

DETECÇÃO DO BOLOR NA LARANJA E MENSURAÇÃO DA ÁREA AFETADA PELO FUNGO

Jaciara Domingos Elisiário, Paulinelly de Sousa Oliveira, Sílvio Nascimento Garcia

Francisco Heider Willy dos Santos

Instituto Federal de Minas Gerais – Campus Bambuí (IFMG)

**Departamento de Engenharia e Computação – Bacharelado em Engenharia de
Computação**

RESUMO

Atualmente, o Brasil ocupa posição de destaque na produção de laranjas e também exportação do suco desta fruta. Para que essa realidade seja mantida e melhorada ainda mais, uma das preocupações dos produtores é colher e separar os melhores frutos para comercialização. Um dos problemas encontrados para esta tarefa é a presença do bolor nas laranjas. Em uma linha de produção, para que se tenha frutos sadios e com grande teor vitamínico, deve-se conseguir identificar de forma rápida e precisa a presença do bolor, para que esta fruta afetada seja retirada das proximidades das demais. Portanto, esse trabalho foi proposto para criar um *software* capaz de identificar automaticamente, através do processamento digital de imagens, a presença do fungo na fruta e também mensurar a área afetada pelo bolor, o que servirá para classificar os fungos e possibilitar estudos sobre as possíveis modificações no processo produtivo: alterações na logística empregada (armazenamento, transporte), manipulação.

Palavras-chave: Laranjas. Processamento digital de imagens. Detecção de bolor. *OpenCV*.

1 INTRODUÇÃO

O Brasil está entre os maiores produtores de laranja, sendo responsável por 30% da produção mundial da fruta e 60% da produção de suco da laranja. (MINISTÉRIO DA AGRICULTURA, 2016).

Cerca de 50% da produção mundial de laranja e 80% da brasileira resultam em sucos industrializados. O principal comprador da bebida brasileira é a União

Europeia que aumenta significativamente o percentual de importação anualmente. A maior parte das importações mundiais, 85%, é absorvida por apenas três mercados: Estados Unidos, União Europeia e Canadá. Da laranja, além do suco, são extraídos óleos essenciais e líquidos aromáticos. O bagaço de citros, com alto teor energético, é um subproduto industrial de expressivo valor econômico, para alimentação animal, sobretudo para ruminantes e, em especial, a vaca de leite. (MINISTÉRIO DA AGRICULTURA, 2016).

Em uma linha de produção, as laranjas que chegam para serem processadas, são classificadas e separadas para comercialização. Em um caso específico, quando a laranja possui bolor, a inspeção é bem visual: depende de alguém a constatação da presença do fungo e a exclusão do fruto contaminado do contato com as frutas sadias, do contrário, as demais frutas também serão contaminadas e sofrerão com o mesmo problema, perdendo-se assim uma quantidade considerável da produção.

O bolor é um fungo que geralmente aparece pós-colheita, por isso, não faz sentido se pensar em uma abordagem com defensivos agrícolas. As causas de seu aparecimento estão relacionadas a transporte inadequado, condicionamento / armazenamento incorretos, temperaturas fora do ideal.

Portanto, para minimizar as perdas é importante que a identificação seja feita em tempo hábil. Sendo um trabalho manual, pode ocorrer falhas durante a separação. Devido a tudo isso, o presente artigo surgiu.

Este trabalho tem por objetivo o desenvolvimento de um *software* para detecção do bolor em uma linha de produção de laranjas para a identificação das impróprias para o consumo. Adicionalmente, calcular a área ocupada pelo bolor, baseado na densidade de pixels das imagens.

Para isso, utilizou-se do processamento digital de imagens, onde automaticamente, através de biblioteca própria, o *software* corretamente implementado e ajustado é capaz de detectar o fungo.

2 MATERIAIS

Os materiais para a realização deste trabalho foram:

- A linguagem de programação escolhida foi C++;
- O ambiente de desenvolvimento *CodeBlocks*;
- A biblioteca de processamento de imagens *OpenCV*;

- Imagens variadas de laranjas com o bolor;

Para algumas imagens foi necessário um pré-processamento para remoção do fundo (transformação da cor de fundo para preto), para estes casos utilizou-se um software editor de imagens.

