



**LUCAS VASCONCELOS PRADO PANISSA
MATEUS APARECIDO DE ANDRADE E SILVA**

**PROCESSAMENTO DE DADOS PARA CLASSIFICAÇÃO DO
GRAU DE MATURAÇÃO DE CAFÉ**

LAVRAS-MG

2023

LUCAS VASCONCELOS PRADO PANISSA
MATEUS APARECIDO DE ANDRADE E SILVA

PROCESSAMENTO DE DADOS PARA CLASSIFICAÇÃO
DO GRAU DE MATURAÇÃO DE CAFÉ

Concepção Básica apresentada a Universidade Federal de Lavras como parte das exigências do Curso de Engenharia de Controle e Automação, para obtenção de título de Bacharel.

Prof. Dr. Danton Diego Ferreira
Orientador

Dr. Luiz de Gonzaga Ferreira Júnior
Coorientador

LAVRAS-MG
2023

**LUCAS VASCONCELOS PRADO PANISSA
MATEUS APARECIDO DE ANDRADE E SILVA**

**PROCESSAMENTO DE DADOS PARA CLASSIFICAÇÃO DO GRAU DE
MATURAÇÃO DE CAFÉ**

**DATA PROCESSING FOR CLASSIFYING
THE DEGREE OF MATURATION OF COFFEE**

Concepção Básica apresentada a Universidade Federal de Lavras como parte das exigências do Curso de Engenharia de Controle e Automação, para obtenção de título de Bacharel.

Prof. Dr. Danton Diego Ferreira
Orientador

Dr. Luiz de Gonzaga Ferreira Júnior
Coorientador

APROVADA em 16 de março de 2023

Prof. Dr. Danton Diego Ferreira UFLA

Dr. Luiz de Gonzaga Ferreira Júnior GONZAGA TREINAMENTOS E CONSULTORIA
AGRÍCOLA LTDA.

Me. Fernando Elias de Melo Borges UFLA

LAVRAS-MG

2023

RESUMO

O objetivo desse trabalho é a análise da possibilidade da utilização de imagens para implementação de um aplicativo que auxiliaria na classificação dos graus de maturação do café em uma lavoura. A classificação de uma amostra de frutos de café na lavoura é útil para os produtores de tal forma a garantir um período de colheita que maximize a quantidade de frutos colhidos em estágios de maturação mais valorizados, influenciando no valor agregado ao produto final e, com isso, em maior ganho da produção. A ideia central é extrair dados úteis de imagens de amostras de café da plantação para identificar os graus de maturação por meio de um algoritmo que classifica áreas na imagem correspondentes a uma base de dados categorizada. A base de dados é categorizada conforme cada grau de maturação, contendo imagens de frutos nos estágios correspondentes, que são certificados por um especialista. Os estágios de maturação envolvidos são: verde, cana, cereja (vermelho ou amarelo dependendo da espécie), passa e seco. O processamento de dados das imagens do banco de dados e da amostra integra parte do algoritmo de classificação e do aplicativo em si. O aplicativo demonstrado nesse trabalho foi elaborado com o *software* Matlab, utilizando de imagens relacionadas às superfícies de frutos de café em cada estágio de maturação para composição do banco de dados. Por fim, a classificação da imagem de análise foi feita utilizando um algoritmo de classificação supervisionado chamado K-Vizinhos mais próximos, da sigla em inglês KNN. Com o aplicativo, foi possível estimar o grau de maturação de imagens utilizadas para análise, sendo gerados um gráfico de pizza que indica cada grau de maturação presente. O aplicativo gera, também, a imagem original com sobreposição de cores, indicando com essas cores a classificação que foi feita na imagem em questão. Além disso, uma análise econômica é feita, a fim de demonstrar a viabilidade da implementação do aplicativo, por meio do levantamento dos custos relativos à implementação do projeto proposto e da projeção de fatores necessários para o cálculo de viabilidade.

Palavras Chave: Grau de maturação do café, classificação, banco de dados, processamento de dados, análise econômica.

ABSTRACT

The objective of this work is to analyze the possibility of using images to implement an application that would help in classifying the degrees of maturation of coffee in a field. The classification of a sample of coffee fruits in the field is useful for producers in order to guarantee a harvest period that maximizes the amount of fruits harvested in more valued maturation stages, influencing the added value to the final product and, with that, in greater production gain. The central idea is to extract useful data from images of coffee samples from the plantation to identify the degrees of maturation through an algorithm that classifies areas in the image corresponding to a categorized database. The database is categorized according to each degree of maturation, containing images of fruits in the corresponding stages, which are certified by an expert. The ripening stages involved are: green, cane, cherry (red or yellow depending on the species), raisin and dry. Data processing of the sample and database images is part of the classification algorithm and the application itself. The application demonstrated in this work was elaborated with the Matlab software, using images related to the surfaces of coffee fruits in each stage of maturation for the composition of the database. Finally, the analysis image classification was performed using a supervised classification algorithm called K-Nearest Neighbors, KNN. With the application, it was possible to estimate the degree of maturation of images used for analysis, generating a pie chart that indicates each degree of maturation present. The application also generates the original image with color overlay, indicating with these colors the classification that was made on the image in question. In addition, an economic analysis is carried out in order to demonstrate the feasibility of implementing the application, by surveying the costs related to the implementation of the proposed project and projecting the necessary factors for the feasibility calculation.

Keywords: Coffee ripeness, classification, database, data processing, economic analysis.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estágios de Maturação do Café.....	11
Figura 2 - Frutos Verdes Contabilizados Manualmente.....	12
Figura 3 – Tábua de Separação para identificação dos graus de maturação.....	13
Figura 4 – Sistema RGB.....	14
Figura 5 – Pasta ‘Cereja Vermelho’	17
Figura 6 – ‘BancoDados.m’ (1ª à 21ª linha do código)	18
Figura 7 – ‘BancoDados.m’ (22ª à última linha do código)	18
Figura 8 – ‘CanaisRGB’ (1ª à 21ª linha do código)	19
Figura 9 – ‘CanaisRGB’ (22ª à última linha do código)	20
Figura 10 – ‘Classificacao.m’ (1ª à 26ª linha do código)	21
Figura 11 – ‘Classificacao.m’ (27ª à última linha do código)	21
Figura 12 – ‘mostrarClassificacao.m’ (1ª à 29ª linha do código)	22
Figura 13 – ‘mostrarClassificacao.m’ (30ª à 56ª linha do código)	23
Figura 14 – ‘mostrarClassificacao.m’ (57ª à última linha do código)	23
Figura 15 – ‘XY.m’ (1ª à última linha do código)	24
Figura 16 – ‘Principal.m’ (1ª à 28ª linha do código)	25
Figura 17 - ‘Principal.m’ (29ª à 57ª linha do código)	26
Figura 18 – ‘Principal.m’ (58ª à última linha do código)	26
Figura 19 – Imagem de análise	27
Figura 20 – Imagem de análise redimensionada	27
Figura 21 – Gráfico gerado para a imagem de análise redimensionada	28
Figura 22 – Imagem de análise redimensionada e sobrescrita	28
Figura 23 – Tempo da execução do programa para o caso em particular	29
Figura 24 – Segunda imagem de análise	29
Figura 25 – Segunda imagem redimensionada	30
Figura 26 – Gráfico gerado pela segunda imagem	30
Figura 27 – Segunda imagem de análise redimensionada e sobrescrita	31
Figura 28 – Tempo da execução do programa para o segundo caso	31
Figura 29 - Fluxos de caixa estimados	36

LISTA DE TABELAS

Tabela 1 - Materiais necessários.....	33
Tabela 2 - Investimentos necessários	33
Tabela 3 - Estimativa de colheita mensal em 2023 na região sudeste (em mil sacas)	35
Tabela 4 - Dados da produção de café e projeção de mercado	36
Tabela 5 - VP de cada período e VPL do projeto	37
Tabela 6 - Cálculo do TIR	38
Tabela 7 - Valores da taxa Selic ao mês no ano de 2022	39
Tabela 8 - Tempo de <i>payback</i>	40

LISTA DE SIGLAS

RGB	<i>Red Green Blue.</i>
KNN	<i>K-Nearest Neighbor.</i>
IDE	<i>Integrated Development Environment</i>
VP	Valor Presente
VPL	Valor Presente Líquido
TIR	Taxa Interna de Retorno
TMA	Taxa Mínima de Atratividade

SUMÁRIO

1	INTRODUÇÃO	10
2	DESENVOLVIMENTO	11
2.1	Identificação do Problema	11
2.2	Análise das Alternativas	12
2.2.1	Alternativa de Análise Visual dos Estágios de Maturação	12
2.2.1.1	Seleção da amostra e classificação ‘fruto em fruto’	12
2.2.1.2	Classificação com Tábuas.....	13
2.2.2	Alternativa de análise e classificação via <i>software</i>	14
2.2.3	Metodologia da alternativa de análise e classificação via <i>software</i>	15
2.2.3.1	Materiais	15
2.2.3.2	Métodos	16
2.2.3.2.1	Construção de um banco de dados	16
2.2.3.2.2	<i>Scripts</i> Auxiliares (funções) - ‘BancoDados.m’	17
2.2.3.2.3	<i>Scripts</i> Auxiliares (funções) - ‘CanaisRGB.m’	19
2.2.3.2.4	<i>Scripts</i> Auxiliares (funções) - ‘Classificacao.m’	20
2.2.3.2.5	<i>Scripts</i> Auxiliares (funções) - ‘mostrarClassificacao.m’	22
2.2.3.2.6	<i>Scripts</i> Auxiliares (funções) - ‘XY.m’	24
2.2.3.2.7	<i>Script</i> Principal - ‘Principal.m’	24
2.2.3.2.8	Execução do programa	27
2.3	Levantamento dos Custos da Alternativa.....	32
2.3.1	Materiais Necessários	32
2.3.2	Serviços	33
2.4	Levantamento dos Benefícios.....	34
2.5	Análise Econômica	35
2.5.1	Valor Presente Líquido VPL	37
2.5.2	Taxa Interna de Retorno TIR	38
2.5.3	Taxa Mínima de Atratividade (TMA)	39
2.5.4	<i>Payback</i>	40
2.5.5	Resultados	41
3	CONCLUSÃO	42
4	REFERÊNCIAS.....	43

1 INTRODUÇÃO

Em uma lavoura de café é de suma importância a capacidade de determinar os períodos favoráveis a colheita dos frutos.

Decidir o momento ideal para colher é muito importante e tem relação direta com a qualidade e os defeitos do café. Uma colheita atrasada acarreta maior quantidade de grãos pretos. Caso a colheita for adiantada, poderá ocorrer maior quantidade de frutos ardidos e verdes, prejudicando também a qualidade final do café. FERREIRA JÚNIOR *et al* (2017).

A análise dos estágios de maturação entra para auxiliar o produtor a verificar o tempo certo de iniciar a colheita, visando o máximo de frutos com maturação mais valorizada.

