

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS
GERAIS – CAMPUS BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

João Vitor Veloso Rodrigues

**COFFECOLOR: DESENVOLVIMENTO DE UM APLICATIVO
PARA DETECÇÃO DO GRAU DE TORRA DE CAFÉ
BASEADO NA COR**

BambuÍ - MG

2022

JOÃO VITOR VELOSO RODRIGUES

**COFFECOLOR: DESENVOLVIMENTO DE UM APLICATIVO
PARA DETECÇÃO DO GRAU DE TORRA DE CAFÉ
BASEADO NA COR**

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Bambuí - MG

2022

Catálogo na Fonte Biblioteca IFMG - Campus Bambuí

R696c Rodrigues, João Vitor Veloso.
CoffeColor: desenvolvimento de um aplicativo para detecção do grau de torra de café baseado na cor. / João Vitor Veloso Rodrigues. – 2022.
73 f.; il.: color.

Orientador: Prof. Dr. Marcos Roberto Ribeiro.
Trabalho de Conclusão de Curso (graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG, Curso Bacharelado em Engenharia de Computação, 2022.

1. Ponto de torrefação. 2. Escala Agtron. 3. Florestas aleatórias. I. Ribeiro, Marcos Roberto. II. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG. III. Título.

CDD 641.877

João Vitor Veloso Rodrigues

COFFECOLOR: DESENVOLVIMENTO DE UM APLICATIVO PARA DETECÇÃO DO GRAU DE TORRA DE CAFÉ BASEADO NA COR

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí para obtenção do grau de Bacharel em Engenharia de Computação.

Aprovado em 29 de novembro de 2022 pela banca examinadora:

Prof. Dr. Marcos Roberto Ribeiro – IFMG – Campus Bambuí – (Orientador)
Prof. Dr. Fábio Pereira Dias – IFMG – Campus Bambuí – (Coorientador)
Prof. Álvaro Antonio Fonseca de Souza – IFMG – Campus Bambuí
Prof. Ma. Paloma de Albuquerque Diesel – IFMG – Campus Bambuí



Documento assinado eletronicamente por **Marcos Roberto Ribeiro, Professor**, em 29/11/2022, às 12:38, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Fabio Pereira Dias, Professor**, em 29/11/2022, às 12:55, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Álvaro Antonio Fonseca de Souza, Professor Visitante**, em 29/11/2022, às 13:04, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Paloma de Albuquerque Diesel, Professora Substituta**, em 29/11/2022, às 13:46, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **1389644** e o código CRC **C14F56E2**.

RESUMO

O café é um dos produtos agrícolas de maior relevância para a economia do Brasil, e sua qualidade final é determinada por diversos fatores, sendo um deles a torra. A torra do café determina as características sensoriais do produto, proporcionando mudanças no sabor, aroma, textura e cor. Durante o processo de torra, a cor dos grãos é utilizada como critério para interromper o processo. O grau de torra, muitas vezes, é determinado de forma subjetiva pelo torrefador. Existem padrões de representação de grau de torra dos grãos aceitos internacionalmente, como a escala Agtron. Entretanto, a utilização da escala Agtron envolve a aquisição de equipamentos consideravelmente caros ou comparação visual com discos impressos, o que, novamente, leva ao problema da subjetividade. O presente trabalho propõe uma metodologia alternativa e automatizada para a análise da cor do café torrado em grãos, com base na escala Agtron, por meio de uma aplicação computacional para dispositivos móveis. Através de experimentos e otimizações, foi desenvolvido um algoritmo de classificação utilizando florestas aleatórias que obteve uma precisão de 92.1% na predição da cor da torra de café em grãos. Os resultados obtidos são promissores e representam uma alternativa viável de baixo custo para avaliar o café torrado em função da cor.

Palavras-chave: Ponto de Torrefação. Escala Agtron. Florestas Aleatórias. Aplicação para dispositivos móveis

ABSTRACT

Coffee is one of the most important agricultural products for the Brazilian economy and its final quality is determined by several factors, one of them being roasting. Coffee roasting determines the sensory characteristics of the product, providing changes in flavor, aroma, texture, and color. During the roasting process, the color of the beans is used as a criterion to stop the process. The determination of the roasting degree is often determined subjectively by the roaster. There are internationally accepted standards for representing the roasting degree of the grains, such as the Agtron scale. However, the use of the Agtron scale involves the acquisition of considerably expensive equipment or visual comparison with printed disks, which, again, falls into the problem of subjectivity. The present work proposes an alternative and automated methodology for the color analysis of roasted coffee beans, based on the Agtron scale through a computational application for mobile devices. Through experiments and optimizations, a classification algorithm was developed using random forests that obtained an accuracy of 92.1% in predicting the color of roasted coffee beans. The results obtained are promising and represent a viable low-cost alternative to evaluate roasted coffee according to its color.

Keywords: Roasting Point. Agtron Scale. Random Forests. Mobile Application

LISTA DE FIGURAS

Figura 1 – Curvas que representam as modificações dos atributos: acidez, aroma e corpo em função dos graus de torra	13
Figura 2 – Amostras de café nos diferentes níveis de torra, de acordo com os valores dos discos Agtron/SCAA	14
Figura 3 – Modelo de cor RGB	16
Figura 4 – Modelo de cor HSV	17
Figura 5 – Modelo de cor LAB	18
Figura 6 – Aplicação do filtro de suavização pela mediana	18
Figura 7 – Comparação na aplicação do filtro de equalização de histograma	19
Figura 8 – Comparação na aplicação de quantização de cores	20
Figura 9 – Exemplo de construção de uma árvore de decisão	22
Figura 10 – Diagrama de decisão do <i>Random Forest</i>	22
Figura 11 – Esquema de uma RNA simples (<i>Perceptron</i>)	23
Figura 12 – Esquema de uma <i>Perceptron</i> de múltiplas camadas	23
Figura 13 – Fluxo do processo de classificação da amostra	29
Figura 14 – Fluxo do processo de extração de características da amostra	29
Figura 15 – Principais etapas do desenvolvimento	33
Figura 16 – Seleção manual dos grãos	34
Figura 17 – Torrador de amostras	35
Figura 18 – Estrutura fixa	36
Figura 19 – Esquema lógico dos arquivos de controle	38
Figura 20 – Processo de corte das imagens	39
Figura 21 – Processo de conversão de imagem	40
Figura 22 – Precisão dos algoritmos em relação à variação de indicativo de claridade	43
Figura 23 – Tempo da execução dos algoritmos em relação à variação de indicativo de claridade	43
Figura 24 – Precisão dos algoritmos em relação à variação no espaço de cor do café	44
Figura 25 – Tempo da execução dos algoritmos em relação à variação no espaço de cor do café	45
Figura 26 – Precisão dos algoritmos em relação à variação na técnica de pré-processamento	46
Figura 27 – Matriz de confusão	47
Figura 28 – Diagrama de fluxo de dados da solução	50
Figura 29 – Estrutura de pastas do projeto	51
Figura 30 – Fluxo de navegação entre módulos	52

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivo geral	11
1.2	Objetivos específicos	11
1.3	Justificativa	11
2	REFERENCIAL TEÓRICO	12
2.1	Cafeicultura	12
2.2	Processo de torra do café	13
2.3	Processamento Digital de Imagens	15
2.3.1	<i>Espaços de cor</i>	15
2.3.2	<i>Pré-processamento de imagens</i>	17
2.4	Visão Computacional	20
2.5	Aprendizado de Máquina	20
2.5.1	<i>Árvores de decisão e florestas aleatórias</i>	21
2.5.2	<i>Redes neurais</i>	23
2.5.3	<i>Naive Bayes</i>	24
2.6	Desenvolvimento multiplataforma para dispositivos móveis	24
2.7	Estado-da-arte	25
3	METODOLOGIA	28
3.1	Classificação da pesquisa	28
3.2	Solução	28
3.3	Materiais e Tecnologias	29
3.3.1	<i>Tecnologias</i>	29
3.3.2	<i>Ambiente de experimento e desenvolvimento</i>	31
3.4	Método e procedimentos	32
3.4.1	<i>Metodologia de Desenvolvimento do Trabalho</i>	32
3.4.2	<i>Amostras de café</i>	33
3.4.3	<i>Processamento Digital de Imagens</i>	36
3.4.4	<i>Classificadores</i>	37
4	DESENVOLVIMENTO	38
4.1	Construção da base de dados	38
4.2	Experimentos para seleção de atributos	41
4.2.1	<i>Variação no indicativo de claridade do papel</i>	42
4.2.2	<i>Variação no espaço de cor do café</i>	44
4.2.3	<i>Experimentos com técnicas de pré-processamento</i>	45
4.2.4	<i>Otimização dos hiper-parâmetros</i>	48

4.3	Desenvolvimento da solução	48
4.3.1	<i>Servidor</i>	49
4.3.2	<i>Aplicação móvel</i>	50
4.3.3	<i>Experimento com tempo de classificação</i>	53
5	CONSIDERAÇÕES FINAIS	55
5.1	Trabalhos futuros	56
	REFERÊNCIAS	58
	APÊNDICE A – HISTÓRICO DO QUADRO <i>KANBAN</i>	64

1 INTRODUÇÃO

Segundo a Associação Brasileira da Indústria de Café (ABIC), o Brasil é o maior produtor e segundo maior consumidor de café no mundo (ABIC, 2021b). Inclusive, o ramo cafeeiro se mantém forte no país, mesmo com os impactos negativos causados pela pandemia da Covid-19. Conforme um estudo realizado pela ABIC, o consumo de café registrou um crescimento de 1,34% em 2020 (ABIC, 2021a). Esse estudo ainda constata que 82% do setor produtivo de café são compostos por micro e pequenas empresas, e 70% operam com administração exclusivamente familiar, sendo o café torrado e moído o produto de maior faturamento destas empresas.

De acordo com Eugênio *et al.* (2011), o processamento térmico do grão de café cru, chamado de torrefação, é de muita valia para a obtenção das características sensoriais do produto final. Isso porque esse processo propicia o desenvolvimento de sabores e aromas característicos do grão, bem como mudanças na cor e na textura, além de ajudar em etapas posteriores, como a moagem. A qualidade sensorial do produto, portanto, depende não somente da qualidade do grão cru, mas também do nível em que este é torrado. A determinação do grau de torra varia de acordo com as notas que o torrefador deseja realçar e é realizada de forma visual, sendo esta uma técnica subjetiva e dependente da experiência do torrefador (MELO, 2004).

Conforme afirma Melo (2004), o consumidor brasileiro acredita que café bom é forte e escuro. Entretanto, a torra mais escura pode ser utilizada para disfarçar defeitos em grãos de menor qualidade ou até procedimentos ilícitos, como adulterações, pois encobre partículas de outros materiais, que, torrados a ponto de carbonizar e misturados ao café em pó, não aparecem na fiscalização por métodos visuais, podendo até lesar a saúde do consumidor.

Para a padronização do grau de torra, a Associação Americana para Cafés Especiais (SCAA), em conjunto com a empresa norte-americana Agtron Inc., adotaram uma escala de representação de grau de torra dos grãos, com padrões aceitos internacionalmente. Essa escala inicia em 100 e, conforme o grau de torra aumenta, seu valor decai até zero (MELO, 2004).

A verificação da torra em relação à escala pode ser realizada por meio de equipamentos especializados, como espectrômetro infravermelho e colorímetros. Porém, tais métodos possuem custo elevado e são inacessíveis para as micro e pequenas empresas, que são maioria no Brasil. Outro método é a utilização do Agtron/SCAA *Roast Classification Color Disk*, que são discos de diferentes tonalidades de marrom associados ao número Agtron. Entretanto, a comparação visual dos grãos torrados com os discos leva novamente ao problema da subjetividade.

O presente trabalho propõe uma alternativa para análise de cor de torra do café por meio de uma aplicação computacional para dispositivos móveis. A ideia é utilizar fotografias obtidas com dispositivo próprio em conjunto com técnicas de Processamento

Digital de Imagens e Inteligência Artificial para detectar o grau de torra do café em relação à escala Agtron. Dessa forma, será possível auxiliar os torrefadores com a redução da subjetividade da análise visual, colaborando para o controle e garantia da cor do café torrado. Além disso, esse aplicativo poderá ser utilizado por consumidores finais, para conferência do grau de torra de seu café.

1.1 Objetivo geral

O principal objetivo do presente trabalho é propor uma metodologia alternativa e automatizada para a análise de cor do café torrado em grãos, com base na escala Agtron, por meio de uma aplicação computacional para dispositivos móveis.

1.2 Objetivos específicos

Para atingir o objetivo principal, foram necessários os seguintes objetivos específicos:

- criar uma base de dados de fotos de diferentes graus de torras de café;
- analisar e avaliar as melhores técnicas de Inteligência Artificial e Visão Computacional para classificar as fotos;
- integrar a técnica melhor avaliada em um aplicativo para dispositivos móveis.

1.3 Justificativa

O processo de torra dos grãos de café é uma parte importante do processo de beneficiamento do produto, pois auxilia na definição da qualidade da bebida. A torrefação é responsável pela obtenção das características sensoriais do produto final. Entretanto, existe uma clara dificuldade em se garantir um padrão de qualidade embasado na escala Agtron, seja pela subjetividade baseada na experiência ou preços elevados de equipamentos especializados. Assim, o presente trabalho visa criar uma alternativa para a análise de cor de torra do café com base na escala Agtron, pois, segundo Melo (2004), é imprescindível que o consumidor passe a receber informações sobre a qualidade do café quanto ao processo de torra e suas respectivas propriedades.

2 REFERENCIAL TEÓRICO

Este capítulo descreve os fundamentos teóricos para o entendimento do presente trabalho. A Seção 2.1 apresenta os conceitos que envolvem a cafeicultura e sua importância para o Brasil. Em seguida, a Seção 2.2 descreve os eventos que envolvem o processo de torra do café e sua classificação baseado na escala Agtron. Já as Seções 2.3 e 2.4 expõem os conceitos de Processamento Digital de Imagens e Visão Computacional, respectivamente. A Seção 2.5 expõe os principais conceitos sobre Aprendizado de Máquina e apresenta os algoritmos de classificação utilizados no presente trabalho. Em seguida, a Seção 2.6 trata do desenvolvimento multiplataforma para dispositivos móveis. Por fim, a Seção 2.7 apresenta estudos que estão intimamente ligados com o tema do presente trabalho e que, de alguma forma, puderam contribuir para seu desenvolvimento.

2.1 Cafeicultura

No Brasil, o café, além de ser um importante produto do agronegócio, é parte integrante da história econômica do país. A história do café no Brasil teve início por volta de 1727, chegando no norte do país pela Guiana Francesa. Desde então, o Brasil tem sido reconhecido como um grande produtor e exportador do produto, sendo a cafeicultura a principal fonte de renda de muitas famílias (ALVES, 2019). Segundo a ABIC (2021a), a evolução do consumo interno de café no Brasil está em constante crescimento, totalizando 21,2 milhões de sacas no ano de 2020, o que representa cerca de 13% da demanda mundial.

Outro ramo cafeeiro em ascensão cujo consumo aumentou em larga escala nos últimos anos foi o de café denominado “especial”. Trata-se de um produto que, além de possuir qualidade superior à dos cafés “tradicionais”, passa por uma avaliação de um *Q-Grader*, um profissional especializado em degustar e classificar cafés. Caso obtenha uma pontuação mínima de 80 pontos após a degustação, o café pode ser considerado especial. Além da avaliação da bebida propriamente dita, os grãos crus e torrados também são avaliados (SILVA, 2021). Deste modo, o produtor necessita estar atento em toda a linha produtiva, desde a colheita do grão até o tipo de preparo. O aumento no consumo desta categoria é um reflexo do perfil do novo consumidor de café, que valoriza a qualidade e demonstra disposição em pagar por ela, proporcionando maior rentabilidade aos produtores (FERREIRA; CARVALHO; NADALETI, 2021).

Uma das etapas cruciais no processamento do café, e que está intimamente ligada à qualidade do produto, é o processo de torrefação. Conforme Melo (2004) relata, esse procedimento está relacionado com as características de sabor, acidez, aroma e corpo da bebida. Além disso, a torra mais forte, que torna a coloração mais escura, pode esconder procedimentos ilícitos, como adulterações por substâncias que podem prejudicar a saúde do consumidor. Portanto, alegar que o café possui determinado grau de torra, com informações claras e precisas, agrega valor e serve como material para estratégias de

marketing, apresentando um produto de melhor qualidade para consumidor.

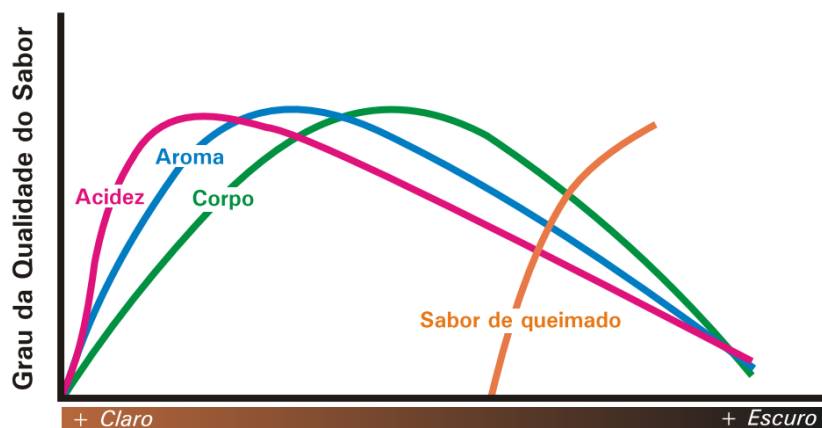
2.2 Processo de torra do café

De acordo com Lima *et al.* (2015), o processo de torra do café pode ser dividido em três etapas, que são a secagem, a pirólise e o resfriamento. Tal processo pode ser descrito como a passagem dos grãos crus por um aquecimento controlado, sendo desencadeadas reações responsáveis pelas características do produto final, como o sabor e o aroma que caracterizam a bebida do café.

As temperaturas, durante o processo de torra, podem variar de acordo com o modelo do torrador, tipo de transferência de calor e grau de torra desejado. O início do processo de torra é caracterizado pela etapa de secagem, quando ocorre a desidratação dos grãos e a mudança de cor, passando de verde para amarelo.

Na segunda etapa do processo de torra, ocorrem diversas reações que resultam na modificação da composição química e física dos grãos. Essas reações, como a caramelização, ou *maillard*, além de outras modificações químicas, contribuem para a alteração do sabor e aroma do café, como pode ser observado na Figura 1. Após os grãos atingirem a coloração desejada, a fim de evitar o escurecimento excessivo, a carbonização e a perda dos compostos de aroma e sabor, os grãos devem ser resfriados, interrompendo o processo de pirólise (MELO, 2004; PIRES, 2019; LIMA *et al.* 2015).

Figura 1 – Curvas que representam as modificações dos atributos: acidez, aroma e corpo em função dos graus de torra



Fonte: Adaptada de MELO, 2004.

O grau de torra pode ser verificado ou calculado por diversos métodos, sendo o mais popular a comparação visual dos grãos. Entretanto, esse método tem sua precisão comprometida, sobretudo pela subjetividade com que o avaliador determina os diferentes níveis de coloração, além da dificuldade de avaliar grãos de tonalidades intermediárias. Sendo assim, foram definidos os padrões de intensidade de torra como clara, média e escura. Outro método é a utilização de equipamentos especializados para a medição do grau de torra de café, como os da marca Agtron.

Os equipamentos Agtron utilizam um feixe de luz infravermelho que mede a refletância dos grãos e, em seguida, apresenta um valor numérico, fundamentado na escala de representação Agtron. A verificação também pode ser realizada através dos discos Agtron, que são acessórios em formato de disco que possui a coloração do café impressa, onde cada tom de cor tem uma numeração preestabelecida (MELO, 2004; WANG; LIM, 2015). A Tabela 1 apresenta os valores da escala Agtron para os níveis de torra de café, e a Figura 2, um exemplo da classificação.

Tabela 1 – Número do disco Agtron em função da descrição de cada nível de torra

Número do disco Agtron	Descrição do nível de torra
#95 a #85	muito clara a clara
#85 a #75	clara a moderadamente clara
#75 a #65	moderadamente clara a média clara
#65 a #55	média clara a média
#55 a #45	média a moderadamente escura
#45 a #35	moderadamente escura a escura
#35 a #25	escura a muito escura

Fonte: Adaptada de PIRES, 2019.

Figura 2 – Amostras de café nos diferentes níveis de torra, de acordo com os valores dos discos Agtron/SCAA



Fonte: Adaptada de PIRES, 2019.

Apesar da influência clara do grau de torra na qualidade da bebida, a sua determinação varia de acordo com os costumes de cada população. No Brasil, em sua grande maioria, os grãos são classificados entre torra clara, média e escura, sendo o grau de torra predominante o escuro (em torno do número Agtron 45). Entretanto, a torra escura favorece as fraudes, pois encobre partículas de outros materiais e substâncias que têm menor ou nenhum valor econômico, que podem não aparecer na fiscalização por métodos visuais (MELO, 2004).

Cada tipo de café possui uma numeração ideal de torra. Para os cafés tradicionais, a numeração se encontra entre 45 e 65 no disco Agtron; o café superior, 50 a

65; e o café *gourmet* deve apresentar o ponto de torra entre os números 60 e 65 no disco (BARACAT, 2020).

2.3 Processamento Digital de Imagens

Uma imagem digital pode ser representada como uma matriz, onde as linhas e colunas identificam um ponto na imagem, sendo *pixel* o termo mais amplamente usado para denotar esses elementos. Para imagens monocromáticas, seus valores são proporcionais à intensidade luminosa (brilho ou nível de cinza) na posição considerada. Já imagens coloridas podem ser representadas como a combinação individual de outras, com as informações de cada imagem variando de acordo com o espaço de cor utilizado (MARQUES FILHO; VIEIRA NETO, 1999; GONZALEZ; WOODS; EDDINS, 2003).

