

**UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

Daniel Elídio Mendes Júnior

**UTILIZAÇÃO DO ALGORÍTIMO K-MEANS NA
SEGMENTAÇÃO E CLASSIFICAÇÃO DE IMAGENS DE
REFEIÇÕES**

Alfenas, 11 de Dezembro de 2015.

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**UTILIZAÇÃO DO ALGORÍTIMO K-MEANS NA
SEGMENTAÇÃO E CLASSIFICAÇÃO DE IMAGENS DE
REFEIÇÕES**

Daniel Elídio Mendes Júnior

Monografia apresentada ao Curso de Bacharelado em
Ciência da Computação da Universidade Federal de
Alfenas como requisito parcial para obtenção do Título de
Bacharel em Ciência da Computação.

[Orientador: Prof. Ricardo Menezes Salgado]

Alfenas, 11 de Dezembro de 2015.

Daniel Elídio Mendes Júnior

**UTILIZAÇÃO DO ALGORÍTIMO K-MEANS NA
SEGMENTAÇÃO E CLASSIFICAÇÃO DE IMAGENS DE
REFEIÇÕES**

A Banca examinadora abaixo-assinada aprova a monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Prof. Luiz Eduardo da Silva

Universidade Federal de Alfenas

Prof. Paulo Alexandre Bressan

Universidade Federal de Alfenas

Prof. Ricardo Menezes Salgado

Universidade Federal de Alfenas

Alfenas, 11 de Dezembro de 2015.

*Dedico este trabalho a todos os
que participaram direta ou
indiretamente desta minha
formação acadêmica.*

AGRADECIMENTO

Agradeço a Deus, pois até aqui, Ele têm me ajudado.

Aos meus pais e demais familiares, pelo incentivo e apoio e sacrifícios.

Ao Pr. Maurício Castro de Magalhães e sua esposa Profa. Dra. Cristiana Schmidt de Magalhães, aos quais têm me ajudado com suas orações e aconselhamentos.

Ao orientador Prof. Dr. Ricardo Menezes Salgado, que de bom grado têm me apoiado e ajudado durante todo o meu tempo dentro e fora desta universidade.

Aos professores que direta ou indiretamente participaram desta minha formação acadêmica.

Aos meus colegas de classe, aos quais muitas vezes, me ajudaram e dos quais tenho aprendido muito.

RESUMO

Métodos para medição de valores nutricionais na ingestão diária de alimentos têm sido cada vez mais procurados, dado que o número de pessoas com problemas relacionados à má alimentação tem crescido muito nos últimos anos.

No Brasil, é encontrada uma grande quantidade de pessoas diagnosticadas com problemas relacionados à má alimentação como hipertensão, diabetes, colesterol alto, obesidade entre outros. Pessoas que são diagnosticadas com esses tipos de doenças tem o controle rígido da alimentação incluído no seu tratamento.

A fim de auxiliar nesse processo de controle nutricional, métodos semiautomáticos que utilizam de um *smartphone* para tirar uma foto do prato de comida e realizar o cálculo dos valores nutricionais da refeição têm sido propostos. Este processo exige que técnicas rápidas e eficazes de reconhecimento de imagens sejam aplicadas, a fim de classificar os alimentos e posteriormente realizar os cálculos de valores nutricionais da refeição em tempo hábil ao paciente.

O *k-means* é um algoritmo de agrupamento que apesar de antigo, tem sido explorado pela comunidade científica na segmentação e classificação de diversos tipos de imagens.

O presente trabalho propõe o estudo do comportamento deste algoritmo durante a fase de segmentação e classificação de imagens de refeições, a fim de verificar seus prós e contras e explorar sua utilização neste domínio de aplicação. Após as simulações realizadas foi possível concluir que o algoritmo *k-means* realizou a tarefa de segmentação por cores e classificação nas três classes de alimentos propostas para a grande parte das imagens utilizadas nos experimentos.

Palavras-Chave: Reconhecimento de Alimentos, Sistema de Cálculo Nutricional, *K-Means*, Segmentação de Imagens de Refeições, Classificação de Imagens de Refeições.

ABSTRACT

Methods for measurement of nutritional values on daily food intake has been increasingly sought, given that the amount of people with problems related to bad diet have grown too much in these last years.

In Brazil, is found a large amount of people diagnosed with problems related to bad eating habits, such like hypertension, diabetes, high cholesterol, obesity and others. The people who are diagnosed with these kind of diseases have the rigorous diet control included on their medical treatments.

In order to assist in this process of nutritional control, semiautomatic methods which use a smartphone to capture a picture from the food dish and then perform the calculation of nutritional values of the meal have been proposed. This process demands that prompt and effective techniques of image recognition to be applied in order to classify and right away perform the calculation of the nutritional values of the meal in a timely manner for the patient.

The k-means is an clustering algorithm which despite being ancient, it has been explored by the scientific community on segmentation and classification of several kinds of images.

This paper aims to study the behavior of such algorithm during the meal images segmentation and classification step, in order to check its pros and cons and even explore its utilization in the field of application. After the simulations performed was concluded that the k-means algorithm accomplished the color segmentation and classification task in the three food classes proposed for much of the food images used in the experiments.

Keywords: Food Recognition, Nutritional Calculation System, K-Means, Food Image Segmentation, Food Image Classification.

LISTA DE FIGURAS

FIGURA 1 - SISTEMA DE ANÁLISE DE IMAGEM. FONTE: TRADUZIDO DE POULADZADEH, SHIRMOHAMMADI E ARICI (2013).....	24
FIGURA 2 - EXECUÇÃO K-MEANS - ENTRADA DOS PADRÕES.	34
FIGURA 3 - EXECUÇÃO K-MEANS - ITERAÇÃO 1 - DEFINIÇÃO DOS VALORES INICIAIS DOS CENTROIDES.	35
FIGURA 4 - EXECUÇÃO K-MEANS - ITERAÇÃO 1 - AGRUPAMENTO DOS PADRÕES.....	35
FIGURA 5 - EXECUÇÃO K-MEANS - ITERAÇÃO 2 - DEFINIÇÃO DOS NOVOS CENTROIDES.	36
FIGURA 6 - EXECUÇÃO K-MEANS - ITERAÇÃO 2 - RESULTADO DO AGRUPAMENTO.	36
FIGURA 7 - EXECUÇÃO K-MEANS - ITERAÇÃO 2 - DESTAQUE DA MUDANÇA ENTRE GRUPOS.	37
FIGURA 8 - EXECUÇÃO K-MEANS - ITERAÇÃO I - RESULTADO DO AGRUPAMENTO.	38
FIGURA 9 - FORMAÇÃO DA BASE DE CONHECIMENTO.....	40
FIGURA 10 - PROCESSO DE DETECÇÃO.....	44
FIGURA 11 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 1 - IMAGEM ORIGINAL.	57
FIGURA 12 - RESULTADO DA SEGMENTAÇÃO - EXEMPLO 1 - ARROZ COM FEIJÃO - SEGMENTO 1.	57
FIGURA 13 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 1 - SEGMENTO 2.	57
FIGURA 14 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 1 - SEGMENTO 3.	58
FIGURA 15 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 2 - IMAGEM ORIGINAL.	58
FIGURA 16 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 2 - SEGMENTO 1.	59
FIGURA 17 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 2 - SEGMENTO 2.	59
FIGURA 18 - RESULTADO DA SEGMENTAÇÃO - ARROZ COM FEIJÃO - EXEMPLO 2 - SEGMENTO 3.	59
FIGURA 19 - RESULTADO DA SEGMENTAÇÃO - FEIJÃO, FARINHA DE MILHO, COUVE, ARROZ, E CARNE DE PANELA - IMAGEM ORIGINAL.	59
FIGURA 20 - RESULTADO DA SEGMENTAÇÃO - FEIJÃO, FARINHA DE MILHO, COUVE, ARROZ, E CARNE DE PANELA - SEGMENTO 1.	60
FIGURA 21 - RESULTADO DA SEGMENTAÇÃO - FEIJÃO, FARINHA DE MILHO, COUVE, ARROZ, E CARNE DE PANELA - SEGMENTO 2.	60
FIGURA 22 - RESULTADO DA SEGMENTAÇÃO - FEIJÃO, FARINHA DE MILHO, COUVE, ARROZ, E CARNE DE PANELA - SEGMENTO 3.	61
FIGURA 23 - FORMAÇÃO DA BASE DE TESTES.	61
FIGURA 24 - TAXA DE ACERTOS DO CLASSIFICADOR.....	63
FIGURA 25 - ARROZ E CARNE DE PANELA - IMAGEM ORIGINAL.....	64
FIGURA 26 - ARROZ E CARNE DE PANELA - SEGMENTO 1.....	64
FIGURA 27 - ARROZ E CARNE DE PANELA - SEGMENTO 2.....	64
FIGURA 28 - ARROZ E CARNE DE PANELA - SEGMENTO 3.....	64
FIGURA 29 - ARROZ E CARNE DE PANELA - SEGMENTO 4.....	64
FIGURA 30 - ARROZ E CARNE DE PANELA - SEGMENTO 5.....	64
FIGURA 31 - ARROZ E FEIJÃO - IMAGEM ORIGINAL.	65
FIGURA 32 - ARROZ E FEIJÃO - SEGMENTO 1.....	65
FIGURA 33 - ARROZ E FEIJÃO - SEGMENTO 2.....	65
FIGURA 34 - ARROZ E FEIJÃO - SEGMENTO 3.....	65
FIGURA 35 - ARROZ E FEIJÃO - SEGMENTO 4.....	65
FIGURA 36 - ARROZ E FEIJÃO - SEGMENTO 5.....	65

FIGURA 37 - ARROZ E FEIJÃO - IMAGEM ORIGINAL.	67
FIGURA 38- ARROZ E FEIJÃO - SEGMENTO 1.	67
FIGURA 39- ARROZ E FEIJÃO - SEGMENTO 2.	67
FIGURA 40- ARROZ E FEIJÃO - SEGMENTO 3.	67
FIGURA 41 - SITUAÇÃO DO PADRÃO xk EM RELAÇÃO ÀS CLASSES 1 E 2.	68
FIGURA 42 - NÚMERO DE ACERTOS DOS SEGMENTOS VÁLIDOS.	69
FIGURA 43 - TAXA DE ACERTO DOS SEGMENTOS VÁLIDOS DE CADA UMA DAS CLASSES.....	70

LISTA DE QUADROS

QUADRO 1 - PSEUDOCÓDIGO K-MEANS. ADAPTADO DE ALPAYDIN (2009).	33
QUADRO 2 - PSEUDOCÓDIGO ALGORÍTIMO K-MEANS.	44
QUADRO 3 - PSEUDOCÓDIGO PROCESSO DE CLASSIFICAÇÃO.	48
QUADRO 4 - ALGORITMO MATLAB PARA SEGMENTAÇÃO DE CORES UTILIZANDO O <i>K-MEANS</i>	50
QUADRO 5 - ALGORITMO MATLAB CLASSIFICADOR.	51
QUADRO 6 - ALGORITMO MATLAB DE TESTES.	53

LISTA DE ABREVIACOES

BoF: *Bag-of-Features*

CIE: *Commission Internationale de l'Eclairage*

CMY(K): *(Cyan Magenta Yellow (Black))*

CRT: *Cathode Ray Tube*

Fast-NCC: *Fast Normalized Cross Correlation*

GPU: *Graphics Processing Unit*

LoG: *Laplacian of Gaussian*

MKL: *Multiple Kernel Learning*

NCC: *Normalized Cross Correlation*

RFID: *Radio Frequency Identification*

RGB: *Red Green Blue*

SVM: *Support Vector Machine*

SUMÁRIO

1 INTRODUÇÃO.....	19
1.1 JUSTIFICATIVA E MOTIVAÇÃO	19
1.2 PROBLEMATIZAÇÃO	20
1.3 OBJETIVOS.....	21
1.3.1 <i>Geral</i>	21
1.3.2 <i>Específicos</i>	21
1.4 ORGANIZAÇÃO DO TRABALHO	21
2 REVISÃO BIBLIOGRÁFICA E FUNDAMENTAÇÃO TEÓRICA	23
2.1 REVISÃO BIBLIOGRÁFICA.....	23
2.1.1 <i>Trabalhos Relacionados</i>	23
2.2 FUNDAMENTAÇÃO TEÓRICA.....	27
2.2.1 <i>Considerações Iniciais</i>	27
2.2.2 <i>Agrupamento</i>	27
3 METODOLOGIA PROPOSTA.....	39
3.1 CONSIDERAÇÕES INICIAIS	39
3.1.1 <i>O Processo de Aprendizado</i>	39
3.1.2 <i>O Processo de Detecção</i>	43
3.1.3 <i>O Processo de Classificação</i>	45
3.2 IMPLEMENTAÇÃO.....	49
3.2.1 <i>Ferramenta</i>	49
3.2.2 <i>A Implementação do Algoritmo K-Means</i>	49
3.2.3 <i>Segmentação de Cores Utilizando o K-Means</i>	49
3.2.4 <i>Algoritmo Classificador</i>	51
3.2.5 <i>Algoritmo de Teste</i>	53
4 RESULTADOS.....	56
4.1 SEGMENTAÇÃO K-MEANS	56
4.2 CLASSIFICAÇÃO K-MEANS	61
4.2.1 <i>Formação da Base de Testes</i>	61
4.2.2 <i>Resultado da Classificação</i>	62
5 CONCLUSÕES E PROPOSTAS FUTURAS.....	71
5.1 CONCLUSÕES	71
5.2 PROPOSTAS FUTURAS.....	72
6 REFERÊNCIAS BIBLIOGRÁFICAS	73

1 Introdução

1.1 Justificativa e Motivação

A má alimentação causada pela falta de instrução ou má observação das atividades alimentícias pode ser considerada um dos fatores que mais contribuem para problemas de saúde comumente encontrados no Brasil como obesidade, diabetes, pressão alta, entre outros. No mundo, os casos de obesidade tem quase dobrado desde 1980, de acordo com Organização Mundial de Saúde (*World Health Organization*) (2015).

De acordo com a Organização Mundial de Saúde (OMS), a obesidade tem alcançado proporções epidêmicas, com cerca de 1.5 bilhões de adultos sofrendo de obesidade em 2008, Villalobos, Almagrabi, *et al.* (2011).

Desde então, a busca por métodos para acompanhamento de dieta tem crescido drasticamente. Métodos manuais têm sido propostos pela comunidade científica, porém, estes métodos são considerados, na maioria dos casos, muito dispendiosos por exigirem muito tempo, memória ou preparação do paciente para realizar tais procedimentos de medição de ingestão nutricional para uma dada refeição. Desde então, métodos semiautomáticos, que exigem o mínimo de esforço do paciente têm sido propostos pela comunidade científica. Alguns destes métodos utilizam de um *smartphone* para obter uma foto da refeição e então é realizado o cálculo dos valores nutricionais desta refeição Pouladzadeh, Shirmohammadi e Arici (2013).