Os estágios de maturação conhecidos dentre as espécies de café que são cultivadas no Brasil são: verde, cana, cereja (vermelho ou amarelo), passa e seco. As nomenclaturas podem variar internamente no vocabulário do país, sendo em alguns locais o café seco chamado de boia, dentre outras variações, no entanto por simplicidade os termos utilizados são os citados anteriormente.

A análise dos estágios de maturação é feita na maioria das vezes de forma visual, o encarregado da análise retira uma amostra da plantação e contabiliza os frutos em cada estágio, de forma a se obter a porcentagem em cada categoria.

Os frutos de maior interesse são os denominados frutos cereja, que possuem características biológicas que conferem um paladar específico no produto final. Uma forma de garantir a colheita priorizada desses frutos é a colheita seletiva, em que os frutos de interesse são identificados no pé e, então, são coletados. No entanto, esse tipo de colheita é trabalhosa e muitas vezes inviável, fazendo com que os produtores prefiram a modalidade de colheita plena, que é a derriça de todos os frutos da lavoura. Nesse caso, os frutos colhidos provavelmente se encontram em diferentes estágios de maturação e, portanto, uma estratégia de maximizar os ganhos da colheita é a observação de um período oportuno do aparecimento de maior quantidade possível dos frutos cereja.

Como a análise dos frutos para determinar um período oportuno de colheita é basicamente um problema de classificação do aspecto dos frutos, relevando principalmente as cores e as relacionando aos estágios de maturação. Para isto, foi proposto uma alternativa que envolve a análise de imagens das amostras de frutos da plantação, que utilizaria um algoritmo de classificação para indicar as proporções de frutos em cada grau de maturação. Servindo, dessa forma, de ferramenta de análise digital para auxiliar na tomada de decisões para colheita.

2 DESENVOLVIMENTO

2.1 Identificação do Problema:

No cenário de uma plantação de café, sabe-se que o período ideal da colheita dos frutos é algo almejado pelos produtores. Na plantação, durante o processo de amadurecimento geral, é verificado, na maioria das vezes, que os estágios de maturação individuais dos frutos não é algo homogêneo. E, ao se optar pela colheita plena, a consequência é colheita de frutos com certa variedade de maturação. O desafio é, portanto, escolher uma época favorável à maior incidência de frutos cereja, que são os frutos mais valorizados no setor cafeeiro, além de garantir o mínimo possível de frutos secos e verdes.

Durante todo o período de maturação, o que se percebe é a mudança de cor dos frutos. Logo, o critério visual é amplamente utilizado para classificação do café. Com essa classificação é possível estimar períodos de colheita ou a qualidade geral da lavoura.

As características das fases de amadurecimento são particulares, e, de forma visual, se pode relacionar os períodos às cores dos frutos. Os frutos verdes, como o nome indica, possuem coloração predominante de verde escuro à um verde claro amarelado. Nos frutos cana, as cores predominantes variam entre o verde claro amarelado e um amarelo claro, com traços avermelhados, dependendo da espécie. Os frutos cereja possuem duas variedades conforme a espécie em questão: frutos que exibem coloração predominante do amarelo claro ao vermelho escuro (cerejas vermelhos), ou frutos que exibem cor amarela intensa (cerejas Amarelos). Os frutos passa possuem coloração predominantemente vermelho escuro ao vinho. Por fim, os secos, em tons amarronzados ou escuros.

Para a estimativa do ponto da colheita deve-se adotar procedimentos que auxiliariam na visão da maturação geral da lavoura, e com esses dados conseguir uma previsão do início da colheita. A análise do nível de maturação da lavoura, é utilizada principalmente para essa finalidade. A Figura 1 ilustra a variação de cores em frutos com diferentes maturações.

Figura 1- Estágios de maturação do café



Fonte: <https://www.cafepoint.com.br/noticias/tecnicas-de-producao/voce-sabe-o-tempo-ideal-de-maturacao-da-sua-lavoura-215991/> (acesso em 03/01/2023).

2.2 Análise das Alternativas

2.2.1 Alternativa de Análise Visual dos Estágios de Maturação

A análise visual é a forma mais recorrente de classificação dos estágios de maturação, no entanto, para que ocorra, são utilizados procedimentos de seleção da amostra e de classificação. Para, então chegar na proporção que representaria o nível médio de maturação da lavoura.

2.2.1.1 Seleção da amostra e classificação ‘fruto em fruto’

Uma das formas de seleção da amostra é a escolha visual de um pé representante, que contenha visualmente frutos que representam a média da maturação da lavoura. Outra forma é escolher até 8 plantas aleatórias que sejam representativas da maturação geral da lavoura, andando entre os talhões em zigue-zague e colhendo os seus frutos. Após coletada a amostra, os detritos grosseiros são retirados manualmente, e, então o restante removido via peneiração (SERVIÇO NACIONAL DE APRENDIZAGEM RURAL - SENAR, 2017)

Tendo a amostra limpa em mãos, os frutos são misturados para se espalharem de forma homogênea. Feito isso, a classificação fruto em fruto pode ser feita de algumas formas: encher uma mão com os frutos e fazer uma média, ou utilizar de um recipiente de meio litro para retirar uma quantidade da amostra e contabilizar os frutos em cada estágio. É recomendável realizar qualquer que for a técnica por pelo menos três vezes (GONZAGA TREINAMENTOS E CONSULTORIA AGRÍCOLA LTDA., 2023).

A proporção em cada estágio é calculada com base na quantidade total de frutos contados ou pelo volume de todos os frutos. Com os valores da quantidade de frutos ou volume em cada estágio é possível realizar um cálculo de regra de três simples, relevando o total dos frutos e com isso determinar as proporções dos estágios de maturação. A Figura 2 mostra frutos classificados visualmente.

Figura 2 – Frutos verdes contabilizados manualmente



Fonte: <https://www.cafepoint.com.br/noticias/tecnicas-de-producao/voce-sabe-o-tempo-ideal-de-maturacao-da-sua-lavoura-215991/> (acesso em 08/01/2023).

2.2.1.2 Classificação com Tábuas

Neste caso é utilizado uma tábua com 100 furos, de modo que ‘mergulha-se’ essa tábua na amostra homogeneizada de café, e, assim que os frutos se fixam nos furos, se realiza a identificação visual. Desta forma a quantidade de frutos identificados em determinado estágio de maturação já representa a porcentagem dos mesmos na amostra. Segundo um especialista da área, também é recomendável que o procedimento de classificação se repita, ao menos, três vezes. A Figura 3 mostra uma tábua utilizada para o processo de classificação.

Figura 3 – Tábua de separação para identificação dos graus de maturação:



Fonte: Gonzaga Treinamentos e Consultoria Agrícola Ltda. (2023).

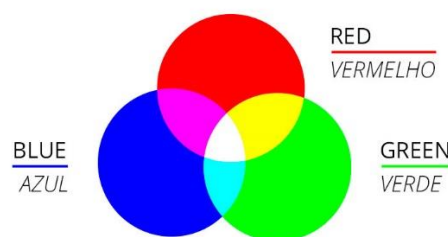
2.2.2 Alternativa de análise e classificação via *software*

Uma imagem colorida pode ser descrita computacionalmente como 3 matrizes com dimensões referentes à altura e largura da imagem. Sendo que cada matriz faz associação da posição do pixel na imagem (índice na matriz) e seu valor do sistema de cores utilizado.

Um pixel indexado a certo valor de x e y dos eixos ortogonais no plano em que a imagem pode ser descrita, possui, também, 3 canais que guardam informação sobre o sistema de cores dos pixels. Um sistema de cores muito utilizado é o RGB, que atribui valores de 0 a 255 para cada canal (R, G ou B), sendo 0 a mínima intensidade para o determinado canal e 255 a máxima. O canal R informa os valores sobre a cor vermelha (R de ‘Red’, que do inglês que significa vermelho) dos pixels, o canal G (G de ‘Green’ que significa verde) informa sobre a cor verde, e o canal B (B de ‘Blue’ que significa azul) informa sobre a cor azul. Cada canal, ao atribuir os valores, à todos os pixels representa uma das 3 matrizes do sistema de cores que compõe a imagem colorida.

A junção das informações dos 3 canais desse sistema modela a cor que é visualizada em um pixel, conforme ilustra a Figura 4. Como a determinação do grau de maturação de uma amostra de café é algo que pode ser feita através da análise das cores presentes, a informação dos canais RGB dos pixels são dados muito úteis para a classificação das imagens digitalmente. A Figura 4 ilustra o princípio da formação de cores no sistema RGB

Figura 4 – Sistema RGB



Fonte: <https://www.afixgraf.com.br/blog/o-que-significa-rgb/> (acesso em 17/01/2023).

Além dessas informações de cores, outros inúmeros dados podem ser retirados da imagem. O estudo de visão computacional e processamento de dados nos dá várias ferramentas de análise que retiram informações úteis de arquivos de imagem conforme o contexto empregado. Informações como variação brusca de cores para identificação de bordas, dentre outros, são exemplos de ferramentas de análise das imagens digitais. No entanto, como o problema da classificação dos graus de maturação é algo muito relacionado à coloração dos frutos, é suficiente, em teoria, a análise das cores da imagem.

Assim como na classificação feita visualmente, na classificação virtual tomam-se referências para categorizar os elementos. Uma base de dados devidamente escolhida fornece a um algoritmo de classificação as referências necessárias para análise de dados a serem classificados.

A base de dados pode ser construída utilizando imagens classificadas visualmente (classificador supervisionado). E, então, os dados RGB das imagens de referência escolhidas podem ser utilizados para classificação de uma imagem a ser analisada.

Para a comparação, uma leitura pixel a pixel da imagem analisada pode ser feita, a fim de informar a um algoritmo de classificação os valores RGB dos mesmos, e comparar aos valores encontrados no banco de dados previamente elaborado. Com isso, pode-se rotular o pixel com uma das classificações existentes.

Diversos algoritmos de classificação, bem como utilização de redes neurais, entre outros, podem auxiliar na classificação de uma imagem e trazer resultados satisfatórios. Existem vários exemplos que corroboram a possibilidade da análise de imagens e de classificações, como no trabalho de BOELL, M. G. *et al* (2020), sobre a classificação de imagens multiespectrais obtidas por equipamentos de satélites.

A classificação pixel a pixel dentro da dimensão total da imagem pode fornecer uma estimativa da porcentagem em cada grau de maturação que a amostra possui fisicamente. Com isso, a proposta da classificação via *software* se torna interessante a ser discutida mais a fundo.

2.2.3 Metodologia da alternativa de análise e classificação via *software*

2.2.3.1 Materiais

- Computador pessoal ou um dispositivo com requisitos necessários para desenvolvimento de programas e processamento de dados.
- *Software* para implementação do programa.

Dispositivo utilizado: Computador Desktop com processador Intel Core i3 – 2100, 3.10 GHz e 6 Gb de memória RAM. Sistema Operacional: Windows 10 pro, 64 bits.

Software utilizado: Matlab R2011a, 64bits.