Para realizar o processamento das imagens, a biblioteca *OpenCV* foi empregada, por se tratar de uma biblioteca livre com várias funções já implementadas e testadas.

Optou-se pela linguagem C++ devido ao fato da equipe de desenvolvimento estar mais familiarizada com a mesma. Por consequência da escolha da linguagem, optou-se pelo ambiente de desenvolvimento *CodeBlocks*, que é gratuito e possui uma boa integração com o *OpenCV*.

3 METODOLOGIA

Foram trabalhadas duas abordagens: a primeira, considerando as imagens no tipo RGB e a segunda com as imagens em formato HSI.

Quando a imagem está em RGB são consideradas três componentes de cores primárias: vermelho (R, do inglês, *red*), verde (G, do inglês, *green*) e azul (B, do inglês, *blue*). A combinação dessas cores é capaz de gerar o amarelo, o magenta e o ciano.

No formato HSI (ou HSV), têm-se um modelo de cores que determina a matiz (H, do inglês *Hue*), a saturação (S, do inglês *saturation*) e a intensidade / brilho (I, do inglês *intensity* ou *value*). A matiz é a componente que determina a componente predominante do aspecto de cores, ou seja, se a cor é azul, vermelho, etc. A saturação determina a profundidade da cor, ou quão pura a cor é, se ela é mais esmaecida ou mais intensa. Já a intensidade determina o nível de claridade / brilho da imagem.

3.1 Imagem RGB

Na abordagem com a imagem no formato RGB, foram feitos os seguintes métodos:

Primeiramente, fez-se uma conversão da imagem colorida para escala de cinza, para reduzir a complexidade das operações que seriam realizadas. A função para essa conversão é a *cvtColor()*.

Em seguida, uma limiarização foi realizada. Limiarizar é o ato de separar da imagem aquilo que é relevante do que não é. A limiarização foi feita com o método *threshold()* com limiar mínimo de 120 e máximo de 250.

Com um elemento estruturante em formato de elipse, fez-se uma dilatação com a função *dilate()*. Continuando, uma transformada de distância com a função *distanceTransform()* e uma normalização pela função *normalize()* foram realizadas.

Um fechamento (dilatação seguida de erosão), uma erosão (método *erode()*) e uma abertura (erosão seguida de dilatação) foram realizadas.

Nova limiarização foi feita, agora com o *threshold* binário, com os limiares .01 como mínimo e 1. como máximo.

Finalizando, utilizou-se a função *findContours()* para detectar os contornos da imagem, para em seguida, através da contagem desses contornos a área fosse encontrada.

O cálculo da área levou em consideração a correspondência de que 1 pixel equivale a 0.026458 cm. Como essa equivalência é linear, multiplicou-se os contornos pelo quadrado da equivalência (contornos multiplicados por 0.026458 ao quadrado).

3.2 Imagem HSI

Com a imagem em HSI, o intuito era escolher um intervalo de cores característico do bolor para que sua área fosse extraída, então o importante era descobrir qual a cor predominante para do bolor para separá-lo do restante da laranja.

Essa abordagem foi desenvolvida utilizando, primeiramente, a conversão de RGB para HSI. Em seguida foram determinados os limiares inferior e superior do *threshold* na função *inRange()*.

Assim obteve-se as regiões com o a área do bolor, porém com muitos ruídos. Para tratar os ruídos presentes na imagens realizamos um fechamento e em seguida uma abertura na imagem.

Após terminada a remoção dos ruídos, realizamos o cálculo de área com a mesma metodologia aplicada no tratamento RGB.

4 RESULTADOS E DISCUSSÕES

4.1 Imagem RGB

Como pode ser verificado na metodologia, os passos para a detecção e obtenção da área afetada pelo bolor com a imagem em RGB foi um pouco mais trabalhosa. Onde mais passos foram necessários para conseguir o objetivo.



Figura 1 - Imagem original



Figura 2 - Imagem após o pré-processamento

Após a aquisição da imagem, foi necessário a remoção do fundo para facilitar a identificação das regiões relevantes da laranja.

