



UNIVERSIDADE
ESTADUAL DE LONDRINA

VINÍCIUS CESAR DOS SANTOS

USO DE IMAGENS TÉRMICAS PARA IDENTIFICAÇÃO
DA COMPOSIÇÃO DE ALIMENTOS

LONDRINA

2022

VINÍCIUS CESAR DOS SANTOS

**USO DE IMAGENS TÉRMICAS PARA IDENTIFICAÇÃO
DA COMPOSIÇÃO DE ALIMENTOS**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Sylvio Barbon Junior

Coorientador: Profa. Me. Jessica Fernandes Lopes

LONDRINA

2022

VINÍCIUS CESAR DOS SANTOS

**USO DE IMAGENS TÉRMICAS PARA IDENTIFICAÇÃO
DA COMPOSIÇÃO DE ALIMENTOS**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Sylvio Barbon Junior
Universidade Estadual de Londrina

Prof. Dr. Jacques Duílio Brancher
Universidade Estadual de Londrina – UEL

Profa. Dra. Helen Cristina de Mattos
Senefonte
Universidade Estadual de Londrina – UEL

Londrina, 27 de junho de 2022.

*Dedico esse trabalho à todos aqueles que, em
algum momento da vida, aspiraram se
tornar um cientista.*

AGRADECIMENTOS

Agradeço ao meu professor orientador, Sylvio, por abrir as portas e me aceitar nesse vasto mundo da visão computacional e do aprendizado de máquina. Também a minha coorientadora, Jessica, por ter me auxiliado no projeto.

Agradeço aos ótimos professores do departamento de computação, que me forneceram toda a base necessária para chegar até aqui.

Agradeço aos meus amigos por sempre estarem ao meu lado, principalmente nos momentos mais difíceis, sendo o alívio necessário para continuar trilhando o caminho.

Por mim, agradeço a minha família, que sempre me apoiou e fez tudo ser possível.

“Os que se encantam com a prática sem a ciência são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino.”
(Leonardo da Vinci)

SANTOS, V. C.. **Uso de Imagens Térmicas para Identificação da Composição de Alimentos**. 2022. 39f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2022.

RESUMO

A indústria alimentícia utiliza diversos métodos para analisar a composição de alimentos, seja para segurança alimentar, regulamentações governamentais, controle de qualidade ou pesquisa e desenvolvimento. Sendo assim, esse trabalho buscará o desenvolvimento de um novo método simples, rápido e barato para análise da composição de alimentos. Usando visão computacional e imagens térmicas, com um modelo de aprendizado de máquina (*machine learning*) que utilizará *random forests* para aprender o padrão de resfriamento natural das amostras e conseguir diferenciá-las.

Palavras-chave: Análise de Alimentos. Visão Computacional. Imagem Térmica. Aprendizado de Máquina. Floresta Aleatória.

SANTOS, V. C.. **Use of Thermal Imaging to identify Food Composition**. 2022. 39p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina, 2022.

ABSTRACT

The food industry uses several methods to analyze the food composition for food safety, government regulations, quality control or research and development. Therefore, this paper intends to develop a simple, fast and cheap method to analyse the food composition. Using computer vision and thermal imaging, with a machine learning model using random forests to detect the natural cooling pattern of the samples and be able to differentiate them.

Keywords: Food Analysis. Computer Vision. Thermal Imaging. Machine Learning. Random Forest.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Fluxo da visão geral do projeto | 13 |
| Figura 2 – Representação matricial de uma imagem digital | 15 |
| Figura 3 – Diferença das tonalidades de cores possíveis de serem representadas dependendo da quantidade de bits necessários para armazená-las | 15 |
| Figura 4 – Cubo de cores RGB de 24 bits | 15 |
| Figura 5 – Divisões do espectro eletromagnético | 16 |
| Figura 6 – Algumas das aplicações da visão computacional | 18 |
| Figura 7 – Diferença da função de um modelo que treinou o necessário para generalizar bem (esquerda) e um modelo que treinou excessivamente, sofrendo um sobreajuste (direita) [1] | 20 |
| Figura 8 – Exemplo simples de uma Árvore de Decisão que decide como um estudante passará a sua noite [1] | 21 |
| Figura 9 – Ilustração de uma Árvore de Decisão com Replicação [2] | 24 |
| Figura 10 – Exemplo da aplicação da Validação Cruzada com 5 <i>folds</i> | 25 |
| Figura 11 – Alguns <i>frames</i> das amostras resfriando conforme o tempo | 27 |
| Figura 12 – Diferença da imagem antes e após reduzir a profundidade de cor | 28 |
| Figura 13 – Diferença de uma distribuição de probabilidades com <i>Skewness</i> positivo e negativo | 28 |
| Figura 14 – Diferença de uma distribuição de probabilidades com <i>Kurtosis</i> (S_{ku}) maior, menor e igual a 0 | 29 |
| Figura 15 – Importância das <i>features</i> agrupando pela Média | 31 |
| Figura 16 – Importância das <i>features</i> agrupando pela Variância | 32 |
| Figura 17 – Importância das <i>features</i> agrupando pelo Desvio Padrão | 32 |
| Figura 18 – Importância das <i>features</i> agrupando pela Soma | 33 |
| Figura 19 – Resultados obtidos nas execuções com agrupamento | 34 |
| Figura 20 – Exemplos de classificações corretas e incorretas em uma das iterações do agrupamento pela soma, sendo um exemplo para cada concentração de fibra | 36 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Regiões do espectro infravermelho | 16 |
| Tabela 2 – Parâmetros fornecidos para o algoritmo de seleção de melhores parâmetros | 30 |
| Tabela 3 – Melhores parâmetros selecionados para o agrupamento com a média . . | 30 |
| Tabela 4 – Melhores parâmetros selecionados para o agrupamento com a variância | 30 |
| Tabela 5 – Melhores parâmetros selecionados para o agrupamento com o desvio padrão | 30 |
| Tabela 6 – Melhores parâmetros selecionados para o agrupamento com soma . . . | 31 |
| Tabela 7 – Resultados das acurácias por agrupamento | 35 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|--------------------------------------|
| CVS | <i>Computer Vision System</i> |
| FIR | <i>Far-infrared</i> |
| IR | <i>Infrared Radiation</i> |
| LWIR | <i>Long-wavelength Infrared</i> |
| MWIR | <i>Mid-wavelength Infrared</i> |
| NIR | <i>Near-infrared</i> |
| PCA | <i>Principal Components Analysis</i> |
| RGB | <i>Red, Green and Blue</i> |
| SVM | <i>Support Vector Machine</i> |
| SWIR | <i>Short-wavelength Infrared</i> |
| TIR | <i>Thermal Infrared</i> |

SUMÁRIO

| | | |
|---------|---|----|
| 1 | INTRODUÇÃO | 12 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 | Análise da Composição de Alimentos | 14 |
| 2.2 | Imagens Digitais | 14 |
| 2.3 | Imagens Térmicas | 16 |
| 2.3.1 | Aplicações na Área Alimentícia | 17 |
| 2.4 | Visão Computacional | 17 |
| 2.5 | Aprendizado de Máquina | 19 |
| 2.5.1 | Árvores de Decisão | 20 |
| 2.5.1.1 | Entropia e o Ganho de Informação | 22 |
| 2.5.1.2 | Coefficiente de Gini | 22 |
| 2.5.1.3 | Critérios de Parada | 22 |
| 2.5.1.4 | Métodos de Poda | 23 |
| 2.5.1.5 | Vantagens e Desvantagens | 23 |
| 2.5.2 | <i>Random Forests</i> | 23 |
| 2.5.3 | Validação Cruzada | 25 |
| 3 | DESENVOLVIMENTO | 27 |
| 3.1 | Base de Dados | 27 |
| 3.2 | Pré-Processamento | 27 |
| 3.3 | <i>Feature Selection</i> | 28 |
| 3.4 | Seleção dos Parâmetros | 29 |
| 3.5 | Importância das <i>Features</i> | 31 |
| 4 | RESULTADOS | 34 |
| 4.1 | Discussão visual das amostras classificadas | 35 |
| 5 | CONCLUSÃO | 37 |
| | REFERÊNCIAS | 38 |

1 INTRODUÇÃO

Analisar a composição dos alimentos é algo necessário para as mais diversas aplicações na indústria alimentícia. Por conta disto, existem vários métodos que vão desde a queima e análise das cinzas, até métodos químicos, mecânicos, espectroscópicos, entre muitos outros. Como cada método é adequado para um objetivo ou composto diferente, existe uma necessidade contínua para o desenvolvimento de novos métodos que sejam mais robustos, eficientes, sensíveis e econômicos [3].

Um desses métodos tradicionais para análise da composição de alimentos é a Análise de Cinzas, que analisa o material obtido a partir da queima do alimento. Apesar dessa técnica ser eficaz em muitos casos, pode também ser cara, demorada ou complexa a ponto de necessitar de considerável perícia técnica para realizá-la [3].

Dito isso, esse trabalho consiste no desenvolvimento de um método que analise a composição dos alimentos, visando ser mais simples, mas igualmente eficaz se comparado com métodos tradicionais, como o da análise de cinzas. Por meio do uso de um Sistema de Visão Computacional, do inglês *Computer Vision System* (CVS), imagens térmicas e aprendizado de máquina.

Com o uso da visão computacional, podemos extrair os dados referentes ao resfriamento natural das amostras utilizando a sequência de imagens térmicas capturadas de cada amostra aquecida. Com esses dados em mãos, podemos treinar um algoritmo de aprendizado de máquina, para que classifique cada amostra. Com isso, conseguiremos diferenciá-las umas das outras de acordo com cada composição.

Sistemas de visão computacional já são amplamente utilizados na área alimentícia, assim como algoritmos de aprendizado de máquina. Como exemplo, [4] avalia a qualidade de maçãs com uma e duas cores utilizando imagens multiespectrais, segmentando as imagens com o algoritmo Fuzzy C-Means e fazendo a classificação com o algoritmo SVM. Já [5] apresenta um método para classificar a fermentação de grãos de cacau a partir de imagens desses grãos e um algoritmo de *random forest*. Essas técnicas também já foram usadas em alimentos de origem animal, como por exemplo para classificação de marmoreio em carne [6].

Utilizar um CVS para análise de alimentos possibilita diversas vantagens. Por ser baseado em imagens, é não destrutivo, pois não é preciso contato com os alimentos para captura das amostras [7]. Possui um custo baixo, já que uma simples câmera térmica é o suficiente para realizar a captura. Também é rápido, no caso do nosso estudo, leva apenas o tempo necessário para capturar as imagens da amostra resfriando. Por fim, possui alta acurácia e precisão, caso sejam utilizados os processamentos e algoritmos adequados. Além

disso, não é necessário pessoal altamente treinado para realizar a análise, pois com pouca instrução uma pessoa já é capaz de manipular uma câmera térmica.

A seguir estão listados os exatos objetivos que buscam ser atingidos com esse trabalho.

- a) extrair as informações de resfriamento da sequência de imagens de cada amostra.
- b) treinar um modelo de aprendizado de máquina que classifique as amostras, com boa acurácia.
- c) conseguir descobrir a composição de novas amostras que não possuem sua composição previamente classificada.

A Figura 1 apresenta o fluxo geral do projeto. A ideia proposta é utilizar as imagens térmicas de alimentos previamente aquecidos, durante seu processo de resfriamento. Então, esses dados seriam processados e fornecidos para um modelo de *machine learning*, que, com base na análise do padrão de resfriamento, classificará o composto do alimento.