Joutou e Yanai (2009) afirmam que nos últimos cinco anos, técnicas de reconhecimento de imagens tem feito grande progresso, graças ao desenvolvimento de novas formas de representação de características como as *bag-of-features* (BoF), e métodos de aprendizagem de máquina, como as *support vector machines* (SVM). Esses tipos de avanços fazem com que sistemas que necessitem de reconhecimento de imagens em tempo hábil se tornem praticáveis.

Técnicas que visam misturar um conjunto de características de uma imagem a fim de melhorar a precisão do sistema de reconhecimento têm sido propostas na literatura com resultados satisfatórios. Outra técnica como as *Convolutional Neural Networks* (CNN), uma técnica já comprovada, mas que até então era impraticável devido à falta de recursos de hardware, como a *Graphic Processing Unit* (GPU), hoje tem sido aplicadas no reconhecimento de imagens e alcançado resultados interessantes.

1.2 Problematização

Para que um sistema automatizado faça os cálculos da informação nutricional de uma dada refeição, é necessário que um bom sistema de reconhecimento de imagens seja aplicado, para que se faça a distinção e o reconhecimento dos vários itens que compõe uma refeição.

Villalobos, Almaghrabi, *et al.* (2012) afirmam que pelo fato de existirem muitas categorias de alimentos, tem sido impossível realizar o reconhecimento de refeições em tempo hábil. Porém, recentemente, pesquisadores têm feito novos avanços na área de representação de características, e aprendizado de máquina, tais como *bag-of-features* (BoF) e *support vector machines* (SVM), como descrito em Joutou e Yanai (2009). Para o problema de reconhecimento de imagens de comida, a adição de características de cor e textura a técnica *bag-of-features* (BoF) têm se mostrado dignas de atenção.

Outro algoritmo é o *k-means*, que tem sido explorado no domínio de reconhecimento de imagens digitais das mais diversas categorias, desempenhando um papel muito importante nas etapas de segmentação - com adaptações para se trabalhar com espaços de cores - e classificação dessas imagens. Na maioria dos casos, este algoritmo tem sido usado em combinação com técnicas, como a *Laplacian of Gaussian* (LoG) e *Prewitt filter* para filtrar bordas em uma imagem, entre outras técnicas, a fim de alcançar uma melhor precisão durante o processo de segmentação e/ou classificação de uma imagem, como descrito em (SUBBIAH e CHRISTOPHER, 2012).

O *k-means* é um algoritmo de agrupamento antigo proposto por Macqueen (1967), de baixo custo computacional e que vem sendo explorado pela comunidade científica em várias aplicações. O presente trabalho visa responder a seguinte questão: É possível utilizar o algoritmo *k-means*, na segmentação e classificação de imagens de refeições?

1.3 Objetivos

1.3.1 Geral

Estudar o comportamento do algoritmo de agrupamento *k-means* durante o processo de segmentação e classificação de imagens, sendo essas, especificamente imagens de refeições e verificar seus prós e contras.

1.3.2 Específicos

Para que o objetivo principal seja alcançado, alguns objetivos específicos são requeridos, entre eles:

- Familiarizar-se com a teoria do algoritmo *k-means*.
- Estudar o comportamento do algoritmo em questão no processo de segmentação das imagens, analisando casos de sucesso e falha durante esse processo.

1.4 Organização do Trabalho

Este trabalho é organizado da seguinte forma:

O Capítulo 1 apresenta uma introdução do trabalho.

O Capítulo 2 apresenta uma revisão bibliográfica sobre o tema e a fundamentação teórica que envolve o projeto.

O Capítulo 3 apresenta a metodologia proposta neste trabalho, os processos que envolvem a formação das bases de conhecimento e testes utilizadas neste projeto, a utilização do algoritmo *k*-means nas fases de segmentação e de classificação das imagens, e detalhes sobre a implementação.

O Capítulo 4 apresenta os resultados dos testes de segmentação e classificação das imagens e um estudo sobre os pontos onde o algoritmo obteve sucesso e pontos onde ele apresentou falhas.

O Capítulo 5 apresenta as conclusões deste trabalho.

O Capítulo 6 finaliza com as referências bibliográficas utilizadas no desenvolvimento deste trabalho.

2

Revisão Bibliográfica e Fundamentação Teórica

2.1 Revisão Bibliográfica

2.1.1 Trabalhos Relacionados

Segundo Pouladzadeh, Shirmohammadi e Arici (2013), métodos para cálculo de valores nutricionais podem ser divididos em quatro categorias: métodos clínicos, assistidos, semiautomáticos e automáticos.

Os métodos clínicos consistem na descrição feita pelo paciente da sua própria alimentação, preenchendo em um formulário ou similar, a fim de um profissional realizar o cálculo dos valores nutricionais consumidos naquela dada refeição. Essa abordagem, porém, traz limitações no que se diz respeito à precisão, pois exige que o paciente tenha uma boa memória, e tempo para preencher formulários.

Os métodos assistidos foram criados para amenizar as limitações geradas pelo método clínico, e consiste em utilizar *smartphone* ou similar para obter uma foto da refeição e enviar online para um profissional treinado que irá realizar os cálculos dos valores nutricionais e então enviar para o paciente. Este método ajuda no fato do paciente não precisar utilizar a memória para preencher formulários sobre sua alimentação. Isto nos leva a um terceiro método, o semiautomático.

O método semiautomático consiste em deixar a maior parte do cálculo dos valores nutricionais para um sistema especializado, o que diminui muito o trabalho e o tempo de resposta com os valores nutricionais.

Finalmente temos o método automático, que consiste em utilizar um sistema especializado para realizar todo o processo de detecção dos alimentos ingeridos e

cálculo dos valores nutricionais. Um método como este, utilizando uma mesa equipada com alguns tipos de sensores e utilizando *Radio Frequency Identification* (RFID) foi proposto por (CHANG, LIU, *et al.*, 2006).

No trabalho proposto por Chang, Lui, *et al.* (2006), o RFID é utilizado para detectar o alimento enquanto os sensores localizados dentro da mesa, detectam o peso da porção. Pouladzadeh, Shirmohammadi e Arici (2013) argumentam que este método é impraticável, dado que requer acessórios para acompanhar a medição, o que o faz não aplicável para situações do dia a dia como comer em um restaurante ou em outro local, onde o paciente não teria acesso a uma mesa equipada como essa.

Pouladzadeh Shirmohammadi e Arici, (2013) propuseram a aplicação de um modelo SVM, juntamente com uma arquitetura (Figura 1) para a construção de um sistema de medição de ingestão alimentícia, utilizando a extração da forma e textura de uma imagem para o reconhecimento dos valores nutricionais de uma dada refeição.

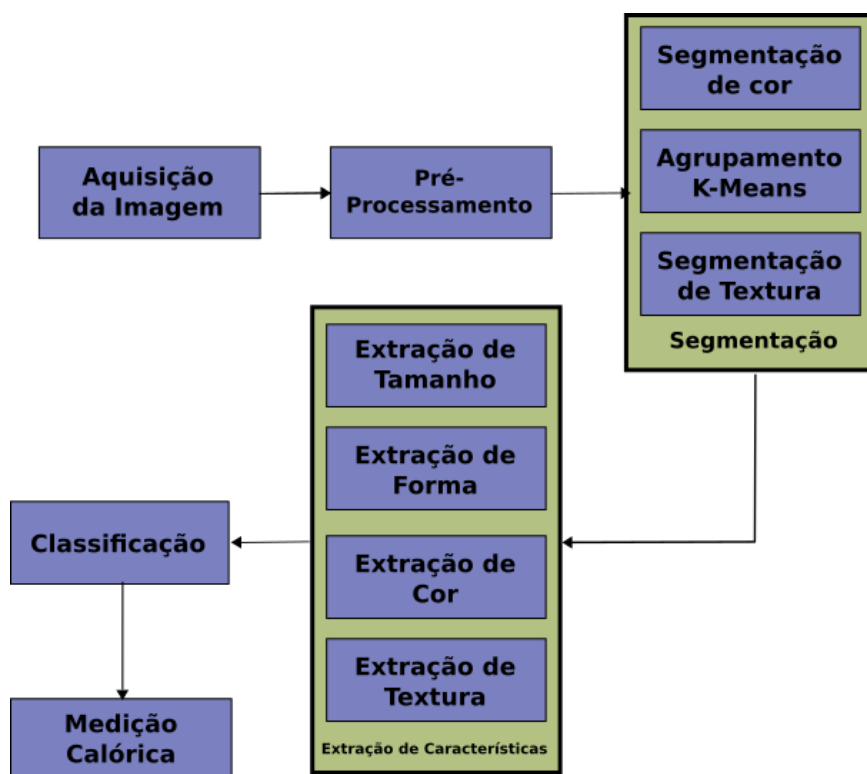


Figura 1 - Sistema de Análise de Imagem.

Fonte: Traduzido de Pouladzadeh, Shirmohammadi e Arici (2013).

Como descrito na Figura 1, na arquitetura proposta por Pouladzadeh, Shirmohammadi e Arici (2013), antes que a imagem chegue à etapa de classificação a imagem, que pode ser adquirida através da câmera de um smartphone, por exemplo, passa por um processo de segmentação de cor e textura. Logo após a segmentação, a imagem passa pelo processo de extração de características, onde as características cor, forma, textura e tamanho são extraídos antes de ir para a etapa de classificação.

De acordo com Sapna Varshney, Rajpal e Purwar (2009), a segmentação é uma das primeiras etapas na análise de uma imagem e consiste em particionar uma imagem em múltiplas regiões, ou seja, conjunto de pixels. Esse particionamento pode ser feito utilizando métodos diferentes como os métodos baseados em agrupamento, limiarização, detecção de bordas e métodos baseados em região. Algoritmos de agrupamento como o *k-means* e o *mean-shift* têm sido explorados, tanto sozinhos como combinados com outras técnicas, na segmentação e na classificação de diversos tipos de imagens.

Uma aplicação do algoritmo *k-means* para realizar a segmentação por cores foi proposta por Sapna Varshney, Rajpal e Purwar (2009) em uma imagem que contém células de sangue a fim de realizar a separação do fundo da imagem das células que compõe a imagem. O método consiste em utilizar o algoritmo *k-means* para distribuir as cores dos pixels da imagem em duas cores dominantes, ou seja, centroides. Um dos centroides resultantes será muito similar a cor de fundo predominante e o outro similar a cor predominante nas células.

Villalobos, Almaghrabi, *et al.* (2012) propuseram um sistema utilizando um esquema de segmentação de cores para reconhecimento das imagens de refeições, utilizando apenas segmentação *Mean-Shift* e calibração utilizando o dedo polegar.

A fim de melhorar a precisão do sistema, foi inserida a segmentação por textura em Pouladzadeh, Villalobos, *et al.* (2012), no sistema que já utilizava a técnica SVM para aprendizado e classificação.

Em Pouladzadeh, Shirmohammadi e Arici (2013) ainda mais uma melhoria no sistema com a inclusão do algoritmo *k-means* durante a fase de segmentação de cores, considerando a média dos resultados obtidos pela segmentação feita pelos algoritmos *mean-shift* e *k-means*, e ainda combinando o resultado da segmentação

de cores com a segmentação de textura incluso na última versão do sistema. O trabalho também utilizou mais dados no treinamento e teste do classificador SVM.

Em Subbiah e Christopher (2012), os autores observam que técnicas de detecção de bordas tem tido um papel significativo no reconhecimento de padrões. O mesmo trabalho então, propõe o uso do algoritmo *k-means* adaptado para trabalhar imagens, combinando com os filtros de bordas *Laplacian of Gaussian (LoG) filter* e *Prewitt filter*. O processo é feito adquirindo uma imagem, e realizando a segmentação utilizando o algoritmo *k-means* com adaptações para trabalhar com imagens, e aplicando os filtros de borda *Laplacian of Gaussian (LoG)* e *Prewitt*, realizando a comparação dos resultados. Um banco de dados de imagens de satélite é utilizado para realizar os testes.

Outro exemplo que utilizou um algoritmo *k-means* adaptado para trabalhar com imagens na segmentação e classificação de imagens de satélite é encontrado em Usman (2013), onde o autor procurou realizar a segmentação de imagens de satélite a fim encontrar fazendas, áreas contrsuídas, terrenos entre outros.

Um benchmark da segmentação de imagens utilizando o algoritmo *mean-shift* e outros três algoritmos é apresentado em (ESTRADA e JEPSON, 2009).

Um classificador baseado em *wavelets* utilizando o algoritmo *k-means* para classificação de imagens de ressonâncias magnéticas cerebrais foi proposto por (GONAL e KOHIR , 2015). O método proposto utilizava *wavelets* na decomposição da imagem e extração das características de textura utiliando matriz de co-ocorrência de tons de cinza em três níveis de resolução da imagem. O algoritmo *k-means* foi utilizado para realizar a classificação das imagens em normais e anormais de acordo com características extraídas da imagem.

Outro método de classificação de imagens utilizando o algoritmo *k-means* foi proposto em Jeyanthi e Jawahar (2009) onde as características da imagem foram obtidas de um histograma de cores da imagem. Jeyanthi e Jawahar (2009) também argumentam que histogramas de cores apresentam características muito superficiais de imagens e por esse motivo, imagens diferentes podem gerar histogramas similares, sendo uma desvantagem para um método de classificação por cores.

2.2 Fundamentação Teórica

2.2.1 Considerações Iniciais

Trabalhos relacionados com este utilizaram técnicas de aprendizado e classificação como *Support Vector Machines*, *Convolutional Neural Networks*, técnicas para agrupamento de cores e texturas como *Mean-Shift* e *K-Means*, técnicas para reconhecimento de imagens como *Template Matching*, *Normalized Cross Correlation* (NCC) e *Fast Normalized Cross Correlation* (Fast-NCC).

O algoritmo de agrupamento denominado *k-means* foi escolhido como objeto principal na realização da segmentação e classificação de imagens de refeições. Uma descrição mais detalhada deste algoritmo e as adaptações para trabalhar com imagens serão feitas na secção 2.2.2.3.

Esta secção inclui uma descrição sobre cada uma das técnicas utilizadas neste trabalho a fim de familiarizar o leitor com o conteúdo abordado.

2.2.2 Agrupamento

2.2.2.1 Definição

Agrupamento ou *clustering* é um processo de classificação não supervisionada que tem a intenção de recuperar objetos que possuem semelhança em uma ou mais característica e rotula-los em grupos distintos. Ou seja, rotular elementos de um conjunto não rotulado, de acordo com a similaridade de uma ou mais características.

