dividir uma base de dados em um número específico de clusters. Os dados são agrupados nos clusters a partir da similaridade do valor da média de seus atributos numéricos (MACEDO et al., 2010). O algoritmo possui duas etapas, primeiramente ele calcula o centroide para cada cluster e na sequência atribui os dados ao cluster que se encontra mais próximo, utilizando normalmente a distância Euclidiana (DHANACHANDRA et al., 2015). Após uma primeira classificação dos dados, os centroides são realocados e classificados novamente. Este processo se repete até chegar a um critério de convergência (MACEDO et al., 2010). A Figura 4 demonstra graficamente a execução do K-Means

Figura 4 - Etapas realizadas pelo algoritmo K-Means



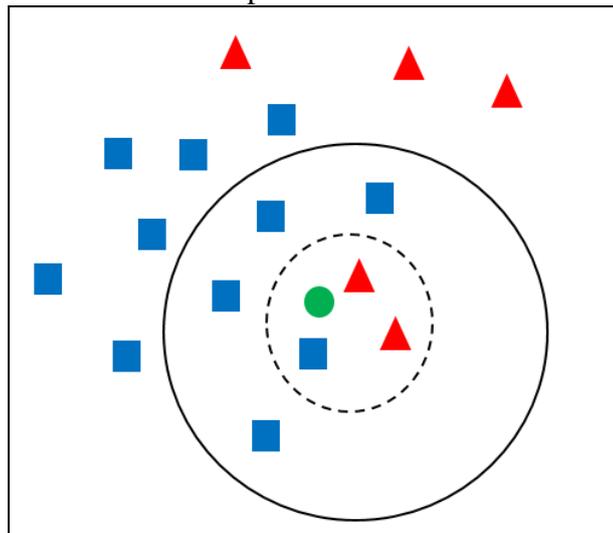
Fonte: adaptado de OpenCV (2014).

Na Figura 4 têm-se as etapas realizadas pelo algoritmo considerando uma base de dados com duas características, sendo divididos em dois grupos. A Figura 4a representa a base de dados ainda não classificada. Na Figura 4b foram adicionados aleatoriamente dois centroides para realizar a primeira classificação (representados pelos quadrados). A Figura 4c exibe o centroide recalculado, e por fim, a Figura 4d traz o posicionamento e classificação final desempenhado pelo K-Means.

### 2.3 KNN

O algoritmo K-Nearest Neighbor (KNN) segundo Almeida e Faceroli (2014) é um dos mais simples e mais utilizados algoritmos de aprendizagem, o qual consiste em classificar um novo exemplo calculando a distância entre os K exemplos mais próximos. Na Figura 5 é exemplificado o uso do valor de K, demonstrando que o número de vizinhos utilizados/analizados, interferem na classificação do novo indivíduo.

Figura 5 - Análise realizada pelo KNN com diferentes valores de K



Fonte: elaborado pelo autor.

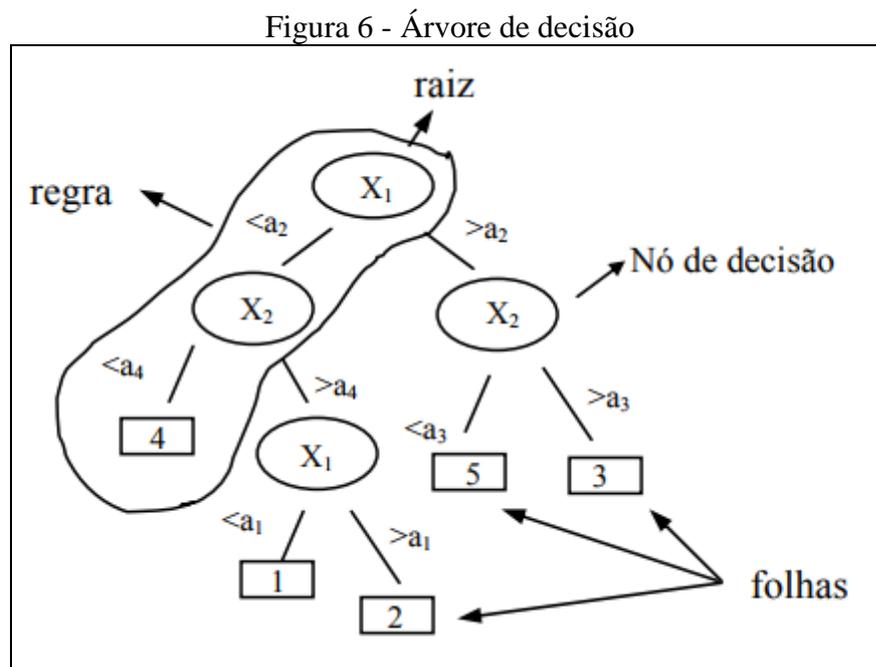
Considerando que o valor de K é 3, delimitado pela linha pontilhada, o resultado é um novo indivíduo, representado pelo círculo, que pertence ao grupo dos triângulos, porém alterando o valor de K para 7, é constatado que a maioria dos vizinhos são quadrados. Dessa forma, evidenciando a importância e a dependência do algoritmo em relação ao valor de K escolhido (OPENCV, 2014). Por fim, para que o KNN funcione corretamente, primeiramente é necessário treiná-lo por meio de exemplos já classificados, ação que constrói uma árvore de busca pela qual os novos indivíduos são classificados (OPENCV, 2014).

O critério de similaridade é um aspecto que possui grande influência no desempenho do método KNN (SANTOS, 2009). Esse critério é composto pela medida de similaridade, ou função de distância e pelo critério de seleção, fator que determina a forma de escolha dos K

vizinhos (SANTOS, 2009). Quando as características são numéricas, a distância Euclidiana pode ser utilizada para comparar os exemplos, porém, quando os valores das características são simbólicos, não ordenados (como a representação de conhecimento especialista em linguagem natural) os métodos do algoritmo KNN não são suficientes (BUANI, 2010).

## 2.4 ÁRVORES DE DECISÃO

Segundo Shiba et al. (2005), as árvores de decisão são amplamente utilizadas em algoritmos de classificação, sendo representações simples do conhecimento e um meio eficiente de construir classificadores que predizem classes baseadas nos valores de atributos de um conjunto de dados. Na Figura 6 é representada uma árvore de decisão onde cada nó de decisão contém um teste para algum atributo, cada ramo descendente corresponde a um possível valor deste atributo. Os conjuntos de ramos são distintos, cada folha está associada a uma classe e, cada percurso da árvore, da raiz a folha corresponde uma regra de classificação (GAMA, 2002).



Fonte: Gama (2002).

Segundo Shiba et al. (2005) as árvores de decisão tendem a crescer muito, de acordo com algumas aplicações, elas são muitas vezes substituídas pelas regras, devido as regras poderem ser facilmente modularizadas. Uma regra pode ser compreendida sem que haja a necessidade de se referenciar outras regras. Uma árvore de decisão tem função de particionar recursivamente um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe, sendo assim necessário comparar e examinar a classes durante sua construção (SHIBA et al., 2005).

A classificação de novas instâncias é feita percorrendo a árvore de forma *top-down* pelos nós de decisão até encontrar a folha, que rotulará essa nova instância, de acordo com a respectiva classe. Halmenschlager (2002) relata algumas das características das árvores de decisão que são utilizadas tanto no meio acadêmico como em aplicações reais:

- a) flexibilidade, não assumem uma distribuição única dos dados, sendo métodos não paramétricos. O espaço da instância é particionado em subespaços e cada subespaço é adaptado a diferentes modelos;
- b) robustez, a seleção interna de características produz árvores que tendem a ser bastante robustas mesmo com a adição de variáveis irrelevantes;
- c) interpretabilidade, todas as decisões são baseadas nos valores (conhecidos) dos atributos usados para descrever o problema;
- d) velocidade, a maioria dos algoritmos constrói as árvores de decisão de forma *top-down*, usando uma abordagem de “dividir para conquistar”, sem *backtracking*<sup>3</sup>.

Dentre os problemas mais comuns pode-se citar a relação feita por Gama (1999):

- a) fragmentação, causa o particionamento de dados em pequenos subconjuntos, o que é indesejável na presença de muitos atributos relevantes;
- b) replicação, é a duplicação de uma sequência de testes em diferentes ligações de uma árvore de decisão, levando a uma representação inconcisa que tende a diminuir a precisão preditiva;
- c) valores desconhecidos, podem causar problemas na decisão de que ligação seguir, pois uma árvore é uma hierarquia de testes;
- d) valores contínuos, a presença de atributos contínuos representa o gargalo de algoritmo, pois é necessário muito tempo de processamento para realizar o tratamento destes atributos;
- e) instabilidade, pequenas variações no conjunto de treinamento podem produzir grande variação na árvore final.

Para percorrer uma árvore de decisão, podem ser utilizados diversos processos de busca, dentre eles a busca em profundidade e a busca por largura. A busca em profundidade escolhe um caminho na árvore de decisão e o percorre até o final, onde irá encontrar o nó folha. Caso esse nó não seja a solução para o problema, irá voltar para o nó pai e percorrer o próximo caminho até o final chegando no próximo nó folha. Já a busca por largura vai testando gradativamente cada possibilidade encontrada, sem se aprofundar demais em cada

um dos caminhos. São avaliados primeiro todos os nós de um determinado nível, antes de prosseguir para a avaliação dos nós do próximo nível.

## 2.5 TRABALHOS CORRELATOS

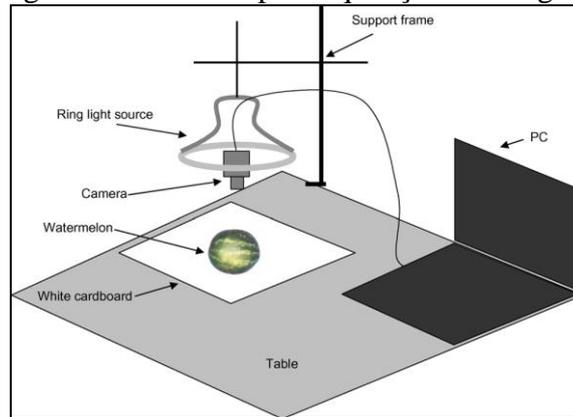
Nesta seção são apresentados três trabalhos os quais apresentam métodos de análise da qualidade de alimentos a partir de imagens pois não foram encontrados trabalhos diretamente relacionados ao objetivo de estudo deste trabalho. O primeiro trabalho, apresentado na seção 2.5.1, demonstra como é feito o cálculo aproximado do volume de melancias baseando-se na elipsoide (KOC, 2007). O trabalho de Simões e Costa (2003) apresentado na seção 2.5.2, utiliza a análise de imagem e redes neurais para verificar se a coloração da fruta está adequada. Por fim, a seção 2.5.3 apresenta uma análise de imagens juntamente com histogramas das cores para identificar a coloração dos frutos (CARVALHO et al., 2014).

### 2.5.1 Determination of watermelon volume using ellipsoid approximation and image processing

Segundo Koc (2007), o tamanho de um produto agrícola é frequentemente representado pelo seu peso, sendo este uma característica relativamente simples de ser obtida. Entretanto, a classificação do fruto pelo monitoramento durante o seu crescimento no campo pode ser um método mais eficiente para se obter o tamanho do fruto do que simplesmente calcular o seu peso. O peso dos produtos agrícolas também pode ser estimado a partir do seu volume quando a densidade do produto for conhecida.

No método desenvolvido por Koc (2007), para aquisição das imagens foi criada uma pequena estrutura que possibilita a captura de imagens em um ambiente controlado. A estrutura criada pode ser observada na Figura 7, sendo composta por uma câmera, um anel de luz fluorescente, um computador e um plano branco para garantir um plano de fundo uniforme. Sem alterar a posição da fruta, ela foi rotacionada e a imagem da superfície da fruta capturada.

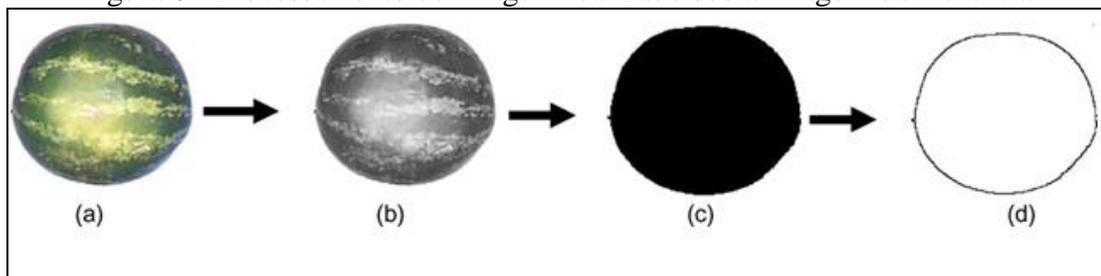
Figura 7 - Estrutura para aquisição de imagens



Fonte: Koc (2007).

Koc (2007), utiliza três abordagens para se medir o volume de melancias. Para fazer a análise inicialmente foi medido o volume da cada melancia utilizando o método de deslocamento de água. Cada melancia foi submersa em um recipiente cheio de água. O volume da água deslocada foi medido utilizando um cilindro graduado com capacidade de 500ml. Para realizar o processamento da imagem foram tiradas duas fotos RGB de cada melancia. Conforme mostra a Figura 8, a imagem original (item a) é convertida em uma escala de cinza de 8 bits (item b). Utilizando a técnica de limiar, a região de interesse é selecionada e convertida para uma imagem binária (item c). A detecção de borda é realizada pelo ImageJ, um editor de imagens multiplataforma (item d).

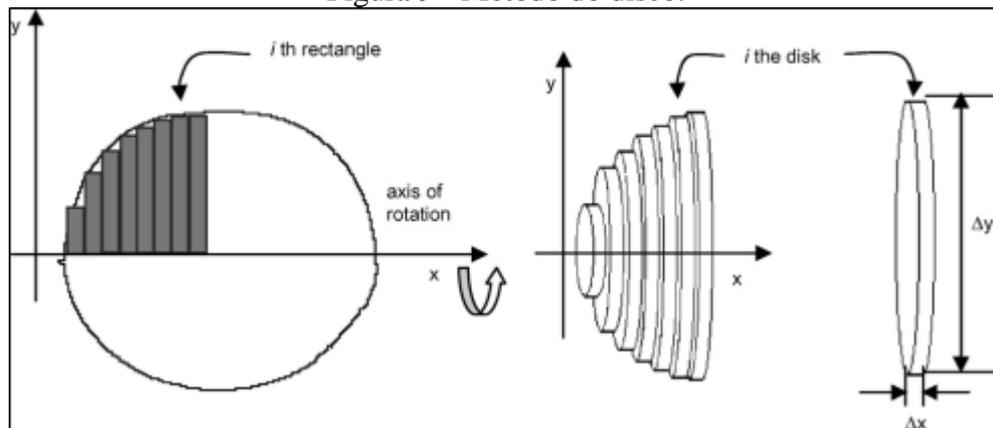
Figura 8 - Processamento de imagem realizado sob a imagem da melancia



Fonte: Koc (2007).

Após a obtenção do contorno da melancia a partir do método do disco, exemplificado na Figura 9, a parte superior da imagem de contorno é preenchida com retângulos que posteriormente são convertidos em discos, dos quais é calculado o volume, obtendo-se o volume total. O número de pixels que representa o comprimento e largura também foram medidos a partir da imagem capturada. As dimensões em pixels foram divididas pelas dimensões em milímetros e um fator de conversão foi calculado para cada melancia. O mesmo fator de conversão foi posteriormente utilizado para estimar o volume de cada melancia.