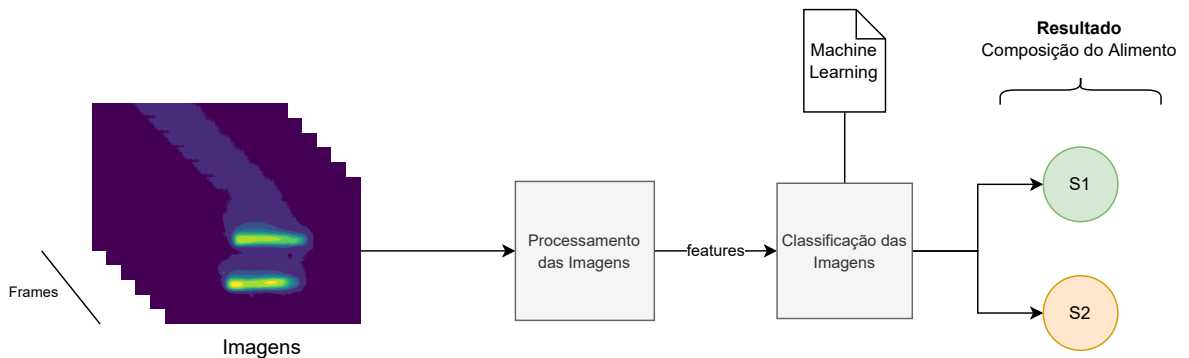


Figura 1 – Fluxo da visão geral do projeto

O Capítulo 2 traz a fundamentação teórica que foi utilizada como base para o desenvolvimento desse trabalho, apresentando conceitos de análise de alimentos, imagens digitais, imagens térmicas e aprendizado de máquina. O Capítulo 3 discute as metodologias utilizadas no decorrer do trabalho, desde a base de dados até o treinamento do modelo de aprendizado de máquina. O Capítulo 4 mostra os resultados que foram obtidos. Por fim, o Capítulo 5 finaliza e propõe estudos futuros para tentar melhorar os resultados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Análise da Composição de Alimentos

Analisar a composição dos alimentos é de extrema importância para diversas aplicações como: Segurança Alimentar, Regulamentações Governamentais, Controle de Qualidade, Pesquisa e Desenvolvimento, além das tendências e demandas dos consumidores que variam de tempos em tempos [3].

Existem várias técnicas para realizar essa análise, cada uma delas com um determinado propósito. A escolha de uma técnica para rotulagem nutricional vai ser diferente se comparada a uma para controle de qualidade, por exemplo. Ademais, o sucesso de um método de análise depende diretamente da seleção e preparação das amostras de alimento, da performance da análise, e dos devidos cálculos e interpretações dos dados obtidos [3].

Entre os vários métodos existentes, os métodos que utilizam imagens como base para a análise possuem vantagens por serem não destrutivos e não possuírem a necessidade de entrar em contato com a amostra para funcionar, preservando-a. Isso é importante pois permite que a amostra seja usada para várias análises [8].

Entre os métodos de análise que utilizam imagens, encontra-se: Imagem Hiperespectral, Imagem de Ressonância Magnética, Imagens de Raio X, Imagens de Ultrassom, Imagem de Fluorescência, Imagens de Refletância de Iluminação Estruturada, Imagens Térmicas, entre outras [8] [7] [9]. O trabalho utilizará imagens térmicas para as análises.

2.2 Imagens Digitais

Uma imagem é uma função de intensidade bidimensional, denotada por $f(x, y)$, em que o valor ou amplitude de f nas coordenadas espaciais (x, y) corresponde à intensidade da imagem naquele ponto [10].

Em uma imagem digital, esse conceito é representado por uma matriz, onde cada célula dessa matriz é chamada de pixel, e armazena um valor de intensidade, como mostra a Figura 2 [10]. Esse valor de intensidade pode variar conforme a profundidade de cor da imagem, ou seja, quantos bits são necessários para armazenar esses valores. Quanto mais bits, maior a quantidade de cores e tons da imagem, conforme mostra a Figura 3.

Cada pixel pode ter mais de um valor, para aumentar ainda mais a gama de cores. No sistema RGB, por exemplo, são armazenados três valores de intensidade. Um para a cor vermelha (em inglês, *red*), outro para a cor verde (em inglês, *green*) e um último para a cor azul (em inglês, *blue*). São essas três cores, pois, se misturadas, elas conseguem

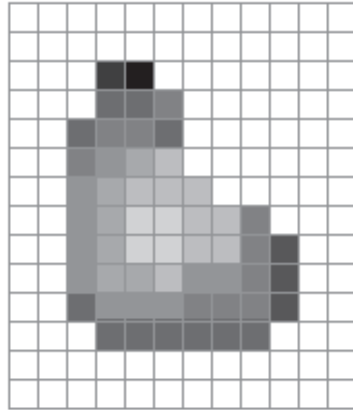


Figura 2 – Representação matricial de uma imagem digital

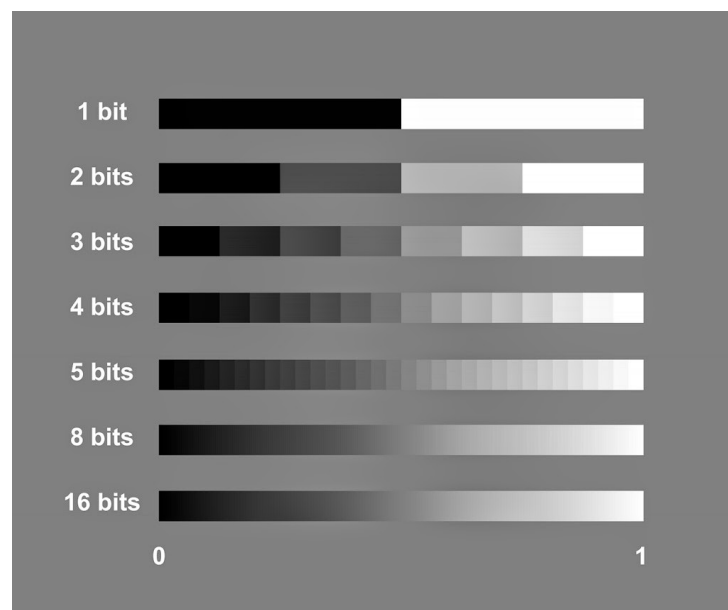


Figura 3 – Diferença das tonalidades de cores possíveis de serem representadas dependendo da quantidade de bits necessários para armazená-las

formar todas as cores visíveis pelo olho humano.

Para uma profundidade de cores de 8 bits, conseguimos armazenar 256 cores (2^8). Utilizando o sistema RGB, que possui 3 camadas, temos 8 bits para cada camada, ou seja, 24 bits. Com isso, obtemos um total de 16.777.216 (2^{24}) cores diferentes, como mostra o cubo da Figura 4 [10].



Figura 4 – Cubo de cores RGB de 24 bits

2.3 Imagens Térmicas

Todo objeto com temperatura acima do zero absoluto emite Radiação Infravermelha (IR, do inglês *Infrared Radiation*), sendo que quanto maior sua temperatura, maior a emissão. É classificado como radiação infravermelha toda onda que possui o comprimento maior que o da luz visível, ou seja, se encontra acima do vermelho, e menor que o das micro-ondas. Na faixa de comprimento aproximada entre $0,7\mu\text{m}$ e $1000\mu\text{m}$ [11]. Isso é ilustrado na figura 5.

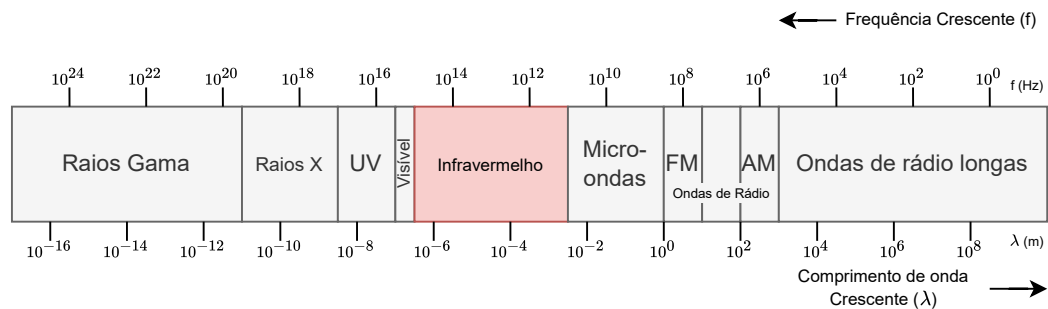


Figura 5 – Divisões do espectro eletromagnético

A própria região do infravermelho pode ser dividida de diferentes formas, sendo a mais comum mostrada na tabela 1. A radiação infravermelha emitida pelas amostras utilizadas no desenvolvimento desse trabalho estão na região conhecida como Infravermelho Térmico (TIR, do inglês *Thermal Infrared*). Essa região contempla o Infravermelho de Ondas Médias e Longas, já que objetos que emitem radiação nessa faixa do espectro possuem temperatura entre 190—1000K (aproximadamente -83 — 726°C) [11]. As imagens que ilustram a radiação do tipo TIR são capturadas por meio de Câmeras Térmicas.

Tabela 1 – Regiões do espectro infravermelho

| Região | Abreviação | Comprimento de Onda (μm) |
|-------------------------------|------------|---------------------------------------|
| Infravermelho Próximo | NIR | 0,7-1,4 |
| Infravermelho de Ondas Curtas | SWIR | 1,4-3 |
| Infravermelho de Ondas Médias | MWIR | 3-8 |
| Infravermelho de Ondas Longas | LWIR | 8-15 |
| Infravermelho Distante | FIR | 15-1000 |

Em muitos casos, as imagens brutas geradas por uma câmera térmica devem passar por um processamento que as refine para análise [12]. Dependendo da aplicação, podem ser aplicados métodos de processamento para diferentes finalidades, desde a eliminação de ruídos até a segmentação para destacar apenas pontos de interesse na imagem. Isso ocorre também em imagens comuns, capturadas dentro do espectro visível da luz. A diferença é que existem alguns fatores específicos que podem prejudicar a qualidade das imagens térmicas, desde a emissão de radiação da própria câmera usada na captura, até

a não uniformidade da superfície onde os dados foram coletados [13]. Esses fatores podem requisitar processamentos específicos para esse tipo de imagem.

2.3.1 Aplicações na Área Alimentícia

As imagens térmicas são amplamente utilizadas nas áreas de segurança, sistemas militares, sistemas de detecção e rastreamento de objetos, detecção de poluição e muitas outras [14]. Na área alimentícia seu uso também é extenso. Existem vários artigos que revisam suas aplicações já exploradas na Agricultura e Engenharia de Alimentos, além de outros que discutem possíveis métodos que ainda podem ser explorados.

[15] apresenta uso no monitoramento e validação da temperatura de alimentos, qualidade pós-colheita, qualidade de grãos e detecção de corpos estranhos, além de discutir as limitações e futuros desenvolvimentos das imagens térmicas na área. [16] aborda algumas dessas mesmas aplicações, discutindo também o uso com estimativas de rendimento em colheitas e detecção de patógenos e doenças em produtos agrícolas. Também traz uma tabela que revisa diversos parâmetros de qualidade em inúmeros alimentos utilizando diferentes técnicas com imagens térmicas. Já [17] explora os avanços recentes para qualidade de diferentes tipos de sementes e cita aplicações como estimativa da viabilidade, detecção de doenças e infecções, detecção de impurezas e danos, e classificação e identificação de variedades. Por fim, [18] apresenta um método para classificação de bananas que foram amadurecidas naturalmente e bananas que foram amadurecidas com o uso de Carbeto de Cálcio, elemento extremamente nocivo para o ser humano.