As técnicas de agrupamento podem ser utilizadas na exploração e análise de padrões de várias maneiras como, tomada de decisões, aprendizado de máquina, mineração de dados, recuperação de documentos, segmentação de imagens e classificação de padrões.

Formalmente, de acordo com Hruschka e F. Ebecken (2003), o processo de agrupamento de um conjunto X de n objetos em k grupos, onde $X = X_1, X_2, \dots, X_n$ onde cada $X_i \in R^p$ é um vetor de p valores reais que dimensionam as características

do objeto, estes devem ser agrupados em grupos disjuntos $C = C_1, C_2, \dots, C_k$, sendo de forma que as seguintes condições sejam atendidas:

1. $C_1 \cup C_2 \cup \dots \cup C_k = X$
2. $C_i \neq \emptyset, \forall i, 1 \leq i \leq k$
3. $C_i \cap C_j = \emptyset, \forall i, j, 1 \leq i, j \leq k$

Dessa forma, todos os objetos devem pertencer a pelo menos um grupo, condição 1; cada grupo deve conter pelo menos um objeto, condição 2; um objeto não pode pertencer a mais de um grupo, condição 3.

2.2.2.2 O Processo de Agrupamento

De acordo com Jain e Dubes (1988), os algoritmos de agrupamento, de um modo geral, passam por cinco estágios básicos:

1. Representação do padrão, que pode opcionalmente incluir extração e/ou seleção de características.
2. Definição de uma métrica de similaridade (proximidade) apropriada para o padrão.
3. Agrupamento.
4. Abstração de dados (se necessário).
5. Avaliação dos resultados (se necessário).

As etapas 1 e 2 são de extrema importância, pois definem a maneira que os dados serão representados e qual a métrica de similaridade adequada para estes dados, isto é, como os padrões relacionam entre si.

Uma formalização sobre os tipos de métricas de similaridade e suas restrições, será descrita na seção 2.2.2.2.1.

2.2.2.2.1 Métricas Para Similaridade de Objetos

Como descrito anteriormente, o agrupamento de dados dá-se pelo cálculo da similaridade entre uma ou mais características de objetos de um dado conjunto. Visto que o cálculo de similaridade é fundamental para a formação de um grupo,

definir métricas precisas de similaridade é primordial para um algoritmo de agrupamento.

De acordo com Jain, Murty e Flynn (1999), as características de um objeto podem variar e como consequência, as maneiras com que essas características são representadas também variam, tanto em forma, como em escala. Portanto, a métrica de similaridade para um dado caso, deve ser escolhida cuidadosamente, caso contrário, o algoritmo entenderá as características dos objetos de maneira que não representam o real e, como consequência, o resultado do agrupamento não representará a verdadeira divisão entre os grupos.

2.2.2.2 Índices de Similaridade

Anderberg (1973) *apud* Jain e Dubes (1988) descreve algumas restrições que devem ser obedecidas ao se considerar uma métrica de similaridade entre objetos. O índice de proximidade (similaridade) entre o i th e k th padrão é descrito como $d(i, k)$. Esta métrica deve obedecer as seguintes restrições:

1. (a) $d(i, i) = 0$, para todo i , no caso de cálculo de disparidade, i.e, dessemelhança entre objetos.
(b) $d(i, i) \geq \max d(i, k)$, para todo (i, k) , no caso de cálculo de similaridade, i.e, semelhança entre objetos.
2. $d(i, k) = d(k, i)$, para todo (i, k) .
3. $d(i, k) \geq 0$, para todo (i, k) .

A restrição 1 diz que, então a distância entre um objeto e ele mesmo, deve ser 0, no caso de cálculo de disparidade, ou seja, eles são iguais, e a distância deve ser o máximo possível, no caso de cálculo de similaridade, ou seja, a semelhança entre eles é a maior possível, indicando que eles são iguais.

A restrição 2 diz que a distância entre um objeto i e um objeto k deve ser a mesma quando calculado a distância do objeto k para o objeto i .

A restrição 3 diz que se dois objetos são ditos distintos, isto é, i e k não são os mesmos objetos, então seu índice de disparidade não é nulo, isto é, eles tem alguma diferença, mesmo que mínima.

Os índices de proximidade (similaridade) entre dois objetos podem ser classificados em dois tipos, nominais e proporcionais.

Dentre as métricas proporcionais, se destaca a métrica de *Minkowski*. Um caso particular desta métrica é chamado de Distância Euclidiana, que por sua vez, é a métrica de similaridade mais usada no algoritmo *k-means*.

Por fazer parte do escopo de interesse deste trabalho, será feita uma breve descrição da métrica de *Minkowski* e a Distância Euclidiana, a fim de facilitar o entendimento do algoritmo *k-means*.

2.2.2.2.1 Métrica de *Minkowski*

A métrica de Minkowski é utilizada para medir a dissimilaridade entre objetos. De acordo com Jain e Dubes (1988), esta métrica se utiliza de uma matriz $[x_{ij}]$ definida da seguinte maneira:

$$x_i = (x_{i1}x_{i2} \dots x_{id})^T, i = 1, 2, \dots, n$$

Onde, $[x_{ij}]$ é uma matriz de padrões onde, x_{ij} representa a j -ésima característica do i -ésimo padrão, d é o número de características representadas na matriz $[x_{ij}]$, n o número de padrões existentes e T uma transposição da matriz.

Conhecida a matriz $[x_{ij}]$, a métrica de Minkowski é definida como:

$$d(i, k) = \left(\sum_{j=1}^n |x_{ij} - x_{kj}|^r \right)^{\frac{1}{r}} \text{ onde } r \geq 1$$

Além das restrições descritas no índice 2.2.2.2.2, todas as métricas baseadas na métrica de *Minkowski* satisfazem as seguintes propriedades:

4. $d(i, k) = 0$, somente se $x_i = x_k$.
5. $d(i, k) \leq d(i, m) + d(m, k)$, paratodo (i, m, k) .

O item quatro diz que a distância entre dois objetos diferentes deve ser 0 se e somente se $x_i = x_k$, i.e, se um objeto for igual ao outro.

O item cinco é chamado de *desigualdade triangular*.

Gower e Legendre (1986) *apud* Jain e Dubes (1988) mostram que para uma matriz de dissimilaridade $[d(i,j)]$, somente as propriedades 1 e 4 se fazem necessárias, dado que, as demais propriedades descritas derivam destas 2.

De um modo geral, algoritmos de agrupamento passam por três estágios básicos durante sua execução, a saber, seleção da característica a ser agrupado, cálculo de similaridade entre dois objetos, e por fim o agrupamento.

O método para se calcular a similaridade entre objetos varia de caso para caso, pois cada característica pode ser representada de uma maneira diferente e exigir cálculos diferentes para sua representação. O método mais comum é a distância euclidiana dada pela Equação 1.

2.2.2.2.2 Distância Euclidiana

A distância euclidiana, Equação 1, é um caso particular da métrica de *Minkowski* com a configuração $r = 2$.

$$d(i, k) = \left(\sum_{j=1}^n |x_{ij} - x_{kj}|^2 \right)^{\frac{1}{2}} = \sqrt{(x_{i1} - x_{k1})^2 + (x_{i2} - x_{k2})^2 + \dots + (x_{in} - x_{kn})^2}$$

Equação 1 - Distância Euclidiana

No processo de agrupamento de dados, quando dois padrões são representados como pontos em um plano cartesiano, a distância entre esses dois pontos representa a similaridade entre esses dois padrões de maneira inversamente proporcional, ou seja, quanto menor a distância entre os pontos, maior a similaridade que esses objetos têm um com o outro.

2.2.2.3 O Algoritmo *K-Means*

2.2.2.3.1 Definição

O algoritmo *k-means* é popularmente conhecido como *Hard C Means*. De acordo com Jain, Murty e Flynn (1999), o *k-means* é um algoritmo de agrupamento, pertencente grupo dos algoritmos particionais, que por sua vez tem a característica de iniciar várias execuções partindo de estados diferentes, e assim que o algoritmo

termina sua execução, a melhor configuração de cada uma destas execuções é usada como o agrupamento final.

De modo geral, o *k-means* inicia sua execução particionando os padrões em clusters. Então esses *clusters* são ajustados até que um critério de convergência seja alcançado. O termo convergência é utilizado quando não há mais realocação de padrão entre os *clusters* durante a fase de ajuste dos *clusters*, (JAIN, MURTY e FLYNN, 1999).

O *k-means* se popularizou por ser um algoritmo de fácil codificação, e por ser relativamente rápido. O maior problema deste algoritmo é que ele é sensível a partição inicial realizada no início do algoritmo, e se essa partição inicial não for bem realizada, a execução pode convergir para o mínimo local da função objetivo especificada, (JAIN, MURTY e FLYNN, 1999). Uma descrição de como a partição inicial foi realizada é apresentada na seção 4.

De acordo com Macqueen (1967), o objetivo do *k-means* realizar a divisão de uma população n-dimensional em *k* grupos. Esse objetivo é encontrado minimizando a distância total entre os membros de um grupo e o centroide representante do grupo.

Formalmente, a função objetivo pode ser descrita como:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

Equação 2 - Função Objetivo Utilizada no *k-means* – Extraído de Usman (2013)

Onde $\|x_i^{(j)} - c_j\|^2$ é a métrica de similaridade escolhida e representa a similaridade entre o ponto $x_i^{(j)}$ e o centroide c_j .

Em detalhes, o *k-means* segue o pseudocódigo apresentado no Quadro 1:

Quadro 1 - Pseudocódigo K-Means. Adaptado de Alpaydin (2009).

k (Número de grupos)

$X = x^1, x^2, \dots, x^n$ (Conjunto de padrões a serem agrupados, com n elementos)

$C = c_1, c_2, \dots, c_k$ (Conjunto de centroides médios)

b_i^t (Matriz que indica se um elemento t está rotulado como pertencente ao grupo i)

Inicializar $c_i \leftarrow x^t \in X$, para todo $i = 1, \dots, k$, x^t é um valor aleatório pertencente a X

Repita

Para Todo $x^t \in X$

$b_i^t \leftarrow 1$ **Se** $\|x^t - c_i\| = \min_j \|x^t - c_j\|$

$b_i^t \leftarrow 0$ otherwise

Fim Para

Para Todo $c_i \in \{1, \dots, k\}$

$c_i \leftarrow \text{somar sobre } t (b_i^t x^t) / \text{somar sobre } t (b_i^t)$

Fim Para

Até Que condição de parada

O vetor C é o vetor de centroides médios de cada cluster; o conjunto X é o conjunto de padrões a serem agrupados; e b contém os rótulos dos grupos estimados para cada elemento, (ALPAYDIN, 2009).

De maneira geral, o algoritmo *k-means* executa as seguintes etapas:

- 1) Inicializa c_i como sendo o elemento médio de cada grupo, para todo $i = 1, \dots, k$.
- 2) Para cada padrão $x^t \in X$, rotula o padrão ao grupo que possui o elemento médio mais próximo de x^t .
- 3) Recalcula os elementos médios m_i baseados no novo estado do grupo, isto é, incluindo os elementos que foram rotulados como pertencentes ao grupo i .
- 4) Repete as etapas 2 - 3 até que a condição de parada seja atingida. Normalmente as condições de parada são definidas

conforme o problema, sendo as mais comuns a estabilidade dos centroides ou quando um número de iterações máxima é atingida.

A configuração final dos grupos, quando m_i converge, é o resultado do agrupamento.

2.2.2.3.2 Um Exemplo Prático

Dado que o algoritmo *k-means* recebe um conjunto de padrões $X = x^1, x^2, \dots, x^n$, definidos, como na Figura 2, e deseja-se agrupar estes padrões em $k = 3$ grupos.



Figura 2 - Execução *K-Means* - Entrada dos Padrões.

O algoritmo inicia sua execução definindo os valores iniciais dos centroides $c_1, c_2, \dots, c_k \in C$ de acordo com um método pré-definido. Neste exemplo, os valores iniciais dos centroides serão definidos aleatoriamente, resultando na configuração apresentada na Figura 3. Cada centroide c_1, c_2, \dots, c_k será então o representante dos grupos 1, 2, ...k respectivamente.

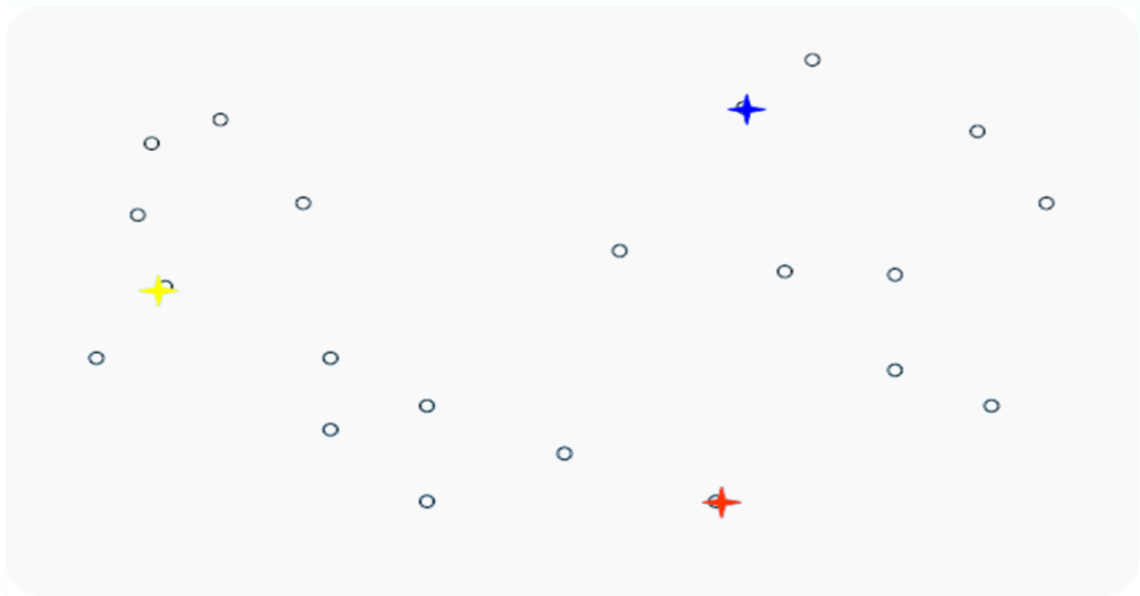


Figura 3 - Execução *K-Means* - Iteração 1 - Definição dos Valores Iniciais dos Centroides.

Uma vez definidos os valores iniciais dos centroides, o algoritmo então define cada um dos padrões como pertencentes ao grupo representado pelo seu centroide mais próximo, ou seja, se o padrão x^i está mais próximo do centroide c_j , então o padrão x^i será rotulado como pertencente ao grupo j . A Figura 4 mostra o resultado do agrupamento inicial.