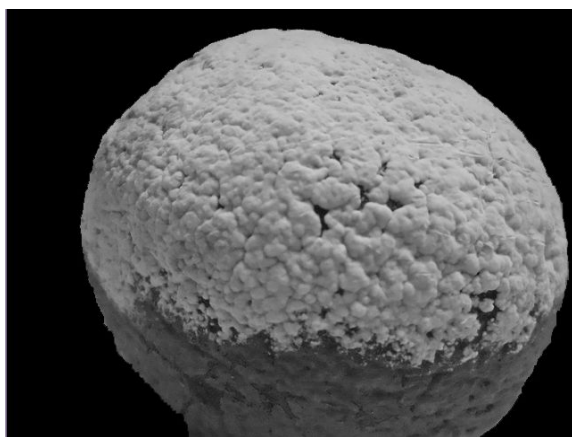


Figura 3.a - Imagem em escala de cinza

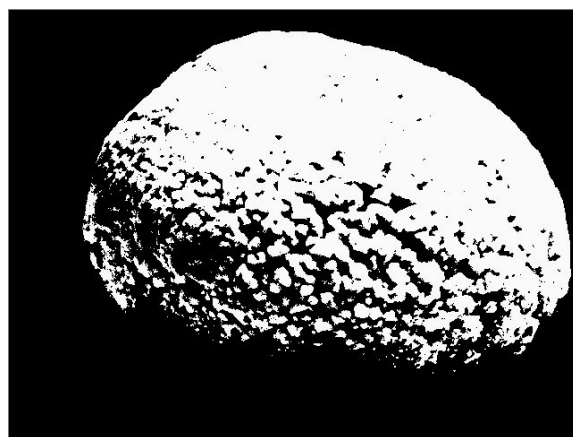


Figura 3.b - Primeiro *threshold* aplicado

Após a conversão para escala de cinza, realizou-se o primeiro *threshold*. Como pode ser verificado na Figura 3.b, o bolor não possui uma textura homogênea e com a limiarização

ficam algumas regiões de bolor desconexas do restante. Diante disso, realizamos a abertura e fechamento na imagem e obtivemos o resultado apresentado na Figura 4.

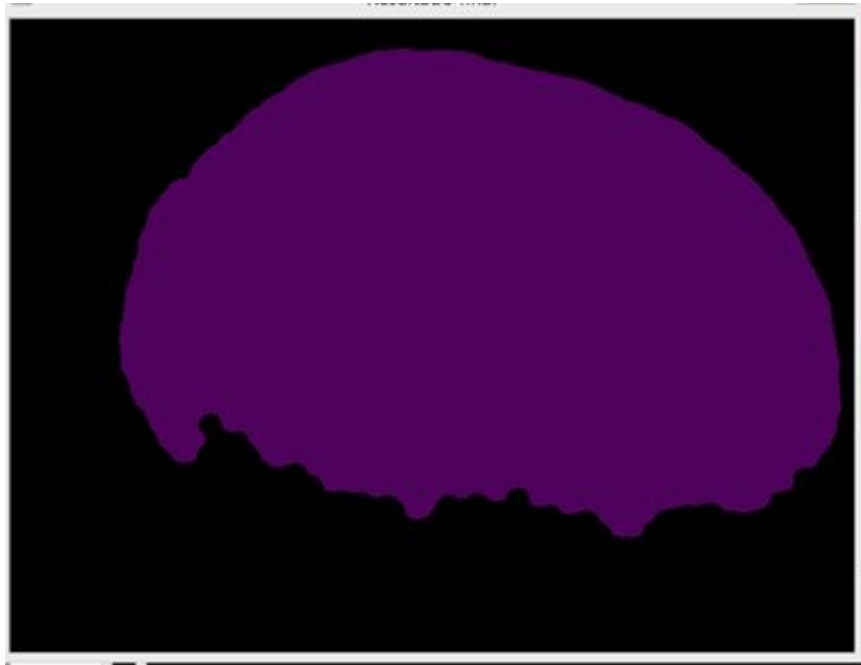


Figura 4 - Resultado a abertura e fechamento.