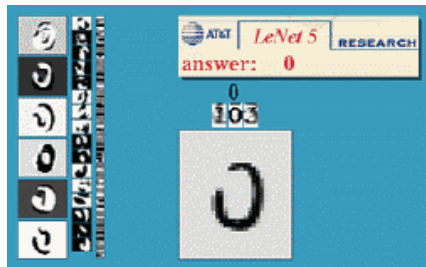
2.4 Visão Computacional

Visão Computacional é um campo da inteligência artificial que busca desenvolver técnicas para representar formas tridimensionais utilizando imagens. Ela busca descrever o mundo que nós vemos em uma ou mais imagens, reconstruindo sua forma, iluminação e distribuição de cor. É basicamente fazer com que computadores consigam enxergar da mesma maneira que nós, seres humanos [19].

No começo da visão computacional, em 1970, ela era vista como um componente de visão e percepção com o propósito de imitar a inteligência humana e dotar os robôs com comportamento inteligente. Com o passar do tempo, foi sendo descoberto que fazer com que um computador recebesse um *input* visual não era uma tarefa tão fácil [19].

Conforme a área foi se desenvolvendo, a atenção foi voltada para técnicas matemáticas mais sofisticadas para realizar análises quantitativas de imagens e cenas. Tópicos como fluxo ótico, *image pyramids*, filtro de Kalman, visão baseada em física, entre muitos outros começaram a ser explorados e aprimorados [19].

O maior avanço na área foi na década de 2010, devido a revolução no desenvolvimento dos algoritmos de reconhecimento de imagens, que aconteceu graças a tendência de usar grandes conjuntos de dados rotulados (supervisionados). Isso foi permitido pelo desenvolvimento de conjuntos de dados de alta qualidade. Esses conjuntos de dados proveram métricas confiáveis para rastrear o progresso de algoritmos de reconhecimento e segmentação semântica, mas mais importante, proveram dados suficientes para desenvolver soluções completas baseadas em aprendizado de máquina. Outra tendência que ajudou todo esse processo foi o aumento do poder computacional disponível para o desenvolvimento de propósito geral [19].



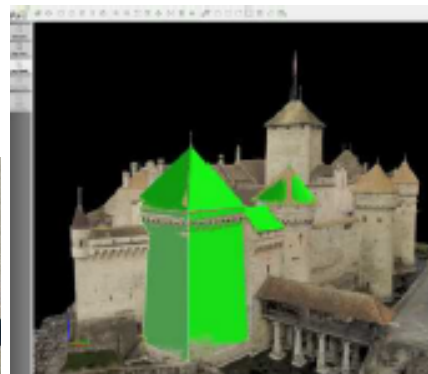
(a) Reconhecimento Óptico de Caracteres



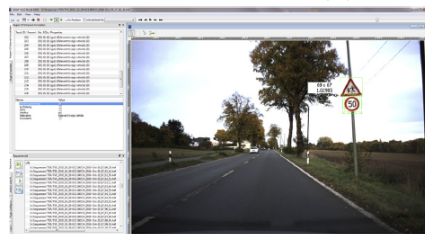
(b) Imagiologia Médica



(c) Carro Autônomo



(d) Fotogrametria



(e) Reconhecimento de Tráfego

Figura 6 – Algumas das aplicações da visão computacional

Atualmente, a visão computacional é utilizada em uma vasta variedade de aplicações do mundo real [19]. Algumas que se destacam são o reconhecimento de textos e números escritos a mão (Figura 6a), análise de imagens médicas para ajudar em diagnósticos de doenças (Figura 6b), veículos autônomos (Figura 6c), construção automática de

modelos 3D (Figura 6d) e também na área de monitoração e segurança, utilizado para detectar intrusos e analisar o tráfego de avenidas, reconhecendo carros, pessoas e placas de sinalização (Figura 6e) [20].

2.5 Aprendizado de Máquina

Aprendizado de Máquina (do inglês *Machine Learning*) são métodos computacionais que usam informação previamente coletada para aumentar sua performance ou para realizar previsões [21]. Ou seja, é a ciência de programar computadores para que eles possam aprender com os dados [22]. Esses dados são informações coletadas via interações com o ambiente, e sua qualidade é de extrema importância para o sucesso das previsões dos algoritmos. Dessa forma, essas técnicas de aprendizado combinam conceitos fundamentais da ciência da computação, estatística, probabilidade e otimização [21].

Algoritmos de aprendizagem de máquina podem ser divididos entre Supervisionados e Não Supervisionados, entre algumas outras classificações. Os algoritmos supervisionados são aqueles que recebem um conjunto de dados para treinamento com as respostas corretas, e baseado nesse conjunto fornecido, o algoritmo generaliza para responder corretamente para todos outras entradas possíveis [1].

Esse tipo de algoritmo pode ser utilizado para resolver problemas de Classificação e de Regressão. Para a classificação, cada uma das entradas recebe uma categoria (ou classe), e o algoritmo deve generalizar para dizer a qual dessas categorias outras possíveis entradas pertencem. Por exemplo, um classificador de imagens deve atribuir aos itens categorias como paisagem, porta retrato, animal [21]. Já a regressão não tenta classificar, mas sim prever um valor real para cada item [21]. Por exemplo, calcular o preço de um carro com base nos dados que foram coletados de outros carros [22].

Para avaliar um modelo de aprendizado de máquina supervisionado, é necessário separar um conjunto de dados que já tem suas classes conhecidas, mas que não foram fornecidos para o algoritmo treinar. Esse é o chamado conjunto de testes. Fornecemos como entrada o conjunto de testes, e comparamos os valores preditos pelo modelo com os valores reais, e validamos conforme o número de previsões realizadas corretamente [1].

Como dito anteriormente, o modelo precisa conseguir generalizar bem, já que ele vai ser treinado e validado com dados conhecidos, mas vai ser realmente utilizado de forma ampla, predizendo valores para entradas desconhecidas. Para isso, precisamos garantir que o algoritmo treine apenas o suficiente, pois um excesso de treinamento pode ser tão prejudicial quanto a falta. Se o modelo treinar excessivamente, ele vai memorizar os dados do conjunto de treino, se superajustando a eles (o que chamamos de *overfitting*), aprendendo inclusive sobre os ruídos e imprecisões dos dados, sendo incapaz de generalizar bem para dados desconhecidos. A Figura 7 ilustra essa diferença de um modelo que se

ajustou bem, para um que superajustou.

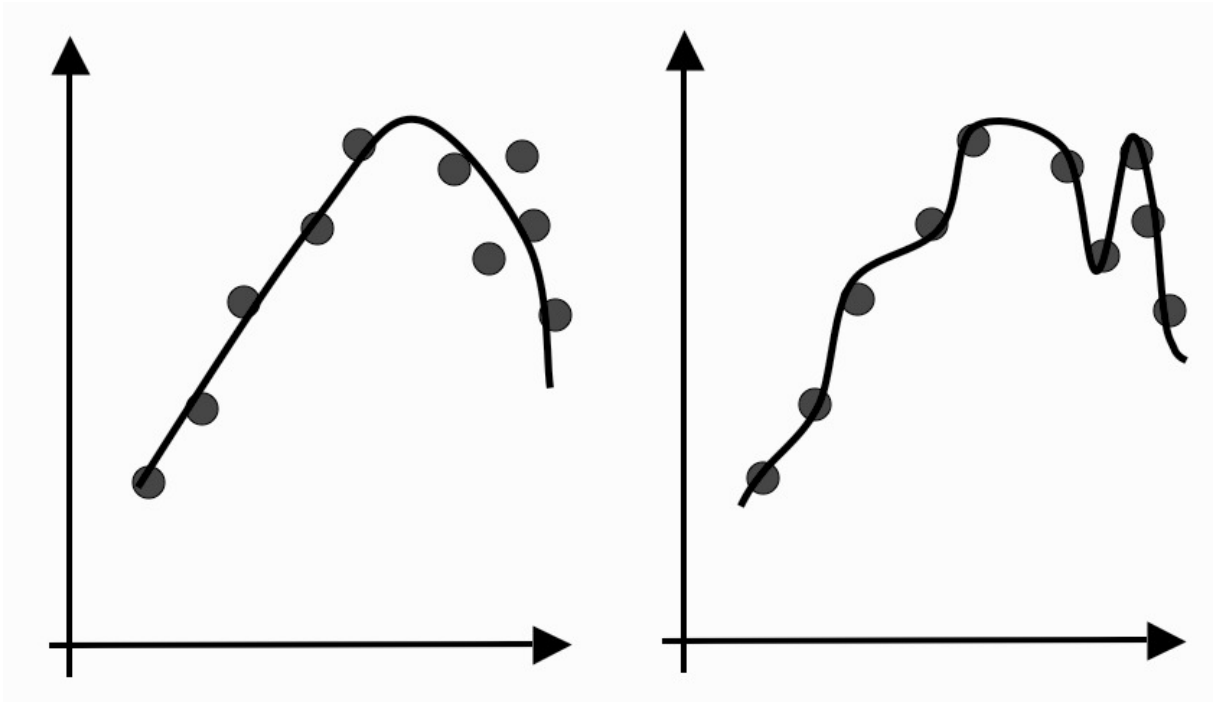


Figura 7 – Diferença da função de um modelo que treinou o necessário para generalizar bem (esquerda) e um modelo que treinou excessivamente, sofrendo um sobreajuste (direita) [1]

Como o problema abordado por esse trabalho é um problema de classificação, utilizaremos um algoritmo supervisionado, o Random Forest, que utiliza Árvores de Decisão como base.

2.5.1 Árvores de Decisão

Uma Árvore de Decisão é um classificador que faz uma série de perguntas sobre as *features* e atributos recebidos como entrada. Cada pergunta é armazenada em um nó, e cada nó interno aponta para um nó filho para cada uma das respostas possíveis para a questão [23], ou seja, uma árvore, sendo que cada nó folha dessa estrutura contém o resultado da classificação.

A Figura 8 ilustra um exemplo didático de árvore de decisão que ajuda um estudante a decidir o que fazer durante a noite.