2.2.3.2 Métodos

A fim de demonstrar a possibilidade da análise e classificação via *software*, um programa foi implementado no Matlab. Por meio de alguns *scripts*, uma imagem que se deseja analisar é classificada por via um algoritmo de classificação do tipo KNN (*'K-Nearest Neighbor'*, ou 'k vizinhos mais próximos' em português), utilizando os valores RGB de um banco de dados de imagens já classificadas.

2.2.3.2.1 Construção de um banco de dados

Primeiramente um banco de dados foi criado a partir de pedaços de imagens de frutos de café. Os frutos são identificados em um determinado estágio de maturação, e então um recorte da figura do fruto é inserido em uma pasta destinada a fornecer os dados RGB da classificação correspondente. Parte das imagens utilizadas no banco de dados foram fornecidas por Gonzaga Treinamentos e Consultoria Agrícola Ltda., outra parte retirada da Figura 1, por meio da seleção de regiões de interesse da imagem.

A pasta 'BancoDeDados' que foi criada agrupa as subpastas 'cerejaAmarelo', 'cerejaVermelho', 'passa', 'seco' e 'verdeEcana' que são utilizadas para armazenar imagens referentes à frutos no estágio de cereja amarelo, cereja vermelho, passa, seco, verde e cana (juntos) respectivamente. A pasta é destinada, como o nome diz, a fornecer o banco de dados total do projeto.

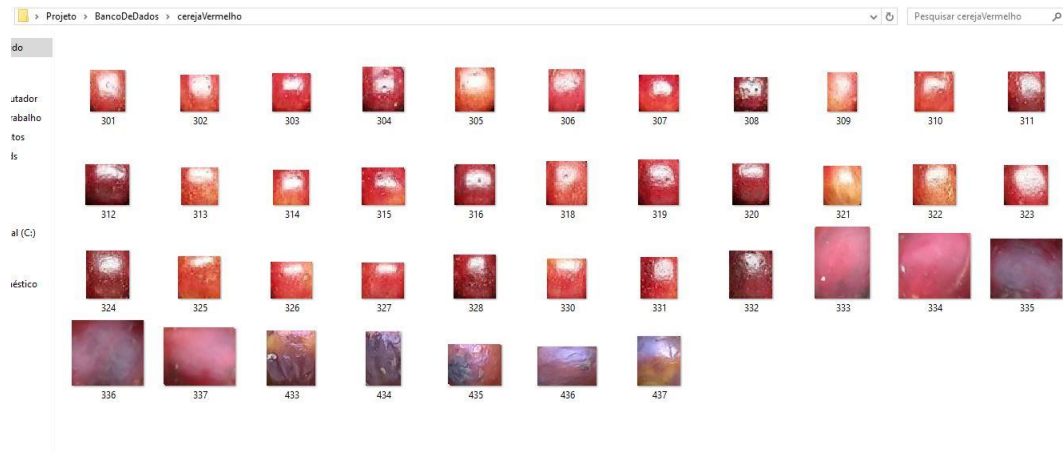
A pasta 'verdeEcana' possui 24 imagens das superfícies dos frutos em estágio verde e cana. A pasta 'cerejaAmarelo' possui 2 imagens relacionadas ao grau de maturação cereja, de espécies que apresentam a cor amarelada nesse estágio. A pasta 'cerejaVermelho' possui 40 imagens relacionadas aos frutos em estágio cereja vermelho. A pasta 'passa' possui 11 imagens e a pasta 'seco' possui 7 imagens.

A quantidade de imagens pode ser aumentada e diminuída porque os códigos foram feitos independentemente da quantidade de imagens por pasta, com isso a possibilidade de melhoria da acurácia do programa é maior, considerando imagens bem selecionadas.

A figura a seguir mostra a pasta 'cerejaVermelho', que foi utilizada para armazenar as imagens dos frutos nesse estágio de maturação. É possível perceber que as imagens são relativas à algumas partes das superfícies dos frutos, uma vez que fatores como luminosidade e o aparecimento do fundo vão gerar dados não representativos para o algoritmo de classificação. A Figura 5 mostra a pasta 'cerejaVermelho', que contém diversas imagens que alimenta o banco

de dados referentes à classificação cereja vermelho, as demais pastas dentro do diretório ‘BancoDeDados’ seguem a mesma lógica.

Figura 5 – Pasta ‘CerejaVermelho’



Fonte: Captura de tela, dos autores (2023).

Os *scripts* foram implementados de modo a serem independentes da quantidade de imagens dentro das pastas do banco de dados. Essa quantidade interfere no tempo de execução do algoritmo de classificação, uma vez que, quanto mais dados, mais comparações deverão ser feitas. A escolha de imagens representativas interfere na classificação da imagem a ser analisada, portanto, a classificação visual das imagens no banco de dados deve ser criteriosa.

O formato utilizado das imagens foi padronizado para JPG, pois a inserção de imagens em diferentes formatos na pasta pode ocasionar classificações erradas no *script*. O Matlab atribui diferentes valores RGB de imagens com diferentes codificações, portanto, a padronização em único formato foi necessária.

2.2.3.2.2 *Scripts* Auxiliares (funções) - ‘BancoDados.m’

A função BancoDados.m criada é responsável por acessar todos os elementos da pasta cujo nome é passado por parâmetro (‘nomePasta’, ou seja, o nome da pasta desejada). Ela foi utilizada para ser possível a organização em pastas do banco de dados. Cada pasta do banco de dados possui uma classificação correspondente e relacionada ao nome da mesma.

A função utiliza de outro *script* chamado ‘CanaisRGB’ para retirar todos os dados RGB de cada imagem em vetores separados, um para cada canal (R, G, B). Feito isso, concatena os valores em uma matriz representativa de todas as imagens da pasta passada por parâmetro. A matriz gerada contém todos os valores RGB de todas as imagens da pasta que foi passada. Essa

matriz possui N linhas, sendo N a quantidade total de pixels de todas as imagens na pasta, e 3 colunas, respectivas aos valores R, G e B. As Figuras 6 e 7 mostram as linhas de código.

Figura 6 – ‘BancoDados.m’ (1ª à 21ª linha do código)

```

1      %função que retorna uma matriz A por B, sendo B = 3, que são
2      %os atributos utilizados como referência para cálculo de knn
3      %A é a quantidade de observações desses atributos
4      %os atributos são os valores r g b de uma imagem
5      function [matriz] = BancoDados(nomePasta)
6
7      %mudança de diretório para a pasta do projeto
8      cd C:\Users\User\Desktop\Projeto
9      %atribuição a variável 'diretorio' a concatenação da pasta do projeto
10     %com o nome da pasta passada por parâmetro e atribuída a 'pasta'
11     diretorio = strcat('C:\Users\User\Desktop\Projeto\BancoDeDados\', nomePasta);
12     %aquisição dos dados da pasta
13     pasta = dir(diretorio);
14     %mudança de diretório para o destino especificado
15     cd(diretorio);
16
17     %leitura da quantidade de elementos na pasta
18     tamanho = size(pasta);
19     %criação de vetor auxiliar
20     aux = [1 1 1];
21

```

Fonte: Captura de tela, dos autores (2023).

Figura 7 – ‘BancoDados.m’ (22ª à última linha do código)

```

22     %varredura de todos os elementos da pasta
23     for i=3:tamanho(1)
24         %aquisição do nome do arquivo de índice i
25         nome = pasta(i).name;
26         %aquisição dos vetores referentes aos valores r g b de cada
27         %imagem da pasta através da função CanaisRGB
28         [vermelho, verde, azul] = CanaisRGB(nome);
29         %concatenação dos vetores adquiridos do elemento anterior com o próximo
30         m = [vermelho;verde;azul];
31         aux = cat(1, aux ,m');
32
33     end
34     %retorno da matriz com os valores rgb de todas as imagens da pasta
35     %matriz com 3 colunas sendo a primeira referente ao valor de r
36     %a segunda do valor de g e a terceira do valor de b
37     %de todas as imagens da pasta
38     %a matriz é cortada a partir do segundo elemento para retirar
39     %o valor do vetor auxiliar 'aux'
40     matriz = aux(2:end, :);
41
42     end

```

Fonte: Captura de tela, dos autores (2023).

2.2.3.2.3 *Scripts Auxiliares (funções) - 'CanaisRGB.m'*

A função `CanaisRGB.m` foi criada para receber o nome de uma imagem como parâmetro ('imagem.formato'). Ela utiliza o comando 'imread' para retirar os dados da imagem em forma de matriz. A matriz possui as duas primeiras dimensões equivalentes à altura e largura da imagem em pixels. As 3 dimensões restantes são os valores da codificação RGB de cada pixel: esses valores podem ter magnitude entre 0 e 255, e indicam o quanto de vermelho, verde e azul o pixel exibe simultaneamente.

Nessa função, a imagem toda é varrida e os valores RGB dos pixels são armazenados nos 3 vetores de retorno, 'vermelho', 'verde', 'azul'.

É a função responsável por retirar da imagem os dados úteis a serem utilizados no programa, uma vez que a informação de cor é o aspecto relevante da imagem para classificação do grau de maturação. As Figuras 8 e 9 mostram as linhas de código.

Figura 8 – 'CanaisRGB' (1ª à 21ª linha do código)

```

1      %função que retorna 3 vetores referente a todos os valores rgb
2      %lidos em ordem linha por linha de uma imagem 'nome' passada
3      %como parâmetro, os 3 vetores são 'vermelho' 'verde' e 'azul'
4      function [vermelho, verde, azul] = CanaisRGB(nome)
5
6      %importando a figura e retirando dados
7      G = imread(nome);
8
9      %aquisição das dimensões espaciais da imagem, altura e largura
10     dimensoes = size(G);
11     largura = dimensoes(1,1);
12     altura = dimensoes(1,2);
13     %quantidade total de pixels
14     qtdPixels = largura*altura;
15
16     %criação de 3 vetores de retorno
17     vermelho = zeros(1,qtdPixels);
18     verde = zeros(1,qtdPixels);
19     azul = zeros(1,qtdPixels);
20     %variável auxiliar de contagem
21     pos = 1;

```

Fonte: Captura de tela, dos autores (2023).

Figura 9 – ‘CanaisRGB’ (22ª à última linha do código)

```

22   %varredura de toda a imagem
23 -   for i = 1:largura
24 -       for v = 1:altura
25           %vetor vermelho recebe o valor do canal R no indice i e v
26 -       vermelho(pos) = G(i,v,1);
27           %vetor verde recebe o valor do canal G no indice i e v
28 -       verde(pos) = G(i,v,2);
29           %vetor azul recebe o valor do canal B no indice i e v
30 -       azul(pos) = G(i,v,3);
31           %incremento da variável de contagem
32 -       pos = pos +1;
33 -       end
34 -   end
35 - end

```

Fonte: Captura de tela, dos autores (2023).