Figura 9 - Método do disco.



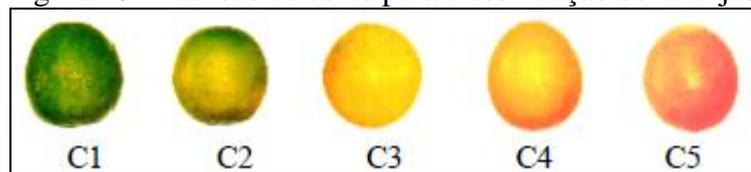
Fonte: Koc (2007).

O volume estimado usando o processamento de imagens e o cálculo da elipsoide foi comparado com o método de deslocamento de água. A diferença entre o volume estimado pelo cálculo da elipsoide e o deslocamento foi estatisticamente significativa, com uma taxa de erro maior que 5%, enquanto a diferença do cálculo do volume obtida pelo processamento de imagem e deslocamento da água não foi estatisticamente significativa. A partir desses resultados, Koc (2007) concluiu que o processamento de imagens proporciona uma metodologia simples, rápida e não invasiva para estimar o volume de melancias e pode ser facilmente implementado no monitoramento da taxa de crescimento de melancias no campo, durante a colheita mecânica estimando o peso de melancias individualmente ou no processamento pós-colheita.

### 2.5.2 Classificação de laranjas baseada em padrões visuais

Simões e Costa (2003) propuseram uma metodologia para classificação de laranjas com foco no tratamento do padrão de cor, adotando como mostrado na Figura 10 o padrão brasileiro de classificação de laranjas estipulado pelo centro de qualidade em horticultura, onde tanto a cor como qualidade são tidas como critérios de avaliação.

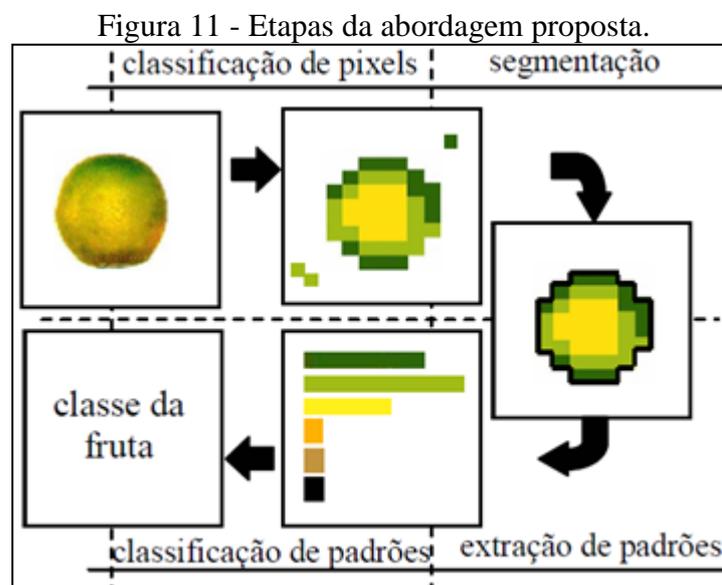
Figura 10 - Padrões de cores para classificação de laranjas.



Fonte: Simões e Costa (2003).

Conforme mostra a Figura 11, a proposta pode ser dividida em quatro partes: classificação das cores dos pixels da imagem, segmentação da imagem, extração de padrões e comparação de padrões encontrados com padrões previamente conhecidos. A classificação de cores foi feita utilizando uma rede neural artificial (RNA) já desenvolvida em trabalhos

prévios (Simões, 2000) que procura separar as cores dos frutos em seis classes, VE (Verde Escuro), VC (Verde Claro), A (Amarelo), LC (Laranja Claro), LE (Laranja Escuro) e M (Manchas). A segmentação das imagens foi realizada com o algoritmo de crescimento da região (SIMÕES; COSTA, 2003), que mapeia pixels individuais a conjuntos de pixels denominados regiões. Após a laranja ser identificada se dá início a análise de suas características, sendo que para este trabalho o vetor de cores é a característica mais importante. A comparação de padrões de cor é dada pela aproximação dos vetores de cores obtidos através da análise da laranja com os vetores de cada uma das classes dos padrões inicialmente definidos.



Fonte: Simões e Costa (2003).

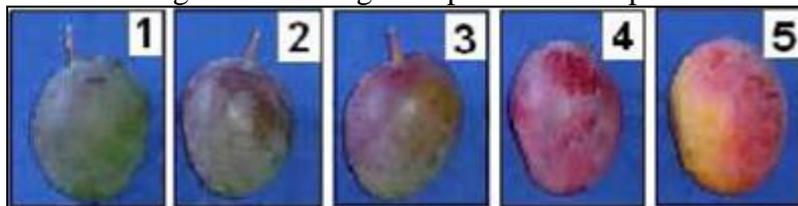
Para a captura de imagens, as laranjas foram colocadas sobre um fundo branco e as imagens foram adquiridas utilizando uma câmera colorida. Em seguida as imagens da câmera foram convertidas para uma imagem de 24 bits RGB com resolução de 200x200 pixels. A RNA foi treinada com 25 exemplos de cada uma das cores importantes no domínio (VE, VC, A, LC, LE e M). O sistema foi testado com 120 imagens de laranjas distintas divididas igualmente nas classes C1 a C5 onde houve 100% de acerto, quando comparado a avaliação humana. A abordagem se mostrou eficaz apresentando grande assertividade na análise das laranjas, sendo assim uma solução robusta a variações nas condições de iluminação do ambiente, a eliminação de rejeitos e contra erros de classificação de pixels. Entretanto, a classificação de padrões por meio da quantização vetorial não é capaz de implementar qualquer função de classificação, sendo neste caso mais indicado a utilização de um classificador Fuzzy.

### 2.5.3 Classificação automática de frutas por análise de imagem - o caso da manga Tommy Atkins

Carvalho et al. (2014) apresentam um método baseado em inspeção visual automática para a classificação de mangas da variedade Tommy Atkins (*Mangifera indica* L.) em padrões previamente definidos, através da avaliação da cor da casca. Os padrões são utilizados para selecionar as frutas pela cor da casca e determinar o seu nível de maturidade.

Para realizar a seleção das mangas utilizando como critério a cor da casca, é estabelecido um padrão de referência conforme exibido na Figura 12. Através desse padrão é possível realizar uma comparação para identificar a manga por meio critérios pré-estabelecidos. A comparação é feita através do modelo de cores Hue, Saturation e Lightness (HSL) que se assemelha a percepção natural do olho humano. Para capturar as imagens foi construída uma câmara escura, que proporciona um ambiente controlado, igualando características como iluminação ambiente, distância da lente ao objeto e posicionamento do objeto para todas as imagens.

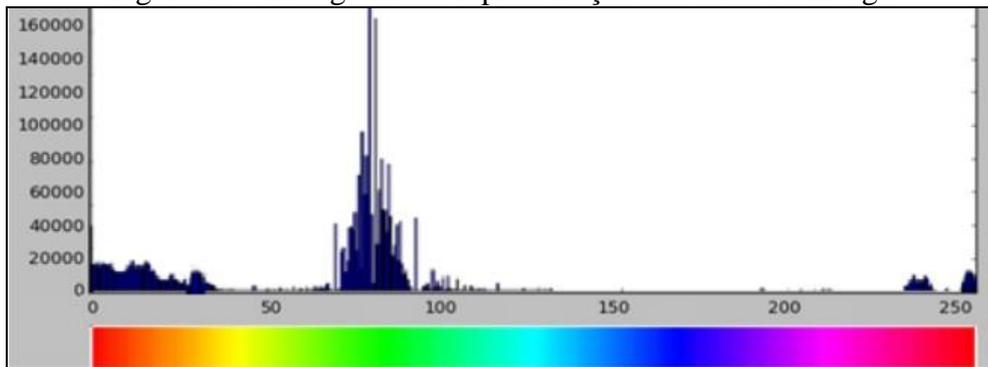
Figura 12 - Mangas do padrão Embrapa.



Fonte: Carvalho et al., (2014).

As imagens são convertidas do padrão RGB para o HSL. Então, são criados histogramas, como o apresentado na Figura 13, para cada componente do modelo, sendo descritos por matrizes que representam a imagem. Considerando o padrão definido pela Embrapa, os níveis de maturação da fruta são definidos pelo percentual de cobertura de três cores: Verde, Vermelho e Amarelo (CARVALHO et al., 2014). Para os autores, a avaliação consiste: na comparação do somatório de componentes de cada cor retirada do histograma da manga a ser classificada, na conversão desse resultado em percentuais de cobertura para cada cor e finalmente, na comparação com a tabela do padrão, com a indicação do respectivo nível.

Figura 13 - Histograma de representação das cores da manga



Fonte: Carvalho et al., (2014).

As avaliações foram realizadas com 170 mangas, que logo após serem higienizadas, foram utilizadas para adquirir as imagens utilizadas para criar o padrão de identificação. Dentre as 170 mangas disponíveis, 80 foram escolhidas aleatoriamente para análise do sistema conforme o padrão Embrapa. Para verificar a consistência dos resultados, mais 60 mangas foram adquiridas e testadas pelo sistema do mesmo modo ao qual as 170 mangas anteriores foram testadas.

O sistema obteve erro médio de 3%, indicando que ele é capaz de criar uma representação genérica para um conjunto amostra, e que essa representação foi suficiente para classificar novos elementos dentro dos níveis criados (CARVALHO et al., 2014). Com isso, a classificação de mangas, atividade extenuante e repetitiva, pode deixar de ser realizada por pessoas para que estas possam assumir atividades com maior complexidade além de ser uma solução de baixo custo.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo tem como objetivo descrever as etapas de desenvolvimento do protótipo proposto. Na seção 3.1 são relatados os requisitos do protótipo. A seção 3.2 apresenta a especificação. A seção 3.3 contém o detalhamento das etapas de implementação e, por fim, a seção 3.4 apresenta os resultados dos testes e sugestões para futuras melhorias.

#### 3.1 REQUISITOS

Os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) do protótipo para a análise da maturação de melancias são exibidos a seguir no Quadro 1 e Quadro 2, respectivamente. Os requisitos funcionais estão relacionados aos casos de uso da Figura 14.

Quadro 1 - Requisitos funcionais

Requisitos funcionais (RF)	Casos de uso (UC)
RF01: identificar a mancha de encosto utilizando o algoritmo K-Means	UC01
RF02: fazer a classificação da mancha de encosto por meio de uma árvore de decisão	UC01
RF03: fazer a classificação da parte verde da casca utilizando o algoritmo de aprendizagem de máquina KNN	UC01
RF04: classificar as melancias como imatura, madura ou muito madura	UC01

Fonte: elaborado pelo autor.

Quadro 2 - Requisitos não funcionais

Requisitos não funcionais (RNF)
RNF01: ser desenvolvido na linguagem C++ para plataforma Desktop
RNF02: utilizar o ambiente de desenvolvimento Microsoft Visual Studio 2017
RNF03: ser desenvolvido utilizando a biblioteca de visão computacional OpenCV

Fonte: elaborado pelo autor.

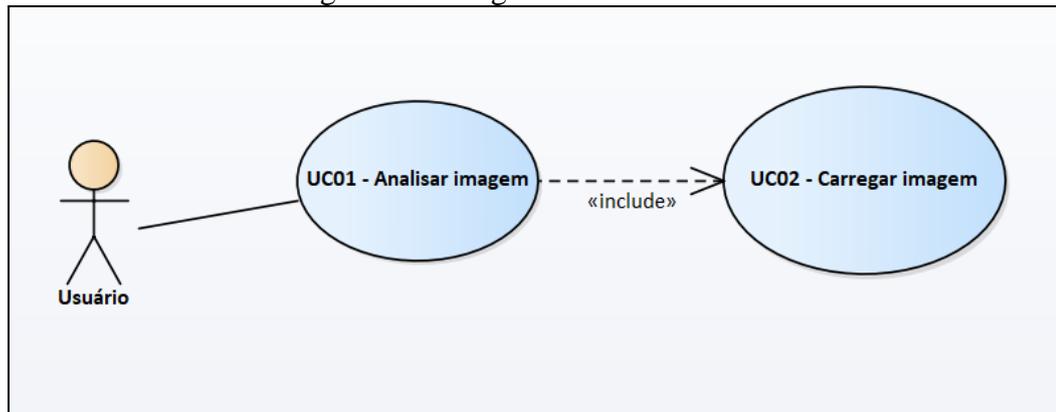
#### 3.2 ESPECIFICAÇÃO

Esta seção descreve a especificação do protótipo por meio da Unified Modeling Language (UML), utilizando a ferramenta Enterprise Architect (EA) com a qual foram desenvolvidos os diagramas de casos de uso, classes e atividades.

##### 3.2.1 Diagrama de casos de uso

Na Figura 14 são apresentados os casos de uso (em inglês Use Case - UC) pertinentes as principais funcionalidades do protótipo. O único ator identificado, o qual é denominado como *Usuário*, é responsável por exercer todas as atividades do protótipo.

Figura 14 - Diagrama de casos de uso



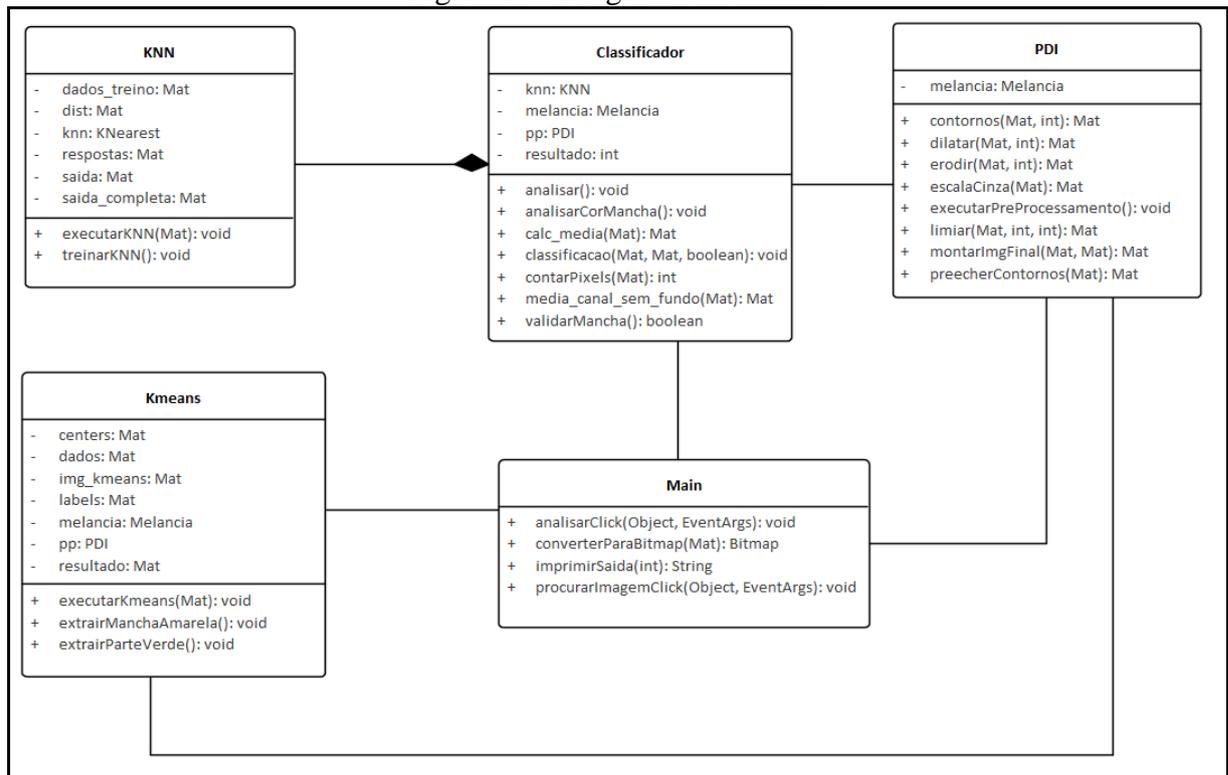
Fonte: elaborado pelo autor.