As árvores de decisão são mais interpretáveis que outros classificadores, já que combinam essas simples perguntas sobre as informações de maneira compreensível [23]. Essa simplicidade faz com que elas possam, inclusive, ser facilmente lidas como um conjunto de disjunções lógicas do tipo *if ... else*, como mostra o seguinte exemplo que usa como base a árvore de decisão da Figura 8:

- a) se houver uma festa, então vá para ela

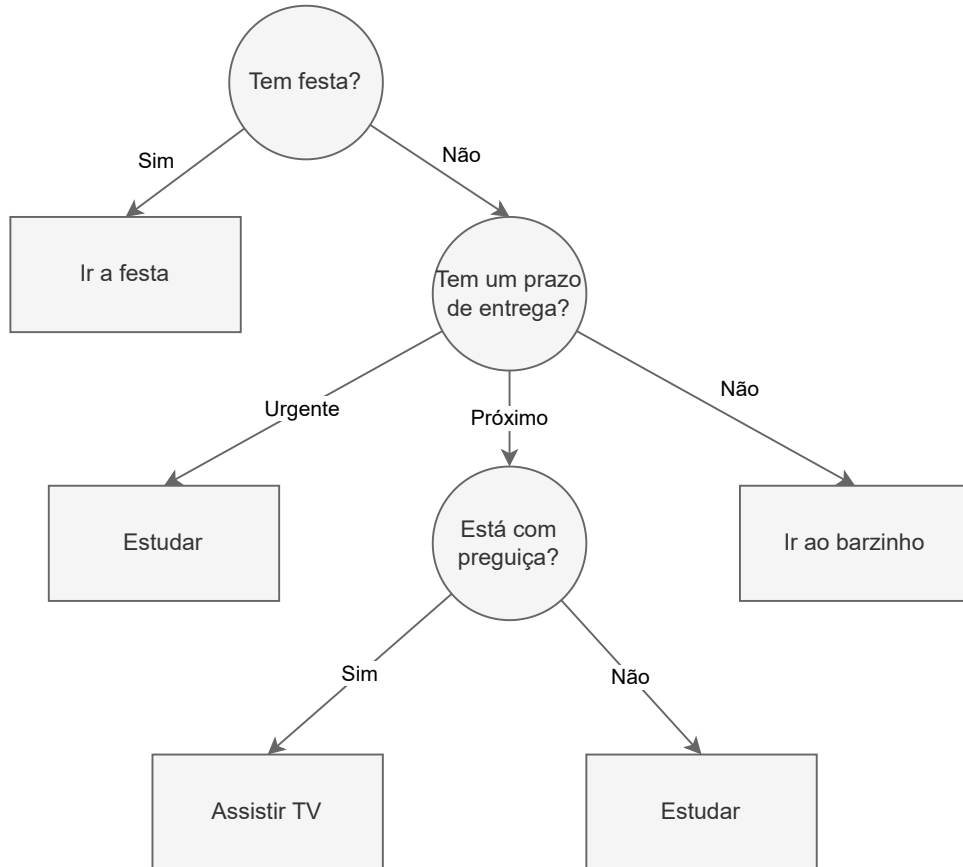


Figura 8 – Exemplo simples de uma Árvore de Decisão que decide como um estudante passará a sua noite [1]

- b) se não houver uma festa e você tem um prazo de entrega urgente, então estude
- c) se não houver uma festa e você não tem um prazo de entrega urgente e você está com preguiça, então assista TV
- d) etc.

Uma árvore de decisão é construída deixando os atributos mais importantes nos níveis inferiores. Portanto, para construir uma árvore de decisão, é necessário definir um critério para medir a importância de cada atributo. Normalmente esse critério é baseado no cálculo da impureza da base de dados.

A impureza é a medida de incerteza de um determinado conjunto de dados. Quanto maior a probabilidade dos itens de um conjunto de dados X pertencerem a uma determinada classe, mais puro (ou menos impuro, menos incerto) é esse conjunto.

Vamos supor que queremos classificar itens em m classes usando um conjunto de treinamento E . Sendo $p_i (i = 1, \dots, m)$ a probabilidade dos itens de E pertencerem a classe i . A impureza é a menor possível quando um único p_i é igual a 1 e todos os outros são iguais a 0, e é a maior possível quando todos os p_i são iguais [23].

Como o objetivo de um classificador é ter a maior certeza possível de qual classe

uma determinada entrada pertence, os atributos que dividirem o conjunto de dados em um subconjunto com a menor impureza, são escolhidos como mais importantes.

Existem vários métodos para realizar o cálculo de impureza, sendo os mais comuns o que utiliza a Entropia, e o que utiliza o Coeficiente de Gini.

2.5.1.1 Entropia e o Ganho de Informação

Ganho de Informação é um critério baseado em impureza que utiliza a entropia como medida de impureza [2]. Quanto menor a entropia, mais puro é um conjunto de dados, e quanto maior a entropia, mais impuro é um conjunto de dados. A Equação 2.1 representa o cálculo da entropia de um conjunto de dados S , onde m é a quantidade de classes e p_i a probabilidade de cada classe.

$$entropia(S) = - \sum_{i=1}^m p_i \log_2 p_i \quad (2.1)$$

A Equação 2.2 descreve o ganho de informação do conjunto S no atributo y . Quanto maior o ganho de informação do atributo y , menor a impureza do subconjunto resultado da divisão do conjunto S original no atributo y .

$$ganho(S, y) = entropia(S) - \sum_{v \in dom(y)} \frac{|S_v|}{|S|} entropia(S_v) \quad (2.2)$$

2.5.1.2 Coeficiente de Gini

O Coeficiente de Gini, definido na Equação 2.3, é um outro critério baseado em impureza que mede as divergências entre as distribuições de probabilidade de um atributo alvo [2].

$$gini(S) = 1 - \sum_{i=1}^m p_i^2 \quad (2.3)$$

O ganho de informação utilizando o Coeficiente de Gini é definido na Equação 2.4

$$GanhoGini(S, y) = gini(S) - \sum_{v \in dom(y)} \frac{|S_v|}{|S|} gini(S_v) \quad (2.4)$$

2.5.1.3 Critérios de Parada

Esse processo de divisão da base de dados de acordo com o ganho de informação de um atributo continua até que um critério de parada seja atingido [2]. Alguns critérios comuns são:

- a) Todas as instancias do conjunto de treinamento pertencem a uma mesma classe.

- b) A árvore chegou num limite máximo de profundidade.
- c) O resultado do melhor critério de divisão não é maior que um certo limite.

2.5.1.4 Métodos de Poda

Adicionar critérios de paradas muito rígidos tendem a criar uma árvore de decisão pequena e com um problema de *underfitting*, quando o classificador está mal ajustado e não consegue encontrar relações entre as entradas durante o treinamento. Ao mesmo tempo, utilizar critérios de parada muito fracos tendem a criar árvores de decisão grandes e com problema de *overfitting*, quando o classificador se ajustou tanto na base de treinamento que não consegue generalizar para outras entradas.

De acordo com as metodologias de poda, durante o treinamento, um critério de parada fraco é utilizado, deixando a árvore de decisão se sobreajustar na base de treinamento. Então, essa árvore é cortada em árvores menores, removendo galhos que não contribuem para precisão da generalização [2]. Alguns métodos de poda são *Cost-Complexity Pruning*, *Reduced Error Pruning*, *Minimum Error Pruning*, *Pessimistic Pruning*, entre outros.

2.5.1.5 Vantagens e Desvantagens

Algumas das vantagens das Árvores de Decisão são:

- a) Árvores de Decisão são autoexplicativas e quando pequenas também são fáceis de entender. Portanto, podem ser entendidas por usuários não profissionais.
- b) Árvores de Decisão podem lidar com atributos nominais e numéricos.
- c) A representação de uma Árvore de Decisão é rica o suficiente para representar qualquer classificador de valor discreto.
- d) Árvores de Decisão são capazes de lidar com bases de dados que possuem erros.
- e) Árvores de Decisão são capazes de lidar com bases de dados que possuem valores faltando.

Por outro lado, a maior desvantagem que pode ser citada das árvores de decisão é que elas tendem a performar bem quando existem alguns atributos altamente relevantes, mas performam pior caso existam muitas interações complexas. Como a maioria das árvores de decisão divide o conjunto de dados em regiões mutuamente exclusivas, em alguns casos a árvore pode conter várias duplicações da mesma subárvore para representar o classificador [2]. Esse problema é ilustrado na Figura 9.

2.5.2 *Random Forests*

Utilizar um conjunto de árvores de decisão para prever uma determinada classe, utilizando entradas aleatórias do conjunto de dados para crescer cada uma das árvores,

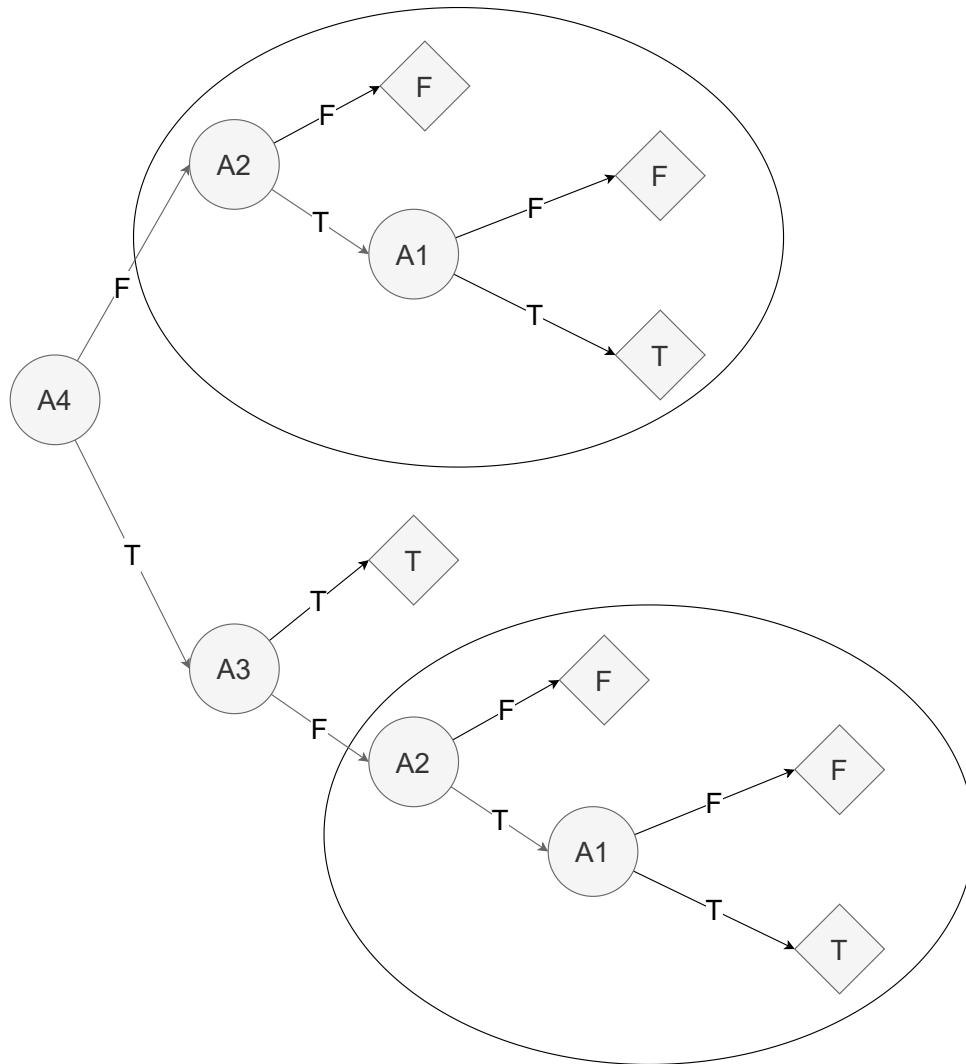


Figura 9 – Ilustração de uma Árvore de Decisão com Replicação [2]

aumentou significativamente a acurácia dos classificadores baseados em árvores de decisão [24].

Dito isso, uma *Random Forest* (do inglês, Floresta Aleatória) é um classificador baseado em um conjunto de árvores, onde cada árvore é criada a partir de entradas aleatórias de um conjunto de dados. Cada subconjunto gerado aleatoriamente do conjunto de dados original é chamado de *Bootstrap Sample*. A randomização envolvida em cada *bootstrap sample* acontece por conta da seleção aleatória de algumas entradas do conjunto de dados original, mas também pela seleção aleatória de alguns dos seus atributos preditores [25]. No caso da classificação, o resultado final é a classe mais votada entre todas as árvores.