2.2.3.2.4 Scripts Auxiliares (funções) - ‘Classificacao.m’

A função `Classificacao.m` recebe como parâmetro uma string relativa ao nome de uma imagem a ser analisada. Recebe também uma matriz ‘x’ que é a matriz formada por todos os valores RGB de todos os pixels de todas as imagens do banco de dados inteiro, tendo número de linhas N igual à quantidade total de pixels lidos em todo o banco de dados e número de colunas igual a 3, que são os canais R, G e B desses pixels. Recebe como parâmetro também um vetor y, com números de linhas igual ao número de linhas da matriz x, cada linha contém a rotulação do pixel em questão. Se os valores RGB na n-ésima linha correspondem ao pixel de uma imagem identificada no banco de dados como sendo do tipo verde ou cana, o valor da linha no vetor y (rótulo) nesse índice é igual a 1, o valor é igual a 2 para dados do tipo cereja vermelho, 3 para cereja amarelo, 4 para passa e 5 para seco.

A função retorna uma matriz de mesma dimensão física da imagem a ser analisada (altura e largura). Cada índice representa a classificação do pixel de mesmo índice na imagem feita na função, com os rótulos citados anteriormente (1 para verde ou cana, 2 para cereja vermelho, 3 para cereja amarelo, 4 para passa e 5 para seco). Para a classificação foi utilizado um modelo de KNN (*K-Nearest Neighbor*), que classifica pixel a pixel a imagem analisada. As Figuras 10 e 11 mostram as linhas de código.

O algoritmo KNN funciona da seguinte forma: são escolhidos alguns atributos que possam auxiliar a classificar as classes de elementos, os atributos observados pertencentes a uma classe recebem o rótulo da classe. Um novo elemento, que possua o mesmo tipo de atributos dos que foram catalogados e rotulados, poderá ser classificado, relevando os

elementos com atributos mais próximos ao seu, ou seja, um cálculo de distância euclidiana entre os valores dos atributos dos elementos já rotulados e os valores dos atributos do elemento que se deseja rotular, tirando como base uma quantidade especificada de vizinhos dos quais se deseja estimar a classificação.

Figura 10 – ‘Classificacao.m’ (1ª à 26ª linha do código)

```

1      %função que gera uma matriz (matrizClassificacao) que armazena a
2      %classificação dada pelo knn de cada pixel da imagem,
3      %a matriz é retornada e possui mesmo tamanho da imagem lida
4      %com cada elemento representando a classificação do pixel
5      %de mesmo índice na imagem original.
6      function [matrizClassificacao] = Classificacao(imagem, x, y)
7
8      %mudança de pasta para a pasta da imagem a ser classificada
9      cd C:\Users\User\Desktop\Projeto\imagemTeste
10     %leitura da imagem a ser classificada
11     i = imread(imagem);
12     %mudança de pasta para a pasta do projeto
13     cd C:\Users\User\Desktop\Projeto
14
15     %criação de um modelo knn com parâmetro de 5 vizinhos mais próximos
16     %recebe o vetor de valores rgb de todos os estágios de maturação
17     %recebe o vetor y com os rótulos correspondetes aos valores de x
18     mdl = fitcknn(x, y, 'NumNeighbors', 5, 'Standardize', 1);
19
20     %informação das dimensões da imagem a ser classificada
21     dimensoes = size(i);
22     %largura da imagem
23     largura = dimensoes(1,1);
24     %altura da imagem
25     altura = dimensoes(1,2);
26

```

Fonte: Captura de tela, dos autores (2023).

Figura 11 – ‘Classificacao.m’ (27ª à última linha do código)

```

27     %inicialização da matriz de retorno da função com valores nulos
28     matrizClassificacao = zeros(largura, altura, 'double');
29
30     %classificação de toda a imagem a ser classificada, pixel a pixel
31     %utilizando os valores rgb lidos da imagem e o modelo knn
32     for j = 1 : largura
33         for k = 1 : altura
34
35             %valor de r lido no pixel da imagem, pixel de indice j e k
36             r = double(i(j,k,1));
37             %valor de g lido do pixel da imagem, pixel de indice j e k
38             g = double(i(j,k,2));
39             %valor de b lido do pixel da imagem, pixel de indice j e k
40             b = double(i(j,k,3));
41
42             %atribuição da posição j e k da matrizClassificacao
43             %à classificação gerada pelo knn da imagem lida
44             matrizClassificacao(j,k) = predict(mdl, [r,g,b]);
45         end
46     end
47
48 end

```

Fonte: Captura de tela, dos autores (2023).

2.2.3.2.5 Scripts Auxiliares (funções) - ‘mostrarClassificacao.m’

A função `mostrarClassificacao.m` recebe o nome da imagem de análise e uma matriz de classificação gerada pela função ‘`Classificacao.m`’, chamada no *script* principal. Ela retorna a porcentagem de cada estágio de maturação presente na imagem de análise dentro das variáveis de retorno: ‘`pVerdeEcana`’, para a porcentagem de pixels classificados como pertencentes aos estágios de maturação verde e cana, ‘`pCereja`’, para os classificados como cereja vermelho, ‘`pCerejaA`’, para os classificados como cereja amarelo, ‘`pPassa`’, para os classificados como passa e ‘`pSeco`’ para os classificados como seco. Além disso retorna uma variável ‘`tela`’ que nada mais é do que a imagem de análise com as cores dos pixels alteradas, indicando o grau de maturação correspondente. Para o grau de verde ou cana a cor inserida no pixel classificado é verde, para o cereja vermelho a cor é o vermelho, para o cereja amarelo a cor é amarelo claro, para o passa é vinho e para o seco é uma tonalidade escura. As Figuras 12, 13 e 14 mostram as linhas de código.

Figura 12 – ‘`mostrarClassificacao.m`’ (1ª à 29ª linha do código)

```

1  %função que retorna a porcentagem em cada grau de maturação
2  %da matriz de classificação gerada pela função 'Classificacao'
3  %retorna tambem a imagem original sobrescrita com cores
4  %para visualização dos pixels classificados na imagem
5  function [pVerdeEcana, pCereja, pCerejaA, pPassa, pSeco, tela] = mostrarClassificacao(imagem, matrizClassificacao)
6  %leitura do tamanho da matrizClassificacao
7  t = size(matrizClassificacao);
8  %inicialização de contadores para o cálculo da porcentagem
9  cverdeEcana = 0;
10 ccereja = 0;
11 ccerejaA = 0;
12 cpassa = 0;
13 cseco = 0;
14 %quantidade total de pixels utilizados para cálculo da porcentagem
15 qtd = t(1,1)*t(1,2);
16 %mudança de diretório
17 cd C:\Users\User\Desktop\Projeto\imagemTeste
18 tela = imread(imagem);
19 %mudança de diretório
20 cd C:\Users\User\Desktop\Projeto
21
22 %vareadura de cada posição da matrizClassificacao
23 for i=1:t(1,1)
24     for j = 1:t(1,2)
25         %caso não foi feito a classificação mostra branco na tela
26         if matrizClassificacao(i,j) == 0
27             tela(i,j,1) = 255;
28             tela(i,j,2) = 255;
29             tela(i,j,3) = 255;

```

Fonte: Captura de tela, dos autores (2023).

Figura 13 – ‘mostrarClassificacao.m’ (30ª à 56ª linha do código)

```

30 - else
31 - %se a classificação indica o rotulo de verde e cana (1)
32 - %sobreescreve verde na imagem original
33 - %incrementa o contador de verde e cana
34 - if matrizClassificacao(i,j) == 1
35 -     tela(i,j,1) = 0;
36 -     tela(i,j,2) = 125;
37 -     tela(i,j,3) = 0;
38 -     cverdeEcana = cverdeEcana +1;
39 - else
40 - %se a classificação indica rotulo de cereja vermelho(2)
41 - %sobreescreve vermelho na imagem original
42 - %incrementa o contador de cereja vermelho
43 - if matrizClassificacao(i,j) == 2
44 -     tela(i,j,1) = 200;
45 -     tela(i,j,2) = 0;
46 -     tela(i,j,3) = 0;
47 -     ccereja = ccereja +1;
48 - else
49 - %se a classificação indica rotulo de cereja amarelo(3)
50 - %sobreescreve cor amarelo na imagem original
51 - %incrementa o contador de cereja amarelo
52 - if matrizClassificacao(i,j) == 3
53 -     tela(i,j,1) = 125;
54 -     tela(i,j,2) = 125;
55 -     tela(i,j,3) = 0;
56 -     ccerejaA = ccerejaA +1;

```

Fonte: Captura de tela, dos autores (2023).

Figura 14 – ‘mostrarClassificacao.m’ (57ª à última linha do código)

```

57 - else
58 - %se a classificação indica rotulo de passa (4)
59 - %sobreescreve cor vinho na imagem original
60 - %incrementa o contador de passa
61 - if matrizClassificacao(i,j)==4
62 -     tela(i,j,1) = 125;
63 -     tela(i,j,2) = 0;
64 -     tela(i,j,3) = 125;
65 -     cpassa = cpassa +1;
66 - else
67 - %se a classificação indica rotulo de seco (5)
68 - %sobreescreve cor escura na imagem original
69 - %incrementa o contador de seco
70 - if matrizClassificacao(i,j) == 5
71 -     tela(i,j,1) = 50;
72 -     tela(i,j,2) = 50;
73 -     tela(i,j,3) = 50;
74 -     cseco = cseco +1;
75 - end
76 - end
77 - end
78 - end
79 - end
80 - end
81 - end
82 - end
83 - %atraves dos contadores e do total de pixels calcula a porcentagem de cada estágio de maturação
84 - pVerdeEcana = cverdeEcana/qtd;
85 - pCereja = ccereja/qtd;
86 - pCerejaA = ccerejaA/qtd;
87 - pPassa = cpassa/qtd;
88 - pSeco = cseco/qtd;
89 - end

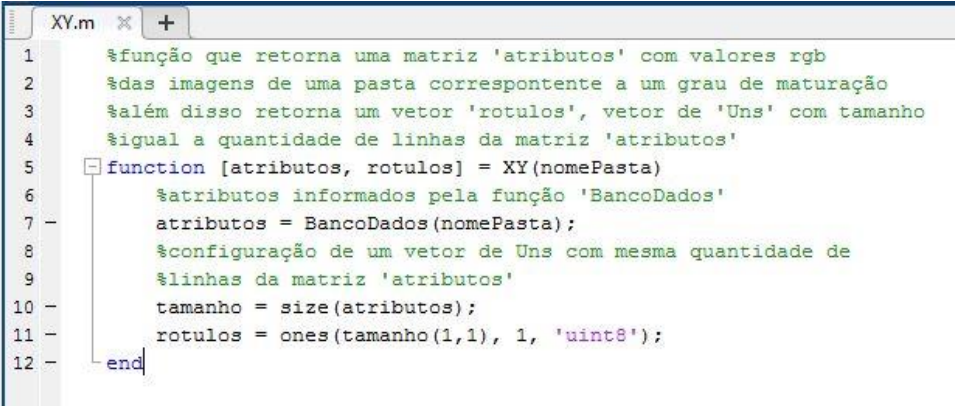
```

Fonte: Captura de tela, dos autores (2023).