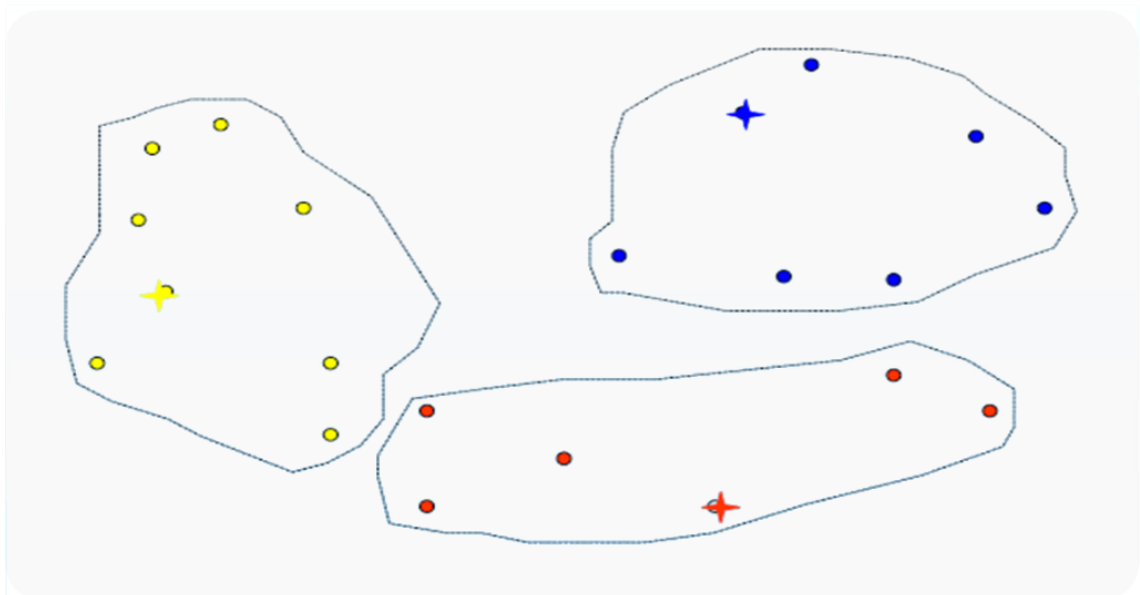


Figura 4 - Execução *K-Means* - Iteração 1 - Agrupamento dos Padrões.

O algoritmo então reinicia sua execução, porém desta vez, os novos valores dos centroides serão definidos como a média dos elementos de seu respectivo grupo. A Figura 5 mostra os novos centroides escolhidos, enquanto a Figura 6 mostra o agrupamento resultante desses novos valores dos centroides.

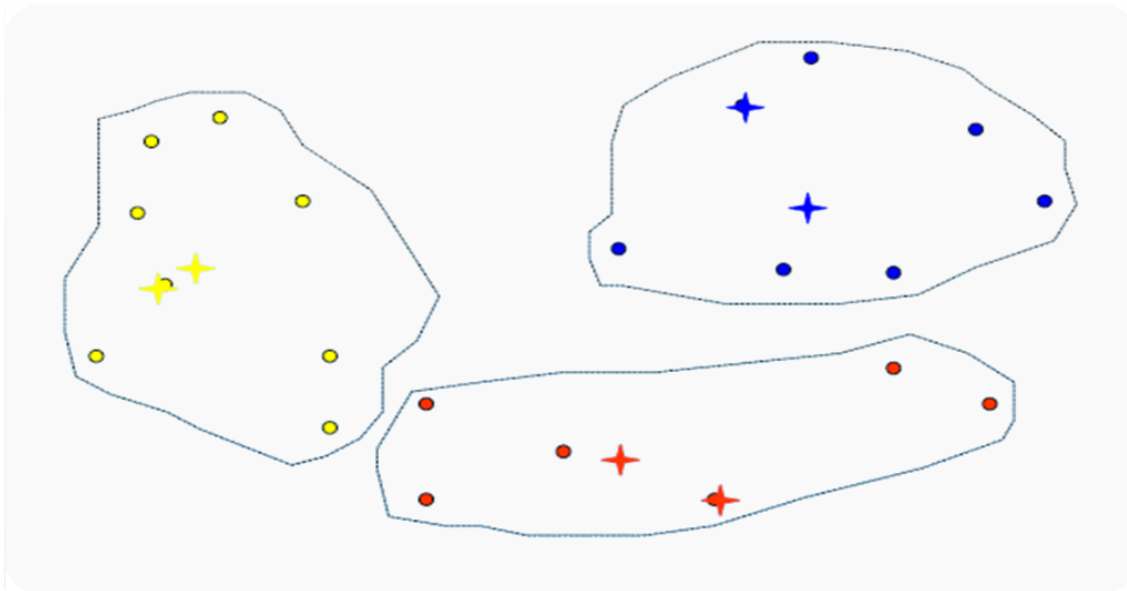


Figura 5 - Execução *K-Means* - Iteração 2 - Definição dos Novos Centroides.

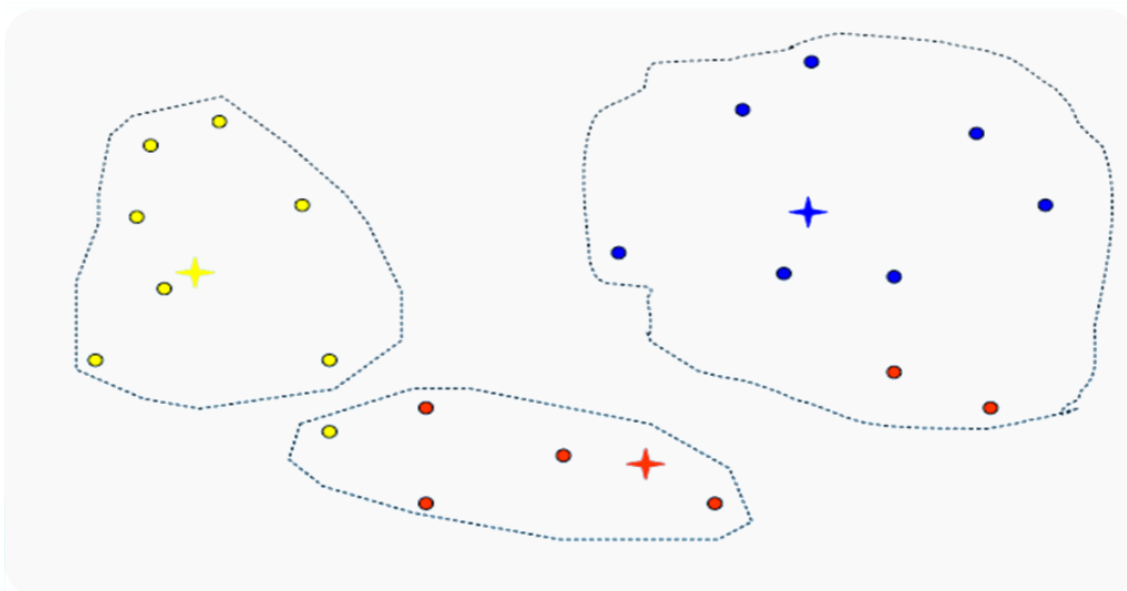


Figura 6 - Execução *K-Means* - Iteração 2 – Resultado do Agrupamento.

Quando um novo centroide é definido, a posição deste centroide é transladada no plano cartesiano, fazendo com que este fique mais perto de algum

padrão pertencente a outro grupo e mais longe de um padrão que pertence ao seu próprio grupo. Como consequência, quando os padrões são reagrupados, pode ocorrer de um ou mais padrões se moverem para grupos representados por centroides que estejam mais próximos deles, como mostrado na Figura 7.

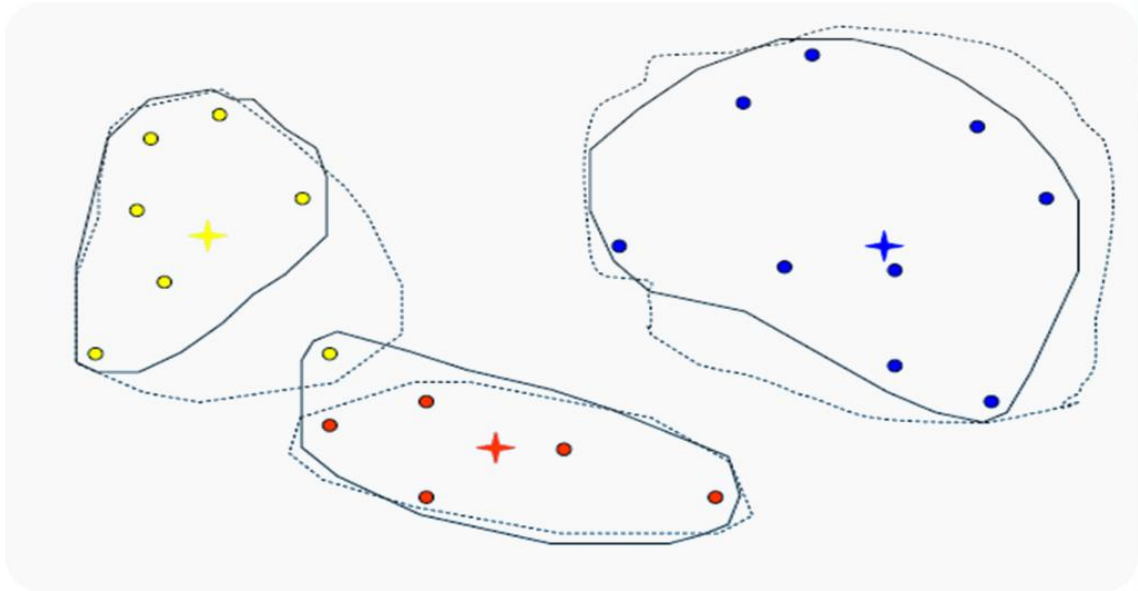


Figura 7 - Execução *K-Means* - Iteração 2 – Destaque da Mudança Entre Grupos.

Essas mudanças entre grupos tendem a ocorrer cada vez menos, dado que *k-means* está minimizando a distância média entre os padrões dos grupos. Quando em uma iteração não ocorre mais essas mudanças, ou seja, nenhum padrão entra e nenhum padrão sai de nenhum grupo, então isso é chamado de convergência, e o algoritmo para.

A configuração final dos grupos é então considerada o resultado do agrupamento feito pelo algoritmo *k-means*, ilustrado na Figura 8.

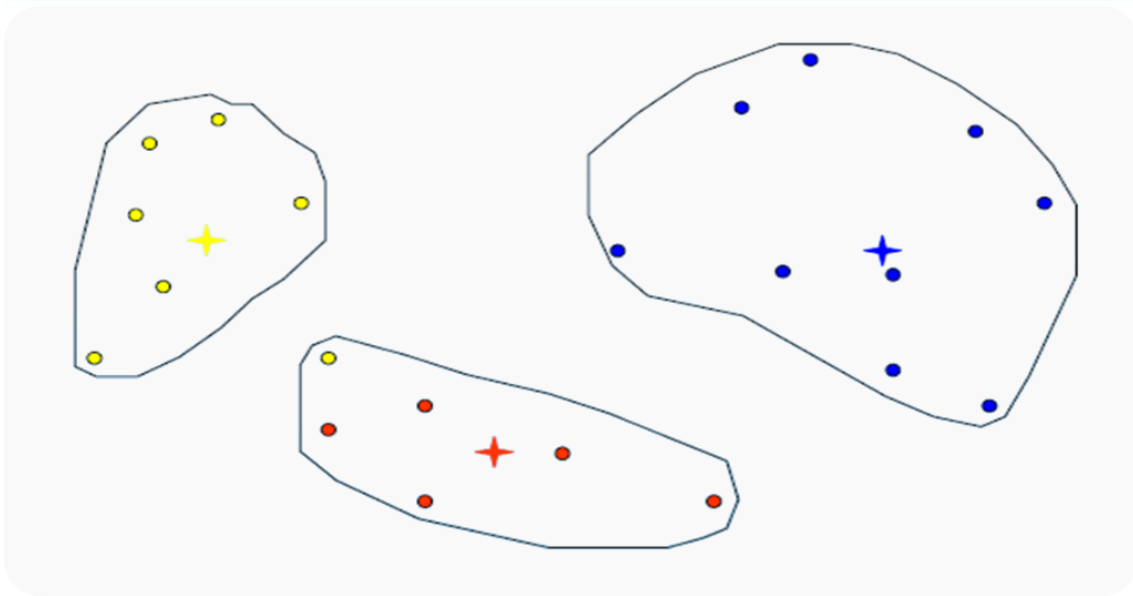


Figura 8 - Execução *K-Means* - Iteração *i* – Resultado do Agrupamento.

Essa característica de reiniciar sua execução várias vezes partindo de estados diferentes, é o que caracteriza o *k-means* como sendo pertencente à classe dos algoritmos particionais.

3

Metodologia Proposta

3.1 Considerações Iniciais

A seguir será descrito o modelo proposto em duas etapas. A primeira etapa refere-se ao processo de aprendizado, onde se mostra todo o processo que envolve a entrada de uma imagem, sua segmentação utilizando o algoritmo *k-means* e geração da base de conhecimento de acordo com os centroides gerados pelo processo de segmentação. A segunda etapa apresenta o processo de classificação, onde se apresenta o processo de entrada de imagem, sua segmentação, comparação com a base de conhecimento previamente adquirida, e resultado da classificação.

3.1.1 O Processo de Aprendizado

3.1.1.1 Descrição

O processo de aprendizado é de extrema importância nos métodos de classificação em geral. De acordo com Salgado, Ohishi, *et al.* (2010), as técnicas utilizadas no processo de reconhecimento de padrões necessitam de uma etapa de aprendizado, também chamada de treinamento. Portanto o processo de aprendizado esse é o primeiro passo para se fazer classificação de padrões em uma aplicação.

De modo geral, o processo de aprendizado consiste na formação de uma base de conhecimento que seja consistente e não contraditória a fim de que esta base de conhecimento represente da melhor maneira possível o domínio da aplicação (SALGADO, OHISHI, *et al.*, 2010).

3.1.1.2 Formação da Base de Conhecimento

Como descrito acima, a base de conhecimento não pode ser contraditória, a fim de que o algoritmo de classificação não entre num estado de indecisão sobre a qual classe um determinado padrão pertence. Nesse sentido foi utilizado um banco de imagens com alimentos parecidos, porém, que representem situações diversas quanto à organização de um prato de comida.

Foi utilizada uma base de conhecimento com três classes para abrigar os padrões associados a elas, isto é, padrões de arroz, padrões de feijão e padrões de carne de panela.