No tratamento RGB obtivemos uma área de 104,72 cm², para realizar os cálculos de área foi utilizada a aproximação de de 1 pixel = 0.026458 cm.

O tratamento por HSI gerou os resultados semelhantes aos resultados obtidos pelo método RGB, porém, como dito anteriormente, com menos passos. A Figura 5.a apresenta a imagem original, a Figura 5.b apresenta a imagem convertida em HSI.

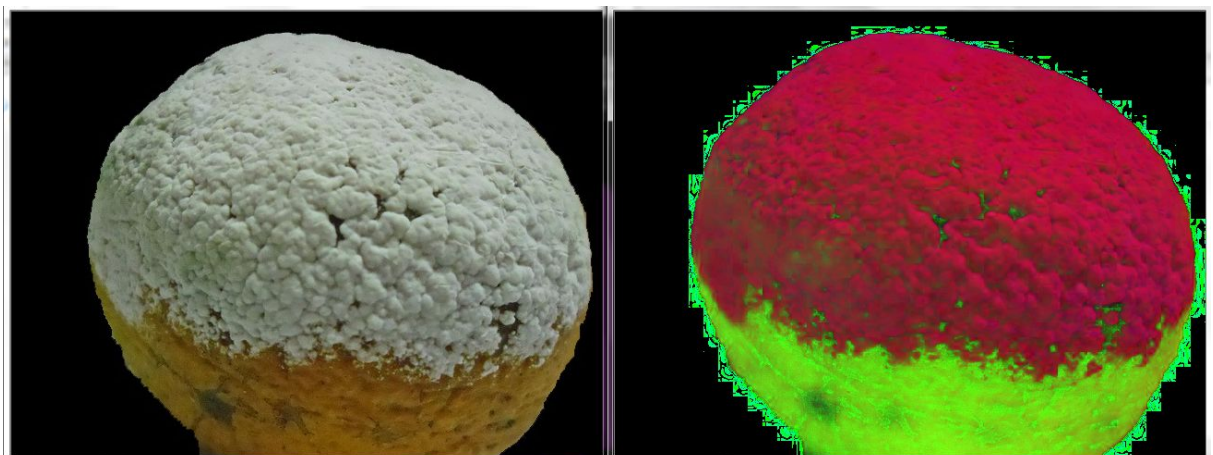


Figura 5.a (esquerda) - Imagem original. Figura 5.b (direita) - Imagem no formato HSI.

A Figura 6.a apresenta o resultado obtido após a execução do *inRange()*, e a Figura 6.b mostra o resultado após o fechamento e a abertura.

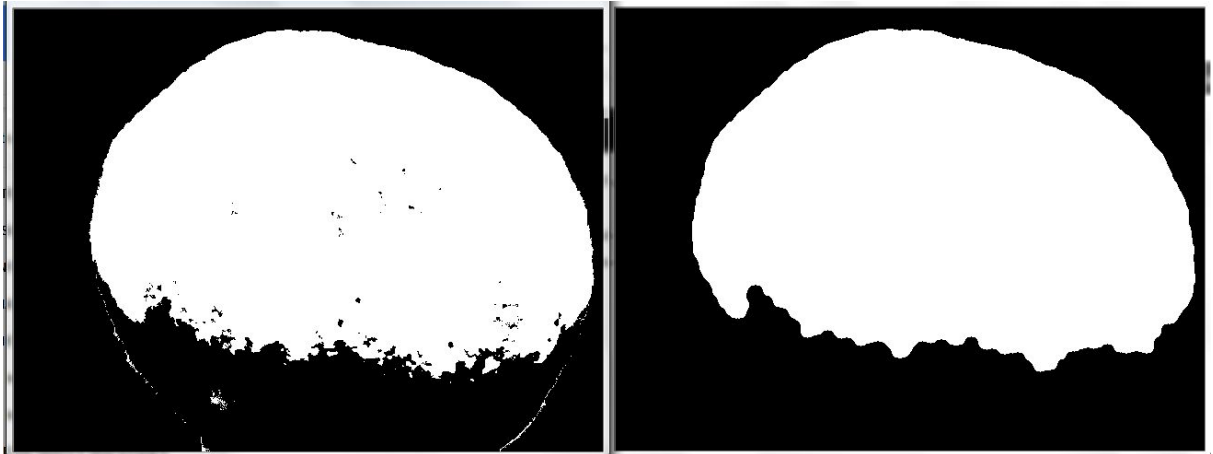


Figura 6.a (esquerda) - Resultado do *Threshold*. Figura 6.b (direita) - Remoção de ruídos.