O caso de uso UC01 - Analisar imagem é responsável por carregar os dois parâmetros utilizados para realizar a classificação da melancia, a imagem da região lateral de uma melancia e um arquivo pré-configurado com extensão CSV, que contém os dados de melancias já classificadas e que serão utilizadas no treinamento do algoritmo KNN. Após o carregamento da imagem e do arquivo CSV é apresentada uma tela onde são exibidas as etapas mais relevantes do processamento de imagem juntamente com o resultado da classificação da melancia. No caso de uso UC02 - Carregar Imagem o protótipo deve exibir uma janela que possibilite a seleção da imagem de uma melancia. Caso a imagem informada pelo usuário seja inválida o sistema deverá abortar o carregamento do arquivo. O arquivo CSV não precisa ser informado pelo usuário, pois o mesmo já está disponível no diretório do programa e é diretamente carregado ao início da análise. Em seguida é realizada a análise do nível de maturação da melancia. O detalhamento dos casos de uso UC01, UC02 pode ser encontrado no Apêndice A.

### 3.2.2 Diagrama de classes

O diagrama de classes do protótipo desenvolvido é apresentado na Figura 15. Os construtores, métodos assessores e secundários de cada classe foram omitidos com o objetivo de proporcionar uma melhor visualização.

Figura 15 - Diagrama de classes



Fonte: elaborado pelo autor.

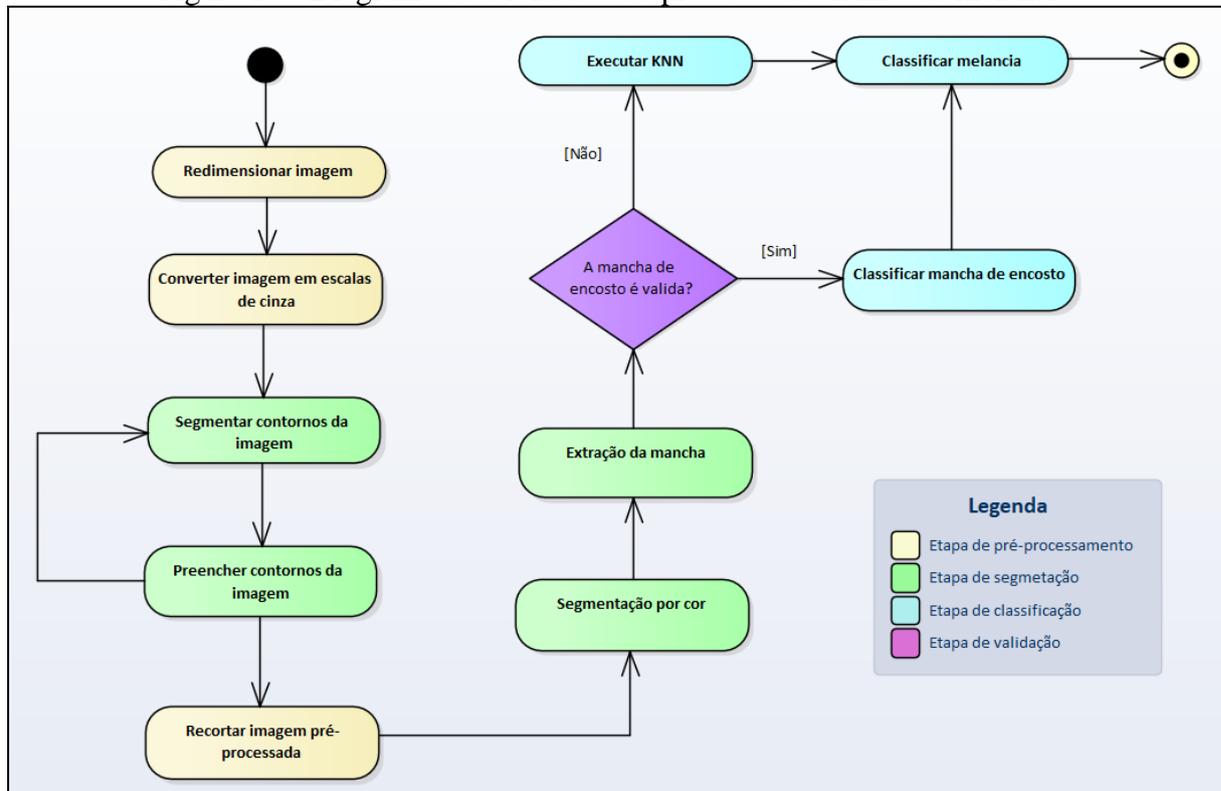
No início da análise são criadas algumas variáveis que serão repassadas para as classes de **PDI**, **Kmeans** e **Classificador**. Ao final do processo, tais variáveis terão os resultados de todas as etapas do processamento de imagens. A função da classe **PDI** é de extrair o plano de fundo através operações morfológicas deixando a região de interesse em destaque com intuito de facilitar a segmentação nas etapas subsequentes. No decorrer do fluxo de processamento alguns métodos da classe **PDI** são requisitados nas classes **Kmeans** e **Classificador** para refinar a segmentação realizada por ambas as classes.

Após a realização das etapas de pré-processamento é iniciada a segmentação da imagem através da classe **Kmeans** que extrai os pixels correspondentes a mancha de encosto, gerando duas novas imagens: a mancha de encosto e a melancia sem a mancha de encosto. Utilizando tais imagens, a classe **Classificador** realiza a classificação da melancia. A imagem que representa a mancha de encosto é analisada na própria classe com o método **classificação**. A análise da imagem sem a mancha de encosto acontece na classe **KNN** que constrói um ambiente de comparação com os dados carregados através de um arquivo **.CSV**. Por fim, as imagens pertinentes a cada etapa do processamento juntamente com o resultado da classificação são exibidas em uma nova janela.

### 3.2.3 Diagrama de atividades

Esta seção apresenta o diagrama de atividade para análise da maturação de melancias partindo de uma imagem de entrada e um arquivo CSV pré-configurado contendo os dados de melancias já classificadas. As etapas estão representadas na Figura 16

Figura 16 - Diagrama de atividades do processo de análise da melancia



Fonte: elaborado pelo autor.

A Figura 16 apresenta o processo de análise da melancia. Ele inicia com o redimensionando a imagem, convertendo-a em seguida para escala de cinza. Na sequência é realizada a extração dos contornos da imagem, tendo como objetivo realçar a borda da melancia. Após a detecção dos contornos, eles são preenchidos. Vale ressaltar que apenas os contornos fechados são preenchidos. Depois desta etapa, obtém-se uma imagem binária com maior contorno, representando a melancia e consecutivamente a região de interesse (em inglês Region of Interest - ROI). Esta imagem binária é utilizada como máscara para gerar a imagem pré-processada apenas com a melancia em cores, ou seja, sem o fundo.

A aplicação do algoritmo de clusterização K-Means é a primeira etapa do processo de extração da mancha de encosto. O algoritmo tem como objetivo criar uma nova representação da imagem a partir de um número de centroides fornecido. O número de centroides é popularmente conhecido como valor de K. Na sequência são realizadas as operações de erosão e dilatação para tornar a ROI contínua. A imagem é novamente convertida para uma

escala de cinza, e então, percorrida para selecionar os pixels mais claros criando uma imagem binária correspondente a mancha de encosto.

Encerrado o processo de extração da mancha de encosto é dado início a análise da mesma. A região é contornada e convertida para suas cores originais para então ter sua cor analisada, verificando se a tonalidade corresponde a um tom amarelado. Caso a mancha de encosto seja composta por mais de uma região, cada região é contornada e analisada individualmente. Com isso a maioria das regiões de verde claro e reflexos são eliminados. À partir desta análise são geradas duas imagens: a imagem da mancha de encosto e a imagem da parte verde, que seria a imagem pré-processada subtraída da mancha de encosto.

A próxima etapa consiste em validar a mancha de encosto. Para ser válida, a mancha de encosto precisa ter uma área superior a 5% da área da melancia. Esta etapa determinará se a mancha de encosto será ou não utilizada na classificação da melancia.

Encerrando o fluxo, tem-se a etapa de classificação da melancia, sendo realizada por meio de uma árvore de decisão. Inicialmente é verificado se a mancha de encosto é válida. Caso seja, ela é categorizada a partir da tonalidade média de sua cor, do mesmo modo, somente a parte verde é analisada e classificada com a técnica de aprendizado KNN, comparando a imagem atual com outras previamente classificadas.

### 3.3 IMPLEMENTAÇÃO

A seguir são contempladas as técnicas e ferramentas utilizadas assim como, a operacionalidade da implementação. Na seção 3.3.1 tem-se o detalhamento das técnicas e ferramentas utilizadas. Concluindo, na seção 3.3.2 é descrita a operacionalidade da implementação.

#### 3.3.1 Técnicas e ferramentas utilizadas

No desenvolvimento do protótipo utilizou-se da linguagem de programação C++ em conjunto ao ambiente de desenvolvimento Visual Studio 2017. As tecnologias utilizadas estão listadas abaixo:

- a) OpenCV: Biblioteca de visão computacional composta por uma série de algoritmos de processamento de imagens e aprendizado de máquinas. Dentre as principais funcionalidades utilizadas para o desenvolvimento do protótipo encontram-se funções para transformação de imagens em escala de cinza, segmentação, operações morfológicas, detecção de contornos, preenchimento de contornos além de algoritmos de clusterização e aprendizagem de máquina;

- b) Windows Forms: Plataforma da Microsoft baseada no Framework .Net, utilizada para criação de interfaces gráficas para plataforma Windows. Por meio desta plataforma foi criada a interface do protótipo;
- c) arquivo .CSV: é um formato simples de armazenamento que agrupa as informações em arquivos de texto separando os valores com vírgula ou outro caractere tabulador. Os resultados da classificação das melancias foram carregados para o protótipo através de um arquivo .CSV.

Nas próximas seções estão descritas as etapas realizadas na análise da maturação de melancias. Para facilitar o entendimento do processo, ele foi subdividido nas seguintes etapas: pré-processamento, identificação e extração da mancha de encosto e, classificação das melancias. Os exemplos de código fonte e resultados associados a cada etapa também são demonstrados.

### 3.3.1.1 Pré-Processamento

O pré-processamento tem como principal objetivo remover o plano de fundo da imagem. Para isso, inicialmente, a imagem original é redimensionada e transformada em escala de cinza. Na sequência é realizada a detecção e preenchimento dos contornos.

#### 3.3.1.1.1 Redimensionamento da imagem

A partir do caminho fornecido pelo usuário a imagem é carregada para memória em conjunto com o arquivo .CSV que é posteriormente utilizado pela classe `KNN`. Com o propósito de melhorar a performance, a imagem é redimensionada para a resolução de 240 x 240 pixels, reduzindo assim a área a ser processada.

O redimensionamento da imagem é feito pelo método `resize` disponibilizado pela biblioteca OpenCV que altera o tamanho da imagem conforme o valor informado no parâmetro `size`. Na linha 2 do Quadro 3 é demonstrada a chamada do método `resize`.

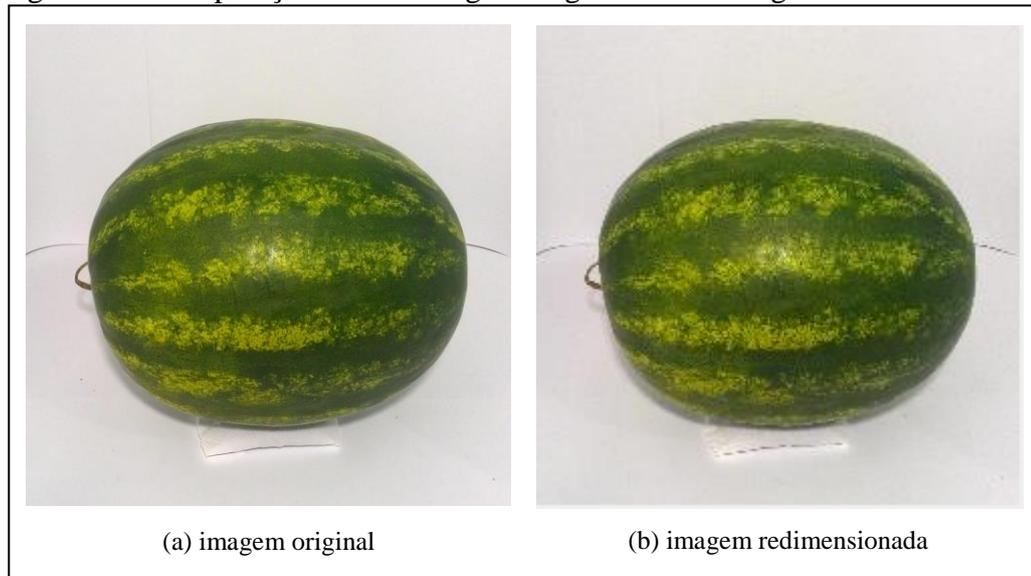
Quadro 3 - Código responsável pelo redimensionamento da imagem

1	<code>imagem = imread(caminhoImagem);</code>
2	<code>resize(imagem, imagemRedimensionada, size);</code>

Fonte: Elaborado pelo autor.

A Figura 17 expõe o resultado do redimensionamento sendo a Figura 17a a imagem original e a Figura 17b a imagem redimensionada. Percebe-se que há alterações em relação aos contornos dos objetos contidos na imagem. Porém, isso não interfere na análise que será realizada.

Figura 17 - Comparação entre a imagem original com a imagem redimensionada



Fonte: elaborado pelo autor.

Após realizar o redimensionamento, a imagem é convertida para escala de cinza. Vale ressaltar que nas etapas de pré-processamento a cor não é relevante por isso foi removida, mantendo apenas sua iluminação. Por meio do método `cvtColor`, exibido no Quadro 4, é realizada a transformação para escala de cinza. A conversão para escala de cinza é definida pelo parâmetro `CV_BGR2GRAY`, já que o método `cvtColor` da biblioteca OpenCV pode ser utilizado para outras conversões.