Utilizando as *Random Forests* é possível utilizar *Out-of-Bag Data* (dados fora do saco, em tradução livre do inglês), para estimar o erro de generalização e a importância dos atributos (em inglês, *feature importance*). *Out-of-Bag Data* são todos os dados que não estão contidos no *Bootstrap Sample* de uma árvore. Ou seja, quando estamos calculando

a acurácia ou o erro de generalização de um dado *Out-of-Bag*, estamos utilizando para o cálculo todas as árvores do modelo que não utilizaram esse dado para sua construção [24] [25]. Isso é útil quando o conjunto de dados não é grande o suficiente, sendo possível então utiliza-lo inteiramente como conjunto de treinamento, e valida-lo utilizando a pontuação dos registros *Out-of-Bag*.

Visto que, durante sua construção, as árvores de decisão automaticamente calculam a importância de cada *feature* com o cálculo do ganho de informação, alocando nas profundidades mais baixas as *features* que possuem mais importância, fica muito fácil reduzir a dimensionalidade de um problema utilizando *Random Forests*. Enquanto em algumas aplicações é padrão rodar uma Análise de Componentes Principais (em inglês, *Principal Components Analysis*, ou PCA), é possível que os componentes principais não capturem a informação importante para o problema de predição. Nesse caso, é preferível obter a *feature importance* diretamente do algoritmo e executar um retreinamento utilizando somente os preditores mais importantes [25].

2.5.3 Validação Cruzada

Validação Cruzada (em inglês, *Cross-validation*) é um método de avaliar a performance de generalização de um modelo, que é mais estável e completo do que usar uma divisão do conjunto de dados entre treinamento e teste [26]. Na validação cruzada, os dados são divididos repetidamente em *folds* e múltiplos modelos são treinados, cada um deles com um *fold* diferente.

Como exemplo, vamos supor que vamos dividir um conjunto de dados em 5 *folds* para realizar a validação cruzada. Em um primeiro momento, o *fold* 1 é separado como conjunto de teste, enquanto todos os outros (2, 3, 4 e 5) são separados como conjunto de treinamento. Na segunda iteração, o *fold* 2 é separado como conjunto de teste, enquanto o restante (1, 3, 4 e 5) como conjunto de treinamento. Esse processo se repete até que todos os dados tenham sido usados como conjunto de teste pelo menos uma vez. O resultado final da avaliação é a média do resultado obtido em cada uma das iterações. A Figura 10 ilustra esse exemplo de validação cruzada com 5 *folds*.

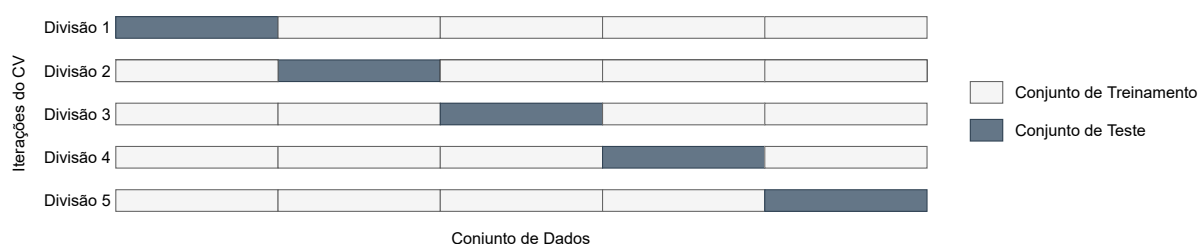


Figura 10 – Exemplo da aplicação da Validação Cruzada com 5 *folds*

Existem diversos benefícios em usar a validação cruzada. Ao utilizar uma divisão entre conjunto de teste e treino, imagine que temos sorte na divisão, e todas as entradas

que são consideradas difíceis de classificar entram no conjunto de treino. Nesse caso, o conjunto de teste irá conter as entradas fáceis, e a acurácia será erroneamente alta. Caso contrário, se tivermos azar na divisão, o conjunto de teste terá as entradas consideradas difíceis de classificar, fazendo com que a acurácia seja erroneamente baixa. Usando a validação cruzada, esse problema não acontece, já que toda entrada estará no conjunto de teste pelo menos uma vez, fazendo com que o modelo tenha que generalizar bem para todas as entradas, para a média da acurácia se manter alta.

Ter múltiplas divisões também provê informação sobre quão sensível o modelo é ao conjunto de treino. Imagine que um conjunto de dados obteve acurácias entre 90% e 100%, esses números fornecem uma ideia sobre como o modelo performa nos piores e melhores cenários.

Outro benefício da validação cruzada é que ele utiliza os dados de forma mais eficaz. Utilizando uma divisão com 5 *folds*, por exemplo, cada iteração utiliza 80% da base para o treinamento. Utilizando uma divisão com 10 *folds*, cada iteração utiliza 90% da base para o treinamento. Mais dados significa modelos mais precisos.

A maior desvantagem da validação cruzada é o aumento do custo computacional, já que agora treinamos k modelos ao invés de somente 1.

3 DESENVOLVIMENTO

3.1 Base de Dados

As imagens térmicas capturadas das amostras de macarrão foram previamente coletadas e fornecidas em parceria por professores da Universidade Estadual de Campinas. Cada amostra de macarrão foi esquentada e teve suas imagens gravadas como um vídeo durante todo o processo de resfriamento natural. Ou seja, para cada amostra de macarrão, possuímos diversas imagens, que são os *frames* do vídeo gravado enquanto o macarrão esfriava.

Quanto as classes, as amostras de macarrão são divididas em duas concentrações diferentes de fibra, com 3,5% e 7% de concentração. No total, são 505636 imagens, de 2001 amostras diferentes, sendo 900 amostras de macarrão com 3,5% de concentração e 1101 amostras de macarrão com 7% de concentração.

As Figuras 11a e 11b mostram 5 *frames* que ilustram o processo de resfriamento do macarrão durante o tempo que foi capturado, com 3,5% e 7% de concentração de fibra, respectivamente.

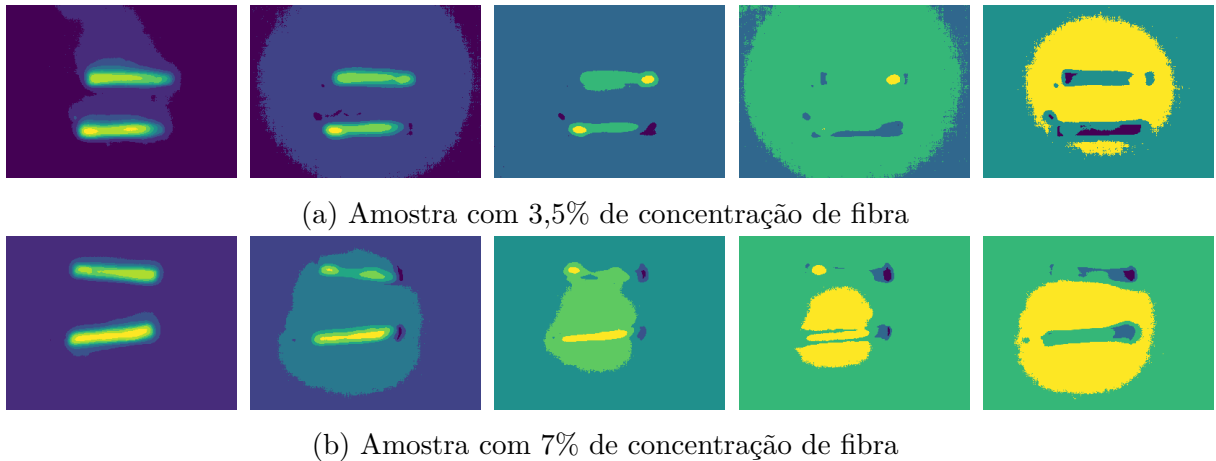
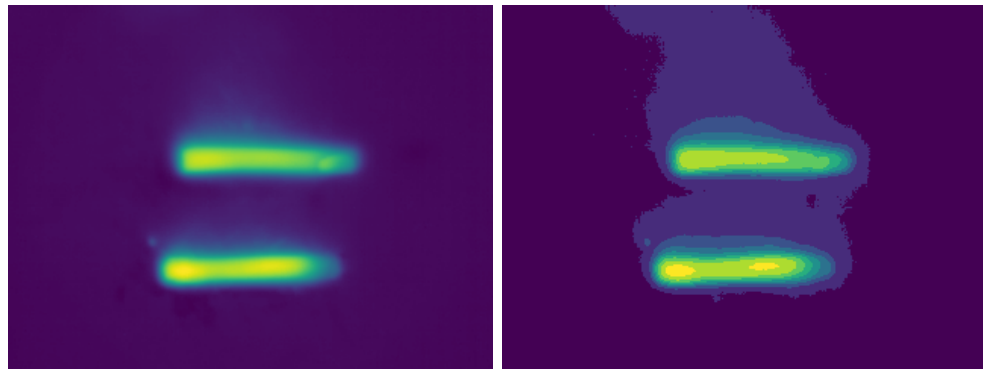


Figura 11 – Alguns *frames* das amostras resfriando conforme o tempo

3.2 Pré-Processamento

O único processamento que foi aplicado nas imagens foi uma redução na profundidade das cores. Todos os *pixels* de todas as imagens foram normalizados, para possuírem valores apenas de 0 até 100. Apesar de deixar as imagens com menos cores e, portanto, menos suaves, esse processo deixa a extração das *features* mais simples, além de deixa-las mais leves, já que agora cada imagem possui apenas 100 cores. A Figura 12 ilustra a diferença da imagem original, com a imagem que possui profundidade de cor reduzida.



(a) Amostra Original

(b) Amostra com a profundidade de cor reduzida

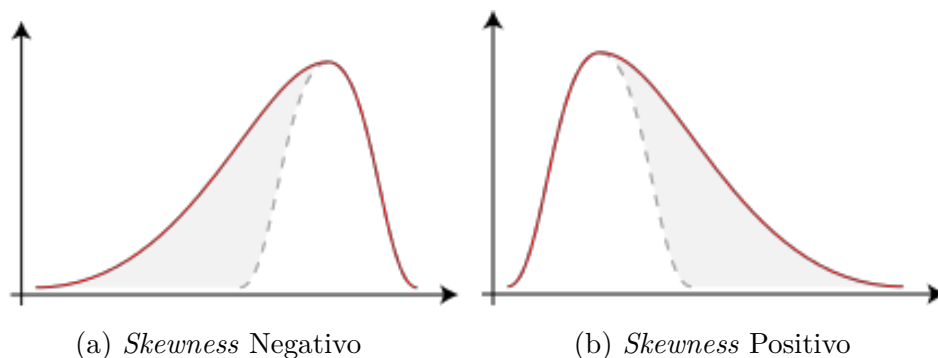
Figura 12 – Diferença da imagem antes e após reduzir a profundidade de cor

3.3 *Feature Selection*

Para cada uma das imagens, foram extraídas 3 *features* espaciais e 3 *features* de histograma, além do próprio histograma da imagem. Como *features* espaciais, foram extraídas a média, o desvio padrão e a entropia das cores da imagem. A entropia mede a incerteza dos dados [27]. Quanto mais distante de 0, mais incertos são os dados, ou seja, maior a quantidade de dados diferentes entre si.