O processamento de imagens consiste na manipulação de uma imagem por computador, possibilitando desde operações simples, como alterações de cores e filtros de redução ruído, até operações complexas, como reconhecimento de padrões, análise e identificação de objetos. A área de Processamento Digital de Imagens vem apresentando crescimento expressivo, permitindo viabilizar aplicações em diversas outras áreas como Astronomia, Segurança, Publicidade, Direito, Biologia, Medicina, entre outras (MARQUES FILHO; VIEIRA NETO, 1999). Além disso, as técnicas de Processamento Digital de Imagens têm sido amplamente utilizadas na indústria de alimentos e agricultura, como evidenciado por Gomes e Leta (2012), Gonzalez Viejo *et al.* (2019) e Wu e Sun (2013), que apresentam revisões bibliográficas sobre inúmeros trabalhos que utilizam técnicas de visão computacional, combinação de espaços de cor e técnicas para medição de cor no controle e inspeção de qualidade de diferentes alimentos e ramos alimentícios.

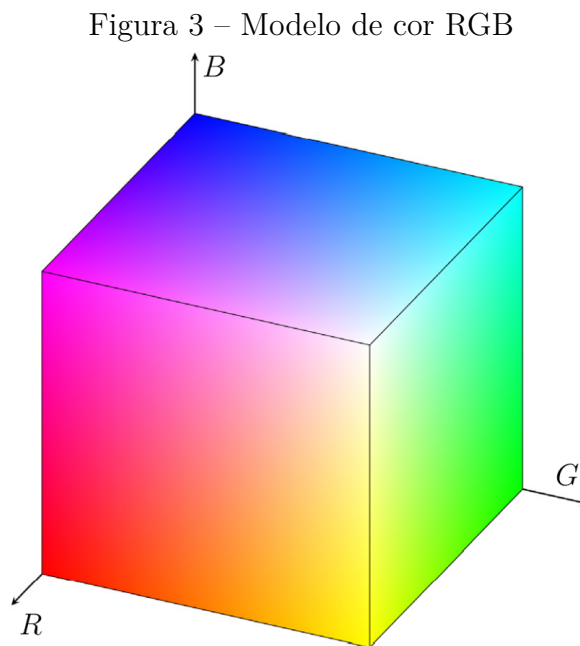
2.3.1 Espaços de cor

Os seres humanos percebem cores por meio de células fotorreceptoras (cones), onde as cores são distinguidas de acordo com o comprimento de onda da luz incidente. A cor azulada é identificada para os picos de comprimentos de onda de curta duração; para os de duração média, esverdeada; e longa, cores avermelhadas (HUNT; POINTER, 2011).

A sensação de cor é formada por uma mistura aditiva de três componentes de cor nos olhos, os quais são chamados de valores tristímulos. Estes valores são gerados com base nos três tipos de cones na medida em que cada um é estimulado (WU; SUN, 2013).

Para lidar matematicamente ou computacionalmente com as cores, foram criados os espaços de cor, que são modelos matemáticos cujo objetivo é associar os valores tristímulos com cada cor. Existem diversos deles, sendo que suas eficiências são dependentes das aplicações em que estão sendo empregados (GONZALEZ; WOODS; EDDINS, 2003; WU; SUN, 2013). Os três principais espaços de cor usados ao longo do presente trabalho foram o RGB, HSV e LAB.

No modelo RGB, as cores são representadas pelos canais R, G e B, que definem, respectivamente, a intensidade com a qual as cores primárias vermelho (*red*), verde (*green*) e azul (*blue*) estão presentes. Qualquer outra cor pode ser obtida pela mistura dessas três cores básicas. Conforme apresentado na Figura 3, este espaço de cor pode ser representado por um cubo em um sistema de coordenadas cartesianas tridimensional, sendo que cada um dos canais varia de 0 a 255. A cor preta está na origem do eixo, enquanto a cor branca está localizada no canto diagonalmente oposto. O restante dos vértices representa as cores primárias e secundárias (CHERNOV; ALANDER; BOCHKO, 2015).



Fonte: CHERNOV; ALANDER; BOCHKO, 2015.

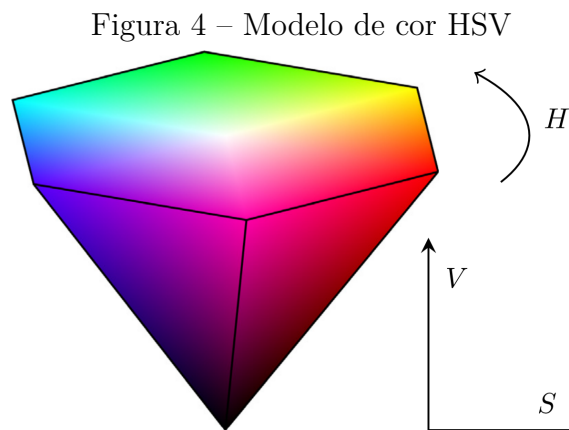
O RGB é importante por ser um dos principais espaços de cor, sendo utilizado em computadores, placas gráficas e monitores. Ele também é empregado no processo de captura de imagens de câmeras digitais. Assim, a grande maioria das imagens são representadas e armazenadas utilizando o RGB. Outra característica importante desse espaço de cor é que qualquer outro espaço de cor pode ser obtido a partir de uma transformação linear ou não linear de RGB (KOLKUR *et al.* 2017; GONZALEZ; WOODS; EDDINS, 2003).

Embora o espaço de cor RGB seja amplamente utilizado para a geração e exibição de imagens coloridas, existem outros espaços de cor, por exemplo, o HSV, que é frequentemente usado em Processamento Digital de Imagens e Visão Computacional (GONZALEZ; WOODS; EDDINS, 2003; CHERNOV; ALANDER; BOCHKO, 2015). O espaço de cor HSV decompõe uma cor em três canais: matiz ou *hue* (H), saturação ou *saturation* (S) e intensidade do valor ou *value* (V).

Os valores do matiz variam de 0° a 360° e dizem respeito ao tipo de cor, como vermelho, azul, amarelo, entre outras. A saturação indica a pureza da cor, e seus valores variam de 0 a 100%, sendo que, quanto mais baixa a saturação, menos se percebe a

presença do matiz na cor; logo, sendo representada mais desbotada e próxima do conjunto de cores da escala de cinza. Por fim, o canal de intensidade do valor está relacionado ao brilho da cor, onde cores mais claras possuem valores mais elevados de V, e cores escuras, valores mais baixos. Seu intervalo também varia de 0 a 100% (CHAVES-GONZÁLEZ *et al.* 2010).

A Figura 4 mostra uma representação dos componentes do HSV, onde o matiz é representado como uma formação cônica da roda de cores, e a saturação, pela distância do centro até a borda do cone. Já a intensidade do valor é a distância da extremidade pontiaguda do cone até o topo.



Fonte: CHERNOV; ALANDER; BOCHKO, 2015.

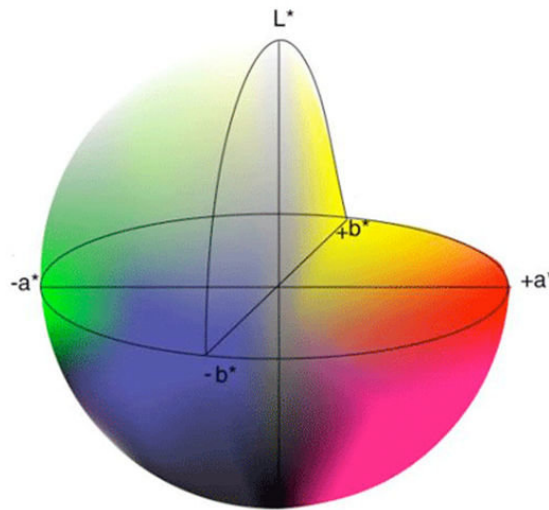
O modelo CIELab, ou $L^*a^*b^*$, representa as cores de acordo com a sua cromaticidade nos eixos x e y , e a sua luminância através do eixo z (Figura 5). Neste modelo, a cor também é formulada por três valores, onde a luminância (L) indica a luminosidade e brilho da cor. Esse valor é expresso em porcentagem, sendo zero para preto e 100 para branco. Os valores A e B variam entre o intervalo de -120 a 120, sendo informações sobre os espectros de cores. O valor A representa uma escala de matiz que varia de verde (valores negativos) a vermelho (valores positivos), e B varia de azul (valores negativos) a amarelo (valores positivos).

Apesar de não ser necessariamente aplicável em monitores e câmeras, o sistema CIELab é perceptivelmente uniforme, em que a distância euclidiana entre duas cores diferentes corresponde, aproximadamente, à diferença de cor percebida no olho humano (HUNT, 1991).

2.3.2 Pré-processamento de imagens

As principais técnicas de pré-processamento de imagens utilizadas no presente trabalho foram a técnica de suavização pela mediana e a técnica de equalização de histograma. Os filtros de suavização são utilizados para redução de ruído ou para criar um efeito de borramento nas imagens, resultando em perda de nitidez, ou seja, minimizando

Figura 5 – Modelo de cor LAB



Fonte: FARRAHI MOGHADDAM *et al.* 2013.

transições bruscas nos níveis de intensidade. Uma importante utilização dos filtros de média é a redução de detalhes que podem ser considerados irrelevantes em casos específicos (GONZALEZ; WOODS; EDDINS, 2003).

A suavização pela mediana é o mais conhecido dos filtros de estatística de ordem, cuja resposta se baseia na substituição do valor do *pixel* central pela mediana dos valores na vizinhança desse *pixel* (GONZALEZ; WOODS; EDDINS, 2003; OPENCV, 2015). Na suavização pela mediana, o elemento central é sempre substituído por algum valor de *pixel* que contém na imagem, ao contrário de outros efeitos que podem fazer com que o *pixel* assuma um novo valor (OPENCV, 2015). Um exemplo de aplicação do filtro de suavização pela mediana pode ser observado na Figura 6.

Figura 6 – Aplicação do filtro de suavização pela mediana



Fonte: Elaborado pelo Autor, 2021.

A equalização de histograma visa mudar a distribuição dos valores de intensidade de um histograma, tornando-os mais uniformes, permitindo uma redução das diferenças acentuadas (GONZALEZ; WOODS; EDDINS, 2003).

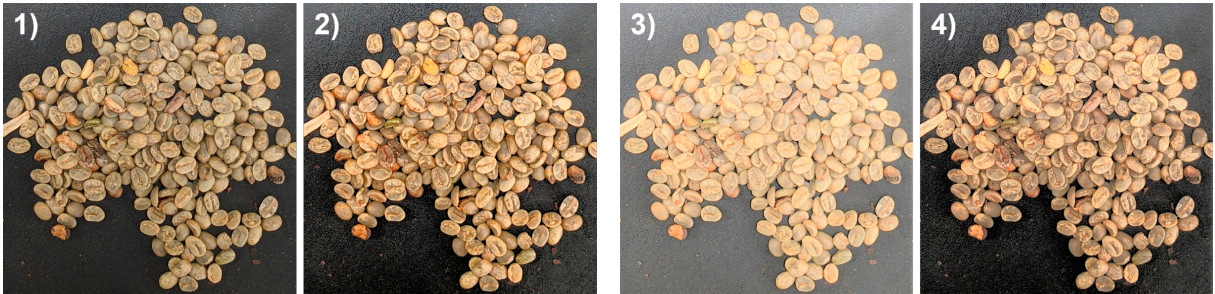
Um histograma de imagem pode ser descrito conforme a Equação (1), r_k é o

k -ésimo valor de intensidade, e n_k , o número de *pixels* na imagem com intensidade r_k . Os níveis de intensidade correspondem ao intervalo entre $[0, L - 1]$, sendo L o número máximo de valores de intensidade possível. Costuma-se normalizar um histograma dividindo cada um desses componentes pelo número total de *pixels* da imagem (GONZALEZ; WOODS; EDDINS, 2003).

$$h(r_k) = n_k \quad (1)$$

A Figura 7 possui um exemplo de aplicação do filtro de equalização de histograma em duas imagens iguais, porém, com intensidades de brilho diferentes. Pode-se observar que os resultados pós-equalização possuem cores menos discrepantes.

Figura 7 – Comparação na aplicação do filtro de equalização de histograma



(1) Imagem normal. (2) Resultado da aplicação do filtro de equalização de histograma na imagem normal. (3) Imagem mais clara. (4) Resultado da aplicação do filtro de equalização de histograma na imagem mais clara.

Fonte: Elaborado pelo Autor, 2021.

A quantização de cores é um processo que reduz o número de cores distintas em uma imagem, normalmente sem perda de qualidade ou de informações importantes. Este método é amplamente utilizado para exibir imagens que contêm muitas cores distinguíveis em dispositivos que permitem que apenas um número limitado de cores seja exibido simultaneamente (ORCHARD; BOUMAN, 1991).

O processo de quantização de cores é composto principalmente por duas fases. A primeira é chamada *design* da paleta de cores, onde ocorre a seleção de um pequeno conjunto de cores que melhor representa determinada imagem. A segunda fase é denominada como mapeamento de *pixels*, onde cada *pixel* da imagem é associado a uma das cores da paleta produzida, reduzindo, assim, o número de cores únicas (CELEBI, 2011).

A Figura 8 possui um exemplo de aplicação de quantização de cores em duas situações. A primeira situação, após a aplicação do algoritmo utilizando oito *clusters*, o número de cores da imagem foi reduzido para oito. A segunda situação, por sua vez, é um caso mais extremo, onde a imagem está sendo representada somente por duas cores. Pode-se observar uma redução significativa de cores distinguíveis, tornando a amostra mais homogênea.

Figura 8 – Comparação na aplicação de quantização de cores



(1) Imagem normal. (2) Resultado da aplicação de quantização com oito cores. (3) Resultado da aplicação de quantização com duas cores.

Fonte: Elaborado pelo Autor, 2021.

2.4 Visão Computacional

A Visão Computacional, área de pesquisa que envolve Processamento Digital de Imagens e Inteligência Artificial, corresponde ao processo de modelagem artificial da visão humana a partir de um conjunto de *hardware* e *software*, com o intuito de replicar suas funções. A sua utilização para medição de cor de alimentos tem sido promissora, tendo as vantagens de ser rápida, objetiva, econômica e não invasiva, pois as informações sobre a cor são extraídas de imagens digitais (WU; SUN, 2013).

Os sistemas que envolvem Visão Computacional, normalmente, são compostos por uma fonte de luz, um dispositivo de captura de imagem e um sistema computacional. Tal sistema possibilita o processamento e compreensão do conteúdo da imagem através da extração de características significativas para alcançar determinados objetivos, tais como classificação, avaliação da qualidade e detecção de defeitos (GOMES; LETA, 2012).

A Visão Computacional pode ser dividida em processos de baixo, médio e alto nível. Processos de baixo nível envolvem operações iniciais de pré-processamento da imagem, como remoção de distorções, redução de ruídos, aprimoramento de contraste e nitidez. Processos de nível médio envolvem tarefas como segmentação, que se propõe a particionar a imagem em regiões ou objetos de interesse. O processo de alto nível envolve a compreensão das características da imagem processada para um determinado fim, seja a identificação ou classificação de um padrão ou objeto (GONZALEZ; WOODS; EDDINS, 2003).

2.5 Aprendizado de Máquina

Aprendizado de máquina, assim como Visão Computacional, é uma subárea da Inteligência Artificial que se dedica a desenvolver métodos capazes de adquirir novos conhecimentos ou habilidades a partir de um conjunto de dados. No entanto, não existe

um único algoritmo com bom desempenho para todos os problemas (MITCHELL, 1997).

A escolha do algoritmo de aprendizado de máquina depende de diversos fatores, como o tipo de problema a ser resolvido, a natureza e a disponibilidade dos dados, seu desempenho, acurácia e/ou precisão. Existem diferentes abordagens de aprendizado. No presente trabalho, utilizou-se um dos tipos mais comuns de aprendizado supervisionado - a classificação, que, a partir de um conjunto de exemplos rotulados, buscou desenvolver um modelo que possa estimar rótulos para exemplos nunca vistos (CAMILO; SILVA, 2009).

As próximas seções apresentam os algoritmos utilizados no presente trabalho. A Seção 2.5.1 aborda os algoritmos, árvores de decisão e florestas aleatórias. A Seção 2.5.2 trata das Redes Neurais Artificiais *MultiLayer Perceptron*, e, por fim, a Seção 2.5.3 descreve o classificador *Naive Bayes*.

2.5.1 Árvores de decisão e florestas aleatórias

Uma árvore de decisão é basicamente um fluxograma estruturado como uma árvore, onde cada nó interno (nó não folha) denota um teste sobre um atributo, cada aresta representa uma saída do teste, e nós folhas representam os rótulos das classes (HAN; KAMBER; PEI, 2012).

Segundo Quinlan (1990), a ideia básica de uma árvore de decisão é utilizar a estratégia de dividir para conquistar, subdividindo-se o conjunto de dados em conjuntos cada vez menores e mais específicos, de modo que os novos subconjuntos tenham uma complexidade menor em relação ao anterior, até atingirem um tamanho simplificado o bastante para ser rotulado. Portanto, as árvores de decisão são construídas a partir de um conjunto de treinamento para o qual os rótulos de classe são conhecidos.

Kingsford e Salzberg (2008) afirmam que, em geral, as árvores de decisão são mais interpretáveis do que outros classificadores porque combinam perguntas simples sobre os dados de maneira compreensível, possuindo, normalmente, respostas categóricas ou de “sim ou não” para valores numéricos, como pode ser observado na Figura 9.

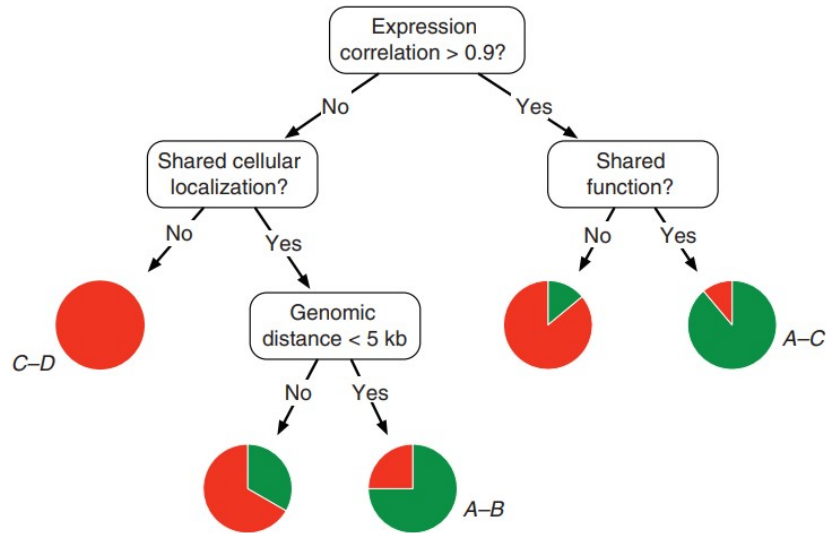
A classificação de um objeto inicia-se no nó raiz da árvore. A cada teste realizado, o resultado aponta para um nó filho que também possui um teste, e este processo é iterado até que se encontre um nó folha. Portanto, um objeto é classificado traçando-se um caminho da raiz da árvore de decisão até uma de suas folhas (QUINLAN, 1990; KINGSFORD; SALZBERG, 2008; HAN; KAMBER; PEI, 2012).

O classificador floresta aleatória, ou *Random Forest*, em inglês, é um algoritmo que utiliza uma técnica conhecida como *ensemble learning*, ou aprendizado conjunto, onde se combina um número finito de modelos para obter uma melhor performance preditiva. Essa característica torna esses algoritmos mais robustos e complexos, acarretando maior custo computacional, que pode ser acompanhado de melhores resultados (XU *et al.* 2019).

Breiman (2001) define *Random Forest* como um classificador constituído de uma coleção de classificadores estruturados em árvores de decisão, que são formadas por

Figura 9 – Exemplo de construção de uma árvore de decisão

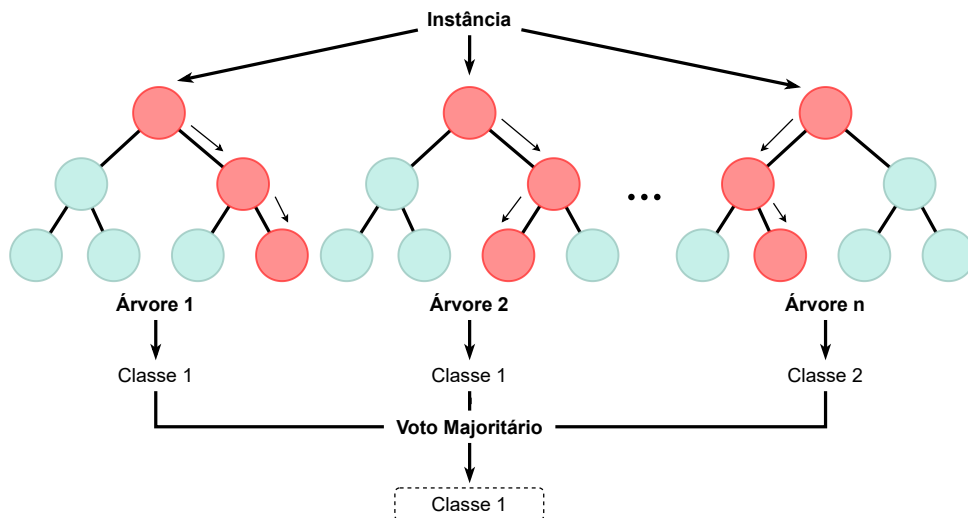
Gene Pair	Interact?	Expression correlation	Shared localization?	Shared function?	Genomic distance
A-B	Yes	0.77	Yes	No	1 kb
A-C	Yes	0.91	Yes	Yes	10 kb
C-D	No	0.1	No	No	1 Mb
⋮					



Fonte: KINGSFORD; SALZBERG, 2008.

um número finito de vetores aleatórios identicamente distribuídos amostrados a partir da base de dados. Durante uma classificação, cada árvore fornece um voto unitário para a classe mais popular de uma entrada, e os resultados individuais gerados por cada modelo são combinados em uma resposta final, conforme pode ser observado na Figura 10.