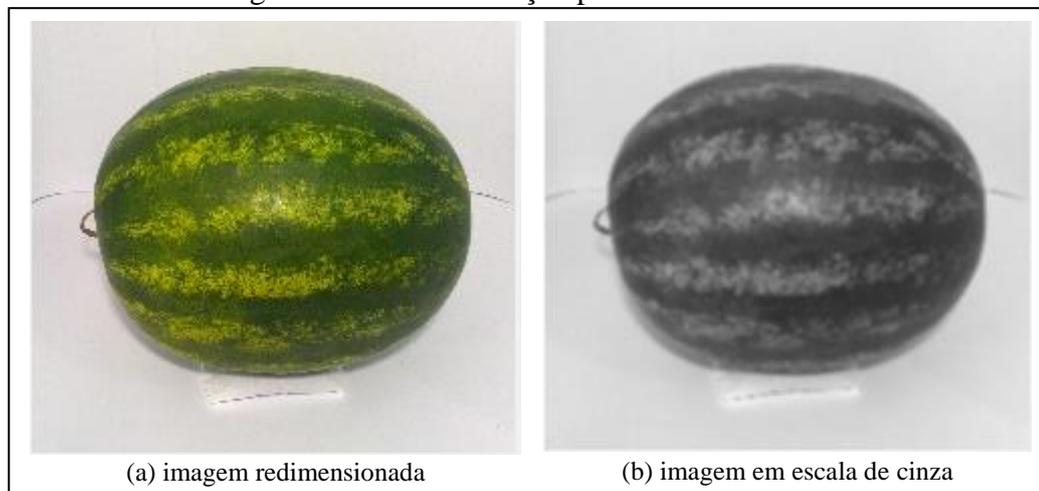
Quadro 4 - Código responsável pela transformação para escala de cinza

01	<code>cvtColor(img, img_cinza, CV_BGR2GRAY);</code>
----	---

Fonte: elaborado pelo autor.

A Figura 18a exibe o resultado da conversão para escala de cinza, onde a Figura 18a corresponde a imagem de entrada e a Figura 18b a escala em cinza.

Figura 18 - Transformação para escala de cinza



Fonte: elaborado pelo autor.

### 3.3.1.1.2 Detecção de contornos

Após converter a imagem para escala de cinza é realizada a detecção de contornos. Nesta etapa são aplicadas 4 técnicas disponíveis na biblioteca OpenCV que são: `blur`, `Canny`, `findContours` e `drawContours`.

Primeiramente é aplicado o método `blur`, que tem por finalidade desfocar a imagem, facilitando assim a detecção dos contornos. Posteriormente, utilizou-se o método `Canny` para localizar as bordas, onde os pixels acima do valor máximo são definidos como bordas e os abaixo são descartados. O método de `Canny` tem como retorno um vetor contendo as bordas da imagem, as quais são repassadas ao método `findContours` que identifica as bordas que formam contornos. Concluindo a detecção de contornos, o método `drawContours` compõem uma nova imagem com os contornos identificados.

O Quadro 5 retrata o processo de detecção de contornos. Na linha 1 é representado o método de `blur`, nas linhas 2 e 4 estão os métodos de `Canny` e `findContours` respectivamente seguidos do método `drawContours`.

Quadro 5 - Código responsável pela detecção de contornos

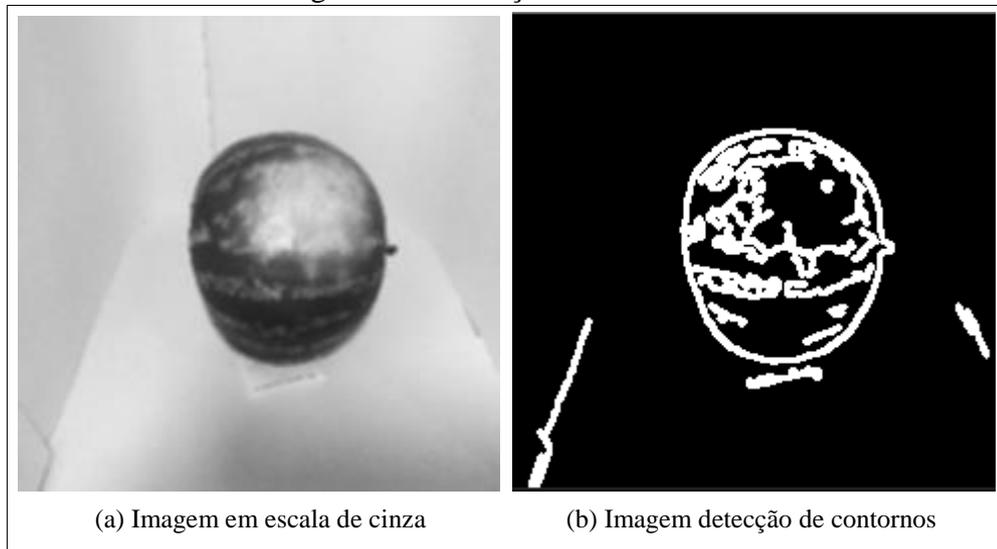
```

1  blur(img_preto_branco, img_desfocada, Size(3, 3));
2  Canny(img_desfocada, canny_output, valor_canny,
3      valor_canny * 2, 3);
4  findContours(canny_output, contornos, hierarchy, CV_RETR_EXTERNAL,
5              CV_CHAIN_APPROX_SIMPLE, Point(0, 0));
6
7  Mat desenho = Mat::zeros(canny_output.size(), CV_8UC3);
8  if(tipo==0){
9      Scalar color = Scalar(255, 255, 255);
10     for (int i = 0; i< contornos.size(); i++){
11         drawContours(desenho, contornos, i, color,
12                     2, 8, hierarchy, 0, Point());
13     }
14 }
15 else if(tipo==1){
16     for (int i = 0; i< contornos.size(); i++) {
17         a = contourArea(contornos[i], false);
18         if (a>maior_area) {
19             maior_area = a;
20             maior_area_indice = i;
21         }
22     }
23     Scalar color = Scalar(255, 0, 0);
24     drawContours(desenho, contornos, maior_area_indice, color,
25
26 }
```

Fonte: elaborado pelo autor.

A Figura 19 apresenta o resultado da detecção de contornos realizada pelos métodos supracitados a partir da imagem em escala de cinza. Na Figura 19b é possível observar a existência de outros contornos provenientes de sombras que precisam ser removidos.

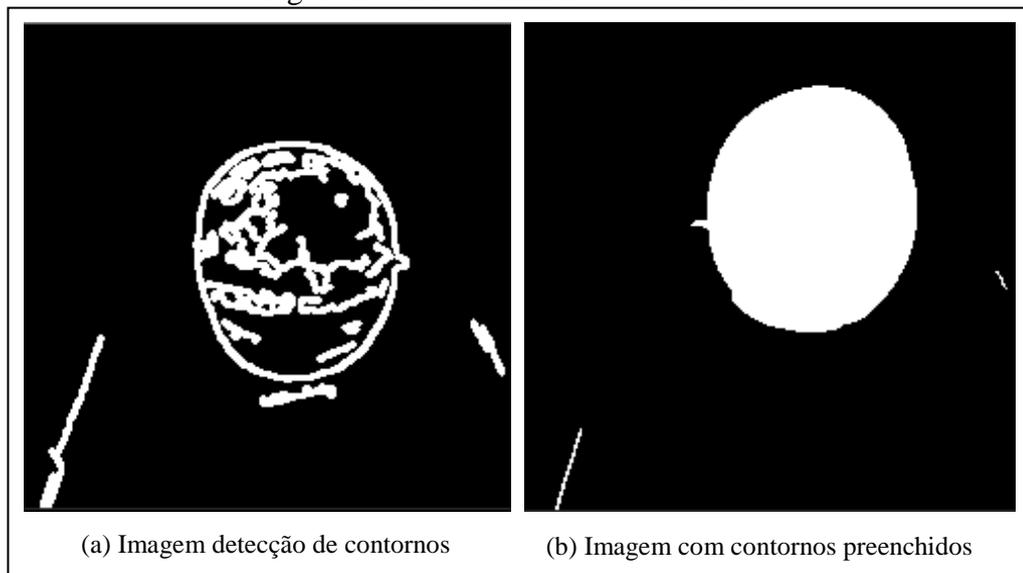
Figura 19 - Detecção de contornos



Fonte: elaborado pelo autor.

Os contornos encontrados são preenchidos através da função `floodFill`. Apesar de eficiente, ocasionalmente são encontradas pequenas regiões compreendidas entre os contornos que não são preenchidas pela função `floodFill`. Para garantir que a máscara empregada na extração da ROI seja composta por uma região homogênea é aplicada a operação morfológica de erosão. A Figura 20 mostra o resultado obtido após a aplicação da função `floodFill` sobre a imagem da detecção de contornos. Na sequência o Quadro 6 apresenta função `floodFill`.

Figura 20 - Preenchimento de contornos



Fonte: elaborado pelo autor.

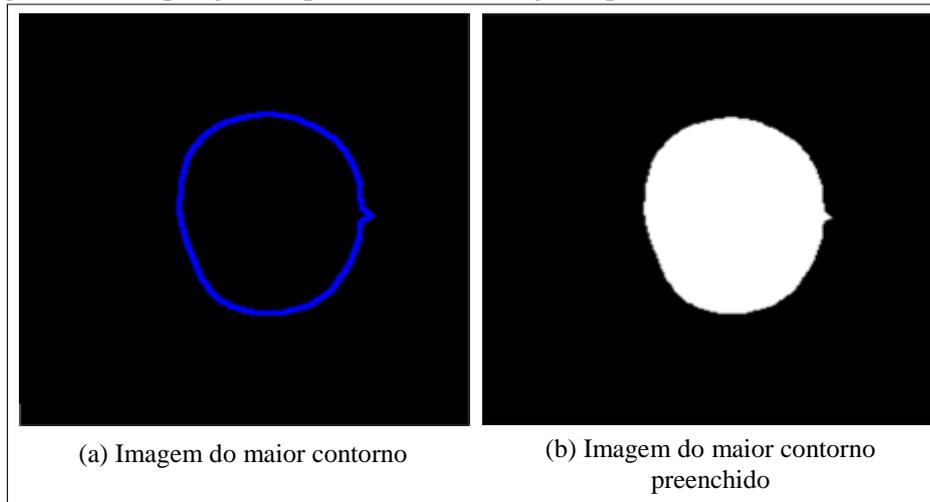
Quadro 6 – Função `floodFill`

01	<code>floodFill(img_preenchida, cv::Point(0, 0), Scalar(255,255,255));</code>
----	---

Fonte: elaborado pelo autor.

Entretanto, algumas regiões podem ser interpretadas como contornos. Neste caso, realizou-se uma nova detecção de contornos com o objetivo de selecionar o contorno com a maior área, conforme ilustrado na Figura 21. Na Figura 21a encontra-se o maior contorno e na Figura 21b, ele preenchido.

Figura 21 - Repetição do processo de detecção e preenchimento de contornos



Fonte: elaborado pelo autor.

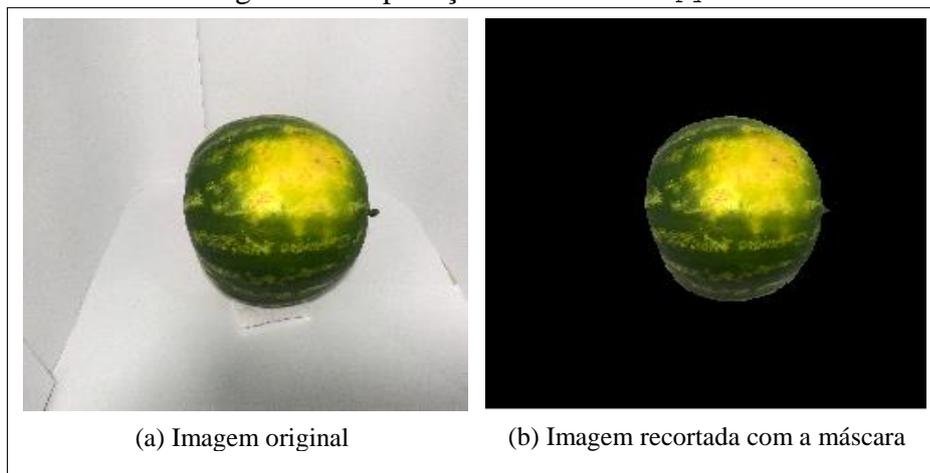
Como a análise da maturação da melancia é baseada em sua coloração, foi aplicada uma máscara sobre a imagem original para ter-se novamente uma imagem no formato RGB. O método `copyTo` da biblioteca OpenCV permite extrair a parte de uma imagem por meio de uma máscara binária. O Quadro 7 mostra o código responsável por criar uma nova imagem através da aplicação de uma máscara binária sob a imagem original

Quadro 7 - Método `copyTo`

1	<code>Mat aux;</code>
2	<code>img.copyTo(aux, mascara);</code>

Fonte: elaborado pelo autor.

A máscara em questão é região referente a melancia representada pela Figura 21b e o resultado de sua aplicação sob a imagem original da melancia pode ser visto na Figura 22.

Figura 22 - Aplicação do método `copyTo`

Fonte: elaborado pelo autor.

### 3.3.1.2 Identificação e extração da mancha de encosto

A identificação da mancha de encosto é realizada por meio do algoritmo de clusterização K-Means em conjunto com as operações morfológicas de erosão e dilatação. O `Kmeans` é um algoritmo de clusterização que reconstrói a imagem com as cores de maior predominância. A quantidade de cores utilizadas para reconstruir a imagem é definida pelo valor de `K`. Quanto mais perto o valor de `K` for do número original de cores, maior será a semelhança da imagem gerada com a imagem original. A biblioteca OpenCV dispõem do `Kmeans`, o qual é exposto no Quadro 8.

Quadro 8 - Código responsável por executar o K-Means

```

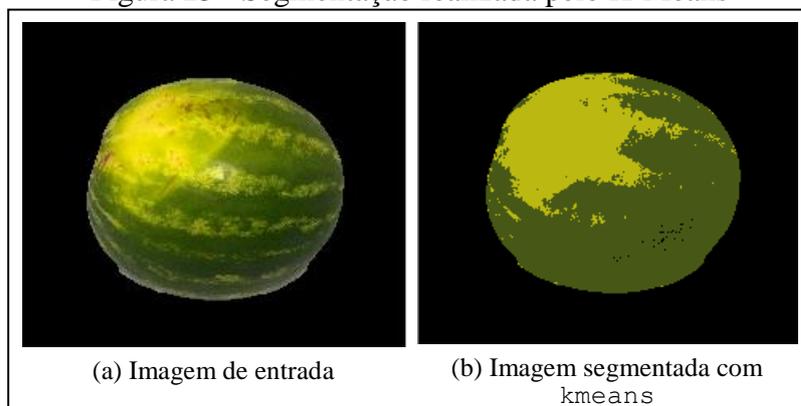
1 dados.convertTo(dados, COLOR_BGR2RGB);
2 dados = dados.reshape(3, dados.cols * dados.rows);
3 dados.convertTo(dados, CV_32FC1);
4 centers.convertTo(centers, CV_32FC1);
5
6 kmeans(dados, 3, labels, TermCriteria(TermCriteria::EPS +
7   TermCriteria::COUNT, 10, 1.0), 5, KMEANS_PP_CENTERS, centers);
8 centers.convertTo(centers, CV_8UC1);
9
10 for (int i = 0; i<labels.rows; i++){
11   if (labels.at<int>(i) == 0) {
12     resultado.push_back(centers.row(0));
13   }else if (labels.at<int>(i) == 1) {
14     resultado.push_back(centers.row(1));
15   }else {
16     resultado.push_back(centers.row(2));
17   }
18 }

```

Fonte: elaborado pelo autor.

Nas linhas 1 a 4 os dados são configurados de modo a estarem aptos a serem utilizados pela função `kmeans` e os vetores que irão armazenar os resultados são convertidos para um formato compatível. A chamada da função `kmeans` ocorre nas linhas 6, 7 e 8 com a passagem dos dados configurados além do vetores que irão armazenar os resultados e o valor de K. As linhas 10 a 18 são responsáveis por reconstruir a imagem a partir do resultado da função `kmeans`. No protótipo desenvolvido, o valor utilizado como K foi 3, dividindo a imagem em 3 partes que representam o plano de fundo da imagem, a parte verde da melancia e a mancha de encosto, conforme mostra a Figura 23.