Já como *features* de histograma, foram extraídos o desvio padrão do histograma, o Kurtosis e o *Skewness*, além do próprio histograma. O *Kurtosis* e o *Skewness* são medidas que podem descrever a forma que um conjunto de dados se encontra [28].

O *Skewness* mede a assimetria de uma distribuição de probabilidade. Se for negativo, a média dos dados se encontra à esquerda do centro da distribuição, ou seja, a média é menor que a moda. Se for positivo, a média dos dados se encontra à direita do centro da distribuição, ou seja, a média é maior que a moda. A Figura 13 ilustra esse conceito.

(a) *Skewness* Negativo(b) *Skewness* PositivoFigura 13 – Diferença de uma distribuição de probabilidades com *Skewness* positivo e negativo

Já o *Kurtosis* mede o quão agrupados em um mesmo ponto estão os dados de uma distribuição de probabilidade. Quanto mais próximo de 0, mais próximo de uma distribuição normal. Quanto mais positivo o valor, mais pontudo e agrupado é o gráfico

dessa distribuição. Quando mais negativo o valor, mais achatado e menos agrupado é o gráfico dessa distribuição. A Figura 14 ilustra esse conceito.

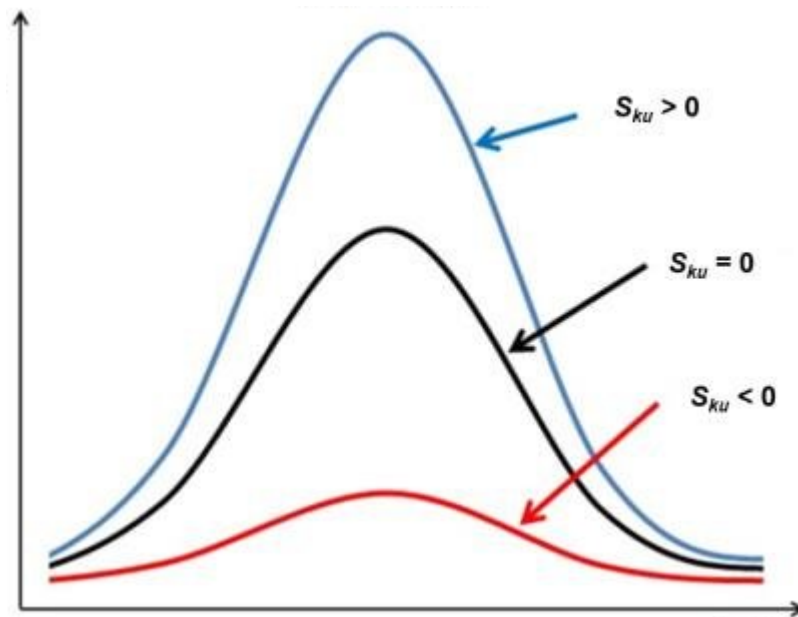


Figura 14 – Diferença de uma distribuição de probabilidades com *Kurtosis* (S_{ku}) maior, menor e igual a 0

Após a extração das *features* de todas as imagens, foi preciso agrupar cada uma delas, de cada *frame* referente a cada amostra de macarrão, para tentar capturar o resfriamento natural da amostra com o passar do tempo. Quanto maior o *frame*, mais tempo a amostra foi resfriada naquela imagem.

Para realizar o agrupamento, foram utilizadas 4 estratégias diferentes, a fim de descobrir qual seria a melhor escolha para a classificação do tipo de fibra dos macarrões. As estratégias utilizadas foram o agrupamento pela média, pela variância, pelo desvio padrão e pela soma das *features*.

Como exemplo fictício, supomos que uma amostra X de macarrão possui diversos *frames* F_i ($i = 1, \dots, t$). Após a extração das *features* de todos os t *frames* da amostra X , essas *features* vão ser agrupadas (de acordo com a estratégia escolhida), para assim, ser montado o conjunto de dados final com todos os agrupamentos dos *frames* de todas as amostras.

3.4 Seleção dos Parâmetros

Existem alguns parâmetros que podem ser ajustados no algoritmo de *Random Forest* fornecido pela biblioteca *Scikit-Learn* a fim de melhorar o resultado final da classificação, entre eles temos: o número de árvores de decisão que serão construídas na floresta. O critério de construção e divisão dos nós de cada árvore. O número mínimo de amostras

requeridas para dividir um nó interno. E o número mínimo de amostras que podem estar em cada nó folha.

Para cada problema, parâmetros diferentes são necessários. Para ajudar na escolha dos parâmetros ideais para o nosso problema, utilizamos a classe `GridSearchCV` disponibilizada pelo *Scikit-Learn*. Essa biblioteca treina o classificador com os parâmetros fornecidos, e valida o modelo, selecionando quais parâmetros obtiveram a melhor acurácia. A Tabela 2 mostra quais foram os parâmetros fornecidos para o algoritmo selecionar.

Tabela 2 – Parâmetros fornecidos para o algoritmo de seleção de melhores parâmetros

| Parâmetros | Valores Testados |
|--|---------------------------------------|
| Número de Árvores de Decisão | 10, 50, 100, 150 |
| Critério de Construção | Impureza de Gini, Entropia de Shannon |
| Número mínimo de amostras para divisão | 2, 5, 10 |
| Número mínimo de amostras nas folhas | 1, 5, 10 |

Os valores para esses parâmetros foram escolhidos observando e testando o funcionamento das *Random Forests*.

As Tabelas 3, 4, 5 e 6 mostram quais foram os melhores parâmetros selecionados para cada estratégia de agrupamento. Os treinamentos dos modelos foram executados utilizando esses valores de parâmetros.

Tabela 3 – Melhores parâmetros selecionados para o agrupamento com a média

| Parâmetros | Melhores Valores |
|--|-------------------------|
| Número de Árvores de Decisão | 10 |
| Critério de Construção | Entropia de Shannon |
| Número mínimo de amostras para divisão | 2 |
| Número mínimo de amostras nas folhas | 10 |

Tabela 4 – Melhores parâmetros selecionados para o agrupamento com a variância

| Parâmetros | Melhores Valores |
|--|-------------------------|
| Número de Árvores de Decisão | 10 |
| Critério de Construção | Impureza de Gini |
| Número mínimo de amostras para divisão | 2 |
| Número mínimo de amostras nas folhas | 10 |

Tabela 5 – Melhores parâmetros selecionados para o agrupamento com o desvio padrão

| Parâmetros | Melhores Valores |
|--|-------------------------|
| Número de Árvores de Decisão | 10 |
| Critério de Construção | Impureza de Gini |
| Número mínimo de amostras para divisão | 2 |
| Número mínimo de amostras nas folhas | 10 |

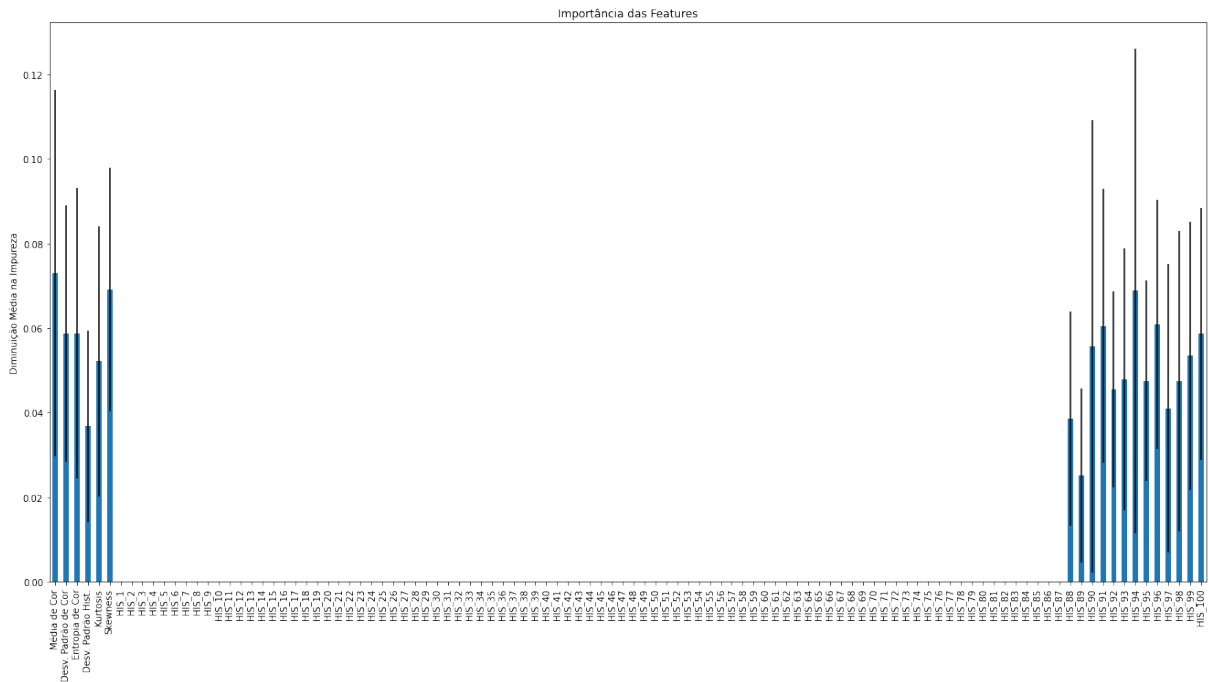
Tabela 6 – Melhores parâmetros selecionados para o agrupamento com soma

| Parâmetros | Melhores Valores |
|--|------------------|
| Número de Árvores de Decisão | 50 |
| Critério de Construção | Impureza de Gini |
| Número mínimo de amostras para divisão | 2 |
| Número mínimo de amostras nas folhas | 10 |

3.5 Importância das *Features*

Com os melhores parâmetros escolhidos, antes de partir para o treinamento da *Random Forest*, foi feita uma análise da importância das *features*, já que isso pode ser feito facilmente utilizando as árvores de decisão. Essa análise foi feita em cada uma das estratégias de agrupamento das *features* citadas anteriormente. As Figuras 15, 16, 17 e 18 mostram o gráfico de cada uma das análises.

Quanto a nomenclatura das *features*, foi adotado o seguinte: Média de Cor, Desvio Padrão de Cor, Entropia de Cor, Desvio Padrão de Histograma, Kurtosis, Skewness, e os valores do histograma, de 1 até 100, antecipados pelo prefixo HIS_.