2.2.3.2.6 *Scripts Auxiliares (funções) - 'XY.m'*

A função XY.m recebe o nome da pasta do banco de dados referente ao grau de maturação que se deseja acessar os elementos. Ela retorna uma matriz chamada 'atributos' que contém os valores RGB, possui N número de linhas sendo N igual à quantidade total de pixels lidos em todas imagens daquela pasta, e 3 colunas referente aos valores RGB dos pixels. Retorna também um vetor de Uns, denominado de 'rotulos', com a mesma quantidade de linhas da matriz 'atributos'. A função é utilizada para auxiliar a aquisição de dados do banco de dados para posteriormente montar a matriz X (utilizada na função 'Classificacao.m'), com os atributos RGB de todos estágios de maturação, e a matriz Y, com os rótulos de cada elemento da matriz X. Nessa função apenas um vetor de Uns é criado para inicializar o vetor 'rotulos' com a mesma quantidade de linhas da matriz 'atributos', no programa principal o devido rótulo é inserido (1,2,3,4 ou 5). A Figura 15 mostra as linhas de código.

Figura 15 – 'XY.m' (1ª à última linha do código)



```

1 %função que retorna uma matriz 'atributos' com valores rgb
2 %das imagens de uma pasta correspondente a um grau de maturação
3 %além disso retorna um vetor 'rotulos', vetor de 'Uns' com tamanho
4 %igual a quantidade de linhas da matriz 'atributos'
5 function [atributos, rotulos] = XY(nomePasta)
6 %atributos informados pela função 'BancoDados'
7 atributos = BancoDados(nomePasta);
8 %configuração de um vetor de Uns com mesma quantidade de
9 %linhas da matriz 'atributos'
10 tamanho = size(atributos);
11 rotulos = ones(tamanho(1,1), 1, 'uint8');
12 end

```

Fonte: Captura de tela, dos autores (2023).

2.2.3.2.7 *Script Principal - 'Principal.m'*

O *script* Principal.m chama a função 'XY.m' para retirar os dados de cada estágio de maturação presente nas pastas cujos nomes são passados como parâmetro, as matrizes de atributos juntamente com a matrizes de rótulos que são adquiridas formam no total parte da matriz 'x' e 'y', utilizadas para classificação.

A imagem de análise deve ser colocada dentro da pasta 'imagemTeste' no diretório 'Projeto' para que o *script* principal efetue a leitura da mesma e execute a classificação. A imagem deve ser recortada, evitando regiões de bordas que não exibem frutos de café, ou regiões em que se aparece o fundo de apoio. Também é indicado uma luminosidade adequada

(sem sombras ou reflexos de luz) nas amostras fotografadas. Para fins de eficiência de execução, as imagens foram redimensionadas a tamanhos menores, já que o algoritmo percorre todos os pixels, e, imagens com dimensões muito grandes estendem o período de processamento de forma considerável. Além disso, como dito anteriormente, o formato padrão adotado das imagens é JPG, porque a inserção de diferentes codificações de imagem no Matlab pode gerar leituras diferentes dos valores RGB.

O *script* faz a chamada da função ‘Classificacao’ para receber a classificação pixel a pixel da imagem analisada. Pela função ‘mostrarClassificacao.m’ adquire dados para plotar um gráfico de pizza com as porcentagens de cada estágio de maturação encontrados na imagem de análise. Além disso, utiliza a variável de retorno da função ‘mostrarClassificacao’, chamada ‘tela’, para exibir a imagem original sobrescrita com cores de identificação definidas para as regiões com determinada classificação. No *script* também é inserido marcador de tempo de execução para avaliação (comando ‘tic’ para início de contagem de tempo e ‘toc’ para encerrar a contagem). As Figuras 16, 17 e 18 mostram as linhas de código.

Figura 16 – ‘Principal.m’ (1ª à 28ª linha do código)

```

Principal.m X +
1  %script principal
2  %Limpando as variáveis
3  - clear all;
4  %Limpando a tela de comandos
5  - clc;
6  %iniciando a contagem de tempo da duração do algoritmo
7  - tic
8  %mudança de pasta para a pasta do projeto
9  - cd C:\Users\User\Desktop\Projeto
10
11 %chamada de função utilizando os nomes das pastas do banco de dados
12 %referentes a cada grau de maturação
13
14 %'matrizX' sendo X referente ao estágio de maturação correspondente
15 %possui os dados rgb das imagens da pasta de nome X
16 %rotulo... é um vetor com mesma quantidade de linhas da matrizX
17 %utilizado para criação dos rotulos numéricos de cada estágio
18 %correspondente
19 - [matrizVerdeEcana, rotuloVC] = XY('verdeEcana');
20
21 - [matrizCereja, rotuloCe] = XY('cerejaVermelho');
22
23 - [matrizCerejaA, rotuloCeA] = XY('cerejaAmarelo');
24
25 - [matrizPassa, rotuloP] = XY('passa');
26
27 - [matrizSeco, rotuloS] = XY('seco');
28

```

Fonte: Captura de tela, dos autores (2023).

Figura 17 – ‘Principal.m’ (29ª à 57ª linha do código)

```

29 %criação da matriz de elementos rgb com todos os estágios de maturação
30 aux1 = [matrizVerdeEcana; matrizCereja; matrizCerejaA; matrizPassa; matrizSeco];
31 x = double(aux1);
32
33 %atribuicao de rotulos para cada grau de maturação
34 %1 para verde e cana, 2 para cereja vermelho, 3 para cereja amarelo
35 %4 para passa, 5 para seco
36 aux2 = [rotuloVC; 2*rotuloCe; 3*rotuloCeA; 4*rotuloP; 5*rotuloS];
37 y = double(aux2);
38
39 %aquisição das informações da pasta 'imagemTeste'
40 pasta = dir('C:\Users\User\Desktop\Projeto\imagemTeste');
41 %mudança de diretorio para a pasta em questão
42 cd C:\Users\User\Desktop\Projeto\imagemTeste
43 %leitura do nome do arquivo da imagem de análise para uso posterior
44 imagem = pasta(3).name;
45 %mudança de pasta para a pasta do projeto
46 cd C:\Users\User\Desktop\Projeto
47
48 %chamada da função de classificação via knn com os dados do
49 %banco de dados (valores rgb, vetor x)
50 %e seus rotulos atribuidos (vetor y)
51 %além da imagem a ser classificada
52 [matrizClassificacao] = Classificacao(imagem, x, y);
53
54 %chamada de função que retorna a porcentagem em cada grau de maturação
55 %e retorna informações para visualizar o local dos pixels classificados
56 [pVerdeEcana, pCereja, pCerejaA, pPassa, pSeco, tela] = mostrarClassificacao(imagem, matrizClassificacao);
57

```

Fonte: Captura de tela, dos autores (2023).

Figura 18 – ‘Principal.m’ (58ª à última linha do código)

```

58 labels = {};
59 dados = [];
60 cor = []; %criação de labels para o gráfico de pizza conforme as porcentagens
61 if pVerdeEcana~=0 %verificação da existencia da porcentagem para gerar gráfico condizente
62 labels = horzcat(labels,'verde e Cana');
63 dados = horzcat(dados, 100*pVerdeEcana);
64 cor = cat(1,cor, [0 0.49 0]);
65 end
66 if pCereja~=0
67 labels = horzcat(labels,'cerejaVermelho');
68 dados = horzcat(dados, 100*pCereja);
69 cor = cat(1,cor, [0.78 0 0]);
70 end
71 if pCerejaA~=0
72 labels = horzcat(labels,'cerejaAmarelo');
73 dados = horzcat(dados, 100*pCerejaA);
74 cor = cat(1,cor, [1 1 0]);
75 end
76 if pPassa~=0
77 labels = horzcat(labels,'passa');
78 dados = horzcat(dados, 100*pPassa);
79 cor = cat(1,cor, [0.49 0 0.49]);
80 end
81 if pSeco~=0
82 labels = horzcat(labels,'seco');
83 dados = horzcat(dados, 100*pSeco);
84 cor = cat(1,cor, [0.19 0.19 0.19]);
85 end
86 %chamada da função de exibição do gráfico de pizza com os labels
87 pie(dados);
88 colormap(cor);
89 title('Porcentagem de frutos em cada estágio de maturação');
90 legend(labels,'Location','southoutside','Orientation','horizontal')
91 %exibição da tela com o local dos pixels classificados
92 figure()
93 imshow(tela);
94 %fim da contagem de tempo do algoritmo
95 toc

```

Fonte: Captura de tela, dos autores (2023).

2.2.3.2.8 Execução do programa

Para exemplificar os resultados, uma imagem de análise foi escolhida para ser classificada pelo programa. Figura 19: arquivo da imagem de análise com 720x1280 pixels, a imagem abaixo é visualização da imagem original reduzida para exibição neste documento.

Figura 19 – Imagem de análise



Fonte: Gonzaga Treinamentos e Consultoria Agrícola Ltda. (2021).

Para fins de fluidez de execução, a imagem foi redimensionada para dimensões menores, e as bordas não relacionadas aos frutos foram recortadas; em seguida, foi colocada dentro da pasta 'imagemTeste' no diretório 'Projeto'. Figura 20: arquivo da imagem de análise com 50x69 pixels, a imagem abaixo é a visualização da imagem redimensionada e ampliada para exibição neste documento.

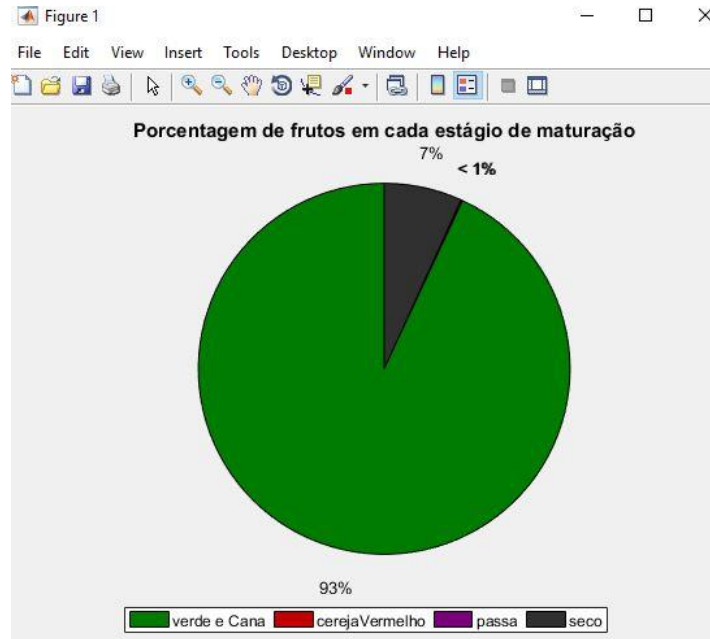
Figura 20 – Imagem de análise redimensionada



Fonte: dos autores (2023).