Figura 10 – Diagrama de decisão do *Random Forest*

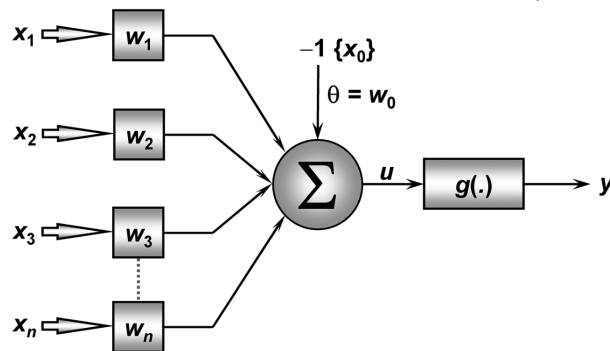


Fonte: Elaborado pelo Autor, 2021.

2.5.2 Redes neurais

Redes Neurais são sistemas computacionais com nós interconectados em redes que apresentam um modelo matemático inspirado no funcionamento dos neurônios do cérebro humano. O *Perceptron* é um modelo simplificado que simula um neurônio real. Sua representação pode ser vista na Figura 11, onde os dados relativos ao problema são inseridos e processados com um conjunto de parâmetros adaptáveis, atribuindo diferentes pesos para esses dados. Os resultados passam por funções degrau, linear ou sigmoide, conhecidas como função limiar de ativação de um nó, que permitem que certas entradas sejam validadas, resultando em uma solução (RAUBER, 2005; DA SILVA; SPATTI; FLAUZINO, 2019).

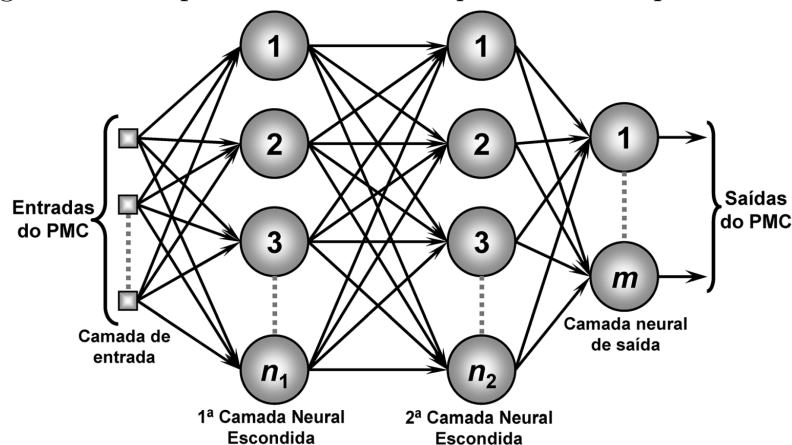
Figura 11 – Esquema de uma RNA simples (*Perceptron*)



Fonte: DA SILVA; SPATTI; FLAUZINO, 2019.

Quando uma rede neural possui pelo menos uma camada intermediária, é denominada como *Perceptron* de múltiplas camadas (PMC) ou *MultiLayer Perceptron (MLP)*. Conforme mostra a Figura 12, as Redes Neurais PMC são divididas entre várias camadas: a primeira é denominada camada de entrada; posteriormente, encontra-se uma ou mais camadas ocultas ou intermediárias; e, por fim, a camada de saída (RAUBER, 2005; DA SILVA; SPATTI; FLAUZINO, 2019).

Figura 12 – Esquema de uma *Perceptron* de múltiplas camadas



Fonte: DA SILVA; SPATTI; FLAUZINO, 2019.

2.5.3 Naive Bayes

Naive Bayes é um método de aprendizado de máquina que utiliza cálculos de probabilidade simples e métodos estatísticos. Tal método baseia-se no teorema de Bayes, demonstrado na Equação (2), onde se calcula a probabilidade de um evento, dado que outro evento já ocorreu, o que é chamado de probabilidade condicional.

O termo “*naive*” (ingênuo) diz respeito à forma como o algoritmo analisa as características de uma base de dados, desconsiderando a correlação entre as variáveis (RASCHKA, 2014).

$$P(A | B) = \frac{P(B | A) * P(A)}{P(B)} \quad (2)$$

Sejam A e B eventos, calcula-se a probabilidade posterior, dada por $P(B | A)$, probabilidade de B acontecer dado que A ocorreu. $P(A)$ é chamado de probabilidade anterior e representa probabilidade de A ocorrer. $P(B)$ é chamado de evidência e representa probabilidade de B ocorrer. Durante o treinamento, as probabilidades posteriores, $P(A | B)$, para cada combinação de A e B com base nos dados de treinamento, são calculadas.

Apesar de simples, *Naive Bayes*, frequentemente, compete bem com classificadores mais sofisticados, pois a decisão de classificação pode, muitas vezes, estar correta, mesmo que suas estimativas de probabilidade sejam imprecisas (RISH, 2001).

2.6 Desenvolvimento multiplataforma para dispositivos móveis

Os dispositivos móveis passaram a ter uma grande relevância no dia a dia das pessoas devido à sua versatilidade. Mesmo com a existência de vários sistemas operacionais para dispositivos móveis, o Android e o iOS são os mais utilizados. De acordo com IDC (2021), o sistema operacional Android está disponível em 83,8% dos dispositivos móveis mundiais, enquanto o iOS possui 16,2% de presença no mercado. No desenvolvimento de aplicações, é importante considerar métodos de desenvolvimento capazes de criar aplicações para diferentes plataformas, atingindo, assim, o maior número possível de usuários.

O desenvolvimento nativo de aplicações para determinado sistema operacional, geralmente, possui desempenho superior, além de proporcionar uma experiência de interface nativa para a plataforma de destino. Entretanto, esse desenvolvimento exige conhecimentos específicos para cada plataforma. Dada a complexidade, muitos desenvolvedores optam por *frameworks* multiplataformas, no intuito de desenvolver a aplicação somente uma vez. Isto permite a sua execução em diferentes sistemas operacionais através de uma única base de código, abrangendo um maior número de dispositivos compatíveis (EL-KASSAS *et al.* 2017; BIØRN-HANSEN; GRØNLI; GHINEA, 2018).

O kit de desenvolvimento de *software* livre e de código aberto Flutter¹, criado

¹ <https://flutter.dev/>

pela Google, viabiliza a criação de aplicativos para dispositivos móveis que operam utilizando os sistemas Android e iOS. O desenvolvimento de *softwares* para Windows, Mac, Linux e *Web* também é viável pela ferramenta, a partir de uma única base de código. O objetivo do Flutter é permitir que os desenvolvedores criem aplicativos de alta performance com uma experiência nativa (FLUTTER, 2021).

O Flutter utiliza o Dart² como linguagem de programação. O Dart é uma linguagem fortemente tipada, orientada a objetos, também desenvolvida pela Google. A utilização dessa linguagem permite que o mesmo código seja utilizado para gerar uma aplicação para as plataformas citadas anteriormente, além de ser possível observar o resultado instantaneamente durante o desenvolvimento (FLUTTER, 2021).

2.7 Estado-da-arte

O trabalho de Solmaz *et al.* (2018) desenvolveu um aplicativo para dispositivos móveis que, através de fotos capturadas com o próprio equipamento, fosse capaz de quantificar o conteúdo de peróxido em tiras de teste colorimétricas. Para a criação do conjunto de treinamento dos classificadores, as imagens foram capturadas com cinco dispositivos móveis diferentes, sob sete condições de iluminação distintas. Foram testados os classificadores *Least-Squares Support-Vector Machine* (LS-SVM) e *Random Forest*, utilizando-se os valores médios para os espaços de cor RGB, HSV e LAB. Para cada espaço de cor, Solmaz *et al.* (2018) também se realizou um teste onde foi aplicada a técnica de constância de cor *grey-world*, a fim de melhorar a detecção de cor, independentemente da fonte de luz. Nos testes realizados, a combinação do classificador LS-SVM, utilizando o espaço de cor HSV e aplicação da técnica *grey-world*, obteve uma média de 90% de precisão na classificação para cores primárias. À vista disso, o trabalho de Solmaz *et al.* (2018) contribui para uma maior compreensão de técnicas, espaços de cor e algoritmos que podem beneficiar o reconhecimento de cores em fotos capturadas com dispositivos móveis, resultando em uma possível melhora na classificação de cores do café.

Oblitas Cruz e Castro-Silupu (2014) projetaram um sistema para avaliar a cor de grãos de café torrados colocando a amostra em uma placa de Petri dentro de uma câmara com uma fonte de iluminação uniforme, gravando-a com uma câmera de vídeo. Foram obtidos das amostras os valores numéricos do espaço de cor CIELab, por meio de um algoritmo, utilizando-se o software Matlab. Uma observação pertinente foi em relação à diminuição do valor de L concordando com a mudança de cor para tons escuros. Portanto, ao observarem um padrão entre o tempo de torra, temperatura do torrefador e a coloração do café, torna-se viável a padronização e generalização da torra com base na cor, permitindo, também, uma análise em tempo real.

Dong *et al.* (2017) teve como objetivo, em seu trabalho, extrair as características

² <https://dart.dev/>

de cor e textura do chá associando-as com escores sensoriais de aparência de especialistas, que incluem cor, limpeza e uniformidade. Para a captura das imagens, foi utilizada uma câmera *reflex*, de lente única, (SLR) da marca Canon. Para a extração das características de cor e textura da imagem, empregou-se o *software* Matlab. Foram obtidas 15 variáveis características de cor e textura da imagem, sendo elas: média de canal vermelho (R), média do canal verde (G), média do canal azul (B), média de matiz (H), média de valor (V), média de saturação (S), média do componente de leveza (L^*), média de componente a (a^*) e componente b (b^*), e, com base nos atributos estatísticos de um histograma da escala de cinza, foram calculadas seis características de textura, como cinza médio (m), desvio padrão (δ), suavidade (r), terceiro momento (μ), uniformidade (U) e entropia (e). Após a extração dos dados, foram testados modelos lineares e não lineares, *Partial Least Squares regression* (PLS), *Extreme Learning Machines* (ELM) e AdaBoost. Os modelos de classificação não linear obtiveram melhores resultados devido a uma capacidade adaptativa e de generalização maior, sendo seu melhor resultado o coeficiente de correlação de previsão de 89%.

O estudo de Pires (2019) propôs três métodos alternativos de predição da cor do café torrado, sendo um deles um sistema de visão computacional utilizando redes neurais. Foram obtidos coeficientes de determinação maiores que 95% para os sete níveis de torra apresentados nos discos Agtron. As imagens foram capturadas utilizando-se um sistema de fotodocumentação especializado, que consiste em um gabinete fechado, com uma câmera digital na parte superior e lâmpadas LED de luz branca em seu interior. O referido estudo utilizou um *software* para a obtenção dos histogramas de cor da imagem, com os valores médios das variáveis RGB, desvio padrão de RGB, R, desvio padrão de R, G, desvio padrão de G, B e desvio padrão de B. Para a classificação dos dados, foram utilizadas Redes Neurais Artificiais de única camada oculta, composta por oito nós na camada de entrada e um nó na camada de saída. Com a finalização do estudo, foi desenvolvido um *software* denominado FRR 1.0 que é capaz de prever o valor Agtron das amostras de café torrado em grão ou moído. Para utilização do *software* em questão, devem ser obtidos os valores de histograma de cor, inseridos manualmente, retornando seu valor Agtron. Dessa forma, o trabalho de Pires (2019) contribuiu para uma maior compreensão de técnicas de obtenção e preparo das amostras de café, e possíveis conceitos podem ser utilizados para o desenvolvimento de base de dados.

Leme (2016) desenvolveu em seu mestrado um sistema integrado para classificação do café utilizando métodos instrumentais de análise digital de imagens por meio de visão computacional e reconhecimento de padrões. Outro objetivo em seu trabalho foi integrar as informações de análises físicas, químicas e sensoriais em uma única base de dados que é consumida por uma aplicação móvel. Para a captura das imagens, foi utilizada uma câmera digital modelo G-12/Canon acoplada em um sistema de fotodocumentação. Para a criação do sistema de visão computacional e de reconhecimento de padrões, utilizaram-se

duas redes neurais, uma responsável por converter os espaços de cores das imagens em RGB para CIELab, e outro modelo de regressão para estimação da tonalidade de torra dos cafés na escala Agtron. Na mensuração dos valores $L^*a^*b^*$, foi utilizado um colorímetro Minolta CR 300, e o *software* Matlab® foi empregado para a extração dos valores RGB da imagem e implementação da Rede Neural Artificial. Por fim, foi desenvolvido o modelo de regressão com a utilização dos valores CIELab, intensidade em escala de cinza e seu respectivo valor Agtron, medidos por um equipamento especializado (M-BASIC II). Leme (2016) obteve um excelente resultado, tendo um coeficiente de determinação de 0,99 para o erro médio quadrático para o café em grãos e moído. Neste trabalho, também foi desenvolvida uma aplicação para dispositivos móveis; entretanto, as informações são preenchidas de forma manual, através de caixas de texto.

O trabalho de Nasution e Andayani (2017), por sua vez, realizou um estudo onde foi possível classificar 16 níveis de grau de torra do café com uma precisão de 97,5%. As fotografias capturadas por um dispositivo móvel passaram por uma redução da dimensão, e os valores RGB da imagem foram transformados para a escala de cinza. Para a extração de características das imagens, utilizou-se o método matriz de coocorrência de níveis de cinza (*Gray Level Co-occurrence Matrix* - GLCM), e os dados extraídos foram normalizados. Por fim, na etapa de classificação, foram utilizadas redes neurais artificiais de multicamadas, mais especificamente o algoritmo de retropropagação de erro (*backpropagation*), que obteve uma boa taxa de precisão. Entretanto, nenhum *software* foi desenvolvido a partir deste estudo, e os padrões determinados de grau de torra, bem como a classificação das amostras, não seguem a escala Agtron. Assim sendo, o trabalho de Nasution e Andayani (2017) serviu de base para amadurecer conceitos e tecnologias que podem ser usados para a classificação do café.

3 METODOLOGIA

O capítulo atual tem como foco descrever a metodologia usada no desenvolvimento do presente trabalho. A Seção 3.1 trata da classificação da pesquisa. Em seguida, a Seção 3.2 explica a solução proposta para resolver o problema identificado. Já a Seção 3.3 aborda os materiais e tecnologias utilizados, e, por fim, a Seção 3.4 aborda os métodos e os procedimentos adotados.

3.1 Classificação da pesquisa

De acordo com as características relatadas por Wazlawick (2009), o presente projeto se enquadra no estilo de pesquisa “Apresentação de algo Presumivelmente Melhor”. Até o momento, apesar de existirem trabalhos que propõem metodologias alternativas para a análise de cor do café torrado, não foi desenvolvida uma aplicação para dispositivos móveis que desempenhasse a função de classificação do grau de torra considerando a escala Agtron.

Quanto à natureza, segundo os critérios estabelecidos por Gerhardt e Silveira (2009), o presente projeto se caracteriza como pesquisa aplicada, tendo como base resolver um problema específico da análise de cor do café torrado, gerando conhecimentos práticos e aplicáveis no âmbito da Cafeicultura.

Segundo Gil (2002), os trabalhos também podem ser classificados quanto aos objetivos e procedimentos. O trabalho em questão é definido como pesquisa exploratória e pesquisa-ação. A classificação como pesquisa exploratória acontece porque foram criadas hipóteses de que é possível realizar a classificação do café torrado por meio de imagens capturadas utilizando um dispositivo móvel. O trabalho é uma pesquisa-ação porque houve participação direta dos pesquisadores para o desenvolvimento de uma base de dados para os vários graus de torra do café e foram realizadas análises de trabalhos correlatos que auxiliam no entendimento e resolução do problema.

Por fim, segundo Gerhardt e Silveira (2009), o presente trabalho, em relação à abordagem, é caracterizado como quantitativo, pois possui a avaliação dos resultados mensurada com base na precisão do modelo de classificação.

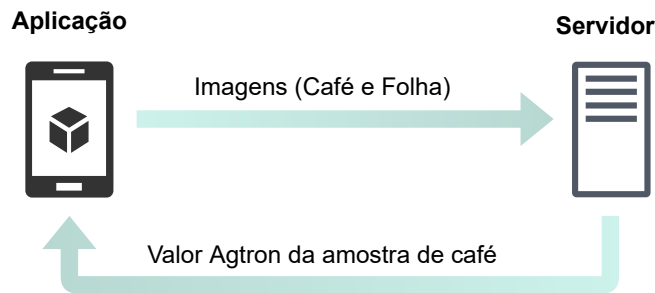
3.2 Solução

O projeto consiste no desenvolvimento de uma aplicação para dispositivos móveis que permitirá que os usuários classifiquem os grãos de café torrado de acordo com a escala Agtron. Juntamente com a classificação, o usuário receberá informações e características específicas sobre o grau de torra analisado.

Para a utilização da aplicação, o usuário poderá selecionar uma foto de sua galeria da amostra de café torrado em grãos e de uma folha de papel da cor branca que

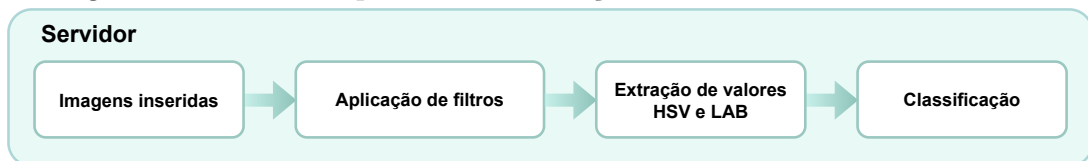
representará a condição de iluminação em que a amostra de café está exposta. Em seguida, a imagem da amostra é cortada e redimensionada para que somente o café seja selecionado. Após esse processo, ambas as imagens são enviadas para um servidor responsável por aplicar filtros que podem auxiliar na classificação, extrair características e classificá-las de acordo com a escala Agtron. Por fim, o servidor retorna a resposta para a aplicação, a qual irá apresentá-la para o usuário, conforme observado nas Figuras 13 e 14.

Figura 13 – Fluxo do processo de classificação da amostra



Fonte: Elaborado pelo Autor, 2021.

Figura 14 – Fluxo do processo de extração de características da amostra



Fonte: Elaborado pelo Autor, 2021.

3.3 Materiais e Tecnologias

A presente seção descreve quais tecnologias e ferramentas foram usadas para se desenvolver o trabalho. A Seção 3.3.1 apresenta as principais tecnologias utilizadas, e a Seção 3.3.2, o ambiente dos experimentos e os equipamentos usados.

3.3.1 Tecnologias

Como linguagem de programação base para o desenvolvimento dos algoritmos de processamento digital de imagens, aprendizado de máquina e servidor Web, utilizou-se Python¹. Em se tratando de uma linguagem de programação interpretada de alto nível, ela oferece suporte a vários paradigmas de programação. O Python pode ser utilizado para processamento de textos e protocolos de Internet, tendo grande importância na Computação Científica, em especial, nas áreas de Análise de Dados e Inteligência Artificial (PYTHON SOFTWARE FOUNDATION, 2021).

A *Open Source Computer Vision Library*² (OpenCV) foi utilizada para o pré-

¹ <https://python.org>

² <https://opencv.org>

processamento das imagens e extração das características para o desenvolvimento da base de dados. A OpenCV é uma biblioteca robusta de código aberto, com módulos focados em processamento de imagens, visão computacional, aprendizado de máquina, estrutura de dados, entre outros aspectos. Outra vantagem é o suporte multiplataforma para linguagens de programação como C++, Java e Python (OPENCV, 2021).

A biblioteca *Scikit-learn*³ foi empregada no desenvolvimento e avaliação dos classificadores, sendo desenvolvida para a linguagem de programação Python e sob a licença *Berkeley Software Distribution* (BSD) simplificada. O *Scikit-learn* é uma biblioteca que possui uma grande variedade de algoritmos de aprendizado de máquina (PEDREGOSA *et al.* 2011). Para a persistência dos modelos de classificação, foi utilizada a biblioteca Joblib⁴.

O *micro-framework* para Python, Flask⁵ foi empregado no desenvolvimento do servidor Web, que irá conter o classificador. Por padrão, o Flask visa manter o núcleo de funções simples, não incluindo uma camada de abstração de banco de dados, validação de formulário, entre outros adicionais, sendo ideal para pequenas aplicações. Entretanto, o *framework* possui diversas extensões para adicionar novas funcionalidades (FLASK, 2021).

Para o desenvolvimento da aplicação para dispositivos móveis que irá extrair características da imagem e se comunicar com o servidor, foi utilizado o kit de desenvolvimento Flutter⁶. Criado pela empresa Google, o Flutter possibilita a criação de aplicativos para diversas plataformas a partir de uma única base de código escrita em Dart⁷, sendo esta uma linguagem de programação fortemente tipada, orientada a objetos, também desenvolvida pela Google (FLUTTER, 2021).