Figura 15 – Importância das *features* agrupando pela Média

De acordo com os gráficos, podemos ver que as 3 *features* espaciais e as 3 *features* de histograma possuem alguma importância. Já os dados do próprio histograma só possuem relevância em valores elevados de cor, em todas as estratégias de agrupamento. Para os agrupamentos pela média, variância e desvio padrão, os valores do histograma abaixo do 88 foram considerados sem importância. Já para o agrupamento pela soma, os valores abaixo de 87 foram considerados sem importância, apesar do valor 87 ter obtido importância

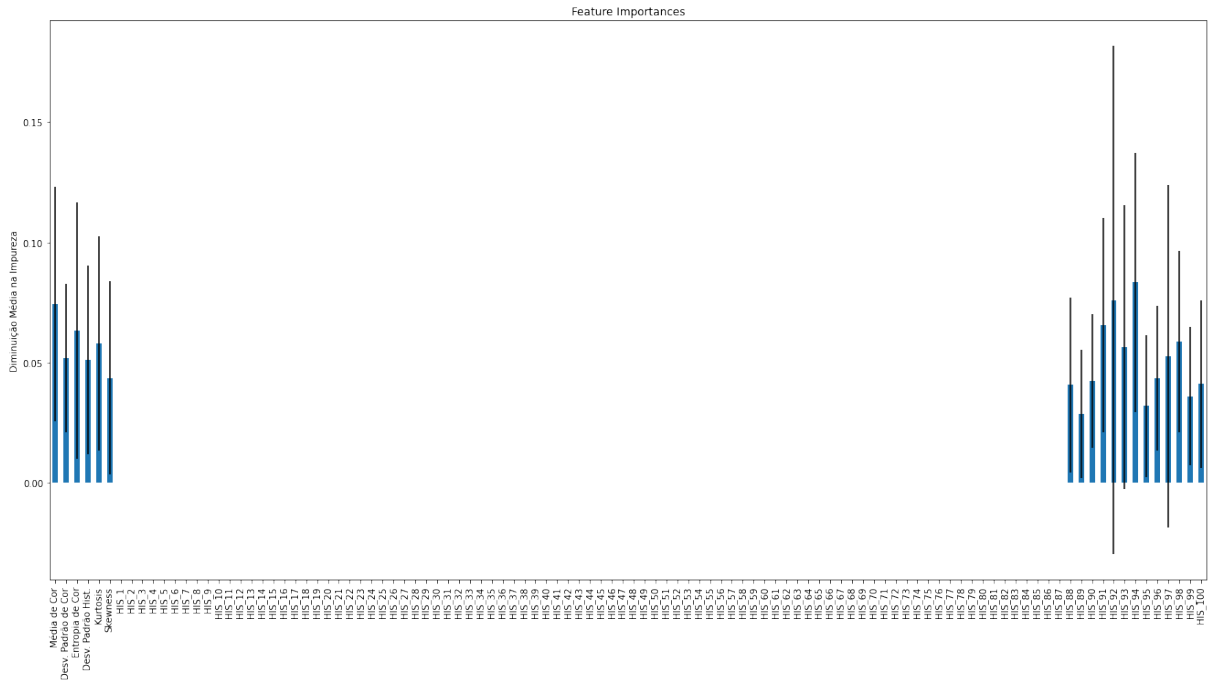


Figura 16 – Importância das *features* agrupando pela Variância

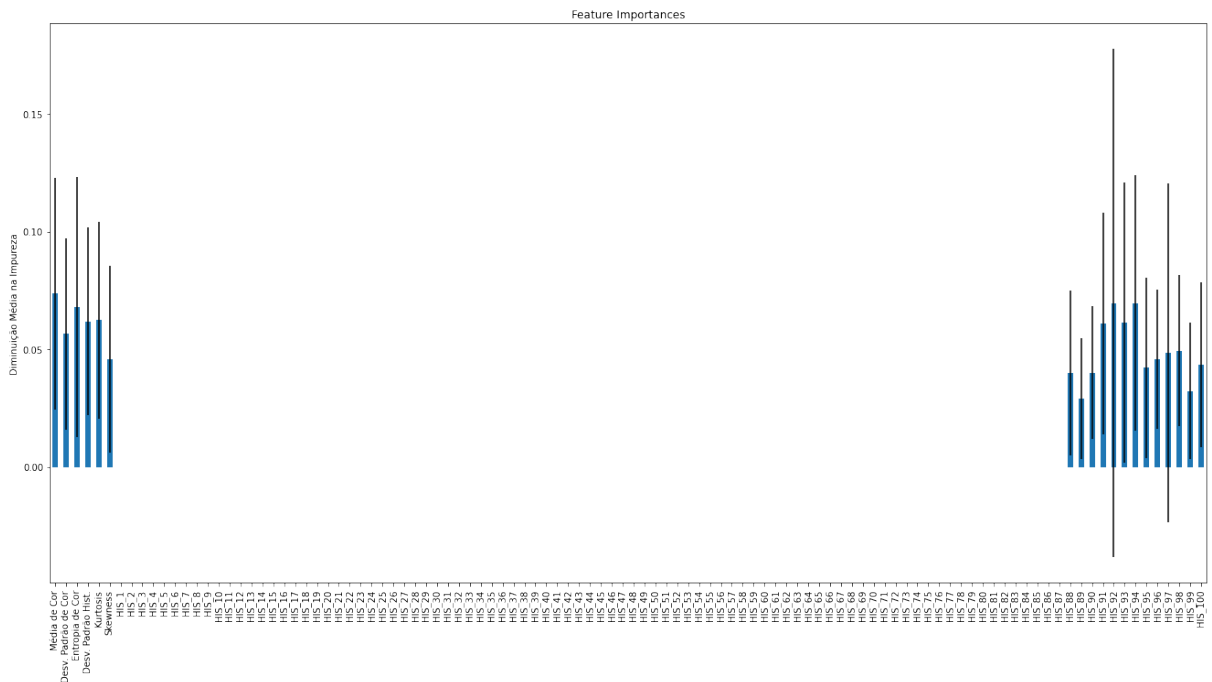


Figura 17 – Importância das *features* agrupando pelo Desvio Padrão

baixíssima.

Outra coisa que podemos notar são as barras de erro bastante elevadas, principalmente nos agrupamentos pela variância, desvio padrão e soma. Essas barras de erro são os desvios padrão da importância das *features* de cada uma das árvores utilizadas na *Random Forest* de cada agrupamento.

Foram removidas todas as *features* consideradas irrelevantes, restando apenas a

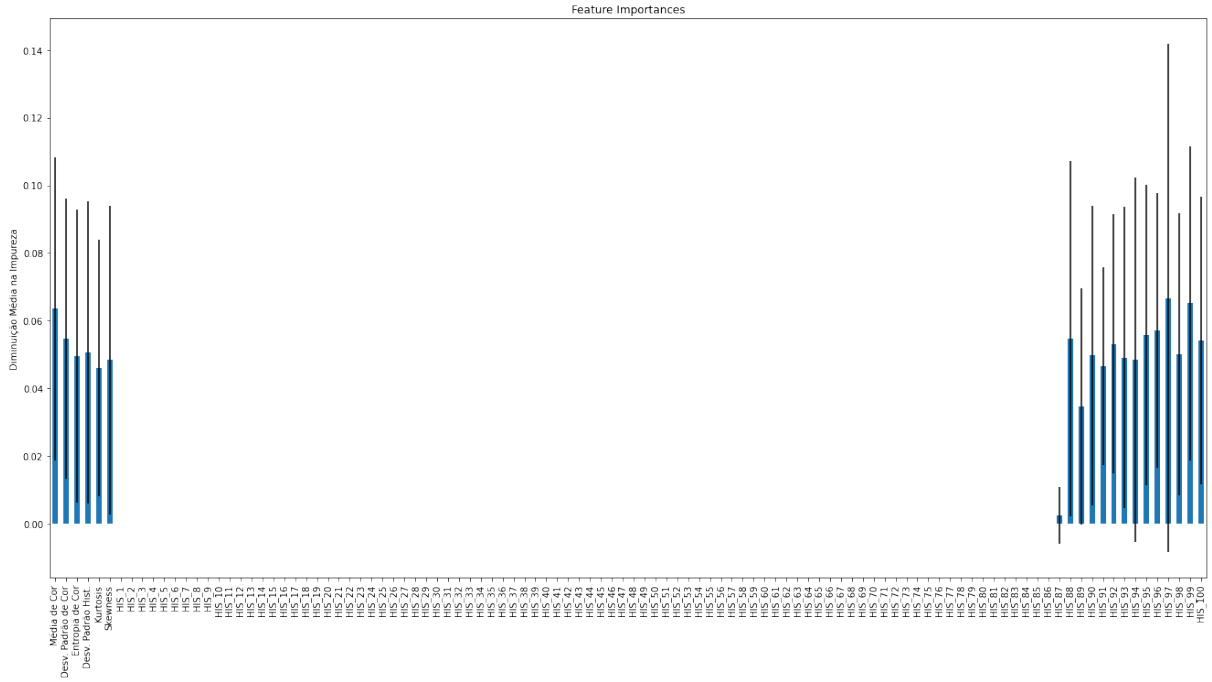


Figura 18 – Importância das *features* agrupando pela Soma

média, desvio padrão e entropia de cor, desvio padrão, *kurtosis* e *skewness* de histograma, e os valores do histograma, do 88 ao 100.

4 RESULTADOS

Para o treinamento dos classificadores de cada um dos agrupamentos, foi utilizada a metodologia de Validação Cruzada, com 10 *folds*. Além disso, a execução dessa validação foi repetida 30 vezes, ou seja, para cada agrupamento foram feitas 30 execuções de teste. A Figura 19 apresenta um diagrama de caixa sobre as acurácias obtidas em cada uma das execuções, para cada um dos agrupamentos.

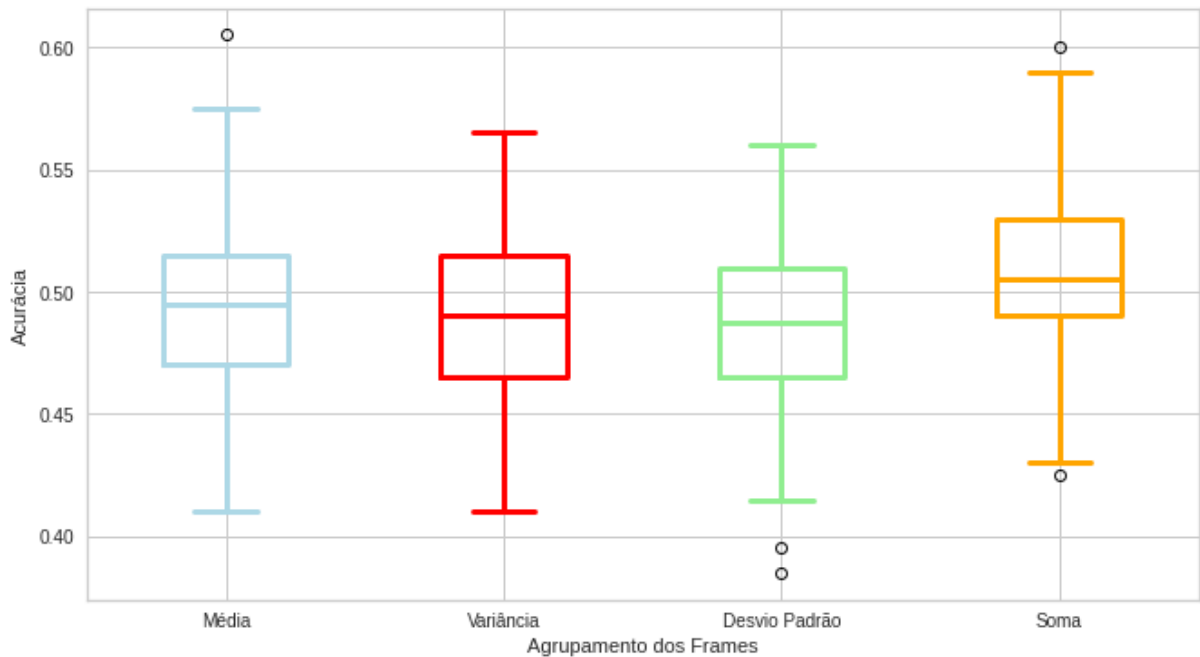


Figura 19 – Resultados obtidos nas execuções com agrupamento

Podemos observar que, no caso do agrupamento pela média e pela soma, chegamos a obter acurácias iguais ou até mesmo superiores à 60%, porém esses valores são considerados *outliers*, ou seja, são valores atípicos, e estão muito distantes do restante dos valores da distribuição. Também podemos notar que o agrupamento pela soma possui os maiores valores de limite inferior, primeiro, segundo e terceiro quartil e limite superior, além de ser o único em que a mediana ultrapassa 50% de acurácia, podendo assim ser considerado o melhor dos quatro agrupamentos.