Após o arquivo da imagem redimensionada ser movido para a pasta determinada, o *script* 'Principal' foi executado, resultando em um gráfico de classificação e na imagem sobrescrita com cores indicativas. A Figura 21 mostra o gráfico e as porcentagens encontradas.

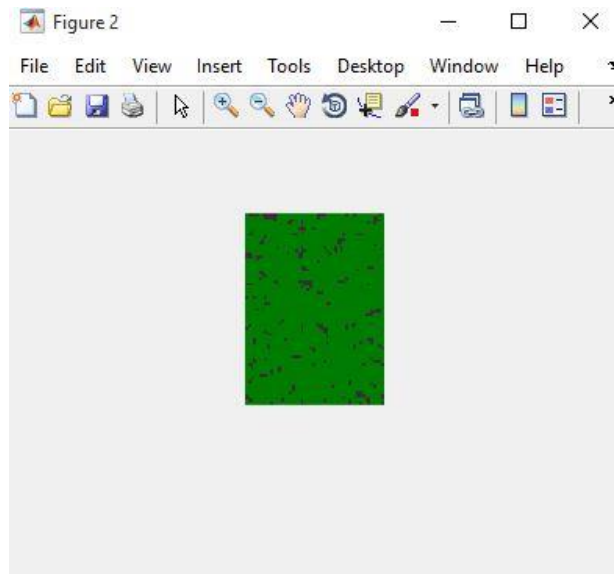
Figura 21 – Gráfico gerado para a imagem de análise redimensionada



Fonte: Captura de tela, dos autores (2023).

Figura 22: a região sobrescrita como verde indica a classificação do grau de maturação como sendo verde ou cana, os pontos escuros indicam classificação como secos, as demais cores são quase imperceptíveis (menos de 1%).

Figura 22 – Imagem de análise redimensionada e sobrescrita



Fonte: Captura de tela, dos autores (2023).

Figura 23: Tempo decorrido da execução de aproximadamente 56 segundos, o tempo de execução varia conforme a extensão do banco de dados, o tamanho da imagem analisada, os parâmetros da função KNN, dentre outros fatores.

Figura 23 – Tempo da execução do programa para o caso em particular

```
Command Window
Elapsed time is 55.785757 seconds.
fx >>
```

Fonte: Captura de tela, dos autores (2023).

É importante destacar que vários fatores influenciam no resultado obtido e no tempo de execução. Um banco de dados criteriosamente elaborado, com imagens bem representativas, leva maior acurácia ao resultado, assim como o fato de que a imagem de análise deve ser feita de forma a evitar sombras, reflexos de luminosidade na superfície dos frutos, e que o fundo de apoio ou bordas não sejam incluídos na mesma.

Para ilustrar o comportamento do programa com uma imagem com vários níveis de maturação, a seguinte imagem foi utilizada para análise após redimensionamento. Figura 24: mesmo arquivo da Figura 1, o arquivo possui 980x552 pixels, a imagem abaixo é a visualização da imagem reduzida para exibição neste documento.

Figura 24 – Segunda imagem de análise



Fonte: <https://www.cafepoint.com.br/noticias/tecnicas-de-producao/voce-sabe-o-tempo-ideal-de-maturacao-da-sua-lavoura-215991/> (03/01/2023).

Para inserção da imagem na pasta e posterior análise do programa, a imagem foi recortada de modo a retirar as bordas que exibem o fundo de apoio, além disso, o seu tamanho foi reduzido para 111x50 pixels. Figura 25: arquivo da imagem de análise com 111x50 pixels, a imagem abaixo é a visualização da segunda imagem redimensionada e ampliada para exibição neste documento.

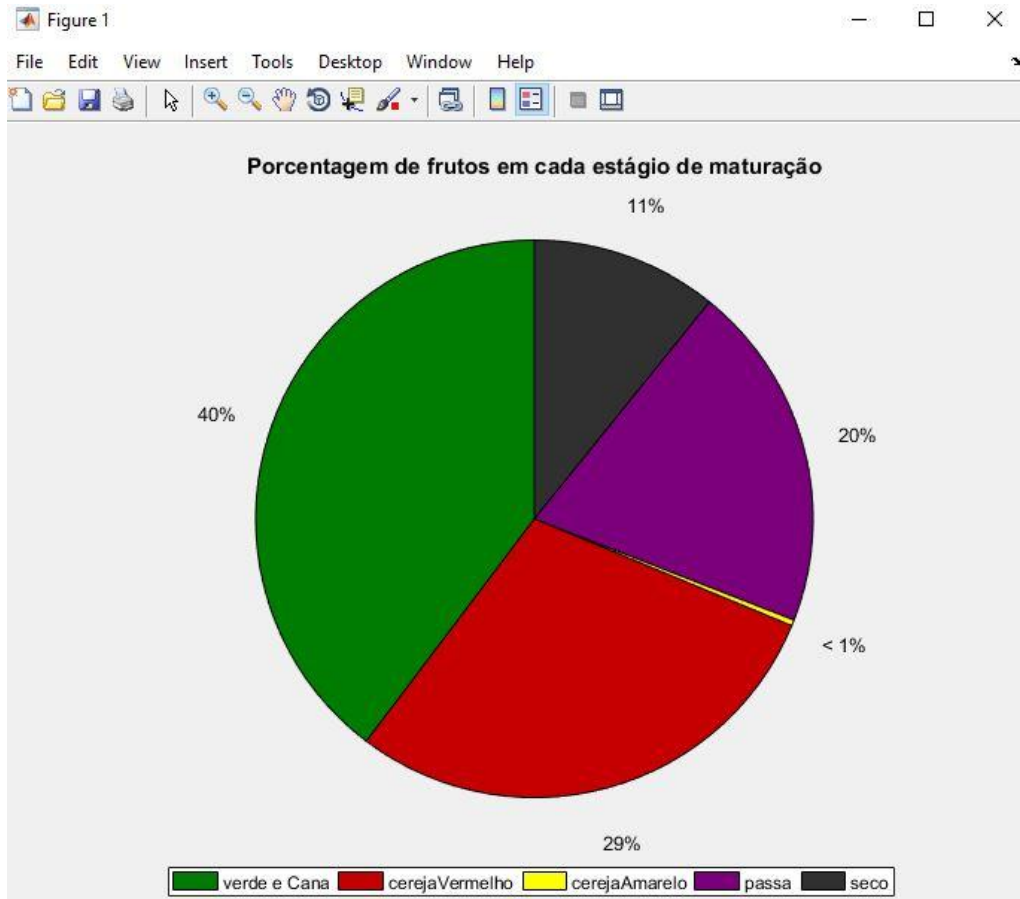
Figura 25 – Segunda imagem redimensionada



Fonte: dos Autores (2023).

Figura 26: mostra o gráfico e as porcentagens encontradas.

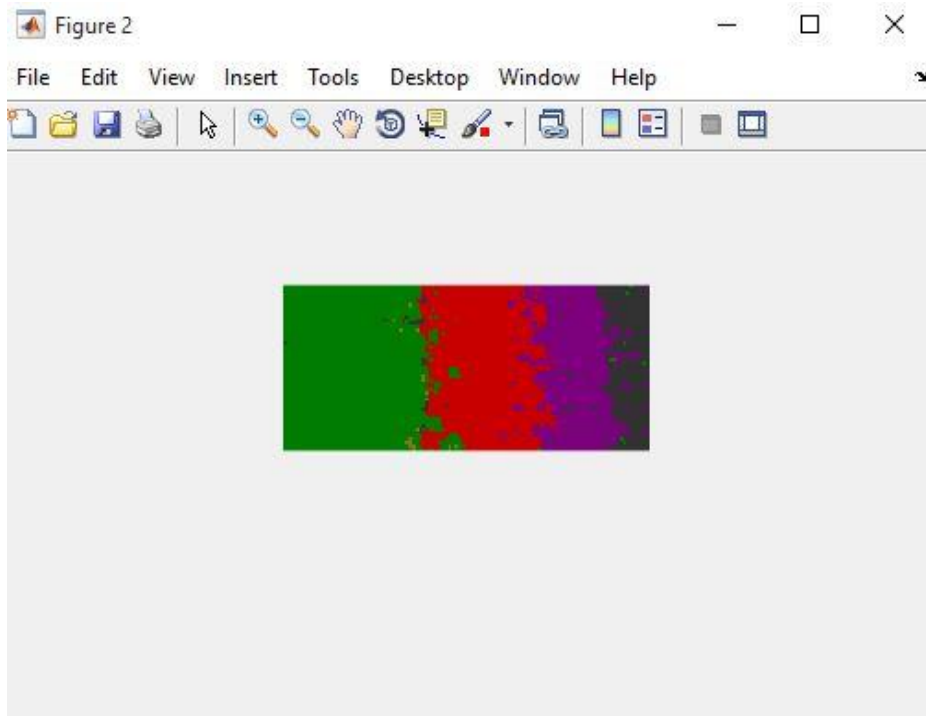
Figura 26 – Gráfico gerado pela segunda imagem



Fonte: Captura de tela, dos autores (2023).

Figura 27: mostra a sobreposição de cores que foi feita, segue as cores da legenda da Figura 26.

Figura 27 – Segunda imagem de análise redimensionada e sobrescrita



Fonte: Captura de tela, dos autores (2023).

Figura 28: mostra o tempo de execução para o segundo caso.

Figura 28 – Tempo da execução do programa para o segundo caso

```
Command Window
Elapsed time is 90.717800 seconds.
fx >>
```

Fonte: Captura de tela, dos autores (2023).

Portanto, para o mesmo banco de dados, as imagens analisadas obtiveram as classificações mostradas nas figuras anteriores e os respectivos tempos de execução.

2.3 Levantamento dos Custos da Alternativa

Para o desenvolvimento de um aplicativo para computadores pessoais ou dispositivos móveis, são usualmente utilizadas plataformas de desenvolvimento de aplicativos que contam com ferramentas úteis como bibliotecas, funções e possibilidade de implementação de interfaces gráficas amigáveis e intuitivas.

Após uma breve pesquisa, foi constatado que a linguagem de programação Python é uma alternativa eficiente para implementação de programas como o que foi proposto nesse trabalho. O programa que foi implementado no *software* Matlab pode ser transposto para a linguagem Python ou desenvolvido de forma otimizada, por se tratar de uma alternativa que conseguiria desenvolver um aplicativo utilizável em diferentes sistemas operacionais diretamente em um programa executável (sem necessidade de compra da licença do Matlab apenas para uso do programa, visto que o *software* contém inúmeras funcionalidades a mais que o necessário para o projeto).

Para a programação em Python é necessário a instalação de uma IDE (*‘Integrated Development Environment’*, ou Ambiente de Desenvolvimento Integrado) compatível à linguagem, além de uma plataforma específica para desenvolvimento de aplicativos. Além da IDE e da plataforma de desenvolvimento de aplicativos, pode-se contabilizar o custo de aquisição de um computador comum para trabalho.