Figura 23 - Segmentação realizada pelo K-Means



Fonte: elaborado pelo autor.

Ao observar a Figura 23 é constatado um excesso de ruído na imagem resultante. Para reduzi-los, foram aplicados os filtros morfológicos de erosão e dilatação. O filtro morfológico de erosão equivale ao aumento das componentes escuras da imagem. O Quadro 9 contém o método utilizado para realizar a operação de erosão, que tem como parâmetros a imagem a ser erodida e o tamanho do elemento estruturante com o qual a imagem deve ser erodida.

Quadro 9 - Método `erodir`

```

1  Mat PreProcessamento::erodir(Mat img_binaria, int tamanho) {
2
3  Mat img_erodida;
4  int erosion_size = tamanho;
5
6  Mat element = getStructuringElement(cv::MORPH_RECT,
7  cv::Size( tamanho, tamanho), cv::Point(-1, -1));
8
9  erode(img_binaria, img_erodida, element);
10 return img_erodida;
11 }

```

Fonte: elaborado pelo autor.

As linhas 6 a 7 compreendem a criação do elemento estruturante através do método `getStructuringElement`. Na linha 9, encontra-se a chamada do método `erode` que é

responsável realçar os objetos escuros. Ambos os métodos, `getStructuringElement` e `erode` estão disponíveis na biblioteca OpenCV. O efeito do método `erode` sobre a imagem pode ser visualizado na Figura 24.

Figura 24 - Imagem erodida



Fonte: elaborado pelo autor.

O filtro morfológico de dilatação funciona de forma semelhante ao de erosão mas com objetivo oposto, aumentando as áreas claras da imagem. Na dilatação assim como na erosão a imagem é percorrida por um elemento estruturante, que neste caso, provoca efeitos de dilatação nas partes escuras da imagem, deixando-as mais claras, como mostra a Figura 25.

Figura 25 - Imagem dilatada



Fonte: elaborado pelo autor.

O método que executa a dilatação é retratado no Quadro 10, recebendo como parâmetros a imagem a ser dilatada e o tamanho do elemento estruturante.

Quadro 10 - Método dilatar

```

1  Mat PreProcessamento::dilatar(Mat img_binaria, int tamanho) {
2
3      Mat img_dilatada;
4      int dilatation_size = tamanho;
5
6      Mat element = getStructuringElement(cv::MORPH_RECT,
7          cv::Size(tamanho, tamanho), cv::Point(-1, -1));
8
9      dilate(img_binaria, img_dilatada, element);
10     return img_dilatada;
11 }

```

Fonte: elaborado pelo autor.

Nas linhas 6 a 7 do Quadro 10, tem-se a criação do elemento estruturante pelo método `getStructuringElement` e na linha 9 a execução do método `dilate`. Tanto método `getStructuringElement` como o método `dilate` são encontrados na biblioteca OpenCV.

A extração da área correspondente a cor mais clara da imagem gerada pelo algoritmo `kmeans` é feita com o método `extrairManchaAmarela` descrito no Quadro 11. O método é dividido em duas partes, primeiramente como mostrado na linha 5 do Quadro 11, a imagem é convertida para escala de cinza. Na sequência, percorre-se a imagem verificando o valor de cada pixel, caso o valor seja maior que 90 e menor 200, assume-se que o pixel faz parte da mancha de encosto da melancia e tem seu valor alterado para 255, do contrário, o valor pixel recebe 0.

Quadro 11 - Método `extrairManchaAmarela`

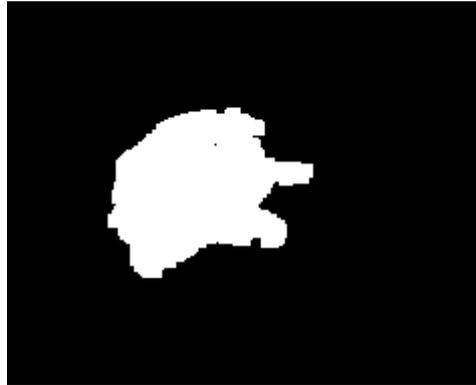
```

1 void Kmeans::extrairManchaAmarela() {
2
3 Mat mancha;
4
5 cvtColor(melancia.getImgKmeans(), mancha, CV_BGR2GRAY);
6
7 for (int i = 0; i < mancha.rows; i++) {
8     for (int j = 0; j < mancha.cols; j++) {
9         if (mancha.at<uchar>(i, j) > 90 &&
10             mancha.at<uchar>(i, j) < 200) {
11             mancha.at<uchar>(i, j) = 255;
12         }
13         else{
14             mancha.at<uchar>(i, j) = 0;
15         }
16     }
17 }
18 melancia.setImgMancha (mancha) ;
19 }

```

Fonte: elaborado pelo autor.

O método `extrairManchaAmarela` retorna uma imagem binária representando a região da mancha de encosto. O resultado do método é ilustrado na Figura 26. Caso não sejam encontrados valores referentes a mancha de encosto o resultado do método é uma imagem totalmente preta.

Figura 26 - Resultado do método `extrairManchaAmarela`

Fonte: elaborado pelo autor.

O processo de validação consiste em verificar dois aspectos da mancha: a tonalidade de cada uma das regiões e a área em relação a superfície da melancia. A análise da tonalidade da mancha inicia com a detecção dos contornos. Diferente da etapa de pré-processamento onde os contornos são seleccionados e preenchidos de uma vez, a análise da tonalidade da mancha detecta cada contorno individualmente, utilizando-o como máscara e reconstruindo cada região com suas respectivas cores. No Quadro 12 tem-se o código associado a criação da imagem em cores de cada região.

Quadro 12 - Código responsável por criar as imagens em cores de cada região

```

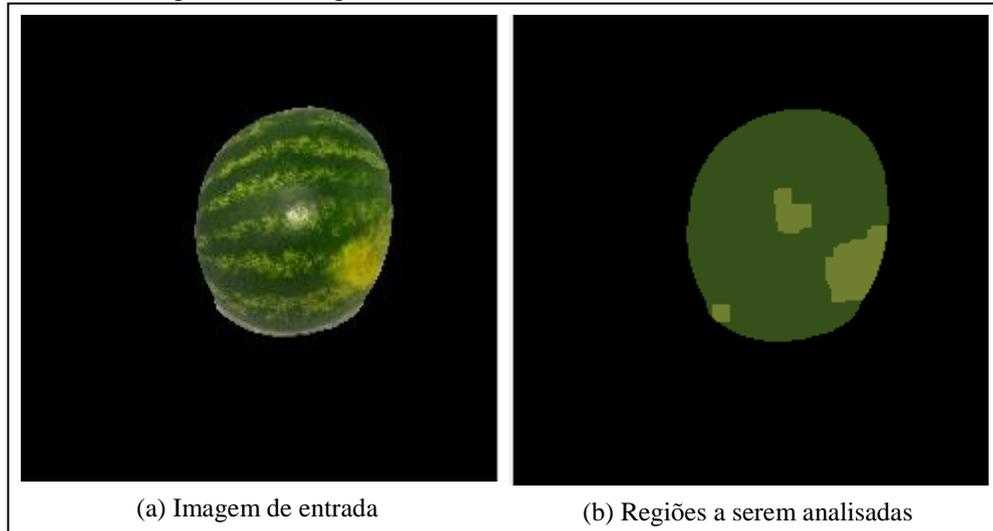
1  for (int i = 0; i < contornos.size(); i++) {
2  Mat subMedia(Mat_<float>(1, 3));
3  Scalar color = Scalar(255, 255, 255);
4  Mat subMancha = Mat::zeros(canny_output.size(), CV_8UC3);
5
6  drawContours(subMancha, contornos, i, color,
7              2, 8, hierarchy, 0, Point());
8
9  subMancha = pp.preencherContornos(subMancha);
10 melancia.getImgFinal().copyTo(subMancha, subMancha);
11 subMedia = media_canal_sem_fundo(subMancha);
12
13 if (subMedia.at<float>(0, 0) > 130 &&
14     subMedia.at<float>(0, 1) > 130 &&
15     subMedia.at<float>(0, 2) < 40) {
16     drawContours(desenho, contornos, i, color,
17                 2, 8, hierarchy, 0, Point());
18 }
19 }
20
21 desenho = pp.preencherContornos(desenho);
22 melancia.getImgFinal().copyTo(desenho, desenho);
23 melancia.setImgMancha(desenho);

```

Fonte: elaborado pelo autor.

Na linha 11 do Quadro 12 ocorre a chamada do método `media_canal_sem_fundo`, que retorna o valor médio de cada canal da imagem. Na Figura 27 é apresentado um exemplo de melancia onde foram identificadas mais de uma região como sendo a mancha de encosto.

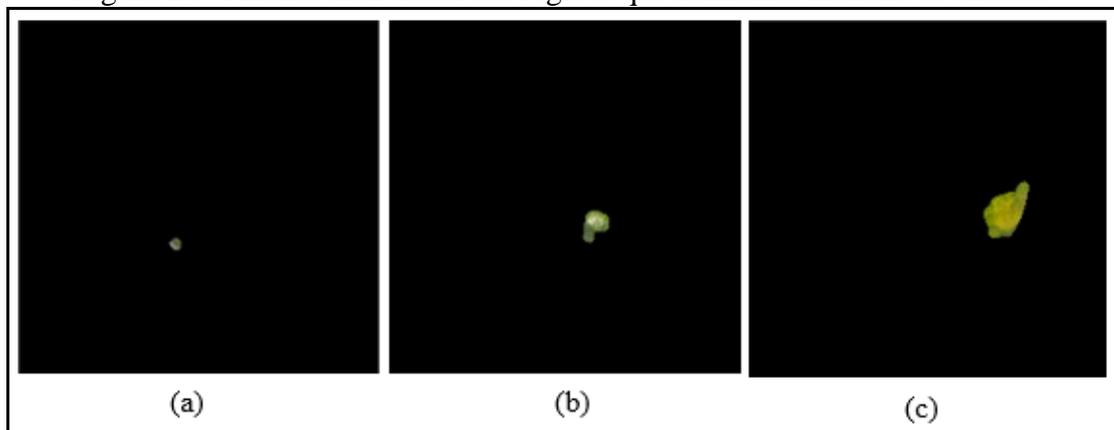
Figura 27 - Regiões a serem analisadas individualmente



Fonte: elaborado pelo autor.

Para determinar as regiões que fazem parte da mancha de encosto, cada região tem sua média analisada individualmente como mostra a Figura 28.

Figura 28 - Análise individual das regiões que formam a mancha de encosto



Fonte: elaborado pelo autor.

O Quadro 13 traz a implementação do método `media_canal_sem_fundo` que converte a imagem para o padrão de cores RGB, como mostra a linha 9. Em seguida tem-se na linha 11 a divisão da imagem em 3 vetores, sendo um para cada canal de cor, utilizando o método `split`. A média de cada canal é calculada pelo método `calc_media`, contabilizando apenas os pixels diferentes de 0. As linhas 13, 14 e 15 mostram a chamada do método `calc_media` para cada um dos vetores obtidos através do método `split`. Os valores resultantes do método `calc_media` serão as médias de cada canal, valores estes utilizados para realizar a validação da mancha de encosto.

Quadro 13 - Método `media_canal_sem_fundo`

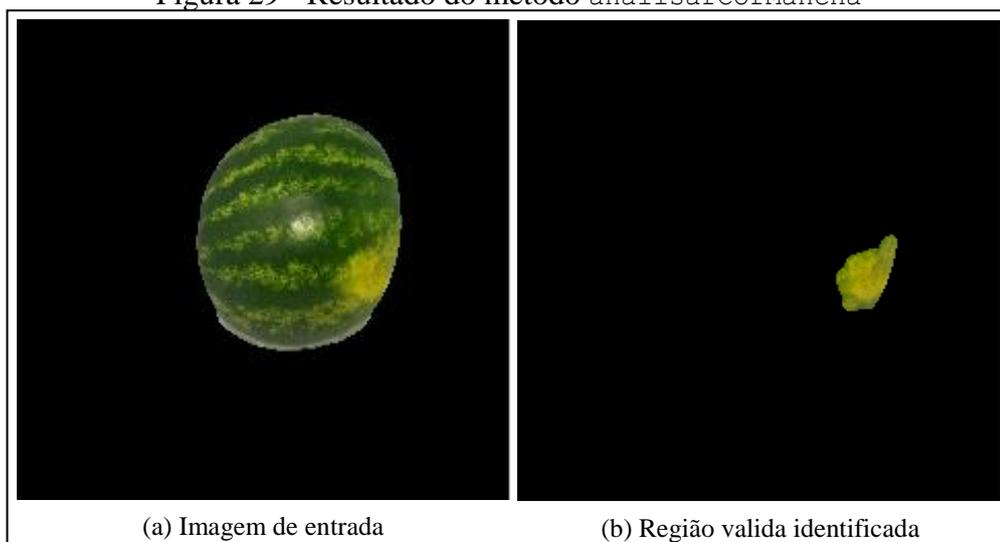
```

1  Mat_<float> Analisador::media_canal_sem_fundo(Mat img) {
2
3  Mat img_media;
4  Mat img_original = img.clone();
5  Mat nova_melancia(Mat_<float>(1, 3));
6  vector<Mat> channels;
7  Scalar media;
8
9  cvtColor(img_original, img_media, CV_BGR2RGB);
10
11 split(img_media, channels);
12
13 nova_melancia.at<float>(0, 0) = calc_media(channels[0]);
14 nova_melancia.at<float>(0, 1) = calc_media(channels[1]);
15 nova_melancia.at<float>(0, 2) = calc_media(channels[2]);
16
17 return nova_melancia;
18
19 }

```

Fonte: elaborado pelo autor.

Posteriormente, é verificado se o valor médio de cada canal tende as tonalidades de amarelo, para então desenhá-lo. O Quadro 12 mostra como foi feita a verificação do valor médio de cada canal, sendo necessário que as imagens possuam um valor médio acima de 130 nos canais vermelho e verde e inferior a 40 no canal azul. A Figura 29 exibe a remoção de regiões inválidas realizada pelo método `analisarCorMancha`.

Figura 29 - Resultado do método `analisarCorMancha`

Fonte: elaborado pelo autor.

Além da verificação da cor da mancha de encosto, é realizada uma validação sobre o seu tamanho em relação a superfície da melancia, conforme é apresentado no Quadro 14. Nas

linhas 10 e 11 é contabilizada a quantidade de pixels referentes a mancha de encosto e a área da melancia. Entre as linhas 17 e 22 é verificado se mancha de encosto pode ser utilizada para classificação da melancia, necessitando ter uma área superior a 5% da superfície da melancia. Tal validação se faz necessária para evitar que o protótipo apresente resultados errôneos baseado na classificação de uma área de tamanho irrelevante.

Quadro 14 - Método `validarMancha`

```

1  bool Analisador::validarMancha() {
2
3  Size size = { 240,240 };
4
5  int totalMancha = 0;
6  int totalMelancia = 0;
7  Mat mancha;
8  Mat img_final;
9
10 cvtColor(melancia.getImgMancha(), mancha, CV_BGR2GRAY);
11 cvtColor(melancia.getImgFinal(), img_final, CV_BGR2GRAY);
12
13 totalMancha = contarPixels(mancha);
14 totalMelancia = contarPixels(img_final);
15
16 if (totalMancha != 0) {
17     if ((totalMelancia * 0.05) > totalMancha) {
18         melancia.setImgMancha(Mat::zeros(size,
19             CV_8UC3));
20         return false;
21     }else{
22         return true;
23     };
24 return false;
25 }

```

Fonte: elaborado pelo autor.