A Figura 9 mostra um fluxograma que descreve o processo de formação da base de conhecimento para o domínio da aplicação.

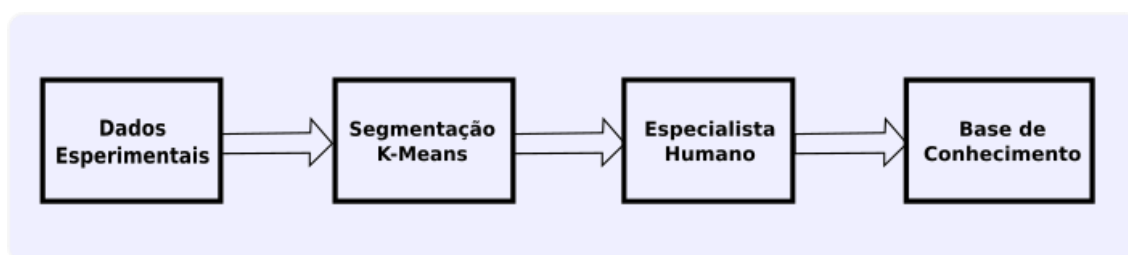


Figura 9 - Formação da Base de Conhecimento.

A saída de cada etapa serve como entrada para a etapa seguinte como será descrito em detalhes abaixo.

Dados Experimentais: Conjunto de imagens utilizadas para realizar o treinamento do algoritmo. Imagens diferentes que contém predominantemente alimentos comuns na refeição brasileira, tais como arroz, feijão, carne de panela.

Esses foram escolhidos por se tratarem de itens considerados básicos na refeição brasileira e por terem algumas propriedades particulares que beneficiam e outras que prejudicam tanto o algoritmo de treinamento quanto o de classificação, o que será discutido em uma seção mais adiante.

Outra preocupação foi a variedade das imagens adquiridas a fim de que a base de conhecimento represente situações diversas quanto ao estado do item no

prato – feijão molhado, feijão seco (popularmente conhecido como tutu de feijão) – e até mesmo a organização do prato de comida – itens que podem ser confundidos com outros por serem semelhantes.

Segmentação *K-Means*: Dada uma imagem dos dados experimentais, esta passa por um processo de segmentação por cor, a fim de dividir a imagem em segmentos que possuem cores semelhantes.

O algoritmo *k-means* foi escolhido por se tratar de um algoritmo extremamente rápido e eficiente. Como descrito na seção 2.2.2.3, o algoritmo recebe um conjunto de padrões X como entrada e realiza o agrupamento de cada um dos padrões de X em k clusters de acordo com a distância entre estes padrões, distância esta que é definida pela métrica de similaridade escolhida para o domínio da aplicação.

No presente trabalho, foi utilizado um algoritmo *k-means* adaptado para trabalhar com padrões RGB de uma imagem. O algoritmo também é adaptado para ter o número de clusters definidos automaticamente. Variações do algoritmo *k-means* que visam encontrar o número de clusters automaticamente podem ser encontradas em (CHEN, LUO e PARKER, 1998).

Como descrito na seção 2.2.2.3, o resultado do algoritmo *k-means* é sensível à escolha dos centroides iniciais. A má escolha desta configuração inicial dos centroides pode fazer com que o algoritmo atinja um mínimo local durante a minimização da função objetivo. Neste trabalho, a mesma estratégia para a escolha dos centroides iniciais foi utilizada em todas as execuções que envolvem o algoritmo *k-means*. Esse tipo de abordagem garante que as imagens que formaram as bases de conhecimento e teste passaram pelo mesmo segmentador gerando resultados comparáveis entre si.

O centroide inicial é definido obtendo-se a cor média da imagem. Apenas 1 centroide inicial é definido, dado que o algoritmo inicia sua execução com 1 cluster e então cresce o número de clusters a cada iteração a fim de achar o melhor número de clusters para a execução.

Outros trabalhos utilizaram o algoritmo *k-means* para realizar a segmentação de imagens. Em alguns destes trabalhos foram acrescentados outros tipos de

segmentação, como por exemplo, em Pouladzadeh, Shirmohammadi e Arici, (2013), onde além da segmentação por cores utilizando o algoritmo *k-means*, foi utilizado segmentação de textura utilizando a técnica de *Gabor Filter* e estimando a média do desvio padrão da energia da imagem filtrada.

O presente trabalho utilizou apenas o método de segmentação por semelhança de cores utilizando o algoritmo *k-means* sem nenhuma outra técnica auxiliar para realizar os testes, a fim de verificar seu comportamento neste domínio de aplicação.

Como descrito na secção 2.2.2.3, o *k-means* utiliza um vetor de centroides $C = c_1, c_2, \dots, c_k$, onde k é o número de *clusters* a serem agrupados. Cada centroide c_1, c_2, \dots, c_k é ajustado a cada iteração até que haja convergência, isto é, não haja realocação de elementos entre os *clusters*.

Os valores dos centroides c_1, c_2, \dots, c_k obtidos no fim do processamento, são os padrões que representam seus respectivos clusters, isto é, cada centroide $c_i \in C$ é um representante do *cluster* i .

Como o algoritmo utilizado foi adaptado para realizar os agrupamentos baseados em padrões RGB, os centroides resultantes em C , são cada um, vetor RGB, isto é, uma 3-upla (r, g, b) , que representa a coloração média do *cluster* c_i .

Especialista Humano: Geradas as imagens segmentadas e os seus respectivos centroides, um especialista humano observa as imagens segmentadas e associa seus centroides a base de dados correspondente a sua classe, isto é, o especialista verifica a primeira imagem segmentada, rotula o centroide com um rótulo apropriado, isto é:

1. Arroz
2. Feijão
3. Carne de Panela

Logo em seguida, o especialista insere o centroide (*representante da imagem*) rotulado, na base de dados.

Base de Conhecimento: Resultado dos centroides rotulados adquiridos seguindo as etapas anteriores.

Foram utilizadas poucas imagens para a criação da base de dados. Isso porque como descrito acima, o algoritmo *k-means*, quando finaliza sua execução, gera centroides que são os representantes de seus *clusters*, isto é, no caso da segmentação por semelhança de cores, o centroide de um padrão arroz, por exemplo, será a cor que representa este cluster. Sendo assim, qualquer centroide de que represente um padrão de arroz, terão seus valores *r*, *g*, *b* semelhantes.

Nesse sentido, não é necessário utilizar várias imagens de arroz parecidas. Porém, a fim de obter uma melhor generalização da base de dados, pode-se formar uma base de dados utilizando imagens que possuem porções de arroz em diferentes situações, isto é, grãos muito próximos e grãos mais separados; arroz com uma coloração um pouco diferente, até mesmo com diferentes configurações de iluminação; todas essas características influenciam na apresentação final do centroide.

Neste trabalho, foram utilizadas 6 exemplos de alimentos para cada uma das classes de interesse, isto é, arroz, feijão e carne de panela. Exemplos com as variações descritas acima, a fim de se obter certo poder de generalização na base de dados.

3.1.2 O Processo de Detecção

O processo de detecção utiliza-se da base de conhecimento previamente obtida pelo processo de aprendizagem descrito na secção 3.1.1. Uma das propostas desse trabalho é utilizar o mesmo algoritmo *k-means* tanto no processo de aprendizagem, para a segmentação e rotulação das imagens, quanto no processo de classificação. O processo é proposto em detalhes no fluxograma da Figura 10.

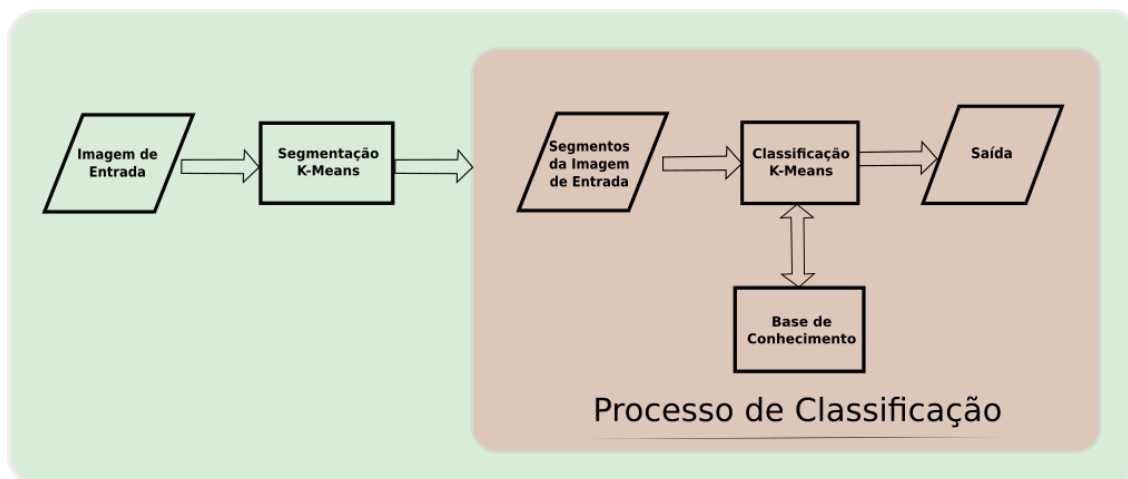


Figura 10 - Processo de Detecção.

Imagem de Entrada: Imagem de que se deseja saber os itens que a compõe.

Segmentação *K-Means*: Semelhantemente ao processo de aprendizagem, a imagem passa pela segmentação por semelhança de cores utilizando o mesmo algoritmo *k-means* adaptado para trabalhar com imagens.

De modo geral o processo de detecção segue o pseudocódigo apresentado no Quadro 2:

Quadro 2 - Pseudocódigo Algoritmo *K-Means*.

S (Vetor de segmentos gerados para a imagem de entrada)
 $C = c_1, c_2, \dots, c_n$, onde $n = |S|$ (Vetor de classificação dos segmentos de S)

Inicializar IMG_{RGB} = Imagem de entrada com pixels em formato *RGB*

I = Vetor de pixels de IMG_{RGB}

$S \leftarrow$ Segmentar(I) (Realiza a segmentação da imagem utilizando o algoritmo *k-means*)

Para Cada $s_i \in S$

$c_i \leftarrow$ Classifica(s_i)

Fim Para

3.1.3 O Processo de Classificação

Os k -centroides retornados pela etapa de segmentação, são enviados para o processo de classificação, onde é realizada a comparação dos centroides obtidos nesta etapa de segmentação com os centroides armazenados na base de conhecimento.

Um problema enfrentado na comparação de cores é que quando dois indivíduos olham para um mesmo objeto, eles podem ter percepções diferentes sobre a coloração do objeto. Essa diferença gera divergências quanto à percepção de diferenças entre cores de um objeto, olhando pela perspectiva de um indivíduo humano.

Existem vários espaços de cores diferentes cuja utilização depende da aplicação. Dentre os espaços de cores mais conhecidos pode-se destacar o RGB (*Red Green Blue*), que é um espaço de cores de fácil implementação e muito utilizado em sistemas que utilizam um monitor CRT (*Cathode Ray Tube*), para mostrar imagens. Apesar de sua facilidade de implementação, este espaço de cores não é linear com a percepção visual humana (FORD e ROBERTS, 1998).

Outro espaço de cores conhecido é o CMY(K) (*Cyan Magenta Yellow (Black)*), que é um espaço de cores muito utilizado no domínio de impressões. Este espaço de cores também não é linear com a percepção visual humana (FORD e ROBERTS, 1998).

Porém, existem alguns espaços de cores conhecidos que são adaptados a percepção visual humana, que são os casos dos espaços de cores XYZ, YUV, YU'V', $L^*u^*v^*$ e $L^*a^*b^*$, definidos pela *Commission Internationale de l'Eclairage* (CIE).

Esses espaços são similares, porém cada um deles adota variações dos seus antecessores de maneira a corrigir ou melhorar propriedades que influenciam percepção visual de um humano sobre esses espaços de cores (FORD e ROBERTS, 1998).

O espaço de cores $L^*a^*b^*$ foi escolhido por ser muito utilizado na literatura no que se diz respeito à comparação de cores em imagens e por possuir métricas de similaridade próprias adaptadas para esse espaço de cores

Neste espaço, a^* e b^* são as coordenadas da cor no espaço, onde a^* representa a intensidade e vermelho, se $a^* > 0$, ou a intensidade de verde, se $a^* < 0$. Do mesmo modo, b^* representa a intensidade e amarelo, se $b^* > 0$, ou a intensidade de azul, se $b^* < 0$, e L^* é o nível de luminosidade, onde $0 \leq L \leq 100$.

Dado que o espaço de cores $L^*a^*b^*$ se mostra mais adequado à percepção humana, a comparação entre as cores é feita utilizando este espaço.

Para realizar a comparação entre cores, espaço $L^*a^*b^*$ possui métricas próprias. Essas métricas foram desenvolvidas ao longo dos anos, sendo aprimoradas a cada versão, acrescentando e/ou removendo componentes do cálculo a fim de deixar a métrica de similaridade entre as cores definidas neste espaço mais precisas e adequadas à visão humana.

No presente trabalho, duas das últimas métricas de distância adaptadas para este espaço de cores foi utilizada para fins de comparação. São estas as métricas, *Delta-E 94*, representada por ΔE_{94}^* e *Delta-E 2000*, representada por ΔE_{00}^* .

Formalmente, função ΔE_{94}^* é definida como:

$$\Delta E_{94}^*(L_1^*, L_2^*) = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2}$$

Onde,

$$\Delta L^* = L_1^* - L_2^*$$

$$C_1^* = \sqrt{a_1^{*2} + b_1^{*2}}$$

$$C_2^* = \sqrt{a_2^{*2} + b_2^{*2}}$$

$$C_{ab}^* = \sqrt{a_2^{*2} + b_2^{*2}}$$

$$\Delta C_{ab}^* = C_1^* - C_2^*$$

$$\Delta H_{ab}^* = \sqrt{\Delta E_{ab}^2 - \Delta L^{*2} - \Delta C_{ab}^{*2}} = \sqrt{\Delta a^{*2} - \Delta b^{*2} - \Delta C_{ab}^{*2}}$$

$$\Delta a^* = a_1^* - a_2^*$$

$$\Delta b = b_1^* - b_2^*$$

$$\begin{aligned}
S_L &= 1 \\
S_c &= 1 + K_1 C_1^* \\
S_h &= 1 + K_2 C_1^* \\
K_L &= \begin{cases} 1 & \text{para aplicações padrão} \\ 2 & \text{para aplicação têxtil} \end{cases} \\
K_1 &= \begin{cases} 0.045, & \text{se aplicação gráfica} \\ 0.048, & \text{se aplicação têxtil} \end{cases} \\
K_2 &= \begin{cases} 0.015, & \text{se aplicação gráfica} \\ 0.014, & \text{se aplicação têxtil} \end{cases} \\
K_C &= 1 \\
K_H &= 1
\end{aligned}$$

Na métrica ΔE_{00}^* foram adicionadas cinco correções com relação à ΔE_{94}^* . Por questões de complexidade, a formalização da função ΔE_{00}^* será omitida. A formalização desta métrica, tais como as diferenciações com relação à métrica ΔE_{94}^* são feitas em, (SHARMA, WU e DALAL, 2005).