Para a estruturação do código da aplicação móvel, utilizou-se a biblioteca Flutter Modular⁸, a qual se propõe a adicionar ao Flutter injeção de dependência modularizada, rotas modulares e uma estruturação divididas por escopo. Com essa estruturação, cada recurso da aplicação possui seu próprio MVC, *Model* (Modelo) *View* (Visão) e *Controller* (Controle), sendo benéfico para a escalabilidade e manutenção (FLUTTERANDO, 2022).

Para o envio das imagens da aplicação móvel para o servidor através de requisições HTTP, foi utilizada a biblioteca Dio⁹, a qual possui suporte para interceptadores, cancelamento de solicitações, download de arquivos e tempo-limite, entre outros aspectos. Em comparação com o pacote *http* padrão fornecido pelo Flutter, a Dio proporciona maneiras de executar tarefas avançadas de rede, como envio de arquivos, de forma mais simples (BISWAS, 2022).

³ <https://scikit-learn.org>

⁴ <https://joblib.readthedocs.io>

⁵ <https://flask.palletsprojects.com>

⁶ <https://flutter.dev>

⁷ <https://dart.dev>

⁸ <https://modular.flutterando.com.br/>

⁹ <https://pub.dev/packages/dio>

A coleção com todas as fotos capturadas foi disponibilizada no Kaggle¹⁰. Esta plataforma é uma comunidade online de ciência de dados e aprendizado de máquina. Os usuários que utilizam essa comunidade podem acessar e publicar conjuntos de dados e modelos de classificação.

O GitHub é uma plataforma de hospedagem de código que permite versionamento, gerenciamento do projeto, hospedagem de sites estáticos e mais. Assim, todos os códigos foram hospedados em uma organização nomeada CoffeeColor¹¹. Com o auxílio desta plataforma, também foi realizada a gerência do quadro *Kanban*, utilizado para aplicação de alguns conceitos do *Scrum*.

Para hospedagem da API Web desenvolvida, foi utilizado o serviço PythonAnywhere¹². Além de fornecer um ambiente de desenvolvimento que pode ser acessado por um navegador web, este serviço permite a hospedagem de projetos baseados em Python. O PythonAnywhere permite a hospedagem de uma única aplicação Web de forma gratuita com algumas limitações, podendo ser acessado através de um domínio informado pelo serviço (PYTHONANYWHERE, 2021).

3.3.2 Ambiente de experimento e desenvolvimento

Todo o desenvolvimento dos componentes da arquitetura proposta (aplicativo móvel, classificadores e API Web) foi realizado em um computador pessoal com as seguintes configurações:

Modelo: Dell Inspiron 15 7000 7567 Gaming.

Processador: Intel® Core™ i5-7300HQ @ 2.50GHz x 4.

Processador Gráfico: NVIDIA® GeForce® GTX 1050, 4GB, GDDR5.

Memória RAM: 16 GB, 2400MHz, DDR4.

Unidade de Estado Sólido: 240 GB.

Arquitetura do Sistema Operacional: 64-bit.

Sistema operacional: Windows 10 Pro.

Para preparação do ambiente de desenvolvimento, foram utilizadas as seguintes ferramentas:

Visual Studio Code: 1.62.2

Python: 3.9.8

¹⁰ <https://www.kaggle.com/>

¹¹ <https://github.com/coffeecolor>

¹² <https://www.pythonanywhere.com/>

OpenCV: 4.5.4.58

Scikit-learn: 1.0.1

Joblib: 1.1.0

Flask: 2.0.2

Flutter: 2.5.3

Dart: 2.14.4

Git: 2.33.0

3.4 Método e procedimentos

A presente seção descreve quais métodos e procedimentos foram seguidos para o desenvolvimento do trabalho. A sequência dos principais passos seguidos é apresentada na Figura 15. Inicialmente, foram efetuadas a coleta e a preparação das amostras. Depois disso, começou a etapa de construção da base de dados, composta pela implementação de algoritmos de processamento digital de imagens e extração de características. A seguir, ocorreram a seleção, treino, teste e avaliações de classificadores. Após, foi realizada a implementação do servidor Web que contém os algoritmos de pré-processamento e o modelo de classificação melhor avaliado. Por fim, a implementação da aplicação móvel, sendo cada um destes passos explicado detalhadamente nas seções a seguir.

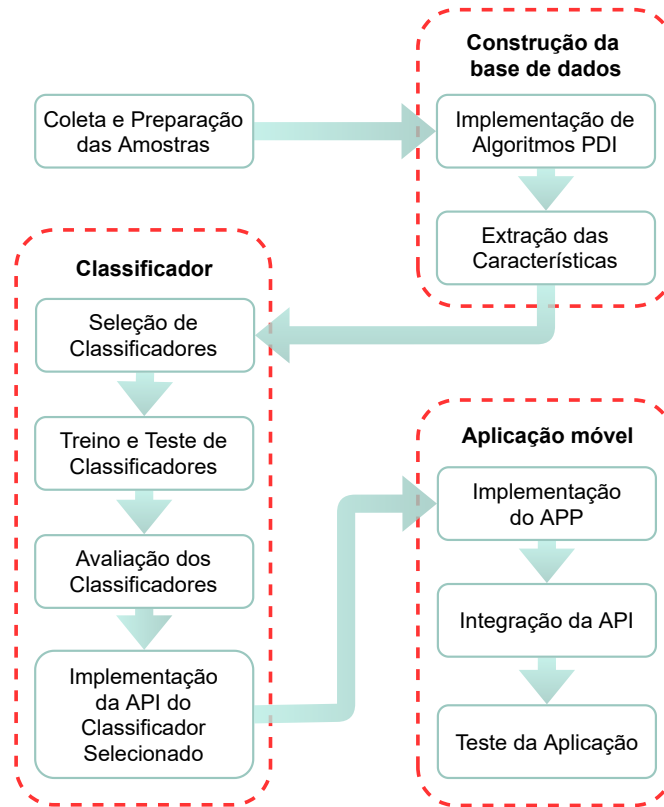
A Seção 3.4.1 descreve a metodologia de desenvolvimento empregada no projeto. Já a Seção 3.4.2 apresenta os passos seguidos para preparação das amostras de café. A Seção 3.4.3 exibe os passos seguidos para a extração de características das imagens, e, por fim, a Seção 3.4.4 expõe como os classificadores são avaliados.

3.4.1 Metodologia de Desenvolvimento do Trabalho

Para auxiliar no processo de desenvolvimento do trabalho, foi empregada a metodologia *Scrum* de desenvolvimento ágil. Segundo Sabbagh (2014), o *Scrum* se tornou a forma mais comum de se trabalhar em projetos de desenvolvimento de *software*, sendo utilizado também em diferentes mercados, como empresas de *marketing* e de desenvolvimento de *hardware*. Além disso, o autor evidencia a redução dos riscos do projeto, maior qualidade no produto gerado, visibilidade do progresso do projeto, redução do desperdício, aumento de produtividade, entre outros, como benefícios da utilização desta metodologia.

O desenvolvimento do projeto foi baseado em algumas ideias principais do *Scrum*. Os *Sprints* são ciclos de trabalho que, ao serem encerrados, geram valor ao produto final, sendo que, no presente projeto, foram divididos em ciclos de sete dias.

Figura 15 – Principais etapas do desenvolvimento



Fonte: Elaborado pelo Autor, 2021.

Para auxiliar no fluxo de desenvolvimento e na aplicação do *Scrum*, foi utilizado o GitHub¹³, que, além das funções de versionamento e hospedagem de repositórios, possui um quadro *Kanban*. O Apêndice A apresenta o quadro *Kanban* e suas alterações ao longo do desenvolvimento do projeto, as quais estão agrupadas quinzenalmente.

3.4.2 Amostras de café

O café utilizado é da espécie *Coffea arabica* L., oriundo da produção do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus Bambuí*, disponível no laboratório de Pós-Colheita e Qualidade do Café. O preparo das amostras do café foi realizado em três etapas. A primeira delas foi o processo de beneficiamento, que visa remover cascas, pergaminhos e as películas prateadas dos frutos, em que se utilizou um descascador de renda elétrico. A segunda etapa envolveu a separação dos grãos por tamanho e forma utilizando peneiras. Foram selecionados somente os grãos chatos retidos nas peneiras de crivos circulares com numeração igual a 16. E, por fim, efetuou-se a seleção manual de grãos sadios com propósito de eliminação de possíveis defeitos extrínsecos e intrínsecos, conforme a Figura 16.

O Programa de Qualidade do Café (PQC), emitido em 2004 pela ABIC, estabelece os requisitos técnicos para a concessão do Símbolo da Qualidade ABIC para

¹³ <https://github.com>

Figura 16 – Seleção manual dos grãos



Fonte: Elaborado pelo Autor, 2021.

café torrado em grão ou moído. Em relação à torra, a ABIC recomenda somente os cafés entre o moderadamente claro e moderadamente escuro (Agtron 75 e 45) (ABIC, 2018). Portanto, no desenvolvimento do projeto, os valores Agtron 95 e 85 não foram utilizados, e, mesmo que não recomendados, os valores 25 e 35 foram mantidos, pois, conforme Melo (2004) evidencia, o café forte e escuro é preferível por uma boa parte dos consumidores brasileiros, mas a torra mais escura pode ser utilizada para disfarçar defeitos em grãos de menor qualidade ou até procedimentos ilícitos.

As amostras de café foram classificadas em três conjuntos de aproximadamente 50,00 g para cada valor correspondente nos discos Agtron/SCAA. Posteriormente, foram submetidas ao processo de torra em um torrador de amostras modelo TP01 com alimentação a gás GLP (Carmomaq, Espírito Santo do Pinhal, São Paulo, Brasil) mostrado na Figura 17. Os diferentes níveis de torra dos grãos foram avaliados visualmente até atingirem a coloração desejada, com referência nos discos Agtron/SCAA. Para interromper o processo térmico, as amostras foram transferidas imediatamente para a bandeja de resfriamento do torrador, que possui um fluxo de ar frio.

As amostras de café torrado em grãos foram submetidas à verificação do valor Agtron por meio da comparação com os discos Agtron/SCAA. Após a análise, seus valores foram anotados e os grãos foram armazenados até a etapa de captura das imagens.

Para a captura das imagens, foram utilizados equipamentos de diferentes modelos, marcas e sensores, conforme podemos observar na Tabela 2. Para cada amostra de café, foram realizadas as capturas das imagens em duas condições de iluminação, em luz ambiente e com *flash* ativo. Para os aparelhos móveis, as configurações de imagem, como temperatura de cor, ISO, tempo de exposição e a velocidade do obturador, foram ajustadas automaticamente pelo *software* interno do aparelho.

Uma característica importante para padronização das imagens é a distância

Figura 17 – Torrador de amostras



Fonte: Elaborado pelo Autor, 2021.

Tabela 2 – Equipamentos em função da resolução da imagem e abertura do sensor.

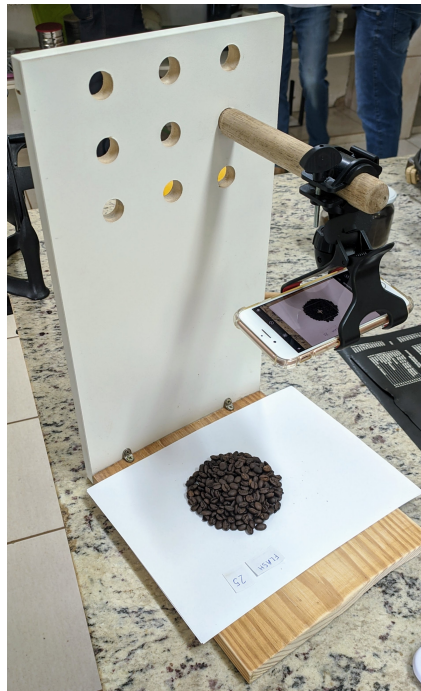
Equipamentos	Resolução da imagem	Abertura do sensor
APPLE iPhone 8	4032 × 3024	$f/1.8$
APPLE iPhone 11	4032 × 3024	$f/1.8$
APPLE iPhone 12	4032 × 3024	$f/1.6$
MOTOROLA X4	4032 × 3024	$f/2.0$
REDMI 5A	3264 × 2448	$f/2.2$
SAMSUNG Galaxy A72	4624 × 3468	$f/1.8$
XIAOMI Mi 9	4000 × 3000	$f/1.8$

Fonte: Elaborado pelo Autor, 2021.

do equipamento e posição do café. Para que todas as imagens fossem obtidas no mesmo padrão, uma massa de aproximadamente 50,00g das amostras de café torrado em grão foi disposta em um prato de plástico descartável com diâmetro de 15,00 cm e 1,00 cm de profundidade. Também foi planejada uma estrutura com as medidas de 27x24 cm na base e 43,00 cm de altura, com furos na lateral para que os equipamentos pudessem ser posicionados a 20,00 cm, 25,00 cm ou 30,00 cm de distância da amostra, conforme Figura 18.

Com o auxílio da função grade de alinhamento presente nos dispositivos, a amostra foi centralizada sobre as folhas visando destacar a amostra de café e facilitar a identificação da escala Agtron correspondente. No fundo da estrutura, foi posicionada uma folha de papel da cor branca, com o número Agtron e a condição de iluminação escritos em sua margem superior. Para cada condição de iluminação, foi realizada uma captura da folha de papel branca sem nenhuma amostra em sua parte superior, visando à identificação da condição de iluminação a que as amostras estavam submetidas.

Figura 18 – Estrutura fixa



Fonte: Elaborado pelo Autor, 2021.

3.4.3 *Processamento Digital de Imagens*

Uma vez que as imagens das amostras haviam sido coletadas, seguiu-se à fase de implementação dos algoritmos de Processamento Digital de Imagens (PDI). A implementação foi realizada utilizando-se a linguagem de programação Python, juntamente com a biblioteca OpenCV, para a manipulação das imagens.

Antes de ocorrer a extração de características das imagens, estas precisaram passar por um pré-processamento. Durante este processo, foram aplicados algoritmos de corte da imagem, preservando somente a amostra de café.

Por padrão, o espaço de cor RGB foi utilizado no processo de captura de imagens por grande parte das câmeras digitais e dispositivos móveis (GONZALEZ; WOODS; EDDINS, 2003). Portanto, foram extraídos das imagens os valores médios dos canais R (Vermelho), G (Verde) e B (Azul).

Conforme Kolkur *et al.* (2017) afirma, os valores RGB podem ser convertidos para qualquer outro espaço de cor. Portanto, também foram extraídos da imagem os valores médios do espaço de cor HSV, LAB e escala de cinza. Uma observação importante é que o espaço de cor HSV não varia de 0° a 360° para H, 0 a 100% para V e de 0 a 100% para S na biblioteca OpenCV, mas sim de 0 a 180, 0 a 255 e 0 a 255 para H, S e V, respectivamente.

Com a base de dados finalizada, em seguida, procedeu-se ao desenvolvimento e avaliação dos classificadores.

3.4.4 Classificadores

Após o desenvolvimento da base de dados, iniciou-se a seleção dos principais algoritmos de classificação. Foram realizados experimentos utilizando diversas configurações de bases e quatro classificadores, com o objetivo de identificar a melhor combinação entre o espaço de cor utilizado e o modelo de classificação.

Para a comparação dos algoritmos de classificação, todos os testes ocorreram através da validação cruzada *k-fold*. Este método particiona o conjunto de dados em um número de k subconjuntos mutuamente exclusivos de mesmo tamanho. Um subconjunto é utilizado para testes, e os $k-1$ restantes, para treino. Este processo é iterado k vezes, alternando os subconjuntos de modo que todos os dados sejam utilizados tanto para teste quanto para treinamento.

Para a seleção e otimização dos hiper-parâmetros dos algoritmos de classificação, empregou-se o algoritmo *Halving Grid Search CV* (HGS), disponível na biblioteca *Scikit-learn*. Inicialmente, é informada a grade de parâmetros dos algoritmos, que se trata de um dicionário com os nomes dos parâmetros como chave e uma lista de possíveis hiper-parâmetros como valores. Este algoritmo utiliza validação cruzada para encontrar a combinação ideal. O HGS adota uma abordagem chamada redução sucessiva à metade, onde, na primeira iteração, ele treina todos os candidatos em uma pequena proporção dos dados de treinamento. Nas iterações seguintes, os candidatos com melhor desempenho têm mais recursos (amostras de treinamento), e os com resultado inferior são eliminados. Ao fim deste processo, somente um conjunto de hiper-parâmetros é mantido.

Assim como Nasution e Andayani (2017) e Solmaz *et al.* (2018), a métrica de precisão foi levada em consideração durante a seleção dos classificadores. Com a precisão, buscamos identificar qual proporção de identificações positivas do nosso modelo estava realmente correta, podendo ser obtida através da Equação (3), onde são utilizados os parâmetros verdadeiros positivos (TP) e falsos positivos (FP).

$$precision = \frac{TP}{TP + FP} \quad (3)$$

Por fim, também foram analisados os tempos de execução de cada algoritmo durante a validação cruzada.

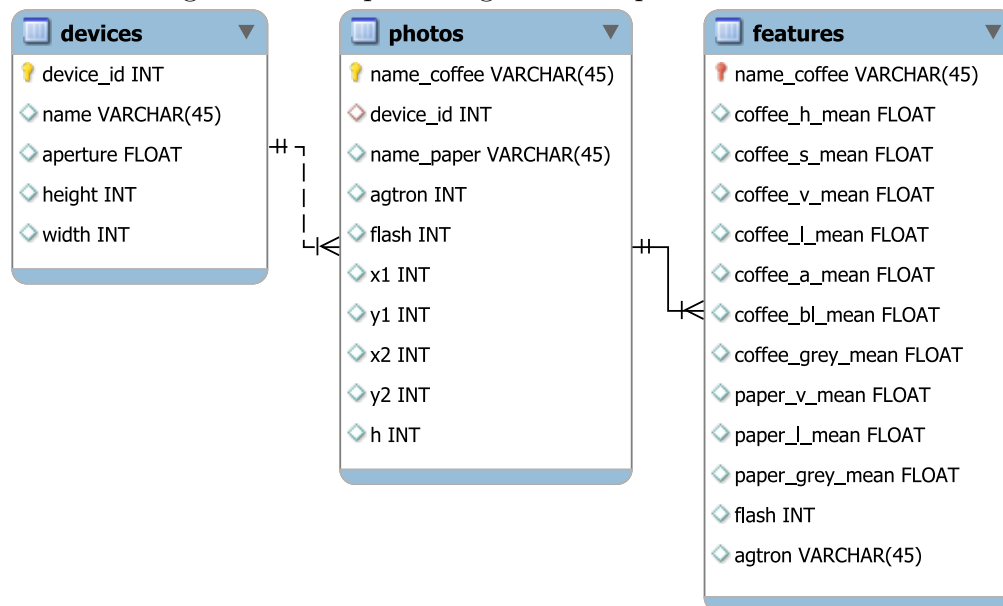
4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da base de dados, de algoritmos e da aplicação CoffeColor para dispositivos móveis. A Seção 4.1 explica como foi realizada a construção da base de dados. Já a Seção 4.2 apresenta os experimentos realizados. Por fim, a Seção 4.3 descreve o desenvolvimento da solução composta pela aplicação para dispositivos móveis e servidor (*backend*).

4.1 Construção da base de dados

Após a obtenção das fotos das amostras de café, conforme descrito no Capítulo 3, foram obtidas 745 imagens, totalizando 2,83 GB de arquivos. A fim de organizar as imagens e suas informações, foi criada uma base de dados contendo informações das imagens capturadas e dos dispositivos utilizados. A Figura 19 exibe o esquema lógico dessa base de dados.

Figura 19 – Esquema lógico dos arquivos de controle



Fonte: Elaborado pelo Autor, 2022.

A base de dados com as informações já processadas está disponível em um repositório¹ do GitHub em arquivos no formato *Comma Separated Values (CSV)*. Devido ao tamanho, a coleção com todas as fotos foi disponibilizada no Kaggle².

A tabela **devices** possui os atributos **device_id** (número único identificador do dispositivo), **name** (nome do dispositivo), **aperture** (abertura da lente da câmera), **height** e **width** (altura e largura das imagens capturadas).

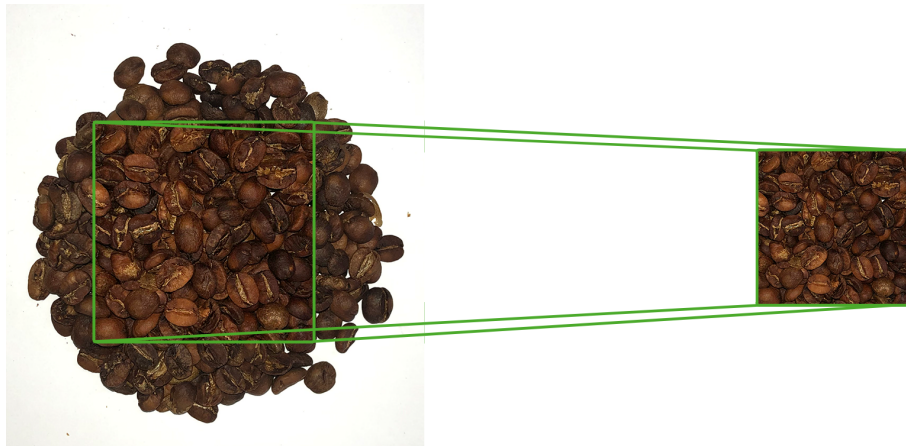
¹ <https://coffeecolor.github.io>

² <https://www.kaggle.com/datasets/joaovvrodrigues/coffee-toasted-agtron>

A tabela `photos` possui os atributos `device_id` (chave estrangeira com o identificador do dispositivo), `name_coffee` e `name_paper` (nome dos arquivos das imagens do café e folha), `agtron` (valor Agtron), `flash` (status do *flash*), `x1`, `y1`, `x2`, `y2` (pontos utilizados na etapa de corte) e `h` (altura final da imagem). O nome da imagem do café é único para cada amostra.