### 3.3.1.3 Classificação

Encerrando o processo da análise da maturação da melancia é executada a classificação. Mediante a utilização de uma árvore de decisão a melancia pode ser classificada em 3 categorias: imaturas, maduras e muito maduras. A melancia é classificada conforme a parte verde ou mancha de encosto, conforme explica-se nas próximas seções.

#### 3.3.1.3.1 Classificação da mancha de encosto

Quando a mancha de encosto é definida como válida pelo método `validarMancha`, os valores da média de cada canal são submetidos a uma série de verificações, procurando

encaixar a melancia em uma das três categorias de classificação. Tem-se no Quadro 15 a exibição das verificações feitas no método `classificacao`, nas quais a melancia é classificada a partir dos valores médios dos canais vermelho e verde, representadas pelas linhas 6 a 14. Devido a não identificação de uma correlação do valor médio do canal azul com o nível maturação da melancia, o valor médio de azul foi descartado.

Quadro 15 - Método `classificacao`

```

1  int Analisador::classificacao(Mat media_sem_mancha,
2                                Mat_<float> media_mancha,
3                                bool mancha_valida) {
4  if (mancha_valida) {
5
6      if (media_mancha.at<float>(0, 0) >= 195.0 &&
7          media_mancha.at<float>(0, 1) >= 180.0) {
8          return 2;
9      } else if (media_mancha.at<float>(0, 0) >= 145 &&
10             media_mancha.at<float>(0, 1) >= 145) {
11         return 1;
12     } else {
13         return 0;
14     }
15 }else {
16 Mat img_nao_relevante = imread("ManchaNaoRelevante.jpg");
17 melancia.setImgMancha(img_nao_relevante);
18
19 knn.treinarKNN();
20 knn.executarKNN(media_sem_mancha);
21 return knn.getSaida().at<float>(0, 0);
22 }
23 }
```

Fonte: elaborado pelo autor.

Quando uma mancha de encosto valida não é identificada, o protótipo efetua a classificação baseado na parte verde da melancia. Para isso, foi elaborado um método de classificação através do algoritmo de aprendizagem de máquina KNN presente na biblioteca OpenCV. A partir de uma base de conhecimento com exemplos rotulados, o KNN compara os valores informados com os presentes em sua base de conhecimento utilizando a distância euclidiana.

Um fator determinante no KNN é o valor K que corresponde ao número de comparações que serão feitas para classificar o novo elemento. A qualidade da base de conhecimento é relevante e, deve conter exemplos bem definidos de todas as categorias de classificação. A base de dados para o desenvolvimento do protótipo foi carregada por meio de

um arquivo .CSV contendo 35 exemplos. A estrutura do arquivo .CSV pode ser visualizada no Quadro 16, onde cada linha representa um exemplar da base. Os três primeiros valores de cada linha são as médias dos canais vermelho, verde e azul respectivamente e o último valor de cada linha é o resultado da classificação feita durante a aquisição de imagens sendo 0 para imatura, 1 para madura e 2 para muito madura.

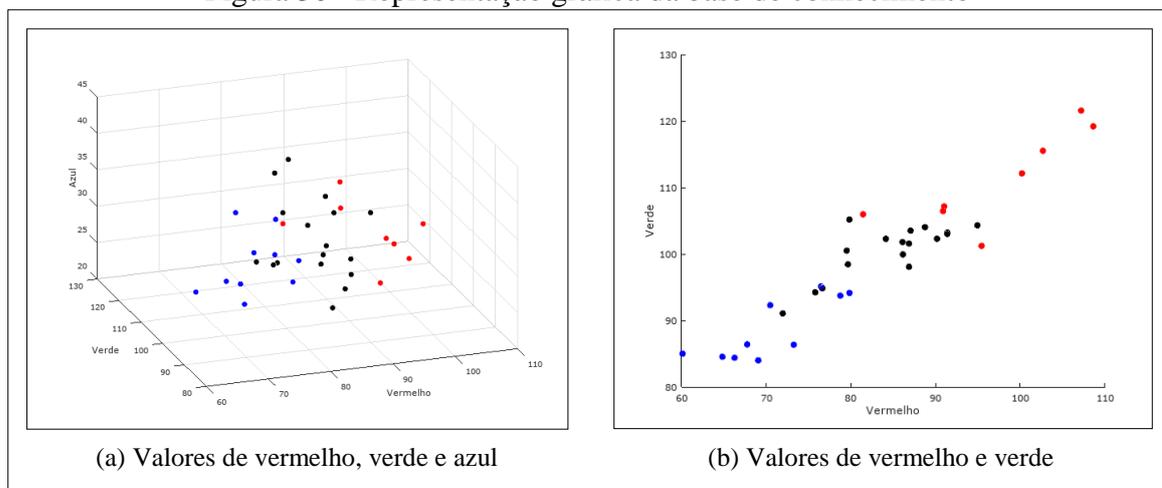
Quadro 16 - Estrutura do arquivo .CSV

66.2777,	84.408,	41.8986,	0
60.1116,	85.0284,	31.4608,	0
78.7535,	93.7545,	28.3337,	0
86.8662,	98.0861,	35.719,	1
86.1439,	99.9658,	37.4906,	1
79.8255,	105.204,	39.7992,	1
100.205,	112.145,	26.6828,	2
102.678,	115.555,	24.6573,	2

Fonte: elaborado pelo autor.

A Figura 30 exibe a base de conhecimento utilizada pelo algoritmo KNN para executar a classificação da parte verde da melancia. A Figura 30a corresponde à disposição dos exemplares de melancia em relação aos valores da média dos três canais RGB. Na Figura 30b é feita a exibição da base dados em uma perspectiva 2D, sem a presença dos valores médios do canal azul, o qual apresenta baixa relação com o nível de maturação. Dessa forma, fica evidente a correlação dos valores médios dos canais verde e vermelho com o nível de maturação da melancia, que quanto maior os seus valores, maior será o nível de maturação da melancia. Os exemplares imaturos, maduros e muito maduras são respectivamente representados pelos pontos de cor azul, preto e vermelho.

Figura 30 - Representação gráfica da base de conhecimento



Fonte: elaborado pelo autor.

Para classificar as melancias de forma correta o KNN precisa primeiramente ser treinado, processo que constrói uma árvore de busca com os dados da base de conhecimento.

No Quadro 17 é possível ver os métodos `treinarKNN` e `executarKNN`, métodos estes que constroem a árvore de busca e efetuam a classificação com o KNN. Observa-se também na linha 10 a definição do valor de `K` com 5, valor que apresentou melhores resultados ao realizar a classificação.

Quadro 17 - Métodos `treinarKNN` e `executarKNN`

```

1 void KNN::treinarKNN(){
2
3     Mat_<float> dados_treino = this-
4     >dados_treino;
5     Mat_<float> respotas = this-
6     >respostas;
7
8     knn->train(dados_treino, ml::ROW_SAMPLE, respotas);
9 }
10
11 void KNN::executarKNN(Mat_<float>
12 novoVizinho){
13     int K = 5;
14     Mat saida;
15     Mat saida_completa;
16     Mat dist;
17
18     knn->findNearest(novoVizinho, K, saida, saida_completa,
19     dist);
20
21     setSaidaCompleta(saida_completa);
22     setSaida(saida);
23     setDist(dist);
24 };
25

```

Fonte: elaborado pelo autor.

A Figura 31 retrata um exemplo de melancia que não possui mancha de encosto que é classificada analisando apenas a parte verde via KNN. O retorno do KNN é um vetor de inteiros com 5 posições. Cada posição é referente a uma das melancias que estão mais próximas dos valores da melancia a ser classificada, sendo que valor de cada posição é correspondente a categoria a qual a melancia pertence. No caso da melancia da Figura 31, o vetor de retorno foi [1, 1, 1, 1, 0], indicando assim que a melancia está madura. Neste caso, a tela de resultados traz no espaço da mancha de encosto a informação de que ela não é relevante.

Figura 31 - Exemplo de melancia sem mancha de encosto

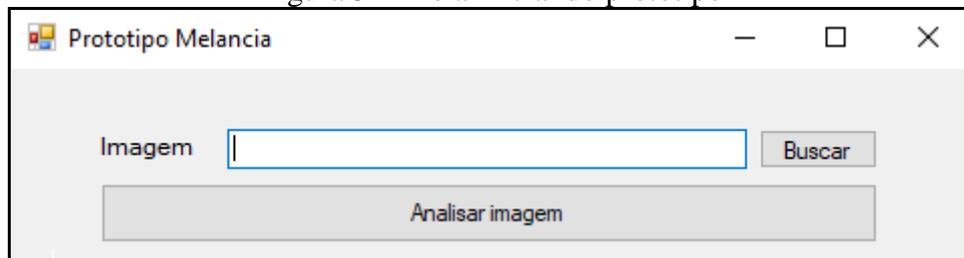


Fonte: elaborado pelo autor.

### 3.3.2 Operacionalidade da implementação

O protótipo foi desenvolvido como uma aplicação desktop. A interface gráfica foi construída para ser simples, requisitando o mínimo de informações ao usuário. A tela inicial ilustrada na Figura 32, apresenta 2 botões. O botão *Buscar*, posicionado na lateral do campo de texto tem a função de selecionar uma imagem para ser analisada. Para dar início a análise da imagem deve-se primeiro fornecer o caminho da imagem e posteriormente clicar sob o botão *Analisar imagem* que carrega o arquivo na memória, iniciando as etapas de processamento de imagens.

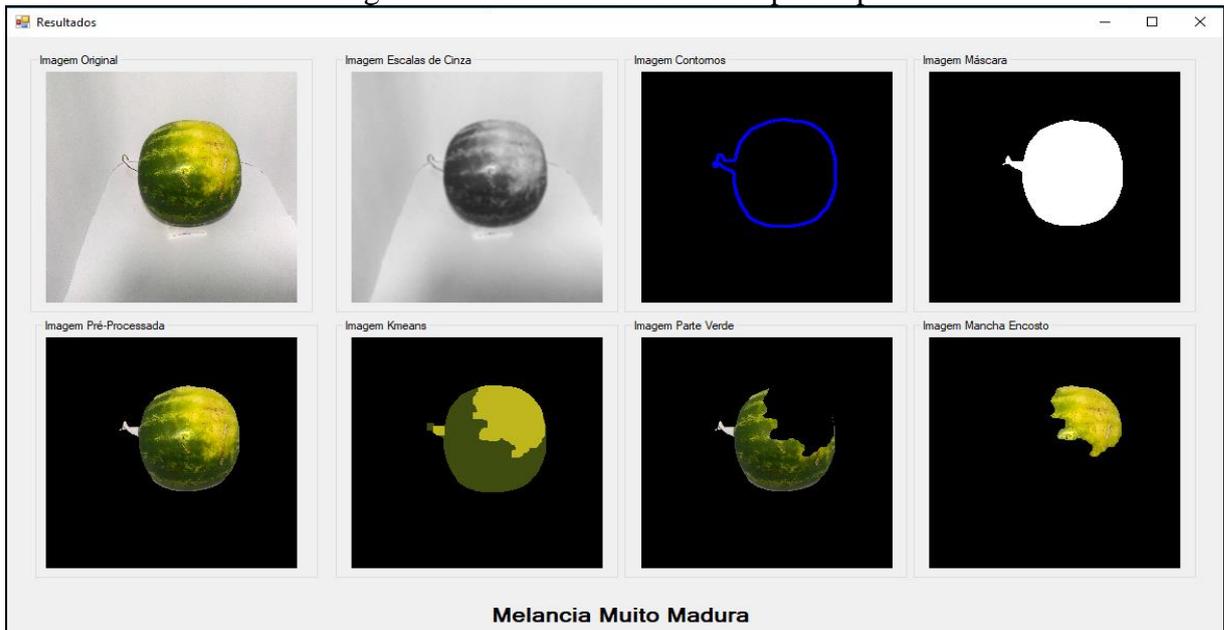
Figura 32 - Tela inicial do protótipo



Fonte: elaborado pelo autor.

Na Figura 33 é possível visualizar a tela de resultados. Nela, estão os resultados de cada etapa do processamento de imagens realizado, mostrando da esquerda para direita e de cima para baixo a imagem original, em escala de cinza, a imagem de contornos, a imagem preenchida, a imagem da melancia em cores sem o plano de fundo, a segmentação realizada pelo K-Means, a parte verde da melancia e a região correspondente a sua mancha de encosto, assim como, os resultados da classificação da melancia.

Figura 33 - Tela de resultados do protótipo



Fonte: elaborado pelo autor.

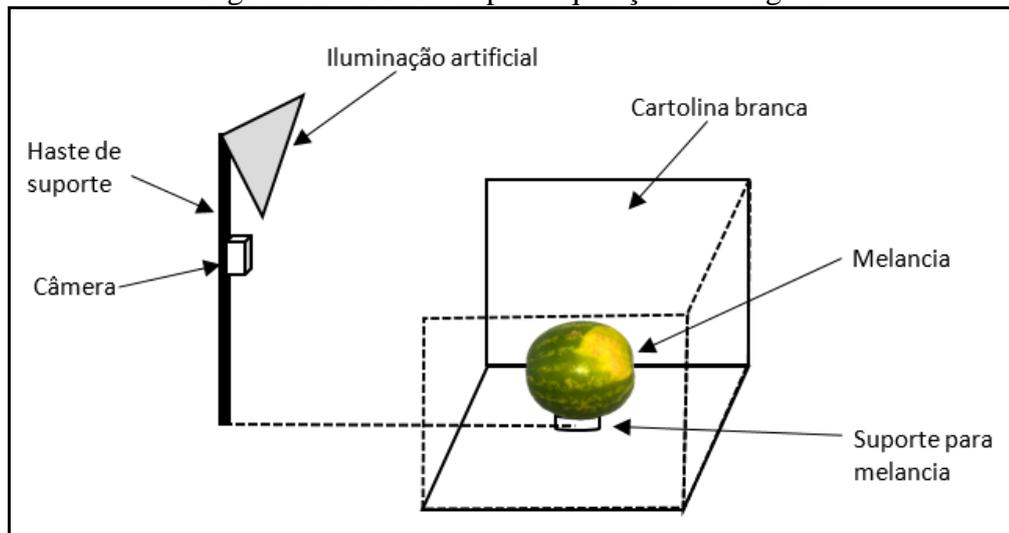
### 3.4 ANÁLISE DOS RESULTADOS

Nesta seção são apresentados os resultados obtidos pelo protótipo. Na seção 3.4.1 é descrito o procedimento de captura/aquisição das imagens que compõem a base de dados. A seção 3.4.2 traz os resultados obtidos apresentando sua taxa de assertividade, principais dificuldades e comparação com os trabalhos correlatos.

#### 3.4.1 Montagem do ambiente, captura das imagens e classificação por degustação das melancias

Na Figura 34 encontra-se o ambiente criado para adquirir as imagens. O ambiente é composto por quatro cartolinas brancas, um suporte para melancia, uma haste de suporte, uma câmera, um sistema de iluminação artificial e a melancia. As quatro cartolinas brancas têm 65 x 50cm e foram dispostas de forma a cobrir as laterais e chão com o propósito de deixar o plano de fundo o mais homogêneo possível. No centro da cartolina inferior foi colocado um suporte para melancia, visto que a presença de deformidades na forma de alguns exemplares dificultava a aquisição da imagem em certos ângulos.