Outro ponto que deve ser considerado é o valor de desenvolvimento do aplicativo em si, que é estimado pela quantidade de horas trabalhadas na implementação do mesmo multiplicado pelo valor cobrado por hora pelo profissional que o está desenvolvendo.

2.3.1 Materiais Necessários:

- Computador pessoal para desenvolvimento.
- IDE para programação em Python.
- Plataforma de desenvolvimento de aplicativos.

Os preços da IDE e plataforma de desenvolvimento foram retirados dos sites referentes, a IDE PyCharm disponível em <https://www.jetbrains.com/pt-br/pycharm/buy/#commercial>, e plataforma de desenvolvimento Kivy disponível em <https://kivy.org/>, ambos acessos em 06/03/2023.

Tabela 1 - Materiais necessários

ITEM	DESCRIÇÃO	VALOR (R\$):
Computador pessoal para desenvolvimento.	Notebook Samsung Core i5-1135G7 8GB 256GB SSD Tela Full HD 15.6" Windows 11 Book NP550XDA-KH2BR	3183,00
IDE para programação em Python	PyCharm, IDE Python para desenvolvedores profissionais	128,72 (Licença mensal)
Plataforma de desenvolvimento de aplicativos.	Kivy, plataforma desenvolvimento de aplicativos Python de código aberto.	Gratuito.

Fonte: Dos autores (2023).

Portanto, para que seja possível o desenvolvimento do aplicativo, serão necessários R\$ 3311,72 de investimento em materiais e licenças de *softwares*.

2.3.2 Serviços

Para levantamento do custo foi considerado um profissional da área de desenvolvimento de aplicativos, a categoria de desenvolvedor Full Stack. Segundo o *site talent*, disponível em <https://br.talent.com/salary?job=desenvolvedor+full+stack#:~:text=O%20sal%C3%A1rio%20m%C3%A9dio%20de%20desenvolvedor,a%20ganhar%20R%2496.000%20anuais> (acesso em 05/03/2023), esse tipo de desenvolvedor de aplicativos ganha em média R\$ 30,77 por hora (levantamento do valor em 2023). Foi estimado que para desenvolver o aplicativo em questão seriam gastos em torno de 40 horas, considerando a elaboração de um banco de dados, tratamento de informações, implementação de códigos, além da construção de interfaces para o usuário. Considerando isso, o valor total dos serviços prestados é de R\$ 1230,80.

Tabela 2 - Investimentos necessários

INVESTIMENTO:	VALOR (R\$)
Materiais	3311,72
Serviços	1230,80
	Total: 4542,52

Fonte: Dos autores (2023).

2.4 Levantamento dos Benefícios

A utilização de um aplicativo para análise do grau de maturação do café possui algumas vantagens em relação à análise visual. Por exemplo, no sistema proposto, o encarregado da operação não precisa saber classificar os graus de maturação, uma vez que o próprio aplicativo é capaz de identificar e mostrar os resultados, sendo apenas necessário a habilidade de tirar uma amostra adequada dos frutos da lavoura e manusear uma câmera digital para obtenção da imagem para análise.

Em relação à classificação ‘fruto em fruto’ o uso do aplicativo tem a vantagem de agilizar o processo. Segundo um especialista, a classificação utilizando apenas uma mão de frutos dura em média 6 minutos (excluído o tempo de obtenção de amostra dos frutos), já a classificação com copo dosador de meio litro leva de 15 a 21 minutos. Além disso qualquer um dos dois procedimentos deve ser repetido ao menos 3 vezes. Utilizando o aplicativo cada operação de classificação necessita de aproximadamente um minuto e meio, desde a importação da imagem para análise no programa até a entrega dos resultados.

No caso da classificação com tábuas, segundo o especialista consultado, o processo é geralmente executado em até 10 minutos, e deve ser repetido como os anteriores. No entanto, assim como o método de utilizar uma mão de frutos, o espaço amostral escolhido é relativamente pequeno, portanto, o nível de representatividade das porcentagens encontradas é reduzido. Além disso, o encarregado deve estar apto a classificar visualmente os frutos, assim como no caso anterior. Então, como se pode verificar com o que foi dito, uma vantagem da classificação automatizada está relacionada ao tamanho do espaço amostral (que é tão grande quanto a quantidade de frutos presentes na imagem), além da classificação feita diretamente pelo aplicativo e da agilidade do processo.

Além disso, o benefício da análise em si é permitir que sejam tomadas decisões a respeito da época de colheita ideal, a fim de aumentar a qualidade do café colhido, e, com isso, a valorização do produto a ser vendido. Com o recurso da análise por aplicativo, a leitura da maturação geral da lavoura é facilitada, uma vez que várias análises em diferentes amostras e épocas podem ser feitas de forma mais ágil, auxiliando o produtor no gerenciamento da plantação.

Por fim, a automação do processo abre espaço para integração dos dados coletados em outros projetos relacionados à área, aumentando o leque de ferramentas de auxílio para os produtores de café.

2.5 Análise Econômica

Para a análise econômica a proposta é que o aplicativo desenvolvido tenha uma taxa mensal de R\$ 9,90 a fim de efetuar cálculos de viabilidade econômica. Os fluxos de caixa pós investimento inicial segue a lógica da quantidade de usuários multiplicadas pelo valor da taxa mensal do aplicativo.

A fim de demonstrar uma análise mais elaborada, foi realizada uma pesquisa que leva em consideração a estimativa mensal de colheita de café em 2023, uma vez que possivelmente a alta da demanda do aplicativo seguiria períodos que antecipam a colheita, visto que a funcionalidade proposta do aplicativo é justamente a análise da maturação para efetuar a colheita.

A quantidade de usuários nessa hipótese tenderia a crescer em períodos antes da colheita e decrescer (cancelamento de assinatura pelos usuários) em períodos depois da colheita.

Foi assumido que a alta na quantidade de sacas estimadas para serem colhidas de um mês para outro indica o período em que o número de usuários aumentaria, enquanto a baixa de sacas estimadas de um mês para outro indicaria uma queda no número de usuários.

Os dados da Tabela 3 foram encontrados no site da CONAB - Companhia Nacional de Abastecimento, 2023, e a região de análise foi escolhida como região sudeste.

Tabela 3 – Estimativa de colheita mensal em 2023 na região sudeste (em mil sacas)

	Mar.	Abr.	Mai	Jun.	Jul.	Ago.	Set.	Out.	Nov.	Dez.
Sudeste	116,4	1223,7	5950,8	14680,9	13748,0	9428,8	1739,6	110,2	29,1	-

Fonte: CONAB (2023).

Como a quantidade estimada de sacas não reflete necessariamente na quantidade de produtores atuantes no período, foi estimado que as quantidades de novos usuários no aplicativo seguiria crescente com proporção definida de forma razoável até o pico da produção do ano. Em seguida, poderia decrescer devido a cancelamento de assinaturas, dentre outros motivos, e tenderia a se estabilizar até novas colheitas. Um número inicial de 30 usuários foi arbitrado para fins de análise.

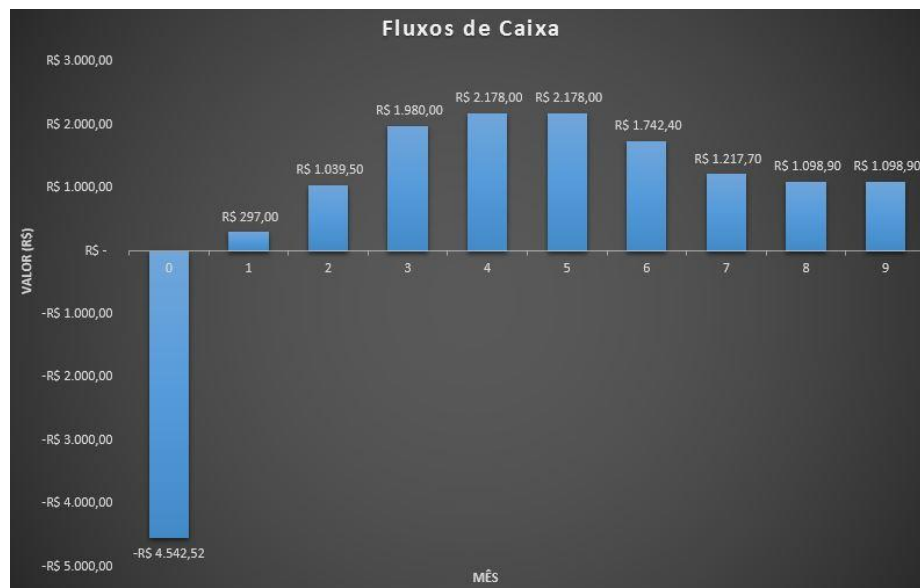
Tabela 4 – Dados da produção de café e projeção de mercado

	Mar.	Abr.	Mai	Jun.	Jul.	Ago.	Set.	Out.	Nov.
Proporção da produção de café	-	1223,7	5960,8	14680,9	13748,0	9428,8	1739,6	110,2	29,1
		÷	÷	÷	÷	÷	÷	÷	÷
		116,4	1223,7	5950,8	14680,9	13748,0	9428,8	1739,6	110,2
		=	=	=	=	=	=	=	=
	10,51	4,86	2,47	0,94	0,68	0,18	0,06	0,26	
Proporção de usuários estimada		3,5	1,9	1,1	1,0	0,8	0,7	0,9	1,0
Total de usuários do aplicativo (Números inteiros)		3,5	1,9	1,1	1,0	0,8	0,7	0,9	1,0
		*	*	*	*	*	*	*	*
	30	30	105	200	220	220	176	123	111
		=	=	=	=	=	=	=	=
	105	200	220	220	176	123	111	111	
Receita estimada em cada mês (R\$)	9,90	9,90	9,90	9,90	9,90	9,90	9,90	9,90	9,90
	*	*	*	*	*	*	*	*	*
	30	105	200	220	220	176	123	111	111
	=	=	=	=	=	=	=	=	=
	297,00	1039,50	1980,00	2178,00	2178,00	1742,40	1217,70	1098,90	1098,90

Fonte: Dos autores (2023).

Figura 29: Gráfico elaborado no *software* Excel

Figura 29 – Fluxos de caixa estimados



Fonte: Dos autores (2023).

2.5.1 Valor Presente Líquido VPL

O Valor Presente Líquido, VPL, é referente à soma do valor presente de todos os fluxos de caixa estimados no projeto de análise (GITMAN, 2017). É utilizado como indicador para avaliar a viabilidade econômica do investimento juntamente com outros fatores.

O Valor Presente, VP, por sua vez, é o valor que se espera obter no futuro trazido ao presente mediante uma taxa de desconto por período, a taxa de desconto é associada a um conceito chamado de Taxa Mínima de Atratividade, TMA, que será tratado posteriormente mais a fundo, mas que, a princípio, relaciona fatores que influenciam no valor efetivo do dinheiro empreendido ao longo do tempo.