Após a criação da tabela `photos`, todas as imagens foram cortadas com o objetivo de selecionar somente a amostra, conforme apresentado na Figura 20. Para a realização desse processo, foi desenvolvido um *script* em Python que realiza o corte das imagens utilizando como referência as coordenadas armazenadas na tabela `photos`. As mesmas coordenadas são usadas para cortar a imagem da folha de papel; portanto, as imagens do café e da folha possuem o mesmo tamanho. Ao fim deste processo, o conjunto de arquivos com as imagens recortadas tinha o tamanho total de 219 MB.

Figura 20 – Processo de corte das imagens



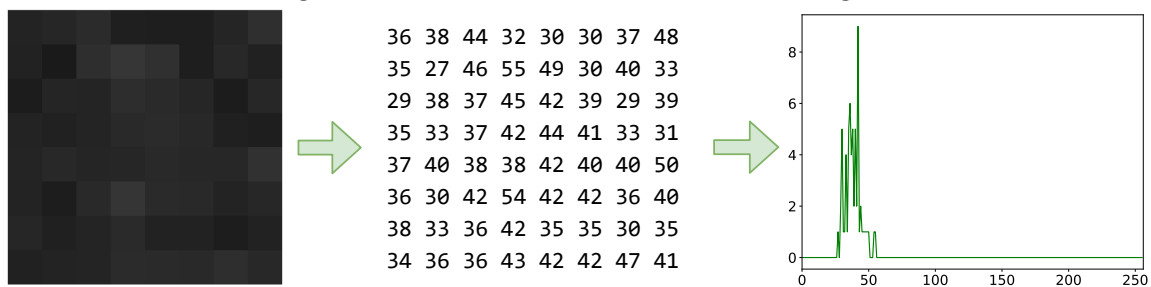
Fonte: Elaborado pelo Autor, 2021.

A partir dos recortes das imagens, foi realizada a etapa de extração de características. Em seu desenvolvimento, empregou-se a linguagem de programação Python juntamente com a biblioteca OpenCV. Nessa etapa foram extraídas informações sobre os espaços de cores (RGB, HSV, LAB e escala de cinza) das amostras de café e da folha de papel que representam a condição de iluminação. As características foram armazenadas na tabela `features`.

Uma imagem digital pode ser representada como uma matriz com valores de intensidade luminosa. Normalmente, é utilizado o espaço de cor RGB, onde as cores primárias (vermelho, verde e azul) são combinadas para formar outras. O número de itens que cada matriz possui varia de acordo com o tamanho da imagem, sendo que cada valor está associado a um *pixel*. Outros espaços de cor também podem ser utilizados para a representação de cores, conforme explicado no Capítulo 2. Portanto, uma imagem colorida, em RGB, é formada por três matrizes com valores de iluminação de cada componente de cor.

A Figura 21 apresenta uma imagem recortada de um grão de café em escala de cinza. A imagem possui 64 *pixels*, portanto, pode ser representada por uma matriz de oito linhas por oito colunas. Outra forma de representar os valores contidos na fotografia é o histograma, que demonstra a distribuição de frequências dos componentes. Além do histograma, também é possível calcular o desvio padrão e a média dos valores que compõem a imagem. A Figura 21 também revela a matriz que representa os *pixels* da imagem e seu histograma correspondente.

Figura 21 – Processo de conversão de imagem



Fonte: Elaborado pelo Autor, 2021.

Ao utilizar o OpenCV, é possível converter imagens de um espaço de cor para outro, bem como separar os componentes. Assim, para a extração de características das imagens, foi utilizada a função *split()*, que permite a divisão do espaço de cor e separa cada matriz. As matrizes são compostas por valores absolutos, sendo que nas matrizes de RGB, por exemplo, os valores variam de 0 a 255.

Com as matrizes extraídas, é possível calcular a média e o desvio padrão de cada componente de cor. Para isso, foi utilizada a biblioteca *numpy*, que fornece uma grande coleção de funções matemáticas para operações com matrizes. As funções utilizadas foram *mean()*, para cálculo da média, e *std()* para o cálculo do desvio padrão, sendo que o mesmo processo se repetiu para cada espaço de cor utilizado. As amostras submetidas a uma mesma condição de iluminação possuem os mesmos valores para os atributos referentes à folha de papel.

Após a extração de características, foi iniciada a construção do conjunto de dados utilizados para treinar os algoritmos de classificação. As informações foram armazenadas na tabela **features**, que possui 29 atributos. Cada imagem da tabela **photos** tem um único correspondente na tabela **features**; portanto, ambas possuem a mesma quantidade de linhas.

A tabela **features** possui os atributos **name_coffee** (nome da imagem do café), média e desvio padrão de cada canal dos espaços de cores RGB, HSV e LAB, bem como a média e desvio padrão da escala de cinza para o café, além dos atributos de indicação de *flash* e valor Agtron. Para a folha de papel, a referida tabela possui os atributos de média e desvio padrão dos canais V (do HSV) e L (do LAB) e da escala de cinza. O Quadro 1 apresenta todos os atributos da tabela **features**.

Quadro 1 – Atributos extraídos das imagens

Atributo	Descrição	Tipo
name_coffee	Nome do arquivo do café	Alfanumérico
coffee_r_mean	R médio do café	Numérico
coffee_r_std	Desvio padrão de R do café	Numérico
coffee_g_mean	G médio do café	Numérico
coffee_g_std	Desvio padrão de G do café	Numérico
coffee_b_mean	B médio do café	Numérico
coffee_b_std	Desvio padrão de B do café	Numérico
coffee_h_mean	H médio do café	Numérico
coffee_h_std	Desvio padrão de H do café	Numérico
coffee_s_mean	S médio do café	Numérico
coffee_s_std	Desvio padrão de S do café	Numérico
coffee_v_mean	V médio do café	Numérico
coffee_v_std	Desvio padrão de V do café	Numérico
coffee_l_mean	L médio do café	Numérico
coffee_l_std	Desvio padrão de L do café	Numérico
coffee_a_mean	A médio do café	Numérico
coffee_a_std	Desvio padrão de A do café	Numérico
coffee_bl_mean	B de LAB médio do café	Numérico
coffee_bl_std	Desvio padrão de B de LAB do café	Numérico
coffee_grey_mean	Média da escala de cinza do café	Numérico
coffee_grey_std	Desvio padrão da escala de cinza do café	Numérico
paper_v_mean	V médio do papel	Numérico
paper_v_std	Desvio padrão de V do papel	Numérico
paper_l_mean	L médio do papel	Numérico
paper_l_std	Desvio padrão de L do papel	Numérico
paper_grey_mean	Média da escala de cinza do papel	Numérico
paper_grey_std	Desvio padrão da escala de cinza do papel	Numérico
flash	Status do flash	Numérico
agtron	Valor Agtron do café	Alfanumérico

Fonte: Elaborado pelo Autor, 2021.

A partir do conjunto de dados da tabela **features**, efetuaram-se experimentos com combinações dos atributos extraídos. O conjunto de dados usado no treinamento dos classificadores tinha 123 KB, contendo 29 atributos e 648 linhas.

4.2 Experimentos para seleção de atributos

Após a extração das características, foram realizados diversos experimentos para seleção de atributos e de algoritmos de classificação. Com base em trabalhos correlatos, foram selecionados os classificadores *Random Forest* e Redes Neurais Artificiais *MultiLayer Perceptron*. Além disso, para comparação de desempenho, foram considerados, também, os algoritmos *Decision Tree* e *Naive Bayes*.

Todos os algoritmos de classificação estão disponíveis na biblioteca *Scikit-learn*. Com o intuito tornar os resultados reprodutíveis, o atributo **random state** dos

classificadores foi fixado em 42.

Em todos os testes, a técnica utilizada para avaliar a classificação do modelo foi a validação cruzada com dez partições. Quanto ao tempo de execução dos algoritmos, efetuaram-se três execuções, considerando-se o tempo médio.

A Seção 4.2.1 apresenta os testes realizados variando o indicativo de claridade do papel. Em seguida, a Seção 4.2.2 relata os experimentos efetuados com a variação dos atributos e espaços de cor utilizados para caracterizar o café. Por fim, a Seção 4.2.3 aborda as técnicas de pré-processamento utilizadas para suavizar a imagem e reduzir as transições bruscas nos níveis de intensidade das cores.

4.2.1 *Variação no indicativo de claridade do papel*

A foto da folha de papel representa a condição de iluminação em que a amostra foi submetida. Assim, os primeiros testes realizados foram para verificar se a adição do indicativo de claridade resultaria em uma melhora na classificação e quais atributos e combinações poderiam ser utilizados. As informações extraídas da folha de papel foram médias e desvio padrão, para os atributos de escala de cinza, canal V do HSV e canal L do LAB. A escala de cinza informa quão clara ou escura está a imagem, o canal V representa o brilho da cor, e o canal L diz respeito à luminosidade. Os algoritmos selecionados foram testados na base de dados com variações na combinação desses atributos, mantendo todos aqueles relacionados à amostra do café.

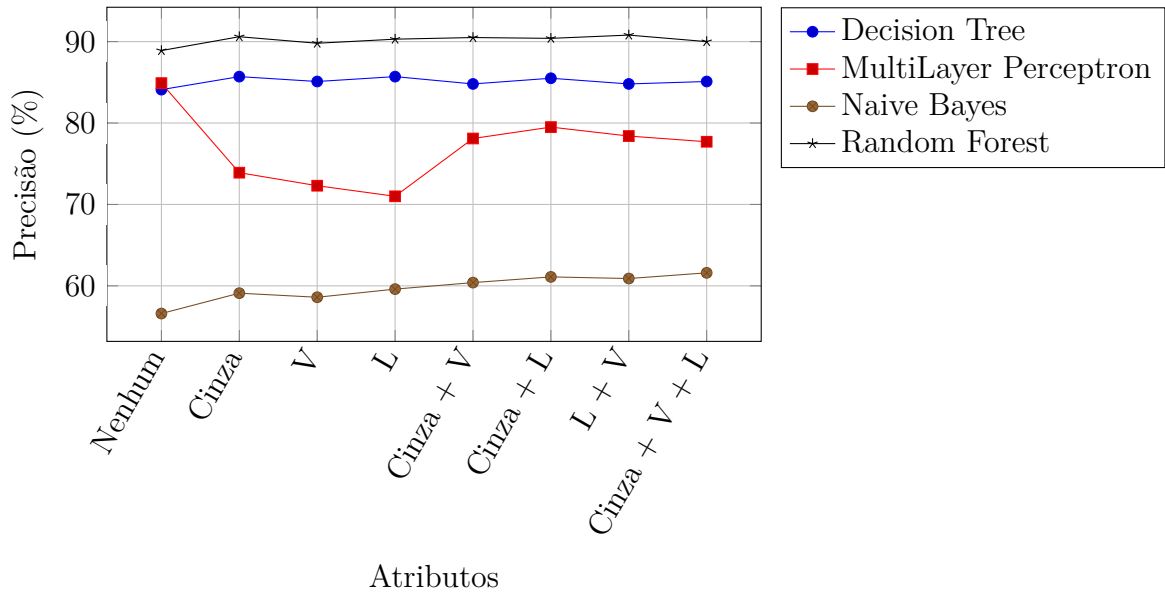
A Figura 22 apresenta a precisão de cada algoritmo considerando as possíveis combinações dos atributos indicativos da claridade do papel, inclusive quando nenhum é utilizado. Os algoritmos *Random Forest*, *Decision Tree* e *Naive Bayes* se comportaram de forma estável mesmo com a variação dos atributos, sendo que todos têm um decréscimo na precisão quando nenhum indicativo de claridade é usado. O *MultiLayer Perceptron*, por sua vez, evidenciou muita oscilação em seus resultados. É interessante notar que, quando não foi utilizado nenhum atributo de claridade, o *MultiLayer Perceptron* atingiu sua maior precisão (84.9%).

Nesse primeiro experimento, o classificador *Random Forest* obteve o melhor resultado em todos os cenários. Em especial, ao utilizar somente os atributos V e L, o algoritmo obteve sua maior precisão (90.8%). Nessa mesma situação, os classificadores *Naive Bayes* e *Decision Tree* atingiram 60.9% e 84.8%, respectivamente.

A Figura 23 apresenta o tempo de execução médio da validação cruzada para cada classificador. O gráfico do tempo de execução está em escala logarítmica porque alguns classificadores obtiveram um tempo muito próximo, fazendo com que, na plotagem não logarítmica, as linhas se sobrepusessem.

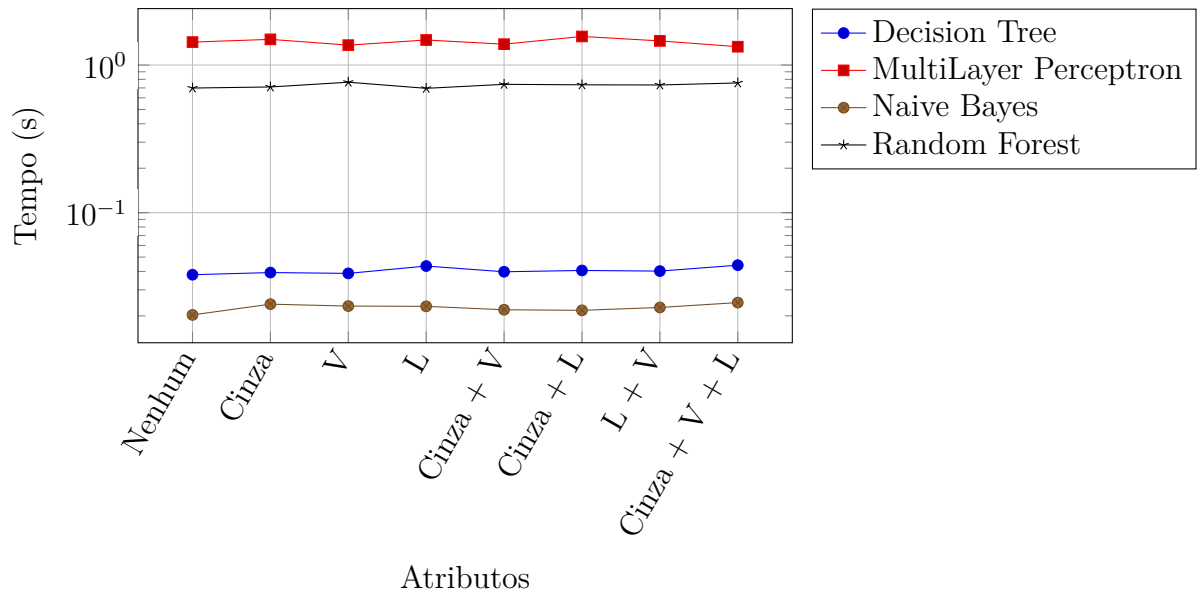
Como esperado, devido à sua simplicidade, o algoritmo *Naive Bayes* obteve o menor tempo em todas as situações, seguido pela *Decision Tree*. O *Random Forest*, por sua vez, devido ao número elevado de árvores, obteve um tempo superior. Por fim, o

Figura 22 – Precisão dos algoritmos em relação à variação de indicativo de claridade



Fonte: Elaborado pelo Autor, 2022.

Figura 23 – Tempo da execução dos algoritmos em relação à variação de indicativo de claridade



Fonte: Elaborado pelo Autor, 2022.

MultiLayer Perceptron, por causa da sua complexidade e número de iterações, atingiu a maior média de tempo de execução. Em geral, o tempo decorrido durante o treino dos classificadores foi estável em todas as situações.

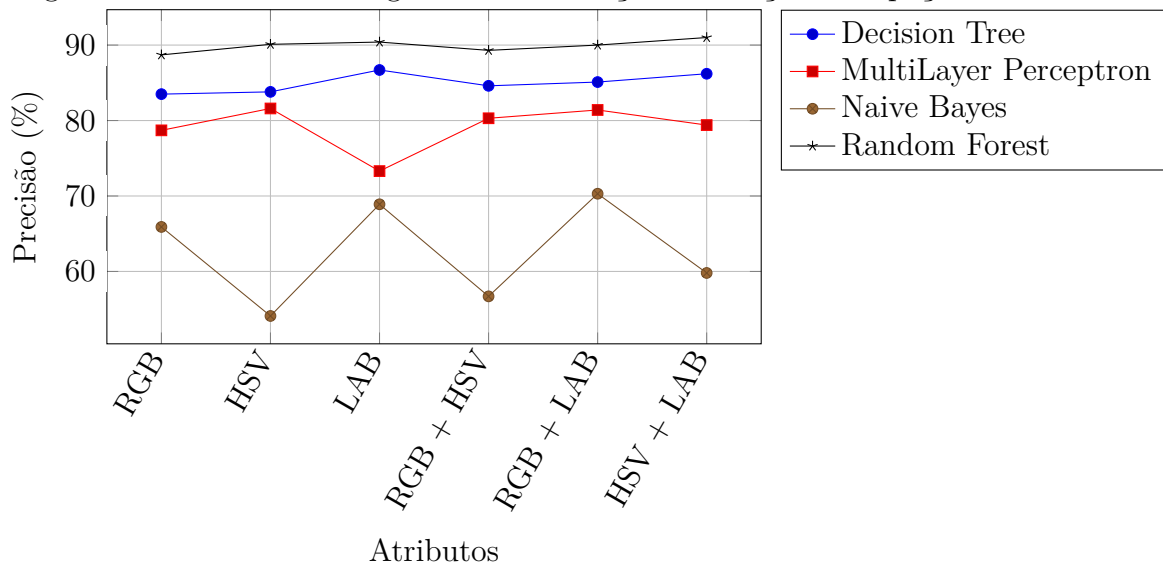
Para os demais experimentos, foram mantidos como atributos de claridade o canal V, do HSV, e canal L, do LAB, somente. Isso foi feito porque esses atributos levaram a uma melhor precisão do melhor algoritmo no primeiro experimento (*Random Forest*).

4.2.2 Variação no espaço de cor do café

Após a seleção dos indicativos de claridade, teve início a etapa de seleção de atributos relacionados às amostras de café. Para isso, foram consideradas combinações das informações dos espaços de cor RGB, HSV e LAB. As informações da escala de cinza foram mantidas em todos os experimentos, assim como o indicativo de claridade selecionado na Seção 4.2.1.

O principal objetivo do experimento descrito na presente seção é encontrar uma combinação de atributos referentes à cor do café que melhore a precisão do classificador *Random Forest*, uma vez que esse foi o melhor classificador no experimento de indicativo de claridade. Para complementar, os demais classificadores também foram executados e comparados.

Figura 24 – Precisão dos algoritmos em relação à variação no espaço de cor do café



Fonte: Elaborado pelo Autor, 2022.

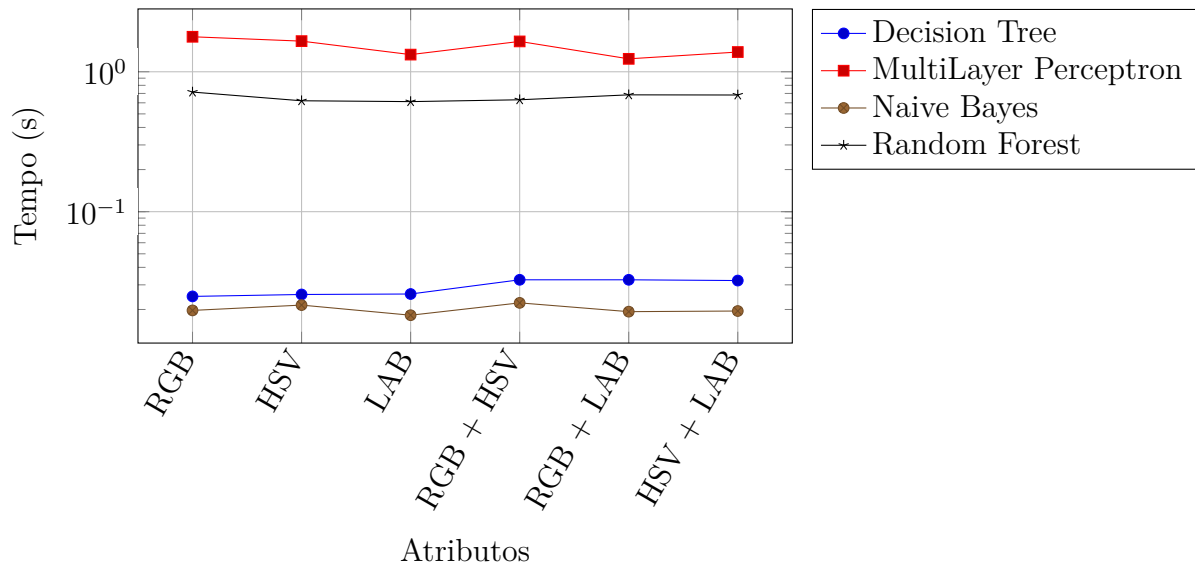
A Figura 24 apresenta a precisão de cada algoritmo considerando as possíveis combinações de espaços de cor. Os algoritmos *Random Forest*, *Decision Tree* comportam-se de forma estável, tendo poucas variações em seus resultados. O *MultiLayer Perceptron*, por sua vez, evidenciou certa oscilação em seus resultados, com uma queda ao utilizar somente o espaço de cor LAB. Por fim, o algoritmo *Naive Bayes* teve uma considerável oscilação, com baixas precisões ao utilizar o HSV.

Novamente, o classificador que obteve o melhor resultado em todos os cenários foi o *Random Forest*. A combinação de HSV e LAB resultou na maior precisão (91%), sendo que, nas mesmas condições, o classificador *Decision Tree* obteve 86.2% de precisão, *Naive Bayes* atingiu 59.8% e *MultiLayer Perceptron* obteve 79.4%.