Figura 34 - Ambiente para aquisição de imagens



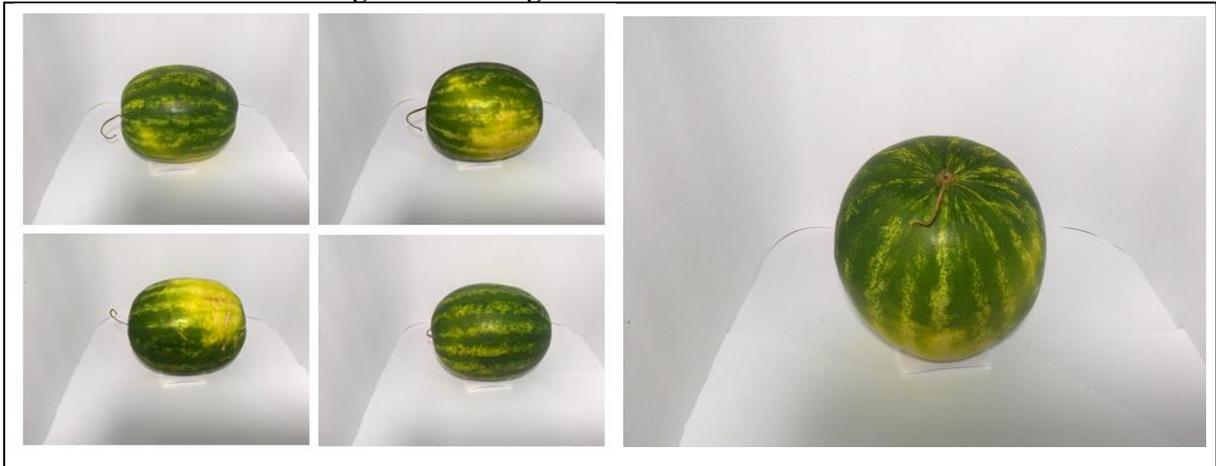
Fonte: elaborado pelo autor.

Na Figura 34 é possível observar uma haste de suporte ajustável posicionada a 65cm do centro do suporte da melancia que em conjunto a uma lâmpada LED compõem o sistema de iluminação artificial. O sistema de iluminação permite ajustar a luz sobre a melancia de modo a minimizar a sombra gerada pelo ambiente externo, garantindo que a iluminação seja equivalente para todas as imagens. A câmera foi presa a haste a 55cm do plano inferior. Como um dos propósitos do protótipo é ter um baixo custo, a câmera utilizada pertence a um *smartphone* configurado para obter as imagens a 5 megapixels.

O método de aquisição de imagens pode ser dividido em três etapas: higienização, a aquisição das imagens externas, correspondentes a superfície da melancia e a aquisição das imagens internas, representado as imagens da polpa da fruta. Para o desenvolvimento do protótipo foram adquiridas 35 melancias, das quais obteve-se 7 imagens de cada exemplar totalizando 245 imagens.

Para evitar que resíduos presos a casca da melancia pudessem influenciar na análise, a primeira etapa do processo de aquisição de imagens foi a higienização de todos os exemplares. A higienização foi feita de forma simplificada, esfregando um pano umedecido sob a superfície da melancia. Depois desta etapa, a melancia é colocada no suporte existente e posteriormente, sofre 3 rotações de 90 graus para que toda a superfície lateral seja fotografada. Em seguida a melancia é posicionada de modo que o caule fique de frente a câmera, adquirindo assim a última imagem da parte externa. A exemplificação das imagens laterais externas e do caule são representadas na Figura 35.

Figura 35 - Imagens laterais externas e caule



Fonte: elaborado pelo autor.

Com a captura das imagens externas concluída, a melancia é dividida ao meio para realizar a classificação *in natura*. Com o auxílio de 6 a 8 voluntários, cada melancia foi classificada por meio da degustação em uma das 3 categorias propostas: imatura, madura e muito madura. Na captura das imagens internas a posição da câmera foi alterada, sendo posicionada a 30cm de distância da melancia. Encerrando o processo, cada metade da melancia é fotografada. A Figura 36 expõem as imagens internas obtidas.

Figura 36 - Imagens internas



Fonte: elaborado pelo autor.

As imagens do caule e da parte interna da melancia não são utilizadas pelo protótipo, porém optou-se em fazer a captura com o intuito de proporcionar outras possibilidades de extensão e disponibilização da base de dados.

#### 3.4.2 Análise de resultados

Para testar o protótipo foram realizados 2 testes, que permitem analisar separadamente a assertividade do KNN utilizado na classificação quando não há mancha de encosto. E da árvore decisão, responsável por classificar a melancia quando a mancha de encosto está presente. Os testes ocorreram somente com as imagens laterais externas da melancia e no caso

do KNN foram desconsideradas as imagens que o protótipo não conseguiu selecionar adequadamente a mancha de encosto, restando das 140 imagens inicialmente disponíveis, 98 imagens aptas das quais 56 foram empregadas nos testes do KNN e 42 nos testes da árvore de decisão. Na Tabela 1, têm-se os resultados da classificação realizada pelo algoritmo KNN, onde 16 das 56 imagens testadas foram classificadas corretamente, obtendo uma taxa de acerto de 28,57%.

Tabela 1 - Resultados dos testes da classificação do KNN

Identificação do exemplar	Foi classificação pela degustação	Classificação pelo Protótipo	Status
M01_F2	Muito Madura	Imatura	FALSO
M01_F3	Muito Madura	Imatura	FALSO
M01_F4	Muito Madura	Imatura	FALSO
M02_F2	Madura	Madura	VERDADEIRO
M03_F4	Muito Madura	Madura	FALSO
M04_F2	Madura	Imatura	FALSO
M04_F4	Madura	Madura	VERDADEIRO
M05_F1	Madura	Madura	VERDADEIRO
M06_F2	Madura	Madura	VERDADEIRO
M07_F4	Madura	Madura	VERDADEIRO
M08_F1	Madura	Imatura	FALSO
M08_F2	Madura	Imatura	FALSO
M08_F3	Madura	Imatura	FALSO
M09_F2	Muito Madura	Madura	FALSO
M09_F4	Muito Madura	Madura	FALSO
M10_F4	Madura	Imatura	FALSO
M11_F1	Imatura	Imatura	VERDADEIRO
M11_F3	Imatura	Madura	FALSO
M11_F4	Imatura	Madura	FALSO
M12_F1	Madura	Madura	VERDADEIRO
M12_F4	Madura	Madura	VERDADEIRO
M13_F1	Madura	Imatura	FALSO
M14_F1	Madura	Imatura	FALSO
M15_F4	Madura	Imatura	FALSO
M16_F1	Muito Madura	Imatura	FALSO
M16_F4	Muito Madura	Imatura	FALSO
M17_F4	Madura	Imatura	FALSO
M18_F3	Muito Madura	Madura	FALSO
M19_F1	Muito Madura	Madura	FALSO
M19_F2	Muito Madura	Imatura	FALSO
M19_F3	Muito Madura	Imatura	FALSO
M20_F1	Muito Madura	Madura	FALSO
M20_F2	Muito Madura	Imatura	FALSO
M20_F3	Muito Madura	Madura	FALSO
M22_F1	Madura	Imatura	FALSO
M22_F2	Madura	Imatura	FALSO
M23_F1	Imatura	Madura	FALSO
M23_F3	Imatura	Madura	FALSO
M24_F4	Madura	Madura	VERDADEIRO

Identificação do exemplar	Foi classificação pela degustação	Classificação pelo Protótipo	Status
M26_F1	Muito Madura	Imatura	FALSO
M26_F2	Muito Madura	Madura	FALSO
M26_F4	Muito Madura	Imatura	FALSO
M27_F4	Imatura	Madura	FALSO
M28_F1	Madura	Imatura	FALSO
M28_F4	Madura	Madura	VERDADEIRO
M29_F1	Madura	Imatura	FALSO
M29_F3	Madura	Madura	VERDADEIRO
M29_F4	Madura	Imatura	FALSO
M30_F4	Imatura	Imatura	VERDADEIRO
M31_F2	Madura	Madura	VERDADEIRO
M32_F3	Imatura	Imatura	VERDADEIRO
M33_F2	Imatura	Imatura	VERDADEIRO
M35_F1	Madura	Madura	VERDADEIRO
M35_F2	Madura	Imatura	FALSO
M35_F3	Madura	Imatura	FALSO
% de acerto			28,57

Fonte: elaborado pelo autor.

A classificação por meio da árvore de decisão apresentou um resultado mais assertivo do que KNN, onde 42,85% dos testes foram corretamente classificados. Os resultados dos testes efetuados para medir a assertividade da árvore de decisão se encontram na Tabela 2.

Tabela 2 - Resultados dos testes de classificação da árvore de decisão

Identificação do exemplar	Foi classificação pela degustação	Classificação pelo Protótipo	Status
M01_F1	Super madura	Madura	FALSO
M02_F3	Madura	Madura	VERDADEIRO
M03_F2	Super madura	Madura	FALSO
M03_F3	Muito madura	Muito Madura	VERDADEIRO
M04_F3	Madura	Imatura	FALSO
M05_F3	Madura	Imatura	FALSO
M05_F4	Madura	Imatura	FALSO
M06_F3	Madura	Madura	VERDADEIRO
M06_F4	Madura	Madura	VERDADEIRO
M07_F2	Madura	Madura	VERDADEIRO
M07_F3	Madura	Madura	VERDADEIRO
M09_F3	Muito madura	Madura	FALSO
M10_F3	Madura	Muito Madura	FALSO
M12_F3	Madura	Madura	VERDADEIRO
M14_F3	Madura	Muito Madura	FALSO
M14_F4	Madura	Madura	VERDADEIRO
M15_F1	Madura	Imatura	FALSO
M15_F2	Madura	Muito Madura	FALSO
M16_F2	Muito madura	Madura	FALSO
M16_F3	Muito madura	Muito Madura	VERDADEIRO
M17_F2	Madura	Imatura	FALSO
M17_F3	Madura	Madura	VERDADEIRO

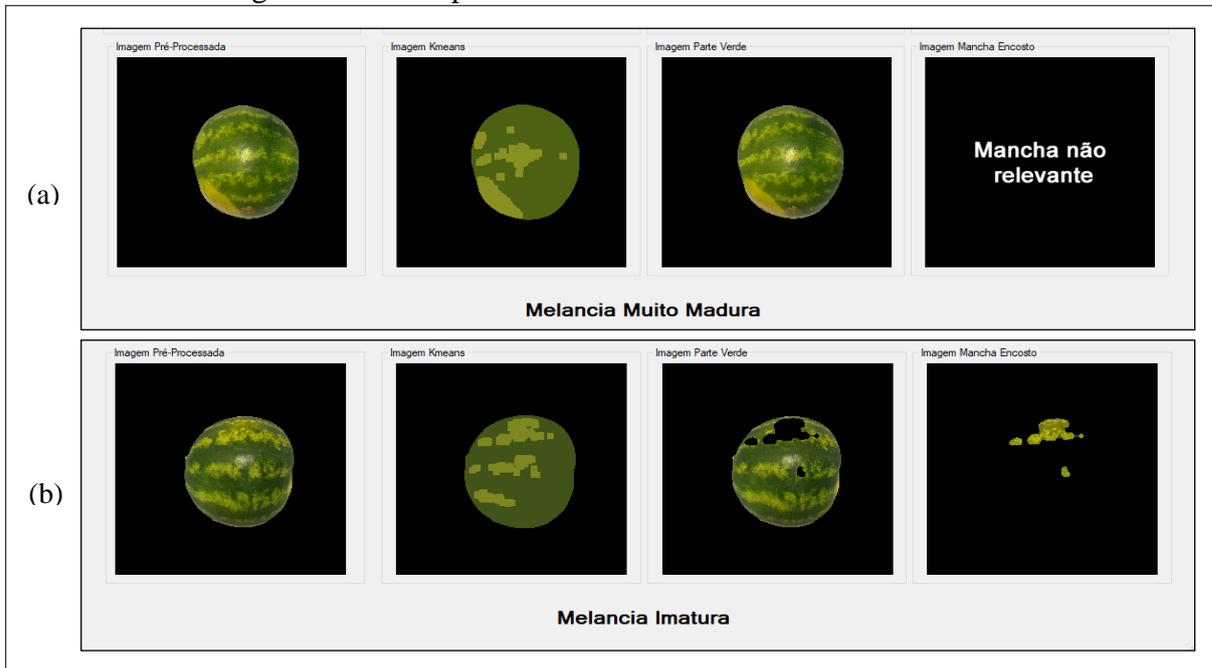
Identificação do exemplar	Foi classificação pela degustação	Classificação pelo Protótipo	Status
M18_F2	Muito madura	Muito madura	VERDADEIRO
M19_F4	Muito madura	Não Acho Mancha	FALSO
M20_F4	Muito madura	Madura	FALSO
M22_F4	Madura	Madura	VERDADEIRO
M23_F4	Imatura	Madura	FALSO
M24_F2	Madura	Madura	VERDADEIRO
M24_F3	Madura	Muito Madura	FALSO
M26_F3	Muito madura	Madura	FALSO
M27_F2	Imatura	Madura	FALSO
M27_F3	Imatura	Madura	FALSO
M28_F2	Madura	Madura	VERDADEIRO
M28_F3	Madura	Muito Madura	FALSO
M29_F2	Madura	Imatura	FALSO
M30_F2	Imatura	Madura	FALSO
M30_F3	Imatura	Imatura	VERDADEIRO
M32_F1	Imatura	Imatura	VERDADEIRO
M32_F2	Imatura	Madura	FALSO
M33_F4	Imatura	Madura	FALSO
M34_F3	Madura	Madura	VERDADEIRO
M34_F4	Madura	Madura	VERDADEIRO
% de acerto			42,85

Fonte: elaborado pelo autor.

A partir dos testes obtidos conclui-se que o protótipo apresentou uma assertividade geral de 34,69%. Um dos principais fatores responsáveis pelo o baixo nível de acerto é a pouca variedade dos exemplares disponíveis para construir da base de conhecimento e definir os valores médios a serem usados na árvore de decisão. No período em que o protótipo foi desenvolvido a melancia não estava em sua época de colheita na região sul, inviabilizando assim a possibilidade da obtenção dos exemplares com um produtor local, o que ocasionaria em uma maior variedade de exemplos. Todos os exemplares necessários para os testes e treinamento do protótipo foram adquiridos em um centro de distribuição de alimentos local, lugar para onde os produtos já vão classificados, dificultando assim principalmente a aquisição de melancias que estejam imaturas. A pouca variedade de exemplares também teve influência sob o método de aprendizagem escolhido, sendo necessário optar por uma técnica simples que pudesse funcionar com poucos exemplares, no caso o KNN.

Outra possível justificativa para o baixo índice de assertos foi a falha do método de segmentação da mancha ao analisar a região extraída pelo K-Means, que conseguiu realizar a segmentação corretamente em 78,58% dos casos, houveram situações como as exemplificadas na Figura 37a em que a mancha de encosto não foi selecionada ou na Figura 37b, onde ela foi identificada uma parte verde.

Figura 37 - Exemplo de mancha de encosto não selecionada



Fonte: elaborado pelo autor.