Segundo Entendendo o Espaço de Cor $L^*a^*b^*$ (2013) o espaço de cores $L^*a^*b^*$ é o mais popular de todos os espaços de cores uniformes, pois este correlaciona consistentemente os valores numéricos de uma cor com a percepção visual.

Com as adaptações feitas no algoritmo *k-means* para trabalhar com imagens, os centroides gerados como resultados da etapa de segmentação são retornados em forma de vetores do espaço de cores *RGB*. Sendo assim, ao se realizar o cálculo de similaridade dos centroides obtidos nesta etapa com os centroides da base de dados, a conversão destes centroides para o espaço de cores $L^*a^*b^*$ se faz necessário a fim de que a comparação entre as cores possa ser feita de forma consistente com a percepção visual humana.

De maneira geral, o processo de classificação segue o pseudocódigo apresentado no Quadro 3:

Quadro 3 - Pseudocódigo Processo de Classificação.

Entrada s (Segmento de Imagem a ser classificada)
c (Centroide representante do segmento s)
metricaSimilaridade = { DeltaE94, DeltaE2000}

Carrega DBArroz = {r₁, r₂, ... }
DBFeijao = {b₁, b₂, ... }
DBCarne = {f₁, f₂, ... }

diferencasParaArroz(i) = metricaSimilaridade(c, r_i),
para todo i = {1, 2, ... |DBArroz|}

diferencasParaFeijão (i) = metricaSimilaridade(c, b_i),
para todo i = {1, 2, ... |DBFeijao|}

diferencasParaCarne(i) = metricaSimilaridade(c, f_i),
para todo i = {1, 2, ... |DBCarne|}

diferencaMediaParaArroz = media(diferencasParaArroz)
diferencaMediaParaFeijão = media(diferencasParaFeijão)
diferencaMediaParaCarne = media(diferencasParaCarne)

menorDiferenca = menorValor(diferencaMediaParaArroz,
diferencaMediaParaFeijão, diferencaMediaParaCarne)

Se menorDiferenca == diferencaMediaParaArroz **então**
 Classe = Arroz

Se não
 Se menorDiferenca == diferencaMediaParaFeijão **então**
 Classe = Feijão

Se Não
 Classe = Carne de Panela

Fim Se

Fim Se

Os resultados obtidos pelos testes gerados serão descritos no Capítulo 4.

3.2 Implementação

3.2.1 Ferramenta

Este trabalho foi desenvolvido utilizando a ferramenta MATLAB, versão estudante R2014b. O MATLAB é uma linguagem de programação de alto nível proprietária da empresa *MathWorks* e utilizada para realizar processamento matemático nas mais diversas áreas como processamento de sinais, computação financeira, processamento de imagens, inteligência artificial, entre outras.

3.2.2 A Implementação do Algoritmo *K-Means*

A implementação do algoritmo *k-means* deste projeto utilizou a biblioteca *Adaptive kmeans Clustering for Color and Gray Image*, uma biblioteca que implementa o algoritmo *k-means* adaptado para trabalhar com imagens de forma que não é necessário especificar o número de *clusters*. Esta biblioteca foi desenvolvida por Ankit Dixit e pode ser encontrada em (DIXIT, 2014).

O algoritmo é chamado pela função *adaptcluster_kmeans(img)*, onde *img*, é uma matriz de pixels de imagem, na forma $[I]_{ij}$, onde i, j representam o número de linhas, colunas da imagem, e cada $p_{ij} \in [I]_{ij}$ é um pixel representado no espaço de cores RGB, isto é, cada $p_{ij} \in [I]_{ij}$ é uma 3-upla onde o primeiro elemento representa a intensidade de vermelho, o segundo a intensidade de verde, e o quarto a intensidade de azul.

3.2.3 Segmentação de Cores Utilizando o *K-Means*

O algoritmo de segmentação recebe o caminho para uma imagem; realiza o agrupamento de cores utilizando o algoritmo *k-means*; configura os segmentos de imagem gerados por esse agrupamento e por fim apresenta os segmentos para o especialista humano.

O algoritmo MATLAB para a segmentação de imagens é apresentado no Quadro 4.

Quadro 4 - Algoritmo MATLAB para Segmentação de Cores Utilizando o *K-Means*.

```
%ESTA FUNÇÃO RECEBE UM CAMINHO PARA UMA IMAGEM E REALIZA A SEGMENTAÇÃO
DA MESMA.
function segmentar(imgPath)
    %OBTÉM A MATRIZ DE PIXELS RGB DA IMAGEM DE ENTRADA.
    img = imread(imgPath);

    %OBTÉM A LARGURA, ALTURA E A PROFUNDIDADE DOS PIXELS DA IMAGEM.
    COMO A IMAGEM FOI LIDA NO FORMATO RGB A PROFUNDIDADE DOS PIXELS
    SERÁ SEMPRE 3, I.E, R, G, B.
    [imgHeight, imgWidth, imgColorChannelSize] = size(img);

    %EXECUTA O ALGORITMO K-MEANS QUE RETORNA A MATRIZ DE PIXELS
    ROTULADOS DA IMAGEM E OS CENTROIDES REPRESENTANTES DE CADA
    CLUSTER.
    [clusterimage, clustercenter] = adaptcluster_kmeans(img);

    [nrSegments, tmp] = size(clustercenter);

    %INICIA A CONFIGURAÇÃO DA MOLDURA ONDE SERÃO EXIBIDAS A IMAGEM
    ORIGINAL E OS SEGMENTOS DA IMAGEM.
    nrFrames = (nrSegments + 1);
    nrWindowRows = ceil(sqrt(nrFrames));
    nrWindowCols = ceil(sqrt(nrFrames));

    %APRESENTA A IMAGEM ORIGINAL
    subplot(nrWindowRows, nrWindowCols, 1), imshow(img),
    title('Imagem Original');

    cont = 1;

    %PARA CADA CLUSTER GERADO PELO K-MEANS, CONFIGURA E MOSTRA O
    SEGMENTO DA IMAGEM CORRESPONDENTE AO CLUSTER.
    for segmentIndex = 1:nrSegments
        imgResult = zeros(imgHeight, imgWidth, imgColorChannelSize);
        for row = 1:imgHeight
            for col = 1:imgWidth
                %OS PIXELS QUE NÃO PERTENCEM AO SEGMENTO ATUAL SÃO DEFINIDOS
                COMO PRETO.
                if clusterimage(row, col) == segmentIndex
                    imgResult(row, col, :) = img(row, col, :);
                else
                    imgResult(row, col, :) = [0, 0, 0];
                end
            end
        end
        imgResult = uint8(imgResult);

        %MOSTRA O SEGMENTO DE IMAGEM GERADO PELO CLUSTER ATUAL.
        subplot(nrWindowRows, nrWindowCols, cont + 1),
        imshow(imgResult), title(strcat('Segmento ', num2str(cont)));
        cont = cont + 1;
    end
end;
end;
```

3.2.4 Algoritmo Classificador

O algoritmo classificador de imagens utilizado neste projeto, chamado de *classifier*, recebe o caminho de uma imagem e a métrica de similaridade de cores a ser utilizada; realiza a segmentação da imagem; e para cada segmento, classifica-o de acordo com a métrica de similaridade de cores definida no parâmetro de entrada.

O algoritmo MATLAB do classificador é apresentado no Quadro 5.

Quadro 5 - Algoritmo MATLAB Classificador.

```
function [resultMap] = classifier(imgPath, deltaEMethod)
    addpath(' ../AdaptClusterKMeans/ ');

    %CARREGA A IMAGEM DEFINIDA NO PARÂMETRO DE ENTRADA.
    img = imread(imgPath);

    %OBTÉM O NÚMERO DE LINHAS, COLUNAS E A PROFUNDIDADE DE PIXELS DA
    IMAGEM. COMO SE TRATA DE IMAGEM NO PADRÃO RGB, A IMAGEM SEMPRE
    TERÁ 3 PIXELS DE PROFUNDIDADE.
    [imgHeight, imgWidth, imgColorChannelSize] = size(img);

    %REALIZA O AGRUPAMENTO POR CORES UTILIZANDO O ALGORITMO K-MEANS.
    [clusterimage, clustercenter] = adaptcluster_kmeans(img);

    %CARREGA A BASE DE CONHECIMENTO PARA CADA UMA DAS CLASSES
    (ARROZ, FEIJÃO E CARNE DE PANELA) TAIS COMO A QUANTIDADE DE
    PADRÕES EM CADA UMA.
    load('./Databases/riceCentroidDatabase.mat');
    load('./Databases/beanCentroidDatabase.mat');
    load('./Databases/fleshCentroidDatabase.mat');

    [nrRiceDatabaseRows, nrRiceDatabaseCols] =
size(riceCentroidDatabase);
    [nrBeanDatabaseRows, nrBeanDatabaseCols] =
size(beanCentroidDatabase);
    [nrFleshDatabaseRows, nrFleshDatabaseCols] =
size(fleshCentroidDatabase);

    %INICIA AS DISTÂNCIAS ENTRE O CENTROIDE REPRESENTANTE DE UM
    SEGMENTO  $s_i$  E UM PADRÃO  $p_j$  DA BASE DE CONHECIMENTO.
    differencesFromRice = zeros(nrRiceDatabaseRows, 1);
    differencesFromBean = zeros(nrBeanDatabaseRows, 1);
    differencesFromFlesh = zeros(nrFleshDatabaseRows, 1);

    [nrCentersRows, nrCentersCols] = size(clustercenter);

    %VETOR ONDE SERÃO ABRIGADAS AS CLASSES CORRESPONDENTES A CADA UM
    DOS VETORES.
    resultMap = zeros(1, nrCentersRows);
```

```

    %PARA CADA CLUSTER RETORNADO PELO K-MEANS, REALIZA A CLASSIFICAÇÃO
    DO CENTROIDE CORRESPONDENTE.
    for i = 1:nrCentersRows
        %CALCULA AS DISTÂNCIAS DO CENTROIDE ATUAL PARA CADA UM DOS
        PADRÕES DA CLASSE ARROZ DA BASE DE CONHECIMENTO.
        for iRice = 1:nrRiceDatabaseRows
            differencesFromRice(iRice, 1) = de94(clustercenter(i,
            :, :, :), riceCentroidDatabase(iRice, :, :, :),
            deltaEMethod);
        end

        %CALCULA AS DISTÂNCIAS DO CENTROIDE ATUAL PARA CADA UM DOS
        PADRÕES DA CLASSE ARROZ DA BASE DE CONHECIMENTO.
        for iBean = 1:nrBeanDatabaseRows
            differencesFromBean(iBean, 1) = de94(clustercenter(i,
            :, :, :), beanCentroidDatabase(iBean, :, :, :), deltaEMethod);
        end

        %CALCULA AS DISTÂNCIAS DO CENTROIDE ATUAL PARA CADA UM DOS
        PADRÕES DA CLASSE ARROZ DA BASE DE CONHECIMENTO.
        for iFlesh = 1:nrFleshDatabaseRows
            differencesFromFlesh(iFlesh, 1) =
de94(clustercenter(i, :, :, :), fleshCentroidDatabase(iFlesh, :, :,
:), deltaEMethod);
        end;

        %CALCULA A DISTÂNCIA MÉDIA PARA CADA UMA DAS CLASSES.
        meanDifferenceFromRice = mean(differencesFromRice);
        meanDifferenceFromBean = mean(differencesFromBean);
        meanDifferenceFromFlesh = mean(differencesFromFlesh);

        %RECUPERA QUAL A CLASSE MAIS PRÓXIMA DO CENTROIDE ATUAL.
        minDifference = min([meanDifferenceFromRice,
        meanDifferenceFromBean, meanDifferenceFromFlesh]);

        if minDifference == meanDifferenceFromRice
            resultMap(1, i) = 1; %CLASSE: ARROZ
        elseif minDifference == meanDifferenceFromBean
            resultMap(1, i) = 2; %CLASSE: FEIJÃO
        else
            resultMap(1, i) = 3; %CLASSE: CARNE DE PANELA
        end
    end
end
end

```

3.2.5 Algoritmo de Teste

O algoritmo de teste é responsável por carregar a base de testes, invocar o processo de classificação, e através dos resultados obtidos, computar o número de acertos para cada uma das classes.

O algoritmo MATLAB para os testes é mostrado no Quadro 6.