Por fim foi determinada a área do bolor utilizando a aproximação de 1 pixel = 0.026458 cm. Na imagem avaliada obtivemos uma área de 105,45 cm².

A Figura 7 apresenta os resultados obtidos utilizando outra imagem de avaliação, através da qual obtivemos áreas de valores $6,54\text{cm}^2$ e $0,33\text{cm}^2$.

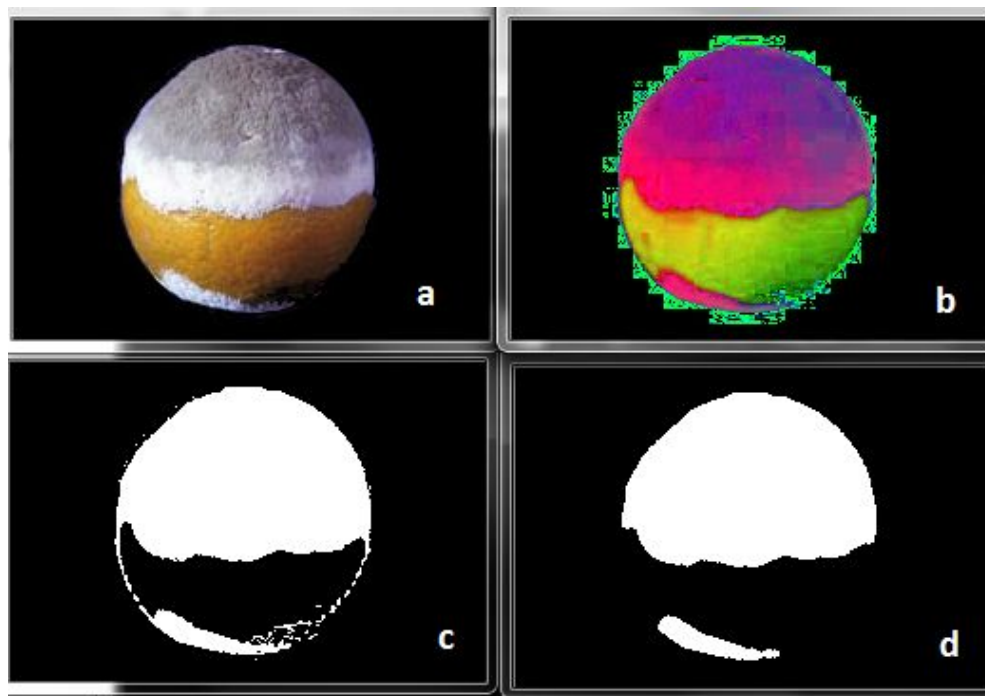


Figura 7: a- Imagem original; b- Imagem HSI; c- Resultado do *Threshold*; d- Remoção de ruídos.

A Figura 8 apresenta o resultado do último teste, onde obtivemos duas áreas, sendo a maior de $58,35\text{ cm}^2$ e a menor de $0,34\text{ cm}^2$.