A Figura 25 apresenta o tempo de execução médio dos classificadores. Podemos observar que o tempo de execução não sofreu grandes alterações, com o algoritmo *Naive Bayes* obtendo menor tempo, seguido pela *Decision Tree*. O algoritmo *Random Forest*,

por sua vez, alcançou um tempo de 0.658 segundos. Por fim, o *MultiLayer Perceptron* obteve a maior média de tempo de execução, chegando a 1.505 segundos.

Figura 25 – Tempo da execução dos algoritmos em relação à variação no espaço de cor do café



Fonte: Elaborado pelo Autor, 2022.

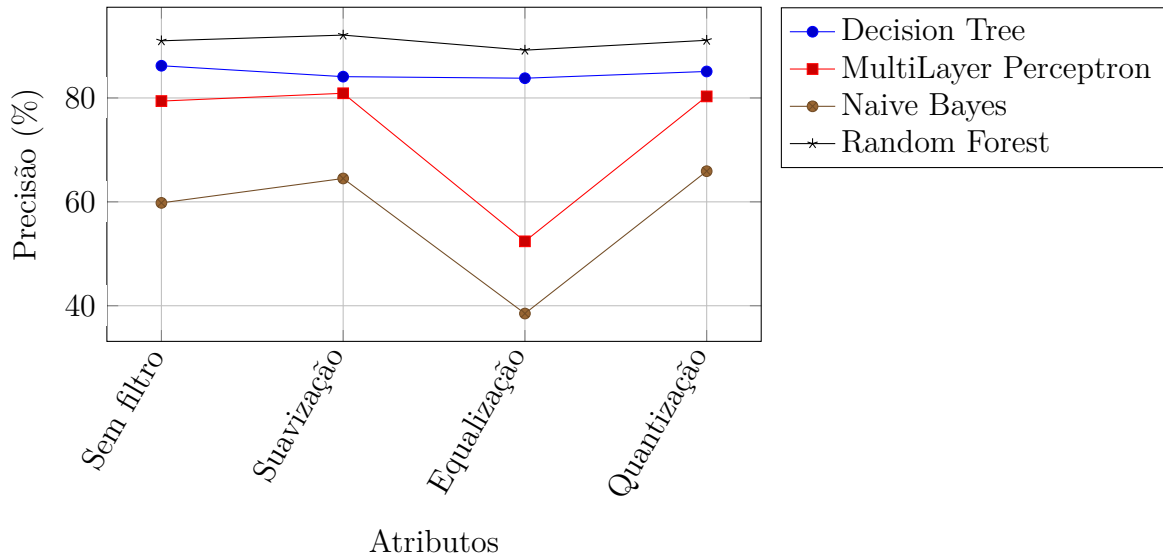
4.2.3 Experimentos com técnicas de pré-processamento

Após as melhorias já alcançadas nos experimentos anteriores, foi realizado um experimento para verificar se a utilização de algoritmos da área de Processamento Digital de Imagens poderia impactar positivamente na precisão dos algoritmos. Nessa etapa, empregou-se a linguagem de programação Python juntamente com as bibliotecas OpenCV e *Scikit-learn*.

A base de dados utilizada no experimento contém os mesmos dados do melhor resultado alcançado na Seção 4.2.2. Entretanto, técnicas de pré-processamento e filtros foram aplicados nas imagens antes da extração de características e execução do experimento com os algoritmos de classificação. Assim, foram utilizados os espaços de cor HSV e LAB e a escala de cinza referente à cor do café e indicativo de claridade da folha de papel. As técnicas aplicadas foram suavização pela mediana, equalização de histograma e quantização de cores. Os resultados podem ser observados na Figura 26.

A técnica de suavização pela mediana foi aplicada com o intuito de reduzir as transições bruscas nos níveis de intensidade das cores, que podem ser causadas por uma não uniformidade da torra ou pela película prateada do café. Com a aplicação desse filtro, o valor de cor de cada *pixel* central é substituído pela mediana dos valores na vizinhança desse *pixel*, tornando as cores mais homogêneas. O parâmetro *ksize*, que define o tamanho linear da abertura do filtro, deve ser ímpar e maior que um. Foram feitos testes com

Figura 26 – Precisão dos algoritmos em relação à variação na técnica de pré-processamento



Fonte: Elaborado pelo Autor, 2022.

variações nos valores deste parâmetro, e, dentre os números testados, o valor 9 retornou um maior ganho de precisão.

Após a aplicação da suavização pela mediana, com exceção da *Decision Tree*, todos os algoritmos tiveram um pequeno ganho na precisão. O *Naive Bayes* obteve um acréscimo na precisão de 4.7%, atingindo 64.5%. O *MultiLayer Perceptron* alcançou 80.9%, e a *Decision Tree*, 84.1%. O *Random Forest*, por sua vez, conquistou a precisão de 92.1%, sendo este o melhor resultado atingido.

Outra técnica aplicada foi a equalização de histograma. As imagens foram convertidas para HSV e, em sequência, os canais de matiz (H), saturação (S) e intensidade de brilho (V) foram separados. Com o intuito de balancear o nível de brilho das imagens, a técnica de equalização de histograma foi aplicada no canal V.

Com a aplicação deste método, todos os algoritmos performaram pior, com destaque para os classificadores *Naive Bayes* e *MultiLayer Perceptron*, que tiveram um grande decréscimo em seus resultados. Atingindo 38.5%, o *Naive Bayes* decaiu 21.3% em sua precisão, e o classificador *MultiLayer Perceptron*, por sua vez, decaiu 27% em seus resultados, obtendo 52.4% de precisão somente. A *Decision Tree* e o *Random Forest* tiveram uma queda inferior, atingindo 83.8% e 89.2%, respectivamente.

Por fim, o último processo testado foi o de quantização de cores, que é aplicado sobre a imagem com o objetivo de reduzir o número de cores distintas nela. Para encontrar as cores que representam a imagem, foi utilizado o algoritmo de agrupamentos, (ou *clustering*) K-Means. Basicamente, esse algoritmo busca encontrar similaridades entre as informações e agrupá-las. Para isso, ele particiona os dados em k grupos (ou *clusters*), e a similaridade é obtida por um cálculo de distância.

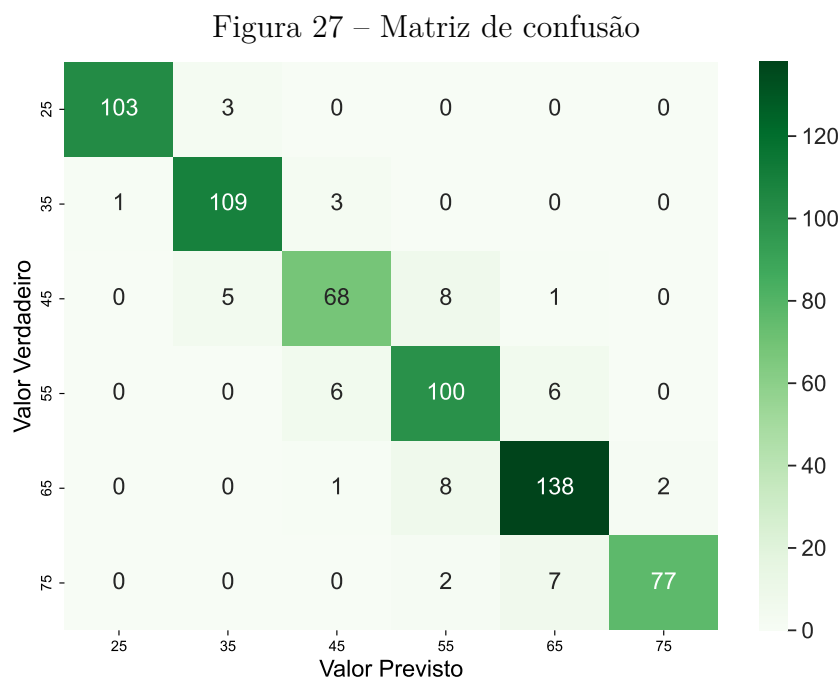
Como o algoritmo K-Means usa a distância euclidiana, a imagem foi convertida para o espaço de cor LAB. Por se tratar de um espaço de cor uniforme, a diferença entre as

cores pode ser calculada a partir desta equação, entre pontos deste espaço. A configuração do K-Means foi efetuada com os parâmetros `random state` e `n clusters` alterados para 42 e 3, respectivamente. A seleção do `n clusters` igual a três foi feita com base em testes, alterando os valores e verificando a precisão final. Portanto, nos testes realizados, o uso deste valor gerou um resultado positivo. Assim, todas as cores distintas da imagem foram divididas em três grupos, ou seja, somente três cores irão compor a imagem ao final do processo.

Com o número de *clusters* definido para três, o classificador *Random Forest* obteve 0.1% de acréscimo em sua precisão em relação à base de dados sem pré-processamento, chegando a 91.1%. A *Decision Tree* atingiu 85.1% de precisão, e o *MultiLayer Perceptron* conseguiu 80.3%. Por fim, o *Naive Bayes* obteve um acréscimo de 6.1%, atingindo 65.9% de precisão total.

Foram testadas combinações entre as técnicas de pré-processamento e alterações nos parâmetros. Entretanto, nenhum resultado foi superior à aplicação da suavização pela mediana.

Na Figura 27, pode ser observada a matriz de confusão obtida no melhor resultado. As informações obtidas por meio da matriz de confusão são pertinentes, pois demonstram o comportamento do algoritmo na classificação, além de mostrarem quais graus de terra possivelmente necessitam de mais amostras para uma maior distinção. O ideal é que todos os graus de terra tivessem a mesma quantidade de imagens. Entretanto, os graus que tiveram maior dificuldade de distinção acabaram contendo mais falsos positivos. Analisando a Figura, foram poucos os erros que ocorreram, sendo em grande parte com graus de terra bem próximos.



Fonte: Elaborado pelo Autor, 2022.

4.2.4 Otimização dos hiper-parâmetros

O classificador *Random Forest*, aquele com melhor resultado nos experimentos, possui parâmetros específicos que podem ser alterados, mudando, assim, seu comportamento durante a classificação. Conforme mencionado no Capítulo 3, para a seleção e otimização dos hiper-parâmetros, foi utilizado o algoritmo *Halving Grid Search* (HGS), disponível na biblioteca *Scikit-learn*.

Inicialmente, é informada a grade de parâmetros, na forma de um dicionário, com os nomes dos parâmetros como chave e uma lista com valores a serem testados. O tempo de execução deste algoritmo pode ser muito elevado, levando em consideração a quantidade de valores testados. Um dicionário de cinco parâmetros, com cada parâmetro tendo cinco valores, possui 3125 possíveis combinações a serem testadas.

O algoritmo HGS tem uma configuração própria, onde foram alteradas a métrica a ser analisada, a quantidade de núcleos do processador e a quantidade de iterações na validação cruzada. A métrica a ser analisada é a precisão, e a quantidade de iterações foi definida como dez. Por fim, determinou-se que o algoritmo pudesse utilizar todos os núcleos disponíveis para melhorar o tempo de execução, rodando os testes em paralelo.

Foram testadas 1.200.000 combinações de parâmetros para o *Random Forest*, sendo que os parâmetros com valores testados foram `num_estimators`, `max_features`, `max_depth`, `min_samples_split`, `min_samples_leaf` e `bootstrap`. Mais detalhes sobre esses e outros parâmetros podem ser encontrados na documentação do *Random Forest* na *Scikit-learn*³.

Com o número elevado de árvores e combinações, o algoritmo executou por mais de 8 horas e não obteve um resultado que superasse a precisão com os hiper-parâmetros no valor padrão. Um conjunto menor de 1.200 combinações foi testado, e, após 22 minutos, o classificador atingiu 91.7%. Também foi testada a alteração manual dos valores, mas não houve melhora.

Desse modo, como não houve melhorias com as mudanças nos hiper-parâmetros, foram mantidos os valores padrões. Com os valores padrões, a precisão máxima alcançada foi de 92.1%, para os espaços de cor HSV, LAB e a escala de cinza referente à cor do café e indicativo de claridade da folha de papel, juntamente com a aplicação da suavização pela mediana.

4.3 Desenvolvimento da solução

A solução proposta necessitava da implementação de uma interface de programação de aplicações (API) desenvolvida em Python, para que as bibliotecas utilizadas para aplicação dos filtros, extração de características e classificação da amostra pudessem

³ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier>

funcionar corretamente. Para isso, foi desenvolvido um servidor Web (*backend*) com uma API Web através da qual a aplicação para dispositivos móveis pudesse se comunicar.

A aplicação para dispositivos móveis permite a seleção das imagens para efetuar o envio ao servidor; este recebe a imagem, faz o processamento necessário e devolve a resposta para a aplicação, a qual recebe a resposta do servidor e apresenta o valor Agtron do café ao usuário. Além disso, a aplicação apresenta instruções de como utilizar o aplicativo e outras informações adicionais sobre café.

A Seção 4.3.1 apresenta a organização e as principais funções utilizadas do servidor. Em seguida, a Seção 4.3.2 aborda o desenvolvimento da aplicação móvel, bem como seu comportamento e comunicação com o servidor. Por fim, a Seção 4.3.3 expõe os testes realizados para mensurar o tempo decorrido durante a classificação de amostras utilizando o aplicativo.

4.3.1 Servidor

Ao final dos experimentos realizados, foi construído um modelo com toda a base de dados considerando os parâmetros, atributos e pré-processamento, que chegaram ao melhor resultado de 92.1% de precisão. O modelo foi exportado utilizando-se a biblioteca *joblib*, com a função *dump()*, que torna possível salvar qualquer objeto como um arquivo.

A API Web baseada em REST implementada proporciona a conexão entre a aplicação móvel e as funções de classificação, sendo que a comunicação é feita por meio de requisições por URIs. O servidor retorna ao requisitante (aplicação móvel) a classificação Agtron do café em formato JSON.

Para o desenvolvimento do servidor, foi utilizado o *micro-framework* Flask, devido à sua simplicidade e estrutura inicial acessível, permitindo que o projeto possua apenas os recursos necessários para sua execução. Como o servidor irá executar somente uma função (classificação da amostra), não há necessidade de se utilizar um *framework* mais complexo.

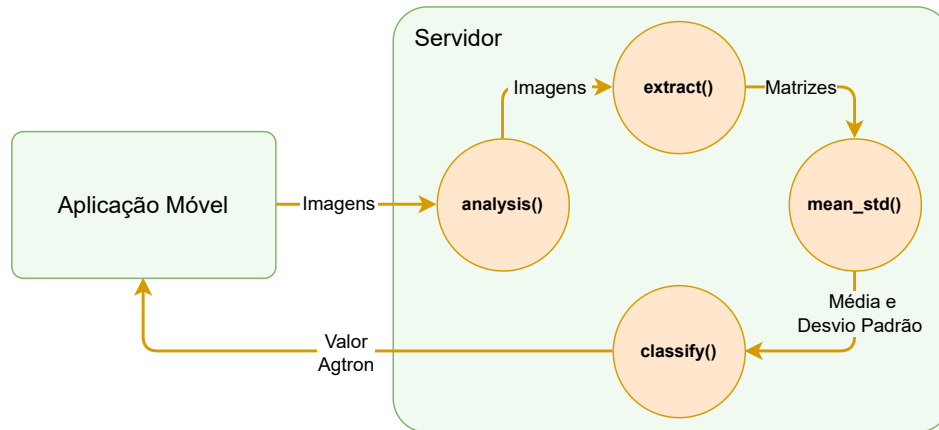
Ao todo, considerando-se a importação de bibliotecas, do modelo, extração de características da imagem e classificação, o código-fonte do servidor chegou a 106 linhas de código.

A Figura 28 demonstra o processo de classificação da amostra de café por meio de quatro funções principais: *analysis()*, *extract()*, *mean_std()* e *classify()*.

Inicialmente, a aplicação móvel envia as imagens e o status do *flash* para o servidor. As imagens são encaminhadas através da função *send()*, permitindo o envio de dados ao servidor. Essa função, primeiramente, verifica se os arquivos existem; caso ocorra algum erro, uma mensagem é retornada para a aplicação.

Após receber os arquivos, a função *analysis()* é iniciada. Os arquivos são carregados com a biblioteca OpenCV por meio da função *imread()*, para que possam ser analisados. Ainda com o auxílio da OpenCV, é aplicado em ambas as imagens o filtro de

Figura 28 – Diagrama de fluxo de dados da solução



Fonte: Elaborado pelo Autor, 2022.

suavização pela mediana. Em seguida, a função nomeada de *extract()* fica responsável por extrair os componentes de LAB, HSV e escala de cinza das imagens. Para isso, a função *split()* da biblioteca OpenCV é utilizada, onde as matrizes que compõem as imagens são retornadas.

Com os componentes extraídos, são calculados a média e desvio padrão na função nomeada de *mean_std()*, sendo que, para a realização dos cálculos, a biblioteca *numpy* foi utilizada. Posteriormente, a função nomeada de *classify()* é iniciada. Através da função *load()* biblioteca *joblib*, o modelo é recuperado, transformando-o novamente em um objeto. Em seguida, a função *predict()* do modelo analisa os novos dados de entrada e retorna o valor Agtron identificado.

Por fim, a informação da classificação, com o valor Agtron, é retornada para a função *send()*, onde é serializada em formato JSON por meio da função *jsonify()* e retornada para a aplicação móvel.

A API Web desenvolvida foi hospedada no serviço PythonAnywhere⁴. Além de um serviço de hospedagem web, também fornece um ambiente de desenvolvimento integrado para Python. Vale ressaltar que existem planos que cobram mensalmente, mas é possível hospedar gratuitamente uma única aplicação Web com algumas limitações. As bibliotecas utilizadas são instaladas por um terminal *bash* interativo disponível no navegador, onde os códigos também são enviados. Por fim, é necessário somente criar o servidor Web com o nome do projeto e indicar os arquivos e pastas que poderão ser acessados.

4.3.2 Aplicação móvel

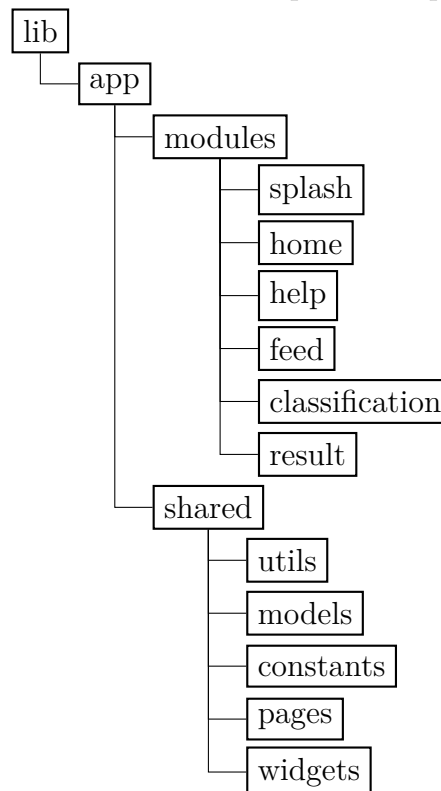
A aplicação para dispositivos móveis, nomeada de CoffeColor, foi desenvolvida com a utilização do *framework* Flutter. Além de artigos informativos, ela fornece ao

⁴ <https://www.pythonanywhere.com/>

usuário um meio para selecionar as imagens capturadas, realizar o corte da imagem do café e comunicar com o servidor. Como a classificação da amostra não ocorre localmente, e sim no servidor, o usuário deve possuir conexão com internet para executar essa tarefa. As outras funcionalidades podem ser acessadas sem conexão.

Um projeto em Flutter possui por padrão uma estrutura de pastas, sendo algumas para configurações relacionadas a sistema operacional, uma pasta para arquivos (*assets*) e a pasta *lib*. Nessa última, fica guardado o código-fonte da aplicação. Para a organização de arquivos, injeção de dependências e controle de estados da aplicação, foi utilizada a biblioteca Flutter Modular, que modulariza toda a aplicação, separando suas dependências, funções e características.