Quadro 6 - Algoritmo MATLAB de Testes.

```
%CARREGA A BASE DE TESTES E RECUPERA O NÚMERO DE IMAGENS A SEREM
TESTADAS.
load('./Databases/testDatabase.mat');
[nrTestDatabaseRows, nrTestDatabaseCols] = size(testDatabase);

%%%%% INICIALIZA AS VARIÁVEIS ESTATÍSTICAS %%%%%%
%IMAGENS COM TAXA DE ACERTO MENORES QUE 50%.
classLessThan50 = cell(0,0);

%IMAGENS COM TAXA DE ACERTO ENTRE 50% E 75%.
classBetween50and75 = cell(0, 0);

%IMAGENS COM TAXA DE ACERTO MAIORES QUE 75%.
classBetween75and100 = cell(0, 0);

%NÚMERO DE SEGMENTOS QUE NÃO FORAM ROTULADOS COMO DESCONHECIDO.
nrTotalValidSegments = 0;

%NÚMERO DE SEGMENTOS ROTULADOS PARA CADA UMA DAS CLASSES.
nrTotalValidSegmentsRice = 0;
nrTotalValidSegmentsBean = 0;
nrTotalValidSegmentsFlesh = 0;

%TAXA DE ACERTO DE CADA UMA DAS CLASSES.
nrTotalValidSegmentsRiceAcertRate = 0;
nrTotalValidSegmentsBeanAcertRate = 0;
nrTotalValidSegmentsFleshAcertRate = 0;

%NÚMERO DE SEGMENTOS QUE FORAM ROTULADOS COMO DESCONHECIDO.
nrTotalNonValidSegments = 0;

%PARA CADA UMA DAS IMAGENS NA BASE DE TESTES, REALIZA OS TESTES
REALIZA OS CÁLCULOS ESTATÍSTICOS.
for i = 1:nrTestDatabaseRows
    %RECUPERA A IMAGEM i DA BASE DE TESTES.
    testImage = testDatabase(i, 1);
    testImage = testImage{1};
    testImage = strcat('./Databases/Teste/', testImage);
    expectedResult = testDatabase(i, 2);
```

```

expectedResult = expectedResult{1};

%REALIZA A CLASSIFICAÇÃO DA IMAGEM INVOCANDO O ALGORITMO DE
CLASSIFICAÇÃO.
classificationResult = classifier(testImage, 'cie00');

[tmp, classificationResultSize] = size(classificationResult);
acertRate = 0;
classificationResultValidSize = 0;

%PARA CADA SEGMENTO DA IMAGEM CLASSIFICADA CALCULA O NÚMERO DE
ACERTOS DO CLASSIFICADOR.
for i = 1:classificationResultSize

    %SE O SEGMENTO NÃO FOI ROTULADO COMO DESCONHECIDO, ENTÃO
    VERIFICA SE O CLASSIFICADOR ACERTOU A CLASSIFICAÇÃO.
    if expectedResult(i) ~= 0
        %INCREMENTA O NÚMERO DE SEGMENTOS QUE NÃO FORAM ROTULADOS
        COMO DESCONHECIDOS PARA A IMAGEM ATUAL.
        classificationResultValidSize =
classificationResultValidSize + 1;

        %INCREMENTA O NÚMERO DE SEGMENTOS ENTRE TODAS AS IMAGENS
        QUE NÃO FORAM ROTULADOS COMO DESCONHECIDOS.
        nrTotalValidSegments = nrTotalValidSegments + 1;

        %INCRESENTA O NÚMERO DE SEGMENTOS QUE NÃO FORAM ROTULADOS
        COMO DESCONHECIDOS PARA A CLASSE ATUAL.
        if expectedResult(i) == 1
            nrTotalValidSegmentsRice = nrTotalValidSegmentsRice +
1;
        elseif expectedResult(i) == 2
            nrTotalValidSegmentsBean = nrTotalValidSegmentsBean +
1;
        else
            nrTotalValidSegmentsFlesh = nrTotalValidSegmentsFlesh
+ 1;
        end

        %VERIFICA SE O CLASSIFICADOR ACERTOU A CLASSIFICAÇÃO DO
        SEGMENTO ATUAL, E CASO TENHA ACERTADO CALCULA OS VALORES
        ESTATÍSTICOS PARA CADA UMA DAS CLASSES.
        if classificationResult(i) == expectedResult(i)
            if expectedResult(i) == 1
                nrTotalValidSegmentsRiceAcertRate =
nrTotalValidSegmentsRiceAcertRate + 1;
            elseif expectedResult(i) == 2
                nrTotalValidSegmentsBeanAcertRate =
nrTotalValidSegmentsBeanAcertRate + 1;
            else
                nrTotalValidSegmentsFleshAcertRate =
nrTotalValidSegmentsFleshAcertRate + 1;
            end
        end
    end
end

```

```
        acertRate = acertRate + 1;
    end
else
    nrTotalNonValidSegments = nrTotalNonValidSegments + 1;
end
end;

%CALCULA A TAXA DE ACERTO DA CLASSIFICAÇÃO PARA A IMAGEM ATUAL.
acertRate = (acertRate / classificationResultValidSize) * 100.00;

%INSERE O RESULTADO DA CLASSIFICAÇÃO ATUAL EM UMA DAS FAIXAS DE
ACERTO DEFINIDAS.
if acertRate <= 50
    [classSize, ~] = size(classLessThan50);
    classLessThan50(classSize + 1, :) = {testImage,
expectedResult, classificationResult, acertRate};
elseif acertRate > 50 && acertRate <= 75
    [classSize, ~] = size(classBetween50and75);
    classBetween50and75(classSize + 1, :) = {testImage,
expectedResult, classificationResult, acertRate};
elseif acertRate > 75 && acertRate <= 100
    [classSize, ~] = size(classBetween75and100);
    classBetween75and100(classSize + 1, :) = {testImage,
expectedResult, classificationResult, acertRate};
end
end
```

4

Resultados

4.1 Segmentação *K-Means*

Como descrito acima, durante a formação da base de testes, foram utilizadas imagens que contenham três dos itens comuns na refeição brasileira, arroz, feijão e carne de panela. Foi descrito também que estes tres itens são capazes de mostrar vantagens e desvantagens de se usar segmentação por cor na classificação de imagens de refeições.

De modo geral os resultados foram satisfatórios. A segmentação utilizando o algoritmo *k-means* adaptado para tabalhar com imagens e número de clusters automático mostrou-se rápido e eficaz, mesmo considerando que o número de padrões no vetor de entrada do algoritmo *k-means* é grande, devido ao fato de trabalharmos com imagens.

Porém, de modo geral a maioria das imagens se mostraram aptas a confundir o algoritmo, gerando segmentos intermediários, isto é, segmentos que representam apenas ruídos pertencentes à porções de mais de um ítem de refeição ao mesmo tempo.

Outros tipos de imagens confundiram o algoritmo quanto a diferenciação entre porções, ou seja, fizeram o algoritmo segmentar por exemplo, arroz e carne de panela como se fossem um único ítem.

E por fim, alguns tipos de imagens geraram o que foi denominado segmentos vazios, que seriam segmentos que continham alguns poucos e minúsculos fragmentos de imagens de uma ou mais porções em geral.

Exemplos dos resultados com explicações de cada caso são mostrados abaixo:



Figura 11 - Resultado da Segmentação - Arroz com feijão - Exemplo 1 - Imagem Original.

Este é um bom exemplo de um prato de arroz e feijão que gera boa segmentação. Neste caso, a luminosidade da imagem é baixa.

Como mostrado, nas Figuras 12 e 14, os itens arroz e feijão foram bem segmentados, com apenas alguns ruídos, porém algo não atrapalha o processo de classificação.

Porém, a imagem também gerou o que foi descrito como segmento intermediário, mostrado na Figura 13, que neste caso abrangeu os pixels que fazem parte da linha de divisão entre arroz e feijão, em união com os pixels que fazem parte do ruído gerados nas Figuras 12 e 14.



Figura 12 - Resultado da Segmentação - Exemplo 1 - Arroz com feijão - Segmento 1.

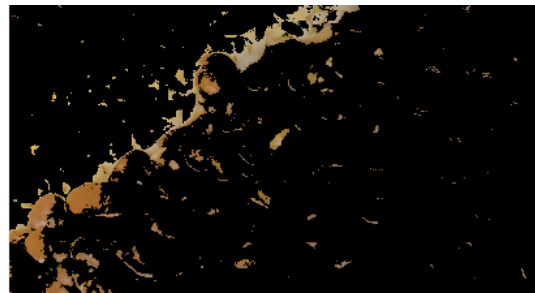


Figura 13 - Resultado da Segmentação - Arroz com feijão - Exemplo 1 - Segmento 2.



Figura 14 - Resultado da Segmentação - Arroz com feijão – Exemplo 1 - Segmento 3.

O exemplo abaixo (Figura 15) mostra um prato semelhante. Neste caso, a luminosidade da imagem é alta.



Figura 15 - Resultado da Segmentação - Arroz com feijão - Exemplo 2 - Imagem Original.

De modo geral, a imagem segmentou bem o item arroz, Figura 18, porém o feijão foi segmentado em 2 tipos, a parte clara (Figura 16) e a parte escura do feijão (Figura 17), provavelmente por conta do ângulo de iluminação da imagem.

As Figuras 17 e 18, geraram pequenos ruídos, que não são suficientes para atrapalhar o processo de classificação, pois de modo geral, esses ruídos tem cor semelhante a cor predominante do segmento em que ela está inserida, o que faz com que o centroide associado a este segmento não se altere muito por causa do ruído gerado no segmento.



Figura 16 - Resultado da Segmentação - Arroz com feijão – Exemplo 2 - Segmento 1.



Figura 17 - Resultado da Segmentação - Arroz com feijão – Exemplo 2 - Segmento 2.



Figura 18 - Resultado da Segmentação - Arroz com feijão – Exemplo 2 - Segmento 3.

Um terceiro caso, mais complexo, é mostrado logo abaixo, nas Figuras 19 a 22.



Figura 19 - Resultado da Segmentação - Feijão, Farinha de Milho, Couve, Arroz, e Carne de Panela – Imagem Original.

Neste caso, a imagem expõe vários itens – feijão, farinha de milho, couve, arroz e carne de panela. Tecnicamente, são imagens de cores bem distintas, exceto o feijão, carne de panela e couve, que possuem cores semelhantes e se diferenciam mais na textura que em cor.

O algoritmo proposto por este trabalho baseia-se na segmentação por cores, por esse motivo, o algoritmo confunde facilmente itens diferentes que possuem cores semelhantes. O resultado pode ser visto nas Figuras 20 a 22.



Figura 20 - Resultado da Segmentação - Feijão, Farinha de Milho, Couve, Arroz, e Carne de Panela - Segmento 1.

A Figura 20, mais uma vez apresenta o feijão sendo segmentado em duas partes, a parte escura, e a parte com mais brilho, que representa as marcas negras (ruídos). O conteúdo dessas marcas negras é mostrado na Figura 21 como pertencendo ao mesmo segmento da farinha de milho. Como a Figura 21 mostra, na farinha de milho a cor predominante é o amarelo, que é uma cor clara, por esse motivo o algoritmo colocou o conteúdo das marcas negras do feijão, Figura 20, como pertencendo à farinha de milho (Figura 21).

O resultado da segmentação da carne, na Figura 20, mostra uma imagem mais inteira, ou seja, apenas minúsculos ruídos. Esses minúsculos ruídos se devem ao fato de na imagem original (Figura 19), as gotículas de gordura da porção de carne refletem a iluminação do local, criando no segmento desta porção, minúsculos pontos de luz, que neste caso, são segmentados como pertencentes ao segmento de arroz, Figura 22. Isso pelo fato de a cor predominante do arroz ser branca, similar ao ponto de brilho gerado na porção de carne.

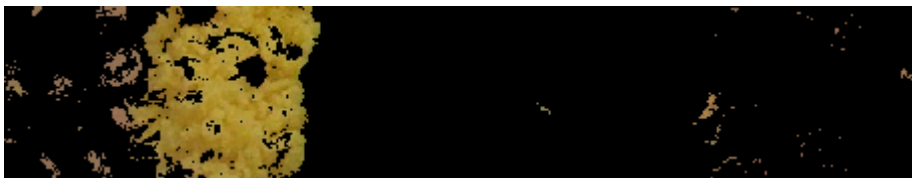


Figura 21 - Resultado da Segmentação - Feijão, Farinha de Milho, Couve, Arroz, e Carne de Panela - Segmento 2.

Um caso semelhante ocorre na Figura 21, que mostra o segmento que corresponde à farinha de milho. Na imagem original, Figura 19, dois grãos de feijão estão no meio da farinha de milho. A maior parte do corpo desses grãos aparecem na Figura 20, onde o feijão está segmentado.

Porém a farinha de milho, assim como o feijão, possui partes mais claras e partes mais escuras, mas ao contrário do feijão, essa diferença de brilho não se dá predominantemente por conta da iluminação, e sim do sombreamento gerado pelos farelos de farinha de milho, que são irregulares quanto à sua forma.

Por existir essa diferença na intensidade de brilho, o mesmo problema ocorrido na Figura 20, com o feijão, ocorre na Figura 21 com a farinha de milho, onde as regiões da farinha de milho que são mais claras aparecem na Figura 22 como pertencentes ao segmento que representa arroz.



Figura 22 - Resultado da Segmentação - Feijão, Farinha de Milho, Couve, Arroz, e Carne de Panela - Segmento 3.

4.2 Classificação *K-Means*

4.2.1 Formação da Base de Testes

A base de testes foi formada por um total de 30 imagens contendo os alimentos-alvo deste trabalho, a saber, arroz, feijão e carne de panela. Essas imagens geraram 76 segmentos. Essas imagens continham combinações de arroz, feijão e carne de panela entre si e algumas delas, combinando com outros alimentos. A Figura 23 mostra um fluxograma que descreve a formação da base de testes.

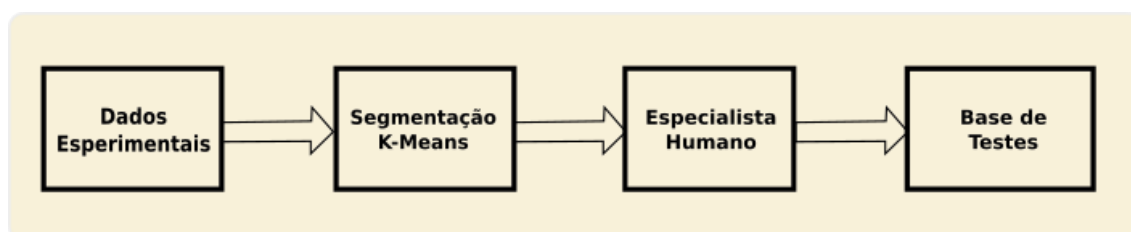


Figura 23 - Formação da Base de Testes.

A formação da base de testes segue o mesmo princípio da formação da base de conhecimento, apresentada na seção 3.1.1.2, diferenciando-se apenas nos dados que são armazenados.

Quando o especialista humano visualiza as imagens resultantes da etapa de segmentação, ele classifica cada um desses segmentos adequadamente e então, armazena o nome da imagem para futuro carregamento dos testes, e quais são as

classes associadas a cada um desses segmentos, isto é, arroz, feijão ou carne de panela. Quando o especialista humano se depara com um segmento difícil de distinguir, ou um segmento que contenha porções de mais de um alimento, então ele o classifica como desconhecido. Neste trabalho, os segmentos que não são classificados como desconhecidos serão chamados de segmentos válidos.

Quando uma imagem é carregada no processo de testes, e a imagem é segmentada, o i -ésimo centroide que representa o i -ésimo segmento da imagem é comparado com o rótulo desse segmento, se o segmento estiver rotulado como desconhecido, então a classificação desse segmento não acontece, e não é contabilizada nos resultados, já que não é um segmento que representa algum alimento de interesse para os testes.

Ao final do processo de formação da base de testes, foram gerados 113 segmentos, sendo destes, 76 segmentos válidos e 37 segmentos desconhecidos. Dentre os segmentos válidos, 32 eram segmentos de arroz, 22 de feijão e 22 de carne de panela.

4.2.2 Resultado da Classificação

De modo geral, o processo de classificação apresentou bons resultados, dado que o classificador acertou em 15 das 30 imagens utilizadas nos testes, mais de 75% dos alimentos que compunham a refeição, como mostrado no gráfico representado pela Figura 24.