Para aplicar os conceitos em dados numéricos, foi atribuído uma taxa de desconto mensal de 3%, em uma seção posterior será debatido a razoabilidade deste valor quando discutirmos a TMA.

A Equação 1 descreve a fórmula de VP:

$$VP = \frac{VF}{(1 + i)^n} \quad (1)$$

Onde ‘VP’ é o valor presente, ‘VF’ é o valor futuro, ‘i’ é a taxa de descontos e ‘n’ o período. A tabela 5 foi elaborada com o auxílio do *software* Excel.

Tabela 5 – VP de cada período e VPL do projeto

Taxa de Desconto (ao mês):		3%
Mês	Fluxos de Caixa	Valor Presente (VP)
0	-R\$ 4.542,52	-R\$ 4.542,52
1	R\$ 297,00	R\$ 288,35
2	R\$ 1.039,50	R\$ 979,83
3	R\$ 1.980,00	R\$ 1.811,98
4	R\$ 2.178,00	R\$ 1.935,12
5	R\$ 2.178,00	R\$ 1.878,76
6	R\$ 1.742,40	R\$ 1.459,23
7	R\$ 1.217,70	R\$ 990,10
8	R\$ 1.098,90	R\$ 867,48
9	R\$ 1.098,90	R\$ 842,22
Valor Presente Líquido (VPL) :		R\$ 6.510,56

Fonte: Dos autores (2023).

Como visto, o Valor Presente Líquido do projeto é igual a R\$ 6510,56, considerando uma taxa de desconto de 3% ao mês. A título de análises de viabilidade, um valor de VPL acima de zero é indicativo de provável viabilidade; no entanto, para fazer melhor estimativa, lançamos mão de mais análises, como vemos a seguir

2.5.2 Taxa Interna de Retorno TIR

O cálculo de VPL anterior releva uma taxa de desconto para o qual o resultado obtido para o VPL foi de R\$ 6510,56, ao aumentarmos a taxa de desconto proposta, o valor do VPL tenderá a diminuir, uma vez que haverá maiores descontos sobre os fluxos de caixa futuros do projeto, e, portanto, o valor líquido decresce. A Taxa Interna de Retorno, TIR, representa em outras palavras a taxa de desconto para o qual o VPL se torna igual a zero (CASTANHEIRA, 2016).

Equação 2, Relação dos Fluxos de Caixa, TIR, Investimento Inicial e os períodos

$$\sum_{i=1}^n \frac{FCi}{(1 + TIR)^i} - Investimento\ Inicial = 0 \quad (2)$$

Em que 'i' representa o índice do período.

'n' representa o último período.

FCi representa o fluxo de caixa no período i.

TIR a Taxa Interna de Retorno.

Para se obter o valor do TIR é necessário manipular a equação 1 de forma a se isolar a variável, considerando todos períodos em questão. A tabela 6 foi feita utilizando o *software* Excel.

Tabela 6 – Cálculo do TIR

Mês	Fluxos de Caixa
0	-R\$ 4.542,52
1	R\$ 297,00
2	R\$ 1.039,50
3	R\$ 1.980,00
4	R\$ 2.178,00
5	R\$ 2.178,00
6	R\$ 1.742,40
7	R\$ 1.217,70
8	R\$ 1.098,90
9	R\$ 1.098,90
TIR:	25%

Fonte: Dos autores.

Portanto o TIR encontrado para o caso de análise foi de 25% ao mês.

2.5.3 Taxa Mínima de Atratividade (TMA)

A taxa mínima de Atratividade, TMA, é um conceito aplicado em análises de viabilidade de um investimento. É utilizado como taxa de descontos no cálculo de VPL. Basicamente, a TMA leva em consideração três fatores: custo de oportunidade, risco e liquidez

O custo de oportunidade é estimado no retorno percentual do capital que poderia ser utilizado em um investimento de baixo risco, como aplicações no tesouro direto, dentre outros.

O risco é um indicador de volatilidade nos preços do ativo a ser investido, nesse caso é calculado utilizando a fórmula de desvio padrão amostral. O risco do investimento em um negócio pode ser estimado com base em parâmetros que situam o empreendimento no mercado geral, como análises de concorrência, sazonalidade do produto ou setor, ameaças externas à organização, dentre outros.

A liquidez pode ser descrita como a capacidade do investimento em retornar valores financeiros ao longo do tempo. Quanto maior a liquidez, mais rápido os retornos financeiros são obtidos.

Para atribuição de dados ao cenário, o custo de oportunidade foi estimado através da média da taxa Selic ao mês referentes ao ano de 2022.

Tabela 7 – Valores da taxa Selic ao mês no ano de 2022

Jan.	Fev.	Mar.	Abr.	Mai	Jun.	Jul.	Ago.	Set.	Out.	Nov.	Dez.
0,73%	0,76%	0,93%	0,83%	1,03%	1,02%	1,03%	1,17%	1,07%	1,02%	1,02%	1,12%

Fonte: <https://www.gov.br/receitafederal/pt-br/assuntos/orientacao-tributaria/pagamentos-e-parcelamentos/taxa-de-juros-selic#Selicmensalmente> (acessado 07/03/2023).

Valor médio da taxa = 0,9775% ao mês.

Com esse valor médio da taxa, o custo de oportunidade considerado para contribuir à TMA do investimento foi de 0,98% ao mês.

Os outros dois componentes foram estimados para se calcular o VPL de forma tal que seus valores fossem próximos ao custo de oportunidade, uma vez que o próprio custo de oportunidade define o limite inferior para a análise quantitativa em um estudo de viabilidade.

No projeto em questão, a contribuição dos fatores risco e liquidez, junto ao custo de oportunidade, definiram um valor de TMA de 3% ao mês.

2.5.4 *Payback*

O *payback* é um conceito que indica o tempo necessário para que o investimento seja restituído através dos fluxos de caixa futuros (ASSAF NETO, 2009).

O cálculo de *payback* pode ser feito através da Equação 3:

$$\textit{Payback} = \frac{\textit{Investimento Inicial}}{\textit{Fluxo de caixa regular}} \quad (3)$$

O investimento inicial do projeto foi de R\$ 4542,52

O fluxo de caixa regular foi estimado como a média de todos os fluxos de caixa futuros em valores presentes, totalizando valor aproximado de R\$ 1228,12.

Com esses dados o tempo de *payback* encontrado foi de 3,70 meses. A 8 tabela foi elaborada no *software* Excel.

Tabela 8 – Tempo de *payback*

Mês	Valor Presente (VP)
1	R\$ 288,35
2	R\$ 979,83
3	R\$ 1.811,98
4	R\$ 1.935,12
5	R\$ 1.878,76
6	R\$ 1.459,23
7	R\$ 990,10
8	R\$ 867,48
9	R\$ 842,22
Média:	R\$ 1.228,12
Investimento Inicial:	R\$ 4.542,52
Payback (meses):	3,70

Fonte: Dos autores.

2.5.5 Resultados

Considerando o investimento inicial apresentado de R\$ 4542,52 e as projeções de fluxos de caixa, a viabilidade do projeto pôde ser conferida através dos indicadores de VPL e TIR, sendo que o Valor Presente Líquido foi positivo e obteve boa margem de receita: valor encontrado de R\$ 6510,56. Além disso, a Taxa Interna de Retorno calculada foi de 25%, valor bem maior que a Taxa Mínima de Atratividade estimada (3%). Essa situação certifica o projeto como viável, sendo que o investimento inicial é compensado em aproximadamente 3,7 meses (tempo de *payback*).

O valor do aplicativo pode sofrer alterações mediante a demanda do mercado e outros fatores. Pode-se, também, criar pacotes de utilização, para que uma conta associada tenha acesso a um número de aparelhos, dentre outras coisas. Além disso, a taxa mensal estipulada no texto foi para fins de projeção, é comum no mercado de *softwares* as licenças serem adquiridas em taxas anuais. Uma mescla de opções de pacotes para o aplicativo, tanto na forma de pagamento, quanto na possibilidade de outros dispositivos poderem ser usados com a mesma conta, pode tornar a solução ainda mais atrativa para o mercado.

3 CONCLUSÃO

A análise do processo de maturação do café, como foi mostrado, é de grande importância na tomada de decisões em uma lavoura. A identificação de um período ideal para uma colheita que é beneficiada pelos frutos em estágio de maturação valorizado, como é o caso do fruto cereja, é algo almejado pelos produtores. Como os procedimentos comumente executados são de análises visuais, sem nenhum tipo de automação, a proposta da análise apresentada nesse trabalho se torna interessante.

A factibilidade da implementação de uma análise baseada em imagens foi demonstrada ao longo do trabalho por meio de um programa feito no *software* Matlab e da demonstração de seu funcionamento.

A transposição do trabalho para outras linguagens de programação é possível e a implementação de um aplicativo comercializável é uma possibilidade para trazer algum nível de automação à análise do processo de maturação.

Através das projeções estimadas e das análises econômicas feitas, foram demonstrados que a viabilidade econômica foi alcançada, uma vez que os indicadores calculados supriram às condições de viabilidade do projeto, como foi exemplificado.

Além disso, o trabalho poderá servir de base para futuras melhorias na área e implementação de níveis de automação cada vez melhores no setor cafeeiro.

4 REFERÊNCIAS

ASSAF NETO, Alexandre. **Finanças Corporativas e Valor**. São Paulo: Atlas, 2009.

BOELL, M. G. *et al.* **Exploiting Feature Extraction Techniques for Remote Sensing Image Classification**. IEEE Latin America Transactions, v. 16, p. 2657-2664, 2018.

CAFÉ POINT. **Como determinar o ponto de colheita do café**. Disponível em < <https://www.cafepoint.com.br/colunas/educapoint/como-determinar-o-ponto-de-colheita-do-cafe-215307/> > Acesso em 22/02/2023.

CASTANHEIRA, N. P. **Cálculo aplicado à gestão e aos negócios**. 1. ed. Curitiba: Intersaberes, 2016.

CNABRASIL. **Café: colheita e pós-colheita**. Disponível em < <https://www.cnabrasil.org.br/assets/arquivos/191-CAF%C3%89.pdf> > Acesso em 18/02/2023.

CONAB - COMPANHIA NACIONAL DE ABASTECIMENTO. **Acompanhamento da safra brasileira de café**. Brasília, DF, v. 10, n. 1, jan. 2023.

FERREIRA JÚNIOR *et al.* **Characterization of the coffee fruit detachment force in crop subjected to mechanized harvesting**. COFFEE SCIENCE, V. 13, p. 71-79, 2018.

FERREIRA JÚNIOR *et al.* **"Efeito dominó": Colheita mecânica do café visando qualidade**. Revista Cultivar, V. 180, p. 28-31, 2017.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems**, 2nd ed. Ed. O'Reilly Media, 2019.

GITMAN, L. J. **Princípios de Administração Financeira**. 14 ed. São Paulo: Pearson, 2017.

HARRISON, Matt. **Machine Learning Pocket Reference: Working with Structured Data in Python**, 1st ed. Ed. O'Reilly Media, 2019.