### 3.4.3 Comparação com trabalhos correlatos

O Quadro 18 apresenta uma comparação entre as principais características dos trabalhos correlatos e do trabalho desenvolvido.

Quadro 18 - Comparativo entre o protótipo desenvolvido e os trabalhos correlatos

trabalhos características	Koc (2007)	Simões e Costa (2003)	Carvalho et al. (2014)	Volkman (2017)
análise da cor dos frutos		X	X	X
análise do volume dos frutos	X			
captura de imagens em ambiente controlado	X	X	X	X
padrão de classificação definido		X	X	
padrão de cor	RGB	RGB	HSL	RGB
análise de forma	X			
método de aprendizagem	-	Redes Neurais	Aprendizagem Indutiva	KNN e Árvore de decisão

Fonte: elaborado pelo autor.

A partir do Quadro 18 é possível observar que o protótipo desenvolvido tem algumas semelhanças com os demais trabalhos correlatos, como a necessidade da aquisição de fotos em um ambiente controlado. O trabalho de Carvalho et al. (2014) se mostrou o único trabalho a não empregar o padrão de cores RGB, comumente utilizado por equipamentos eletrônicos, e sim o padrão HSL. Outro aspecto a se observar é de que o trabalho desenvolvido é o único que realiza a análise de cor e não conta com um padrão de classificação definido. Para encontrar um padrão de características a serem analisadas foi feito contato com o setor de agricultura da cidade de Pomerode, que desconhece a existência de um padrão e no caso da

mancha de encosto, acredita que uma classificação específica não exista, pelo dela ser uma característica do manejo da fruta e não de seu estado maturação. Portanto, os padrões de classificação com os quais a mancha de encosto e a parte verde da melancia foram categorizados tiveram de ser criados baseando-se em testes realizados durante o desenvolvimento.

## 4 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um protótipo para a plataforma desktop que por meio de métodos não invasivos partindo da imagem de uma melancia fornecida pelo usuário realiza a classificação da mesma empregando técnicas de aprendizagem de máquina para analisar a coloração da casca. Para tornar a utilização das técnicas de aprendizagem de máquina viável, fez-se necessário a criação de uma base de dados com imagens de melancia extraídas a partir de ambiente controlado criado especificamente para este propósito.

O protótipo foi desenvolvido na IDE Visual Studio 2017 com a linguagem de programação C++. A interface foi construída com a biblioteca Windows Forms. Para aplicação dos métodos referentes ao processamento de imagens e técnicas de aprendizagem de máquina foi utilizada a biblioteca OpenCV. Os dados necessários para treinar os algoritmos de aprendizagem foram carregados através de um arquivo CSV.

Entre os principais obstáculos encontrados ao realizar este trabalho pode-se mencionar a aquisição de exemplares variados de melancia. Ter um conjunto de exemplos variados com exemplares bem definidos em cada uma das categorias em que o protótipo realiza a classificação é de suma importância, possibilitando assim um treinamento mais robusto por parte dos algoritmos de aprendizagem de máquina. Como a aquisição dos exemplares de melancia não foi feita diretamente com o produtor e sim através de um centro de distribuição de alimentos, exemplos melancias muito imaturas ou extremamente maduras não se fazem presentes na base de dados. Outro empecilho para realização deste trabalho foi a definição dos níveis de maturação das características analisadas, os quais tiveram de ser criados sem a ajuda de um especialista da área, através da análise dos valores médios dos níveis de coloração da casca da melancia.

Apesar de ter atingido todos os objetivos, o protótipo não teve um desempenho satisfatório, tendo uma taxa de assertividade geral de apenas 34,69%. Analisados separadamente o KNN e a árvore de decisão tiveram uma taxa de acerto de 28,57% e 42,85% respectivamente, níveis muito abaixo do ideal.

Por fim, conclui-se que para tornar-se efetivo, o protótipo necessita da adição de exemplares mais variados a sua base de dados com intuito de refinar os níveis de maturação criados além de possibilitar a utilização de métodos de aprendizagem mais eficazes como por exemplo, uma rede neural artificial. Apesar não ter tido resultado esperado, o protótipo pode servir de base para futuros trabalhos relacionados análise do nível de maturação frutas.

#### 4.1 EXTENSÕES

Para o protótipo desenvolvido propõem-se as extensões abaixo:

- a) adicionar uma maior variedade de exemplares a base de dados com melancias imaturas e muito maduras organizando-os em um banco de dados relacional;
- b) melhorar a técnica de detecção da mancha de encosto evitando que a partes em verde claro possam ser identificadas como sendo a mancha de encosto;
- c) substituir o método de aprendizagem KNN por uma técnica mais robusta, como por exemplo uma rede neural artificial;
- d) aperfeiçoar a segmentação da melancia de modo que as imagens não precisem mais ser capturadas em um ambiente controlado;
- e) transformar o protótipo em um App para dispositivos móveis;
- f) realizar o tratamento para remoção do reflexo das imagens.

## REFERÊNCIAS

- ALMEIDA, Domingos. P. F. **Melancia**. Porto, 2003. Disponível em: <<http://dalmeida.com/hortnet/Melancia.pdf>> Acesso em: 12 Novembro. 2017.
- ALMEIDA, Roberto C.; FACEROLI, Silvana T. Análise comparativa das técnicas KNN e rede neural MLP na classificação de padrões mioelétricos. In: CONGRESSO BRASILEIRO DE ENGENHARIA BIOMÉDICA, 24., 2014, Uberlândia-MG. **Anais...** Uberlândia-MG: UFU, 2014. p. 685-688.
- ATLAS SOCIOECONOMICO. **Atlas Socioeconômico**: Rio Grande do Sul. Porto Alegre, 2015. Disponível em: <<http://www.atlassocioeconomico.rs.gov.br/melancia-e-melao>>. Acesso em: 20 nov. 2017.
- BUAINAIN, Antônio M.; BATALHA, Mário O. **Cadeia Produtiva de Frutas**. 7. ed. Brasília: MAPA; SPA; IICA. 2007.
- BUANI, Bruna E. Z. **Aplicação da logica Fuzzy KNN e análises estatísticas para seleção de características e classificação de abelhas**. 2010. 135 f. Tese (Mestrado em Engenharia de Computação e Sistemas Digitais) – Pós-Graduação em engenharia de computação e sistemas digitais, Escola Politécnica da Universidade de São Paulo.
- CARVALHO, Joelson N. et al. Classificação automática de frutas por análise de imagem: o caso da manga Tommy Atkins. In: WORKSHOP DE TECNOLOGIA ADAPTATIVA – WTA, 8., 2014, São Paulo. **Anais...** São Paulo: Escola Politécnica da Universidade de São Paulo, 2014. p. 77-91.
- CONFEDERAÇÃO DA AGRICULTURA E PECUÁRIA DO BRASIL. Fruticultura. **Balanco 2016 Perspectivas 2017**, [S.l.], p. 93-97, jul. 2016.
- COSTA, Nivaldo. D; LEITE, Wêydjane M. Potencial agrícola do solo para o cultivo da melancia. In: CURSO DE MANEJO E CONSERVAÇÃO DO SOLO E DA ÁGUA, 2007, Barreiras. **Palestras...** Barreiras: MAPA; SFA-BA: Embrapa Semi-Árido; Embrapa Solos - UEP Recife, 2007. 1 CD-ROM.
- DHANACHANDRA, Nameirakpam; MANGLEM, Khumanthem; CHANU, Yambem J. Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. **Procedia Computer Science**, Manipur, v. 54, p.764-771, 2015.
- FACON, Jacques. **Processamento e Análise de Imagens**. Curitiba: EBAI, 1993.
- GAMA, João. **Árvores de Decisão**, Porto, 2002. Disponível em [http://www.dcc.fc.up.pt/~ines/aulas/MIM/arvores\\_de\\_decisao.pdf](http://www.dcc.fc.up.pt/~ines/aulas/MIM/arvores_de_decisao.pdf)>, Acesso em: 21 nov. 2017.
- GAMA, João M. P. **Combining classification algorithms**. 1999. 195 f. Tese (Doutorado) – Departamento de Ciência de Computadores, Universidade do Porto, Lisboa.
- GONZALEZ, Rafael C.; WOODS, Richard E. **Digital image processing**. 3rd ed. Upper Saddle River: Pearson Prentice Hall, 2008.
- HALMENSCHLAGER, Carine. **Um algoritmo para indução de árvores e regras de decisão**. 2002. 112 f. Tese (Mestrado em ciências da computação) - Pós-Graduação em ciências da computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- INCRIVEL, **Como escolher uma melancia**, [S.l.], 2016, Disponível em: <<https://incrivel.club/inspiracao-dicas/como-escolher-uma-melancia-120610/>> Acesso em: 26 nov. 2017.

- KOC, Ali B. Determination of watermelon volume using ellipsoid approximation and image processing. **Postharvest Biology and Technology**, [S.l.], v. 45, n. 3, p. 366-371, set. 2007.
- LEEMANS, Vincent; MAGEIN, Hugo; DESTAIN, Marie F. Defects segmentation on 'Golden Delicious' apples by using colour machine vision. **Computers And Electronics In Agriculture**, [S.l.], v. 20, n. 2, p. 117-130, jul. 1998.
- \_\_\_\_\_. Defect segmentation on 'Jonagold' apples using colour vision and a Bayesian classification method. **Computers And Electronics In Agriculture**, [S.l.], v. 23, n. 1, p. 43-53, jun. 1999.
- MACEDO, Maria J. H. et al. Analysis of the standardized precipitation index for the Paraíba state, Brazil. **Ambiente e Agua - An Interdisciplinary Journal Of Applied Science**, [S.l.], v. 5, n. 1, p.204-214, 30 abr. 2010.
- MATHIAS, João. **Revista Globo Rural**, [S.l.], 2016. Disponível em: <<http://revistagloborural.globo.com/vida-na-fazenda/gr-responde/noticia/2016/09/qual-e-hora-de-colher-melancia.html>>. Acesso em: 29 nov. 2017.
- MEDEIROS, Roberto D.; ALVES, Admar B. **Informações Técnicas para o Cultivo de Melancia em Roraima**. Boa Vista: Embrapa, 2016.
- MORI, Emilia E. M. **Suco de Melancia [Citrullus lanatus (Tunberg) Matsumura and Nakai]: processamento, formulação, caracterização física, química, microbiológica e aceitabilidade**. 1996. 131 f. Tese (Doutorado em engenharia de alimentos) - Pós-Graduação em engenharia de alimentos, Universidade Estadual de Campinas, Campinas.
- OPENCV, **Understanding k-Nearest Neighbour**. [S.l.] 2014. Disponível em:< [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_ml/py\\_knn/py\\_knn\\_understanding/py\\_knn\\_understanding.html#knn-in-opencv](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_ml/py_knn/py_knn_understanding/py_knn_understanding.html#knn-in-opencv)>. Acesso em: 01 nov. 2017.
- PROJETODEHORTA, **Um Projeto de Horta**, [S.l.], 2014. Disponível em: < <http://projetodehorta.blogspot.com.br/2014/07/como-escolher-melancias.html>>. Acesso em: 27 nov. 2017.
- RESENDE, Geraldo M.; COSTA, Nivaldo D. Características produtivas da melancia em diferentes espaçamentos de plantio. **Horticultura Brasileira**, Brasília, v. 21, n. 4, p. 695-698, outubro-dezembro. 2003.
- RODRIGUES, Jonathan C.; LAVOIER FILHO, João M. Análise de qualidade de frutas por imagens multiespectrais. **Rev. Científica Eletrônica UNISEB**, Ribeirão Preto, v. 1, n. 1, p. 91-110, jan./jun. 2013.
- SANTOS, Fernando C. **Variações do Método kNN e suas Aplicações na Classificação Automática de Textos**. 2009. 96 f. Tese (Mestrado em Ciências da Computação) - Pós-Graduação em Ciências da Computação, Universidade Federal de Goiás, Goiânia.
- SHIBA, Marcelo H. et al. Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 12., 2005, Goiânia. **Anais...** Goiânia: INPE, 2005. p. 4319-4326.
- SIMÕES, Alexandre S. **Segmentação de imagens por classificação de cores: uma abordagem neural**. 2000. 171 f. Tese (Mestrado em engenharia elétrica) - Pós-Graduação em engenharia elétrica, Escola Politécnica da Universidade de São Paulo, São Paulo.

SIMÕES, Alexandre S.; COSTA, Anna H. R. Classificação de laranjas baseada em padrões visuais. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI), 6., 2003, Bauru. **Anais...** Bauru: Sociedade Brasileira de Automática, p. 77-81, set. 2003.

SYAZWAN, N. A.; RIZAM, M. S. B. S.; NOORITAWATI, M. T. Categorization Of Watermelon Maturity Level Based On Rind Features. **Procedia Engineering**, [S.l.], v. 41, p.1398-1404, 2012.

## APÊNDICE A – Detalhamento dos casos de uso

Esta seção é responsável por detalhar os casos de uso. No Quadro 18 acontece o detalhamento do UC01 - Analisar imagem, caso de uso principal referente a análise do nível de maturação da melancia.

Quadro 19 - Caso de uso UC01 Analisar imagem

<b>UC01 – Analisar imagem:</b> realiza a análise do nível de maturação da melancia	
<b>Descrição</b>	Permitir que o usuário realize a análise do nível de maturação da melancia.
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Usuário deve possuir a imagem da melancia no computador
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Usuário seleciona a opção <i>Buscar</i>;</li> <li>2. Vai para o caso de uso UC02 - Carregar imagem;</li> <li>3. Usuário clica no botão <i>Analisar imagem</i>;</li> <li>4. Sistema imprime na tela a imagem de todas as etapas;</li> <li>5. Sistema imprime na tela o resultado da classificação;</li> <li>6. Fim do caso de uso.</li> </ol>
<b>Fluxo de exceção</b>	<p>E01 – Arquivo inválido</p> <p>No passo 3 do fluxo principal, caso o usuário selecione um arquivo inválido.</p> <ol style="list-style-type: none"> <li>1. Usuário seleciona um arquivo com extensão diferente de *.jpg, *.png ou *.bmp ou deixa o campo <i>Imagem em branco</i> e pressiona o botão <i>Analisar imagem</i>;</li> <li>2. Sistema exibe a mensagem “Arquivo inválido”;</li> <li>3. Retorna para o passo 1 do Fluxo principal.</li> </ol>
<b>Pós-condições</b>	Melancia é classificada pelo sistema

Fonte: elaborado pelo autor.

Por fim, o Quadro 20 apresenta as especificações do UC02 - Carregar imagem, processo pelo qual o usuário seleciona a imagem a ser analisada.

Quadro 20 - Caso de uso UC02 Carregar imagem

<b>UC02 – Carregar imagem:</b> carrega a imagem da melancia no sistema	
<b>Descrição</b>	Permitir que o usuário informe uma imagem ao protótipo.
<b>Ator</b>	Usuário
<b>Pré-condições</b>	Usuário deve possuir a imagem da melancia no computador
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. Sistema exibe a janela para seleção de arquivos;</li> <li>2. Usuário seleciona a imagem desejada e clica em <i>Abrir</i>;</li> <li>3. Sistema fecha a janela de seleção e exibe o caminho da imagem no campo <i>Imagem</i>;</li> <li>4. Fim do caso de uso.</li> </ol>
<b>Pós-condições</b>	Disponibiliza caminho para carregar a imagem da melancia

Fonte: elaborado pelo autor.