Figura 29 – Estrutura de pastas do projeto

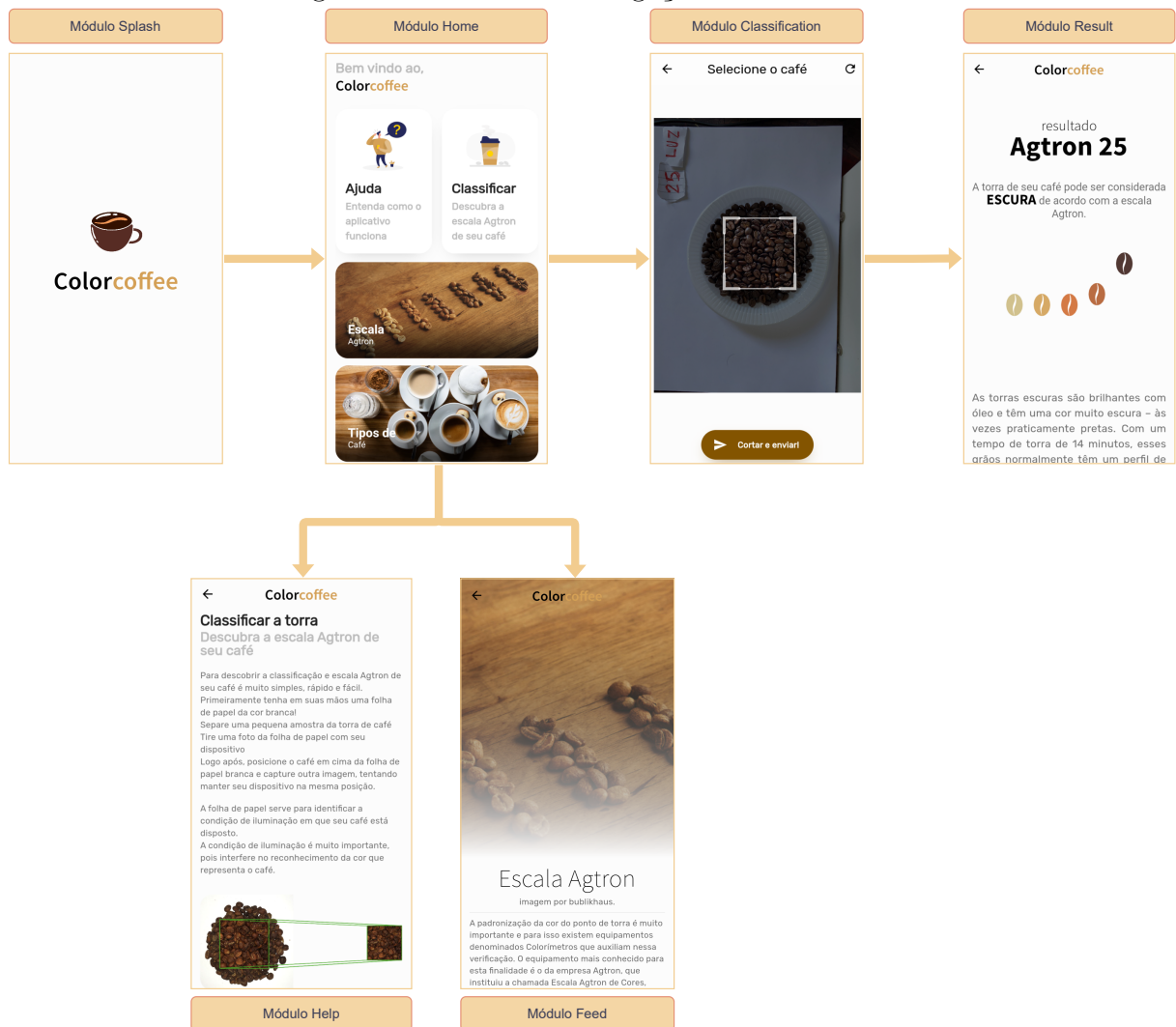


Fonte: Elaborado pelo Autor, 2022.

A Figura 29 apresenta a estrutura de pastas da aplicação. A pasta *modules* contém todos os módulos que compõem o projeto (*splash*, *home*, *help*, *feed*, *classification* e *result*). A pasta *shared* possui arquivos que são comuns entre os módulos, como modelos de classes disponíveis na pasta *models*, componentes de tela (*pages* e *widgets*), variáveis constantes (*constants*), entre outros códigos (*utils*).

Foram criados seis módulos principais. A Figura 30 apresenta cada página que o módulo integra, bem como o fluxo de navegação entre elas. A aplicação começa com o módulo *splash*, que possui uma animação de carregamento indicando que o aplicativo está iniciando. Logo após, o módulo *home* é carregado e mostra a tela principal do aplicativo.

Figura 30 – Fluxo de navegação entre módulos



Fonte: Elaborado pelo Autor, 2022.

Na tela principal, o usuário tem acesso aos módulos *help*, *classification* e *feed*. O módulo *feed* tem como intuito apresentar informações úteis e curiosidades sobre o café, suas classificações, tipos e modos de preparo. O módulo *help* apresenta um texto informativo ensinando como utilizar a aplicação, sendo basicamente um tutorial com mensagens e imagens que ajudam o usuário a realizar corretamente o processo de classificação da amostra de café. O módulo *feed*, por sua vez, apresenta textos e imagens sobre curiosidades e peculiaridades do café e escala Agtron.

No módulo *classification*, é possível enviar as imagens para classificação. Ao iniciar este módulo, aparece uma mensagem indicando que o usuário deve cortar a imagem selecionando a região onde se encontra o café. A aplicação indica que deve ser selecionada primeiramente a foto do café e, em seguida, a foto com a folha de papel branco. As imagens são carregadas diretamente dos arquivos armazenados no dispositivo. Logo após, o processo de corte é iniciado, e o aplicativo identifica os pontos de corte selecionados, aplicando a mesma área na imagem com a folha de papel. Portanto, é importante que

as imagens sejam capturadas em uma distância similar. Caso o usuário deseje reiniciar o processo, existe um botão no canto superior direito da tela.

Para determinar o status do *flash*, a aplicação analisa os metadados das imagens originais. Os metadados de uma imagem podem possuir detalhes técnicos, como data, tamanho e a situação do flash. Existem 27 estados possíveis⁵ para o *flash*, os quais podem ser analisados para determinar se ele estava ativo durante a captura da imagem.

Posteriormente, ao finalizar o processo de corte, as imagens são convertidas em uma lista de comprimento fixo de inteiros sem sinal de 8 bits, com valores no intervalo de zero a 255, adicionados em uma estrutura do tipo *map*. Por fim, os arquivos e o status do *flash* são enviados para o servidor através do método de requisição HTTP POST para a função *send()*. Para efetuar o envio dos arquivos com o método POST, foi utilizada a biblioteca Dio, que é um cliente HTTP para Flutter. Tal biblioteca conta com diversos incrementos, permitindo que o envio de arquivos possa ser realizado mais facilmente.

Após receber a resposta do servidor, o módulo *result* é iniciado, para apresentar o valor Agtron identificado, bem como outras conclusões sobre a amostra de café. Na parte superior da tela, é exibido o valor detectado e, abaixo, de forma mais simples, como a torra em questão pode ser considerada. Por fim, um texto apresenta as principais características sobre o grau de torra verificado. Caso ocorra algum erro durante o processo de envio, uma mensagem de erro é apresentada para o usuário.

O tema de cores utilizado no aplicativo varia de acordo com o tema do dispositivo utilizado. Portanto, caso o aparelho esteja com o modo escuro ativo, o tema da aplicação será alterado automaticamente.

4.3.3 Experimento com tempo de classificação

O tempo de execução partindo do corte da imagem até o retorno da resposta pode variar de acordo com diversos fatores, como a localização do servidor, tamanho das imagens, velocidade de conexão da internet, processador do dispositivo, memória, entre outros. Assim, foram realizados alguns testes para analisar esse tempo.

Como o processo de corte das imagens ocorre localmente no dispositivo móvel, o poder computacional disponível também pode interferir no tempo de execução. Um dos dispositivos utilizados para teste foi um Motorola Moto G7 Play com processador *Snapdragon* 632, 2 GB de memória RAM e Android 10. O outro foi um Samsung *Galaxy* A72, com processador *Snapdragon* 720G, 6 GB de RAM e Android 12. Foram realizados quatro testes em modo de depuração com os dispositivos. Como a comunicação com o servidor depende da rede, consideraram-se apenas o corte e o processamento da imagem efetuados nos dispositivos.

Apesar da diferença de *hardware*, ambos os dispositivos obtiveram performance equiparável. O aparelho da Samsung obteve média de 11,60 segundos; já o Moto G7 levou

⁵ <https://exiftool.org/TagNames/EXIF.html#Flash>

13,72 segundos para cortar ambas as imagens, extrair metadados, enviar para o servidor e receber a resposta. As funções que mais demoram para executar são aquelas responsáveis pelo corte das imagens.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo propor uma metodologia alternativa e automatizada para a análise de cor do café torrado em grãos, com base na escala Agtron, por meio de uma aplicação computacional para dispositivos móveis. Com os experimentos realizados, foi possível identificar, no cenário estudado, a importância e diferença dos espaços de cor, comportamento dos algoritmos utilizados e como as técnicas de pré-processamento podem afetar na classificação de imagens.

Com o objetivo de criar uma base de dados de fotos de diferentes graus de torras de café, foram capturadas 745 imagens com diferentes graus de torra de café em grãos. A captura das imagens foi realizada com sete dispositivos móveis de diferentes modelos. Seu desenvolvimento foi fundamental para o treinamento e comparativo dos algoritmos. A base desenvolvida está disponível de forma *online* e gratuita na plataforma Kaggle, podendo ser utilizada para replicar os experimentos, realizar experimentos com outros algoritmos, bem como desenvolver outros trabalhos, por exemplo, a detecção automática da região do café.

Com o objetivo de analisar e avaliar as melhores técnicas de Inteligência Artificial e Visão Computacional para classificar as fotos, foram realizados diversos experimentos com espaços de cor, algoritmos, parâmetros e técnicas de pré-processamento.

Foram extraídos das imagens atributos para os espaços de cor RGB, HSV, LAB e escala de cinza. Em seguida, efetuaram-se os experimentos com o objetivo de encontrar uma combinação de atributos que resultasse em um bom desempenho na classificação. Técnicas de pré-processamento também foram testadas, visando balancear as cores e brilhos, tornando as amostras mais uniformes.

Com base em trabalhos correlatos e com o intuito comparativo, foram selecionados os algoritmos *Random Forest*, *MultiLayer Perceptron*, *Decision Tree* e *Naive Bayes*. Com exceção do *Naive Bayes*, todos os algoritmos performaram bem na classificação, com médias acima de 70% de precisão.

Por fim, o melhor resultado obtido foi com o algoritmo *Random Forest*, onde a precisão final de 92.1% foi atingida. Para obter este resultado, os espaços de cor HSV, LAB e escala de cinza foram utilizados, bem como a técnica de suavização pela mediana para diminuir as transições bruscas nos níveis de intensidade de cores. Também se efetuou a otimização dos hiper-parâmetros, mas sem alteração na precisão final.

Já para integrar a técnica melhor avaliada em um aplicativo para dispositivos móveis, foi desenvolvido um servidor Web que realiza a interface entre a classificação e a aplicação móvel desenvolvida. No aplicativo, além de se ter acesso a artigos informativos, o usuário pode selecionar as imagens do café e da folha de papel, realizar o corte da imagem na área desejada e enviar os arquivos para o servidor. Já o servidor Web é responsável por carregar o modelo pré-treinado, aplicar técnicas de pré-processamento nas imagens e extrair as características selecionadas, com base nas informações obtidas nos experimentos executados. Por fim, os dados são classificados e o valor Agtron da amostra do café é

retornado para o aplicativo, onde é apresentado para o usuário juntamente com algumas informações sobre o valor identificado.

Com base nos resultados da criação da base de dados, experimentos e desenvolvimento da aplicação, é possível concluir que o presente trabalho atingiu o objetivo principal. Os resultados expostos ao longo do desenvolvimento deste estudo mostraram que é possível realizar a classificação do café torrado com base na cor utilizando um dispositivo móvel.

Com o desenvolvimento do presente trabalho, pequenos e médios produtores de café torrado em grãos podem ter acesso a uma forma rápida e de baixo custo para verificar a escala Agtron de seus produtos, reduzindo, assim, a subjetividade da análise visual, colaborando para o controle e qualidade da torra. Os consumidores finais do café torrado em grãos também podem se beneficiar do uso da aplicação, conferindo o grau de torra e aprendendo um pouco mais sobre a escala Agtron, suas características e importância do grau de torra.

Com relação às dificuldades para a concepção do trabalho, destaca-se a etapa de desenvolvimento da base de dados. Realizar a torra das amostras de café em determinada escala Agtron e capturar um número elevado de imagens com diversos dispositivos móveis levou um tempo maior que o esperado.

5.1 Trabalhos futuros

Como possíveis trabalhos futuros, podem-se apontar a aplicação da metodologia desenvolvida para o café moído, aprimoramentos na classificação e melhorias para a aplicação móvel. O café torrado em um determinado valor Agtron, após ser moído, possui uma coloração um pouco mais clara em relação aos grãos. Portanto, para que a aplicação funcione corretamente com o café moído, é necessário que novas imagens sejam capturadas e a metodologia seja reaplicada. Assim, os atributos extraídos irão representar corretamente a escala Agtron correspondente do café torrado e moído.

Possivelmente, para atingir resultados superiores na classificação, a técnica de seleção de hiper-parâmetros pode ser aplicada em todos os algoritmos, para todos os experimentos realizados, pois os hiper-parâmetros de um algoritmo, assim como os atributos selecionados, podem alterar sua precisão final. Os algoritmos como *Naive Bayes* e *Decision Tree* podem não sofrer uma grande alteração em seus resultados, devido ao baixo número de parâmetros que podem ser alterados. Entretanto, o *MultiLayer Perceptron* possui um número elevado de parâmetros e deve atingir precisões superiores às obtidas neste projeto em sua configuração padrão. Além da seleção de hiper-parâmetros, outras técnicas de pré-processamento também podem ser testadas.

Para possíveis melhorias na aplicação móvel e usabilidade, destaca-se a opção de realizar a segmentação da imagem do café, não sendo necessária a seleção manual do usuário. Com a identificação do café, as informações de indicativo de claridade podem

ser extraídas de uma única imagem, utilizando somente a imagem do café sobre uma folha de papel branca. Outra possível melhoria é a etapa de captura da imagem ocorrer diretamente da aplicação. Por fim, a eliminação do servidor pode ser considerada; sendo assim, toda a classificação ocorreria diretamente no dispositivo móvel, sem a necessidade de conexão com a internet.

REFERÊNCIAS

- ALVES, J. J. **Café brasileiro de qualidade**. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia de Alimentos) – Universidade Federal de Uberlândia (UFU), Uberlândia, MG, Brasil, 2019. Disponível em: <https://repositorio.ufu.br/handle/123456789/25554>.
- ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DE CAFÉ (ABIC). **Indicadores da indústria de café - 2020 - ABIC**. 2021. Disponível em: <https://estatisticas.abic.com.br/estatisticas/indicadores-da-industria/indicadores-da-industria-de-cafe-2020/>. Acesso em: 1 ago. 2021.
- _____. **Norma de qualidade recomendável e boas práticas de fabricação de cafés torrados em grão e cafés torrados e moídos**. 2018. Disponível em: <https://www.abic.com.br/wp-content/uploads/2021/06/Norma-PQC-26-06-2018-rev-30.pdf>. Acesso em: 28 ago. 2022.
- _____. **O café brasileiro na atualidade**. 2021. Disponível em: <https://www.abic.com.br/tudo-de-cafe/o-cafe-brasileiro-na-atualidade/>. Acesso em: 21 ago. 2021.
- BARACAT, M. N. d. S. **Café brasileiro de qualidade**. 2020. Trabalho de Conclusão de Curso (Graduação em Engenharia Agrônômica) – Universidade Federal de São Carlos (UFSCar), São Carlos, SP, Brasil, 2020. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/13544>.
- BIØRN-HANSEN, A.; GRØNLI, T.-M.; GHINEA, G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development. **ACM Computing Surveys**, v. 51, p. 1–34, 2018.
- BISWAS, S. **Networking in Flutter using Dio**. 2022. Disponível em: <https://blog.logrocket.com/networking-flutter-using-dio/>. Acesso em: 29 jul. 2022.
- BREIMAN, L. Random forests. **Machine learning**, Springer, Nova York, Estados Unidos, v. 45, n. 1, p. 5–32, 2001.
- CAMILO, C. O.; SILVA, J. C. d. **Mineração de dados: conceitos, tarefas, métodos e ferramentas**. 2009. Universidade Federal de Goiás (UFG), Goiânia, GO, Brasil.
- CELEBI, M. E. Improving the performance of k-means for color quantization. **Image and Vision Computing**, v. 29, n. 4, p. 260–271, 2011.
- CHAVES-GONZÁLEZ, J. M. *et al.* Detecting skin in face recognition systems: a colour spaces study. **Digital Signal Processing**, v. 20, n. 3, p. 806–823, 2010.
- CHERNOV, V.; ALANDER, J.; BOCHKO, V. Integer-based accurate conversion between RGB and HSV color spaces. **Computers & Electrical Engineering**, Elsevier Science, Amsterdã, Países Baixos, v. 46, p. 328–337, 2015.
- DA SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. 2. ed. São Paulo, SP, Brasil: Artliber, 2019.

DONG, C.-w. *et al.* Sensory quality evaluation for appearance of needle-shaped green tea based on computer vision and nonlinear tools. **Journal of Zhejiang University-SCIENCE B**, Zhejiang University Press, Zhejiang, China, v. 18, n. 6, p. 544–548, 2017.

EUGÊNIO, M. H. A. *et al.* Expansão volumétrica de grãos de café arábica submetidos a diferentes pontos de torração. In: CONGRESSO BRASILEIRO DE PESQUISAS CAFEEIRAS (CBPC), XXXVII., 2011, Poços de Caldas, MG, Brasil. **Anais [...]**. Brasília, DF, Brasil: Embrapa Café, 2011. P. 1–2. Disponível em: <http://www.sbicafe.ufv.br/handle/123456789/5831>. Acesso em: 3 ago. 2021.

FARRAHI MOGHADDAM, R. *et al.* A maximal-information color to gray conversion method for document images: toward an optimal grayscale representation for document image binarization, 2013.

FERREIRA, A. D.; CARVALHO, G. R.; NADALETI, D. H. S. Caracterização agronômica e sensorial de diferentes genótipos de Bourbon visando à produção de cafés especiais. **Boletim de Pesquisa e Desenvolvimento**, Embrapa Café, Brasília, DF, Brasil, n. 2, p. 1–40, 2021. Disponível em: <http://ainfo.cnptia.embrapa.br/digital/bitstream/item/223040/1/Embrapa-Cafe-Boletim-n2-VersaoFinal.pdf>.

FLASK. **User's Guide**. 2021. Disponível em: <https://flask.palletsprojects.com/en/latest/>. Acesso em: 10 nov. 2021.

FLUTTER. **Flutter documentation**. 2021. Disponível em: <https://docs.flutter.dev/>. Acesso em: 13 out. 2021.

FLUTTERANDO. **What is Modular?** 2022. Disponível em: <https://modular.flutterando.com.br/docs/intro/>. Acesso em: 28 jul. 2022.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Porto Alegre, RS, Brasil: Editora da UFRGS, 2009.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo, SP, Brasil: Editora atlas, 2002.

GOMES, J. F. S.; LETA, F. R. Applications of computer vision techniques in the agriculture and food industry: a review. **European Food Research and Technology**, Springer, Berlin, Alemanha, v. 235, n. 6, p. 989–1000, 2012.

GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. **Digital Image Processing Using MATLAB**. Nova Jersey, Estados Unidos: Prentice-Hall, Inc., 2003.

GONZALEZ VIEJO, C. *et al.* Emerging Technologies Based on Artificial Intelligence to Assess the Quality and Consumer Preference of Beverages. **Beverages**, v. 5, n. 4, 2019. Disponível em: <https://www.mdpi.com/2306-5710/5/4/62>.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. Burlington, Massachusetts, Estados Unidos: Morgan Kaufmann Publishers, 2012.

HUNT, R. W. G. Chapter 9 - Colour Displays and Colorimetry. In: JACKSON, K.; TOWNSEND, G. (Ed.). **TV and Video Engineer's Reference Book**. Oxônia, Reino Unido: Butterworth-Heinemann, 1991. P. 1–12.

HUNT, R. W. G.; POINTER, M. R. **Measuring colour**. Reino Unido: John Wiley & Sons, 2011.

INTERNATIONAL DATA CORPORATION (IDC). **Smartphone Market Share**. 2021. Disponível em: <https://www.idc.com/promo/smartphone-market-share>. Acesso em: 14 out. 2021.

EL-KASSAS, W. S. *et al.* Taxonomy of Cross-Platform Mobile Applications Development Approaches. **Ain Shams Engineering Journal**, v. 8, n. 2, p. 163–190, 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2090447915001276>.

KINGSFORD, C.; SALZBERG, S. L. What are decision trees? **Nature Biotechnology**, v. 26, n. 9, p. 1011–1013, 2008.

KOLKUR, S. *et al.* Human Skin Detection Using RGB, HSV and YCbCr Color Models. Atlantis Press, Amsterdã, Holanda, v. 137, p. 324–332, 2017.

LEME, D. S. **Sistema Computacional para Integração de Dados na Análise de Cafés Especiais**. 2016. Dissertação (Mestrado em Engenharia de Sistemas e Automação) – Universidade Federal de Lavras (UFLA), Lavras, MG, Brasil, 2016. Disponível em: <http://www.sbicafe.ufv.br/handle/123456789/8505>.

LIMA, R. R. D. *et al.* Relação entre os sistemas colorimétricos para café torrado e moído com diferentes tonalidades. In: SIMPÓSIO DE PESQUISA DOS CAFÉS DO BRASIL (SPCB), IX., 2015, Curitiba, PR, Brasil. **Anais [...]**. Brasília, DF, Brasil: Embrapa Café, 2015. P. 1–6. Disponível em: <http://www.sbicafe.ufv.br/handle/123456789/3605>. Acesso em: 11 out. 2021.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

MELO, W. L. B. A importância da informação sobre do grau de torra do café e sua influência nas características organolépticas da bebida. **Embrapa Instrumentação Agropecuária: Comunicado Técnico**, Embrapa Instrumentação Agropecuária, São Carlos, SP, Brasil, v. 58, p. 1–4, 2004. Disponível em: <https://agris.fao.org/agris-search/search.do?recordID=BR2004130170>.

MITCHELL, T. M. **Machine Learning**. 1. ed. Nova York, Estados Unidos: McGraw-Hill, 1997. (McGraw-Hill International Editions).

NASUTION, T. H.; ANDAYANI, U. Recognition of Roasted Coffee Bean Levels using Image Processing and Neural Network. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, Bristol, Reino Unido, v. 180, p. 1–8, 2017. Disponível em: <https://doi.org/10.1088/1757-899x/180/1/012059>.

OBLITAS CRUZ, J. F.; CASTRO-SILUPU, W. Computer vision system for the optimization of the color generated by the coffee roasting process according to time, temperature and mesh size. **Ingenieria y Universidad**, v. 18, n. 2, p. 355–368, 2014. DOI: 10.11144/Javeriana.IYU18-2.cvso.