Os outros três agrupamentos, pela média, variância e desvio padrão, tiveram resultados bastante próximos. O que mais se destaca é o agrupamento pelo desvio padrão, que possui dois *outliers* abaixo dos 40% de acurácia, além de possuir os menores valores de limite superior, primeiro, segundo e terceiro quartil, possivelmente obtendo o pior resultado entre os quatro agrupamentos.

Observando a Tabela 7, que mostra os valores mínimos, máximos e de média obtidos por cada agrupamento, podemos verificar o que já havíamos concluído observando

a Figura 19. O agrupamento dos *frames* que obteve o melhor resultado foi o agrupamento pela soma, com acurácia média de 50,91%. Apesar do agrupamento pela média ter obtido acurácia máxima de 60,50%, vimos que esse valor se trata de um *outlier*, além de que sua média ficou em 49,27%.

Podemos confirmar também que o pior resultado foi o do agrupamento pelo desvio padrão, com a menor acurácia entre todos, de apenas 48,76%, mas também menor valor mínimo e menor valor máximo, 38,50% e 56% respectivamente.

Tirando o agrupamento pela soma, os demais não atingiram nem 50% de acurácia média. Apesar disso, os resultados foram próximos, porém, nenhum dos agrupamentos resultaram em acurácias satisfatórias.

Tabela 7 – Resultados das acurácias por agrupamento

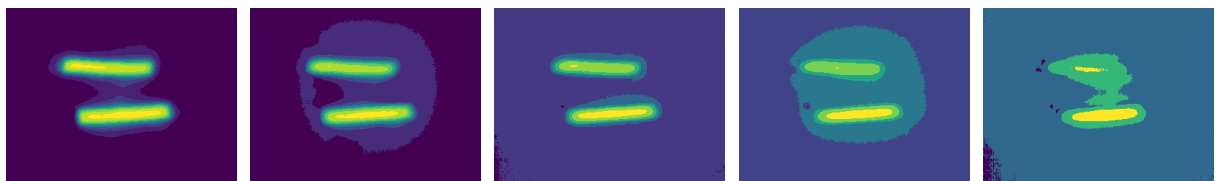
| Agrupado pelo(a) | Acurácia | | |
|------------------|----------|--------|--------|
| | Média | Mínimo | Máximo |
| Média | 49,27% | 41,00% | 60,50% |
| Variância | 48,91% | 41,00% | 56,50% |
| Desvio Padrão | 48,76% | 38,50% | 56,00% |
| Soma | 50,91% | 42,50% | 60,00% |

4.1 Discussão visual das amostras classificadas

A Figura 20 mostra alguns *frames* das amostras dos diferentes níveis de concentração de fibra que foram classificados corretamente e incorretamente, em uma das iterações do agrupamento pela soma.

É possível notar que a amostra com 3,5% de concentração de fibra demorou bastante para ter seu calor dissipado, ainda possuindo em seu último *frame* uma grande concentração da cor azul escura, mostrando a discrepância de temperatura entre o alimento e o ambiente. Esse pode ter sido um motivo que fez com que a amostra tenha sido classificada incorretamente, já que na amostra com mesma concentração que foi classificada corretamente, no último *frame*, o alimento já dissipou bem mais calor.

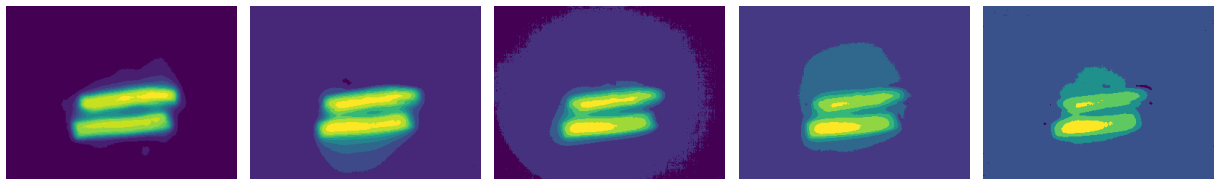
É difícil de se fazer essa análise visual para tentar extrair algum possível motivo que tenha feito alguma delas ser classificada da maneira que foi. Sendo assim, não foi possível extrair nenhuma outra informação relevante apenas olhando os *frames* da amostra com 7% de concentração de fibra.



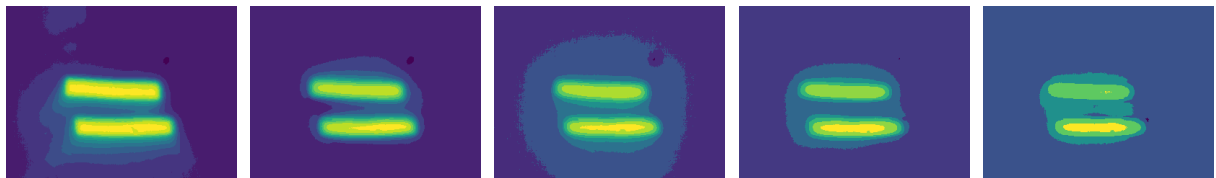
(a) 3,5% de fibra classificado corretamente



(b) 3,5% de fibra classificado incorretamente



(c) 7% de fibra classificado corretamente



(d) 7% de fibra classificado incorretamente

Figura 20 – Exemplos de classificações corretas e incorretas em uma das iterações do agrupamento pela soma, sendo um exemplo para cada concentração de fibra

5 CONCLUSÃO

Esse trabalho apresentou uma nova metodologia de análise da composição de alimentos, através de imagens térmicas capturadas dos alimentos aquecidos e fornecidas como entrada a um modelo de *Machine Learning* que utiliza um algoritmo de *Random Forest*.

Os resultado obtidos não foram satisfatórios, sendo 50,91% a melhor acurácia média obtida. Como estudos futuros a fim de melhorar esses resultados, podem ser propostas outras técnicas de agrupamento dos *frames* das amostras para capturar a ideia de resfriamento com o tempo. Também pode ser estudada uma maneira de combinar as *features* que foram utilizadas, ou até mesmo propor novas *features* que consigam utilizar a informação temporal das amostras.

Além disso podem ser utilizados outros métodos para o pré-processamento das imagens, de forma que elas fiquem mais adequadas para serem fornecidas como entrada ao algoritmo. Também pode ser analisada a aplicação do problema para outros algoritmos de *machine learning*, que podem resultar em uma acurácia melhor que a obtida com a *Random Forest*.

REFERÊNCIAS

- [1] MARSLAND, S. *Machine learning: an algorithmic perspective (2nd ed.)*. [S.l.]: Chapman and Hall/CRC, 2014.
- [2] ROKACH, L.; MAIMON, O. Decision trees. In: *Data mining and knowledge discovery handbook*. [S.l.]: Springer, 2005. p. 165–192.
- [3] NIELSEN, S. S. *Food analysis 5th Edition*. [S.l.]: Springer, 2017.
- [4] BHARGAVA, A.; BANSAL, A. Quality evaluation of mono & bi-colored apples with computer vision and multispectral imaging. *Multimedia Tools and Applications*, Springer, v. 79, n. 11, p. 7857–7874, 2020.
- [5] OLIVEIRA, M. M. et al. Classification of fermented cocoa beans (cut test) using computer vision. *Journal of Food Composition and Analysis*, Elsevier, v. 97, p. 103771, 2021.
- [6] BARBON, A. P. A. da C. et al. Development of a flexible computer vision system for marbling classification. *Computers and Electronics in Agriculture*, Elsevier, v. 142, p. 536–544, 2017.
- [7] AHMED, M. R. et al. Imaging technologies for nondestructive measurement of internal properties of agricultural products: A review. *Journal of Biosystems Engineering*, Korean Society for Agricultural Machinery, v. 42, n. 3, p. 199–216, 2017.
- [8] CHEN, Q. et al. Recent advances in emerging imaging techniques for non-destructive detection of food quality and safety. *TrAC Trends in Analytical Chemistry*, Elsevier, v. 52, p. 261–274, 2013.
- [9] DU, Z. et al. Recent advances in imaging techniques for bruise detection in fruits and vegetables. *Trends in Food Science & Technology*, Elsevier, v. 99, p. 133–141, 2020.
- [10] GONZALEZ, R. C.; WOODS, R. C. *Processamento digital de imagens*. [S.l.]: Pearson Educação, 2010.
- [11] GADE, R.; MOESLUND, T. B. Thermal cameras and applications: a survey. *Machine vision and applications*, Springer, v. 25, n. 1, p. 245–262, 2014.
- [12] URREHMAN, S.; MOHSINZAKAI, F.; ADEEL, M. Inspection on infrared-based image processing. In: IEEE. *2018 IEEE 21st International Multi-Topic Conference (INMIC)*. [S.l.], 2018. p. 1–6.
- [13] IBARRA-CASTANEDO, C. et al. Infrared image processing and data analysis. *Infrared physics & technology*, Elsevier, v. 46, n. 1-2, p. 75–83, 2004.
- [14] SOSNOWSKI, T.; BIESZCZAD, G.; MADURA, H. Image processing in thermal cameras. In: *Advanced technologies in practical applications for national security*. [S.l.]: Springer, 2018. p. 35–57.

- [15] GOWEN, A. et al. Applications of thermal imaging in food quality and safety assessment. *Trends in food science & technology*, Elsevier, v. 21, n. 4, p. 190–200, 2010.
- [16] ALI, M. M. et al. Emerging non-destructive thermal imaging technique coupled with chemometrics on quality and safety inspection in food and agriculture. *Trends in Food Science & Technology*, Elsevier, v. 105, p. 176–185, 2020.
- [17] ELMASRY, G. et al. Emerging thermal imaging techniques for seed quality evaluation: Principles and applications. *Food Research International*, Elsevier, v. 131, p. 109025, 2020.
- [18] ANSARI, S.; SALANKAR, S. An overview on thermal image processing. *RICE*, p. 117–120, 2017.
- [19] SZELISKI, R. *Computer vision: algorithms and applications 2nd Edition*. [S.l.]: Springer Science & Business Media, 2022.
- [20] STALLKAMP, J. et al. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, Elsevier, v. 32, p. 323–332, 2012.
- [21] MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of machine learning*. [S.l.]: MIT press, 2018.
- [22] GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: " O'Reilly Media, Inc.", 2019.
- [23] KINGSFORD, C.; SALZBERG, S. L. What are decision trees? *Nature biotechnology*, Nature Publishing Group, v. 26, n. 9, p. 1011–1013, 2008.
- [24] BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- [25] CUTLER, A.; CUTLER, D. R.; STEVENS, J. R. Random forests. In: *Ensemble machine learning*. [S.l.]: Springer, 2012. p. 157–175.
- [26] MÜLLER, A. C.; GUIDO, S. *Introduction to machine learning with Python: a guide for data scientists*. [S.l.]: " O'Reilly Media, Inc.", 2016.
- [27] SHANNON, C. E. A mathematical theory of communication. *The Bell system technical journal*, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.
- [28] GROENEVELD, R. A.; MEEDEN, G. Measuring skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, Wiley Online Library, v. 33, n. 4, p. 391–399, 1984.