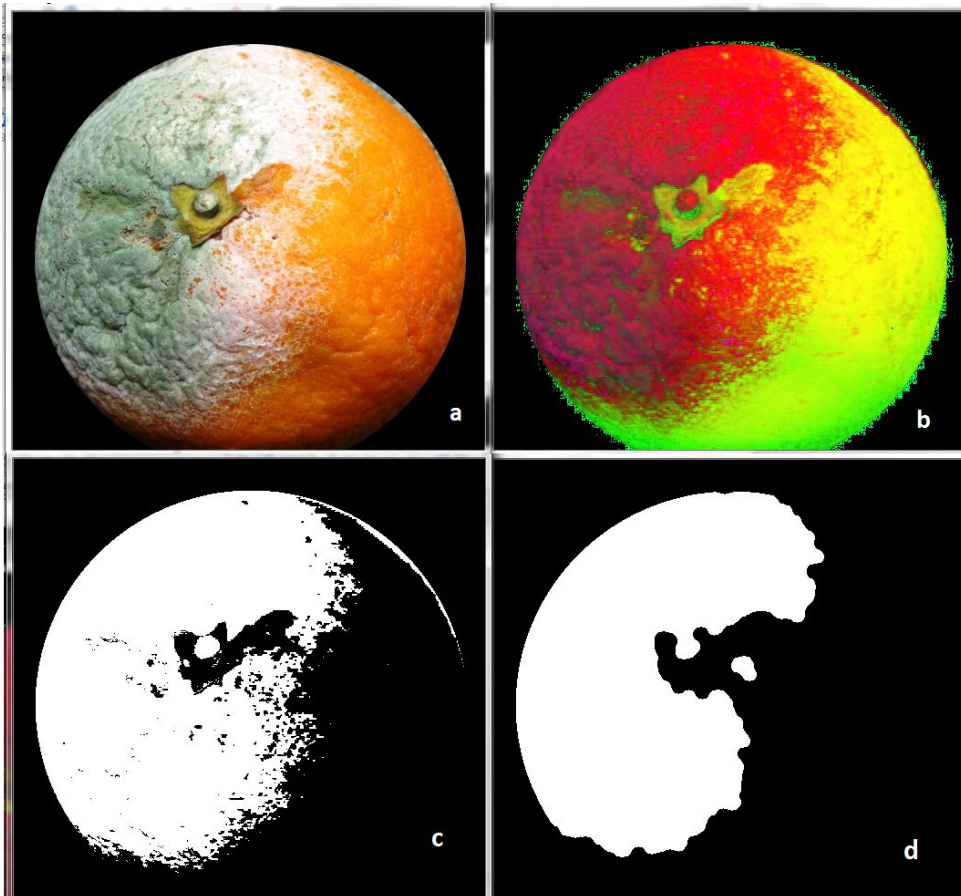


Figura 8: a- Imagem original; b- Imagem HSI; c- Resultado do *Threshold*; d- Remoção de ruídos.

4 CONCLUSÃO

Após realizados os passos constatamos que não é possível garantir um resultado bem próximo da realidade devido ao fato de que trabalhamos com imagens em 2D sendo que na realidade as imagens são em 3D, o que gera uma certa perda na área no cálculo da área. As maiores dificuldades foram em determinar o elemento estruturante (ou *kernel*) usado nos tratamentos de abertura e fechamento, determinar o limiar de busca (devido as cores variadas em que as colônias do fungo apresentam) para o formato HSI e também interpretar a área que foi dada em pixels e transformá-la em grandeza real (nesse caso centímetros quadrados). Contudo nos dois formatos de imagens (RGB e HSI) constatamos que ambos fornecem resultados bem aproximados e com uma margem de erro aceitável dada a natureza das análises das imagens, porem o formato HSI exigem menos tratamentos em relação ao RGB, o

que torna o HSI mais indicado para tratamento de imagens em escala industrial para esse problema com o bolor.

5 REFERÊNCIAS

PECORARO, Wesley. JORGE, L. A. C. BARBATO, D. M. L., GROOTE, Jean-Jacques de. (2010). **Detecção e classificação de fungos em laranjas pós-colheita por meio de processamento digital de imagens**. Disponível em: Biblioteca Digital Brasileira de Computação <<http://goo.gl/WjU686>>. Acessado em jun. 2016.

MINISTÉRIO DA AGRICULTURA. [S. I.]. **Citrus**. Disponível em: <<http://goo.gl/YzMCYx>>. Acesso em jul. 2016.

MARTINEZ, Marina. [S. I.] **Laranja**. InfoEscola, [S. I.]. Disponível em: <<http://goo.gl/EcuWsQ>>. Acesso em jul. 2016.