OPEN SOURCE COMPUTER VISION LIBRARY (OPENCV). **About OpenCV**. 2021. Disponível em: <https://opencv.org/about/>. Acesso em: 10 nov. 2021.

OPEN SOURCE COMPUTER VISION LIBRARY (OPENCV). **Image Blurring**. 2015. Disponível em: https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html. Acesso em: 16 nov. 2021.

ORCHARD, M.; BOUMAN, C. Color quantization of images. **IEEE Transactions on Signal Processing**, Citeseer, Pensilvânia, Estados Unidos, v. 39, n. 12, p. 2677–2690, 1991.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIRES, F. d. C. **Métodos instrumentais alternativos para a predição da cor do café torrado**. 2019. Dissertação (Mestrado em Ciência dos Alimentos) – Universidade Federal de Lavras (UFLA), Lavras, MG, Brasil, 2019. Disponível em: <http://repositorio.ufla.br/jspui/handle/1/38010>.

PYTHON SOFTWARE FOUNDATION. **General Python FAQ**. 2021. Disponível em: <https://docs.python.org/3/faq/general.html>. Acesso em: 13 out. 2021.

PYTHONANYWHERE. **Get a Python website in minutes**. 2021. Disponível em: <https://www.pythonanywhere.com/>. Acesso em: 22 set. 2022.

QUINLAN, J. Decision trees and decision-making. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 20, n. 2, p. 339–346, 1990.

RASCHKA, S. *Naive Bayes and Text Classification I: Introduction and Theory*, 2014.

RAUBER, T. W. *Redes neurais artificiais*. **Universidade Federal do Espírito Santo (UFES)**, Vitória, ES, Brasil, 2005.

RISH, I. An Empirical Study of the Naïve Bayes Classifier. In: INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE (IJCAI), XVII., 2001, Seattle, Washington, Estados Unidos. **Proceedings** [...]. Seattle, Washington, Estados Unidos, 2001. v. 3, p. 41–46.

SABBAGH, R. **Scrum: gestão ágil para projetos de sucesso**. São Paulo, SP, Brasil: Editora Casa do Código, 2014.

SILVA, I. S. **Obtenção de cafés especiais pela fermentação**. 2021. Trabalho de Conclusão de Curso (Graduação em Engenharia de Alimentos) – Universidade Federal de Uberlândia (UFU), Patos de Minas, MG, Brasil, 2021. Disponível em: <https://repositorio.ufu.br/handle/123456789/32488>.

SOLMAZ, M. E. *et al.* Quantifying colorimetric tests using a smartphone app based on machine learning classifiers. **Sensors and Actuators B: Chemical**, v. 255, p. 1967–1973, 2018.

WANG, X.; LIM, L.-T. Physicochemical Characteristics of Roasted Coffee. In: PREEDY, V. R. (Ed.). **Coffee in Health and Disease Prevention**. San Diego, Califórnia, Estados Unidos: Academic Press, 2015. cap. 27, p. 247–254.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. Rio de Janeiro, RJ, Brasil: Elsevier, 2009.

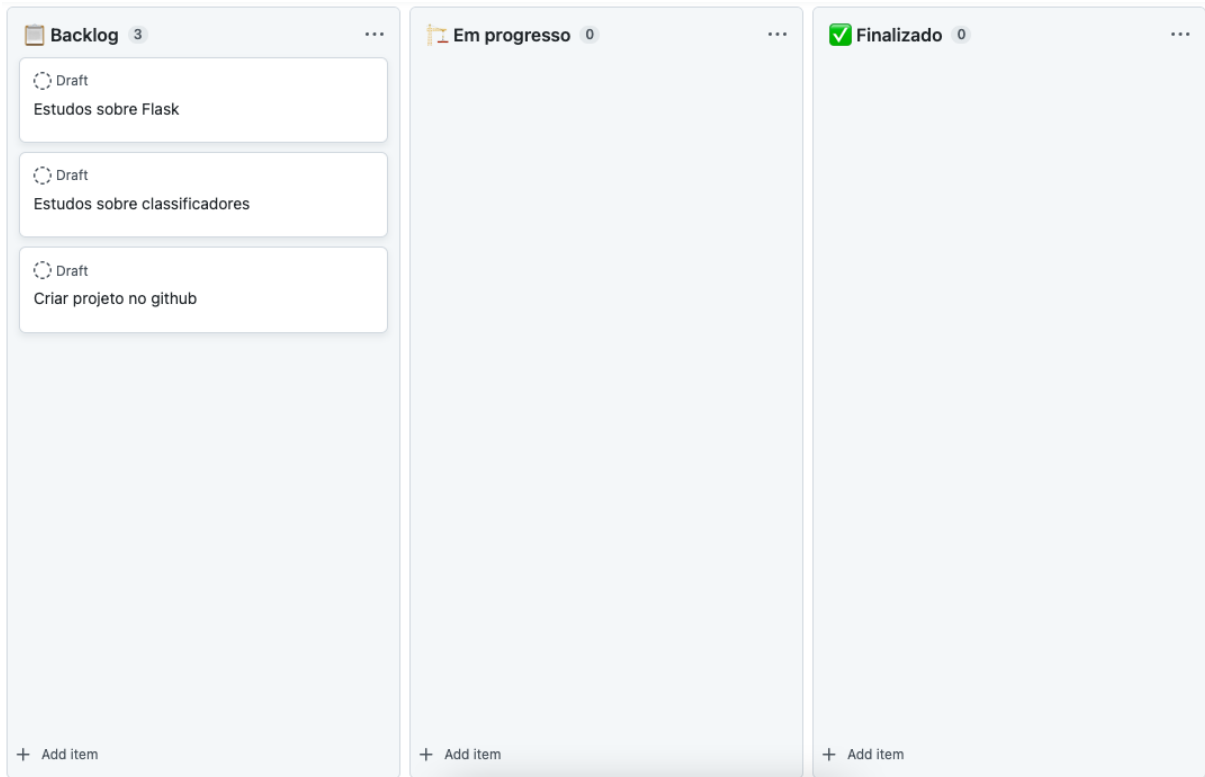
WU, D.; SUN, D. W. Colour measurements by computer vision for food quality control: a review. **Trends in Food Science & Technology**, Elsevier Science, Amsterdã, Países Baixos, v. 29, n. 1, p. 5–20, 2013.

XU, G. *et al.* Bearing Fault Diagnosis Method Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning. **Sensors**, v. 19, n. 5, 2019.

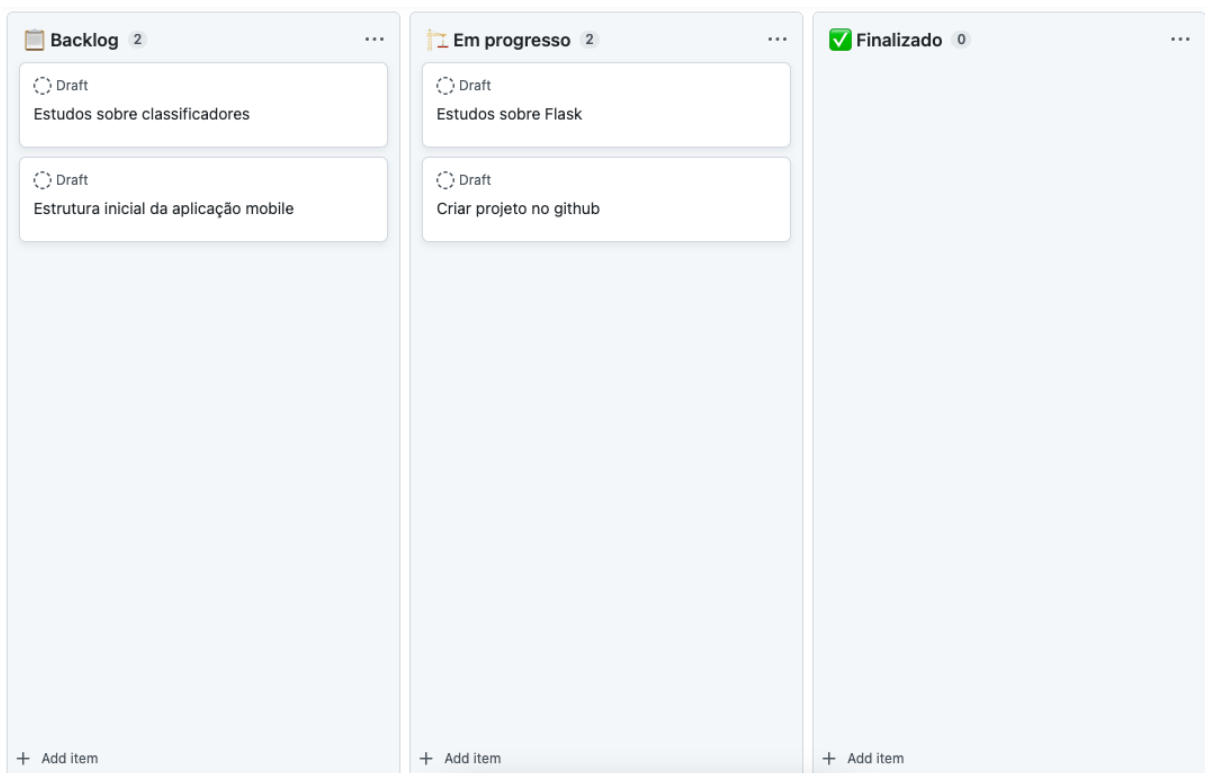
APÊNDICES

APÊNDICE A – HISTÓRICO DO QUADRO *KANBAN*

Semanas 1 e 2



Semanas 3 e 4



Semanas 5 e 6

The Kanban board for Semanas 5 e 6 is organized into three columns: Backlog, Em progresso, and Finalizado. Each column contains two draft items.

Column	Item 1	Item 2
Backlog (2)	Draft: Estudo sobre espaços de cores	Draft: Estudos sobre beneficiamento e seleção do café
Em progresso (2)	Draft: Estrutura inicial da aplicação mobile	Draft: Estudos sobre classificadores
Finalizado (2)	Draft: Estudos sobre Flask	Draft: Criar projeto no github

Semanas 7 e 8

The Kanban board for Semanas 7 e 8 is organized into three columns: Backlog, Em progresso, and Finalizado. Each column contains two draft items.

Column	Item 1	Item 2		
Backlog (2)	Draft: Estudos sobre conversão de espaços de cores	Draft: Separação das amostras de café		
Em progresso (2)	Draft: Estudo sobre espaços de cores	Draft: Estudos sobre beneficiamento e seleção do café		
Finalizado (4)	Draft: Estudos sobre Flask	Draft: Criar projeto no github	Draft: Estrutura inicial da aplicação mobile	Draft: Estudos sobre classificadores

Semanas 9 e 10

A Kanban board with three columns: Backlog (3 items), Em progresso (3 items), and Finalizado (5 items). Each item is a 'Draft' task.

Column	Count	Task
Backlog	3	Draft: Iniciar torra do café
		Draft: Construir estrutura para captura das imagens
		Draft: Separar dispositivos para captura
Em progresso	3	Draft: Estudo sobre espaços de cores
		Draft: Separação das amostras de café
		Draft: Estudos sobre conversão de espaços de cores
Finalizado	5	Draft: Estudos sobre Flask
		Draft: Criar projeto no github
		Draft: Estrutura inicial da aplicação mobile
		Draft: Estudos sobre classificadores
		Draft: Estudos sobre beneficiamento e seleção do café

Semanas 11 e 12

A Kanban board with three columns: Backlog (3 items), Em progresso (3 items), and Finalizado (8 items). Each item is a 'Draft' task.

Column	Count	Task
Backlog	3	Draft: Estudar como persistir o modelo de classificação
		Draft: Codificar algoritmos de conversão de espaços de cores
		Draft: Codificar algoritmos de extração de características
Em progresso	3	Draft: Iniciar torra do café
		Draft: Construir estrutura para captura das imagens
		Draft: Separar dispositivos para captura
Finalizado	8	Draft: Criar projeto no github
		Draft: Estrutura inicial da aplicação mobile
		Draft: Estudos sobre classificadores
		Draft: Estudos sobre beneficiamento e seleção do café
		Draft: Estudo sobre espaços de cores
		Draft: Estudos sobre conversão de espaços de cores
		Draft: Separação das amostras de café

Semanas 13 e 14

Backlog 2

- Draft: Codificar algoritmos de extração de características
- Draft: Iniciar captura das imagens

+ Add item

Em progresso 3

- Draft: Construir estrutura para captura das imagens
- Draft: Estudar como persistir o modelo de classificação
- Draft: Codificar algoritmos de conversão de espaços de cores

+ Add item

Finalizado 10

- Draft: Estudos sobre classificadores
- Draft: Estudos sobre beneficiamento e seleção do café
- Draft: Estudo sobre espaços de cores
- Draft: Estudos sobre conversão de espaços de cores
- Draft: Separação das amostras de café
- Draft: Separar dispositivos para captura
- Draft: Iniciar torra do café

+ Add item

Semanas 15 e 16

Backlog 3

- Draft: Construção do dataset para testes
- Draft: Testes de persistência do modelo
- Draft: Extração de características das imagens

+ Add item

Em progresso 2

- Draft: Iniciar captura das imagens
- Draft: Codificar algoritmos de extração de características

+ Add item

Finalizado 13

- Draft: Estudos sobre conversão de espaços de cores
- Draft: Separação das amostras de café
- Draft: Separar dispositivos para captura
- Draft: Iniciar torra do café
- Draft: Construir estrutura para captura das imagens
- Draft: Estudar como persistir o modelo de classificação
- Draft: Codificar algoritmos de conversão de espaços de cores

+ Add item

Semanas 17 e 18

Backlog 2	Em progresso 3	Finalizado 14
Draft Construção do dataset para testes	Draft Iniciar captura das imagens	✓ Separação das amostras de care
Draft Estudos sobre validação cruzada	Draft Testes de persistência do modelo	Draft Separar dispositivos para captura
	Draft Extração de características das imagens	Draft Iniciar torra do café
		Draft Construir estrutura para captura das imagens
		Draft Estudar como persistir o modelo de classificação
		Draft Codificar algoritmos de conversão de espaços de cores
		Draft Codificar algoritmos de extração de características
+ Add item	+ Add item	+ Add item

Semanas 19 e 20

Backlog 3	Em progresso 2	Finalizado 17
Draft Testes com classificadores	Draft Construção do dataset para testes	✓ Construir estrutura para captura das imagens
Draft Implementação da validação cruzada	Draft Estudos sobre validação cruzada	Draft Estudar como persistir o modelo de classificação
Draft Escolha de métricas de avaliação		Draft Codificar algoritmos de conversão de espaços de cores
		Draft Codificar algoritmos de extração de características
		Draft Iniciar captura das imagens
		Draft Testes de persistência do modelo
		Draft Extração de características das imagens
+ Add item	+ Add item	+ Add item

Semanas 21 e 22

Backlog 3

- Draft: Captura das imagens
- Draft: Implementação do algoritmo de extração de matriz de confusão
- Draft: Implementação das métricas de avaliação

+ Add item

Em progresso 3

- Draft: Testes com classificadores
- Draft: Implementação da validação cruzada
- Draft: Escolha de métricas de avaliação

+ Add item

Finalizado 19

- Codificar algoritmos de conversão de espaços de cores
- Draft: Codificar algoritmos de extração de características
- Draft: Iniciar captura das imagens
- Draft: Testes de persistência do modelo
- Draft: Extração de características das imagens
- Draft: Construção do dataset para testes
- Draft: Estudos sobre validação cruzada

+ Add item

Semanas 23 e 24

Backlog 1

- Draft: Captura das imagens

+ Add item

Em progresso 3

- Draft: Testes com classificadores
- Draft: Implementação das métricas de avaliação
- Draft: Implementação do algoritmo de extração de matriz de confusão

+ Add item

Finalizado 21

- Iniciar captura das imagens
- Draft: Testes de persistência do modelo
- Draft: Extração de características das imagens
- Draft: Construção do dataset para testes
- Draft: Estudos sobre validação cruzada
- Draft: Implementação da validação cruzada
- Draft: Escolha de métricas de avaliação

+ Add item

Semanas 25 e 26

Backlog 2	Em progresso 1	Finalizado 24
<ul style="list-style-type: none"> Draft: Testes com cameras digitais Draft: Estudos sobre algoritmos de pré-processamento 	<ul style="list-style-type: none"> Draft: Incrementar base de dados 	<ul style="list-style-type: none"> Draft: Construção do dataset para testes Draft: Estudos sobre validação cruzada Draft: Implementação da validação cruzada Draft: Escolha de métricas de avaliação Draft: Implementação do algoritmo de extração de matriz de confusão Draft: Implementação das métricas de avaliação Draft: Testes com classificadores
+ Add item	+ Add item	+ Add item

Semanas 27 e 28

Backlog 3	Em progresso 3	Finalizado 24
<ul style="list-style-type: none"> Draft: Criar organização no github Draft: Desenvolvimento do aplicativo Draft: Escolha de atributos 	<ul style="list-style-type: none"> Draft: Incrementar base de dados Draft: Testes com cameras digitais Draft: Estudos sobre algoritmos de pré-processamento 	<ul style="list-style-type: none"> Draft: Construção do dataset para testes Draft: Estudos sobre validação cruzada Draft: Implementação da validação cruzada Draft: Escolha de métricas de avaliação Draft: Implementação do algoritmo de extração de matriz de confusão Draft: Implementação das métricas de avaliação Draft: Testes com classificadores
+ Add item	+ Add item	+ Add item

Semanas 29 e 30

Backlog 2	Em progresso 5	Finalizado 25
Draft Extração de informações sobre flash	Draft Incrementar base de dados	Draft Estudos sobre validação cruzada
Draft Testes com algoritmos de pré-processamento	Draft Estudos sobre algoritmos de pré-processamento	Draft Implementação da validação cruzada
	Draft Escolha de atributos	Draft Escolha de métricas de avaliação
	Draft Criar organização no github	Draft Implementação do algoritmo de extração de matriz de confusão
	Draft Desenvolvimento do aplicativo	Draft Implementação das métricas de avaliação
		Draft Testes com classificadores
		Draft Testes com cameras digitais
+ Add item	+ Add item	+ Add item

Semanas 31 e 32

Backlog 1	Em progresso 3	Finalizado 28
Draft Extração de informações sobre flash	Draft Incrementar base de dados	Draft Implementação do algoritmo de extração de matriz de confusão
	Draft Desenvolvimento do aplicativo	Draft Implementação das métricas de avaliação
	Draft Testes com algoritmos de pré-processamento	Draft Testes com classificadores
		Draft Testes com cameras digitais
		Draft Criar organização no github
		Draft Escolha de atributos
		Draft Estudos sobre algoritmos de pré-processamento
+ Add item	+ Add item	+ Add item

Semanas 33 e 34

Backlog 2

- Draft: Estudo sobre algoritmos de seleção de hiperparâmetros
- Draft: Desenvolvimento da API Web

+ Add item

Em progresso 2

- Draft: Desenvolvimento do aplicativo
- Draft: Extração de informações sobre flash

+ Add item

Finalizado 30

- Draft: Testes com classificadores
- Draft: Testes com cameras digitais
- Draft: Criar organização no github
- Draft: Escolha de atributos
- Draft: Estudos sobre algoritmos de pré-processamento
- Draft: Incrementar base de dados
- Draft: Testes com algoritmos de pré-processamento

+ Add item

Semanas 35 e 36

Backlog 1

- Draft: Hospedagem da API

+ Add item

Em progresso 2

- Draft: Estudo sobre algoritmos de seleção de hiperparâmetros
- Draft: Desenvolvimento da API Web

+ Add item

Finalizado 32

- Draft: Criar organização no github
- Draft: Escolha de atributos
- Draft: Estudos sobre algoritmos de pré-processamento
- Draft: Incrementar base de dados
- Draft: Testes com algoritmos de pré-processamento
- Draft: Extração de informações sobre flash
- Draft: Desenvolvimento do aplicativo

+ Add item

Semanas 37 e 38

Backlog 1

- Draft: Implementação dos algoritmos de seleção de hiperparâmetros

+ Add item

Em progresso 2

- Draft: Hospedagem da API
- Draft: Estudo sobre algoritmos de seleção de hiperparâmetros

+ Add item

Finalizado 33

- Draft: Escolha de atributos
- Draft: Estudos sobre algoritmos de pré-processamento
- Draft: Incrementar base de dados
- Draft: Testes com algoritmos de pré-processamento
- Draft: Extração de informações sobre flash
- Draft: Desenvolvimento do aplicativo
- Draft: Desenvolvimento da API Web

+ Add item

Semanas 39 e 40

Backlog 0

+ Add item

Em progresso 1

- Draft: Implementação dos algoritmos de seleção de hiperparâmetros

+ Add item

Finalizado 35

- Draft: Incrementar base de dados
- Draft: Testes com algoritmos de pré-processamento
- Draft: Extração de informações sobre flash
- Draft: Desenvolvimento do aplicativo
- Draft: Desenvolvimento da API Web
- Draft: Hospedagem da API
- Draft: Estudo sobre algoritmos de seleção de hiperparâmetros

+ Add item

Semanas 41 e 42

The image shows a Kanban board with three columns: Backlog, Em progresso, and Finalizado. The Backlog column is empty with 0 items. The Em progresso column is also empty with 0 items. The Finalizado column contains 36 items, all marked as Draft. The items in the Finalizado column are:

- Testes com algoritmos de pré-processamento
- Draft: Extração de informações sobre flash
- Draft: Desenvolvimento do aplicativo
- Draft: Desenvolvimento da API Web
- Draft: Hospedagem da API
- Draft: Estudo sobre algoritmos de seleção de hiperparâmetros
- Draft: Implementação dos algoritmos de seleção de hiperparâmetros

Each column has an 'Add item' button at the bottom.