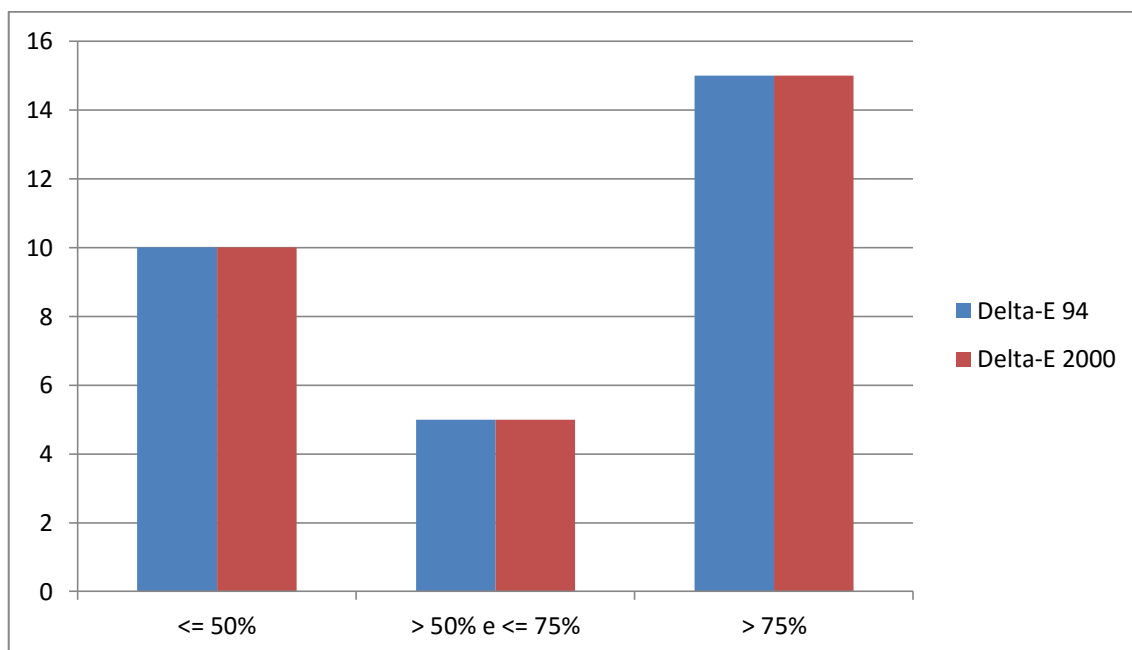


Figura 24 - Taxa de Acertos do Classificador.

De modo geral as imagens que compõe o conjunto que obteve uma taxa de acerto acima dos 75%, não são necessariamente imagens que obtiveram uma boa segmentação uma boa segmentação, isto é, uma separação perfeita entre cada um dos alimentos que compõe aquela imagem, mas imagens que gerarm segmentos de qualidade, ou sejam segmentos bem distinguíveis de modo que o especialista humano poderia facilmente classifica-los durante a geração da base de conhecimento e/ou testes, como é o caso da segmentação da Figura 25, apresentada nas Figuras 26 a 30.



Figura 25 - Arroz e Carne de Panela – Imagem Original.



Figura 26 - Arroz e Carne de Panela – Segmento 1.

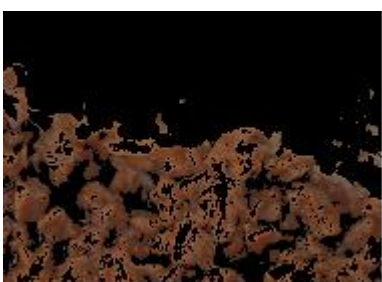


Figura 27 - Arroz e Carne de Panela – Segmento 2.

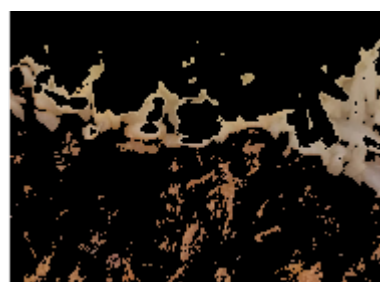


Figura 28 - Arroz e Carne de Panela – Segmento 3.



Figura 29 - Arroz e Carne de Panela – Segmento 4.



Figura 30 - Arroz e Carne de Panela – Segmento 5.

Como apresentado nas Figuras 26 a 30, a Figura 25 não gerou necessariamente uma boa segmentação, pois houveram 3 segmentos cujo conteúdo não é distinguível facilmente, que são os casos das Figuras 26, 28 e 29. Porém as Figuras 27 e 30 são facilmente distinguíveis como sendo porções de feijão e arroz respectivamente.

Esses segmentos bem distinguíveis geram centroides consistentes com suas respectivas classes na base de conhecimento, gerando uma possibilidade maior de acerto pelo classificador.

O gráfico apresentado na Figura 24 também mostra que o classificador acertou em 5 das 30 imagens utilizadas nos testes, entre 50% e 75% dos alimentos que compunham a refeição.

Este conjunto compunha predominantemente imagens que possuíam tanto segmentos bem distinguíveis quanto segmentos que não eram necessariamente descartáveis, mas segmentos que o especialista humano precisaria de um pouco mais de atenção para classifica-los, como é o caso da segmentação da Figura 31, apresentada nas Figuras 32 a 36.



Figura 31 - Arroz e Feijão – Imagem Original.



Figura 32 - Arroz e Feijão – Segmento 1.



Figura 33 - Arroz e Feijão – Segmento 2.

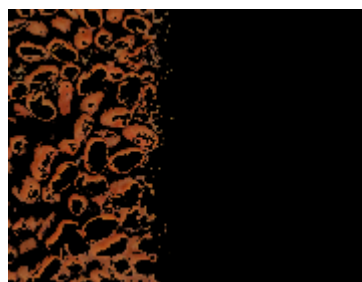


Figura 34 - Arroz e Feijão – Segmento 3.

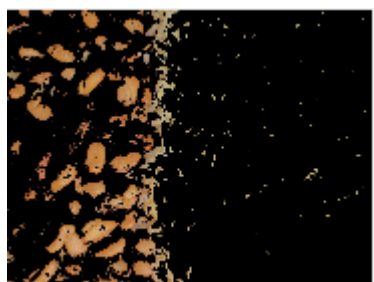


Figura 35 - Arroz e Feijão – Segmento 4.



Figura 36 - Arroz e Feijão – Segmento 5.

A Figura 36 mostra um exemplo de segmento que seria facilmente classificado pelo especialista humano. Um segmento inteiro de arroz, que geraria um centroide consistente com sua respectiva classe na base de conhecimento.

A Figura 33, mostra um segmento que seria prontamente rotulado como desconhecido pelo especialista humano durante a fase de formação da base de testes. Este segmento seria despresado da fase classificação.

Porém as Figruas 32, 34 e 35 mostram segmentos que não são segmentos perfeitos de feijão, mas o especialista humano, com um pouco de atenção, não os rotularia como desconhecido durante a formação da base de testes.

Esse tipo de segmento gera centroides que não são tão consistentes com a sua respectiva classe na base de conhecimento quanto um segmento perfeito como o de arroz da Figura 36 geraria para a classe arroz na mesma base, ou seja, o centroide resultante desse segmento pode estar disperso dos demais centroides de sua classe aumentando a possibilidade de o classificador confundir esse centroide como pertencente uma outra classe mais similar.

Por último, o classificador acertou menos de 50% em 10 das imagens utilizadas na fase de testes.

Este conjunto era formado predominantemente por imagens cujos segmentos eram na sua maioria ruins, ou seja, segmentos que na sua maioria seriam facilmente rotulados como desconhecidos pelo especialista humano durante a formação da base de testes. Como é o exemplo da segmentação da Figura 37 apresentada nas Figuras 38, 39 e 40.



Figura 37 - Arroz e Feijão – Imagem Original.

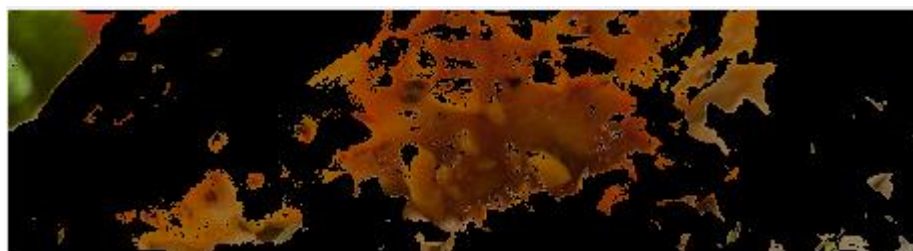


Figura 38- Arroz e Feijão – Segmento 1.



Figura 39- Arroz e Feijão – Segmento 2.



Figura 40- Arroz e Feijão – Segmento 3.

As Figuras 38 e 39 apresentam segmentos que seriam rotulados como desconhecidos pelo especialista humano por se tratar de um pedaço de frango, alimento que não pertence a nenhuma das classes da base de conhecimento.

A Figura 40 apresenta um segmento de arroz com buracos e ruídos pertencentes a porção de frango, o que faz com que o centroide gerado por esse segmento não seja consistente com a base de conhecimento, induzindo o classificador ao erro.

Um ponto importante a ser considerado é que, como descrito na seção 3.1.3, para determinar a qual é a classe mais próxima de um segmento, isto é, determinar a qual classe um segmento pertence, obtém-se primeiro a cor média que representa os centroides para cada classe do banco de testes; e logo em seguida, o centroide que representa o segmento em questão é comparado com cada uma dessas médias; a classe do segmento em questão será aquela que possuir o centroide médio calculado, mais próximo do centroide do segmento em questão.

O problema de se realizar um cálculo como este é que, para toda média, existe uma taxa de variação. No caso de uma média entre cores, no espaço L^*a*b^* a taxa de variação seria descrita geometricamente como o raio do círculo que envolve a dada cor. Um exemplo descrito no plano da Figura 41.

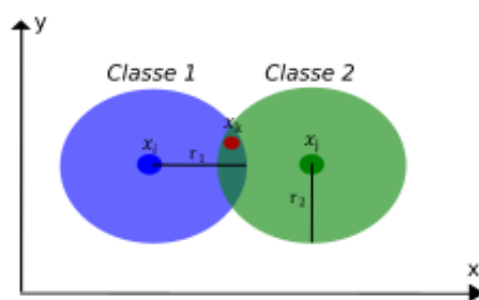


Figura 41 - Situação do padrão x_k em relação às classes 1 e 2.

É observado que se os círculos que envolvem os centroides x_i e x_j interceptarem seus pontos, e o padrão x_k for um centroide que representa uma cor que está dentro dessa intersecção, então durante a fase de classificação, o centroide x_k pode ser rotulado como pertencente à classe um, quando na verdade pertence a classe 2 e vice-versa, pois não é possível fazer uma distinção exclusiva entre as classes 1 e 2.

Pelo fato de na maioria das segmentações, como mostrado na seção 3.2, existirem fragmentos (ruídos) de itens que deveriam pertencer a um segmento, dentro de outro segmento, esses excessos de ruídos, que possuem cores um tanto diferentes do que deveria ser o real representante da classe, causam o desvio do centroide médio da classe para a direção da sua cor, fazendo com que a cor do centroide médio resultante não seja tão precisamente a cor que deveria representar a classe em questão.

Este tipo de erro pode ser minimizado removendo-se os ruídos gerados em cada segmento, de maneira a corrigir o centroide representante do segmento, antes deste ser armazenado na base de dados. O mesmo processo pode ser feito durante a formação da base de testes a fim de se obter centroides mais consistentes com o que eles representam, fazendo com que o classificador possa retornar um resultado mais preciso quanto à classificação do centroide.

A Figura 42 mostra que o classificador acertou mais os segmentos de arroz que os demais segmentos.

Isso se deve ao fato de que na maioria dos casos, o algoritmo *k-means* conseguiu segmentar as porções de arroz com maior precisão que as porções de feijão e carne de panela, o que, como explicado acima, gera centroides mais consistentes com sua respectiva classe na base de conhecimento, não gerando assim centroides muito diferentes um dos outros, facilitando o trabalho do classificador proposto neste trabalho, que classifica um centroide obtendo-se a menor distância deste centroide para os centroides médios de cada uma das classes na base de conhecimento.

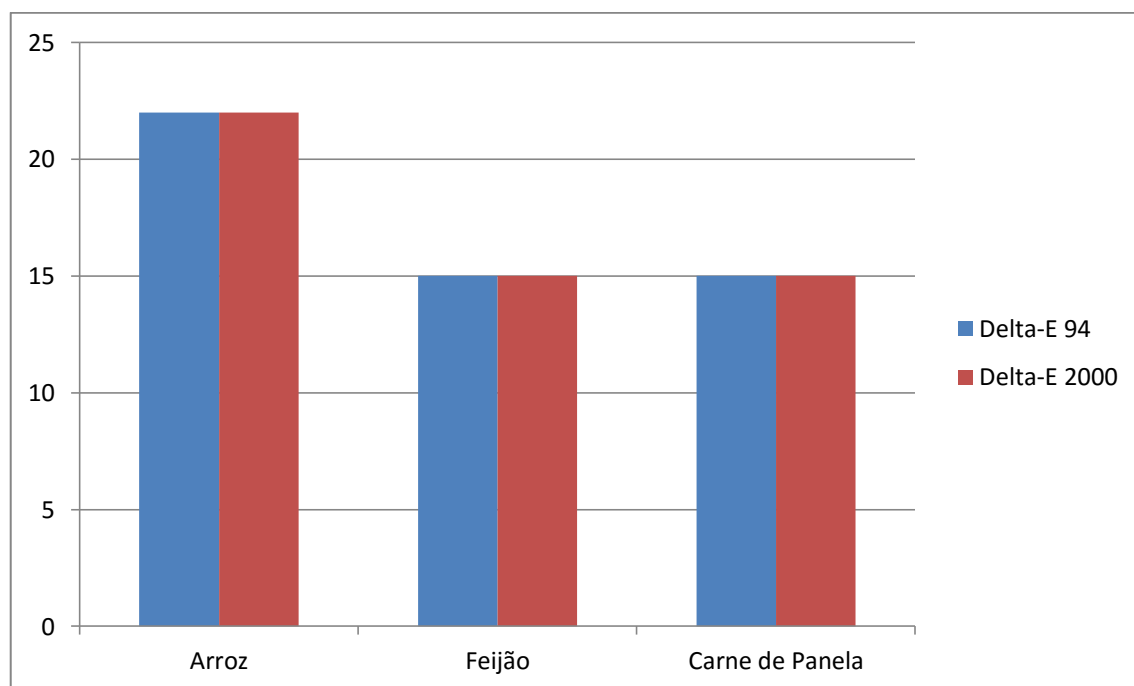


Figura 42 - Número de Acertos dos Segmentos Válidos.

Considerando-se o número de acertos de segmentos pertencente a cada uma das classes e o montante de segmentos de cada uma das classes, o classificador obteve uma média de acertos similar em cada uma das classes, variando décimos de uma classe para outra, como mostrado na Figura 43.

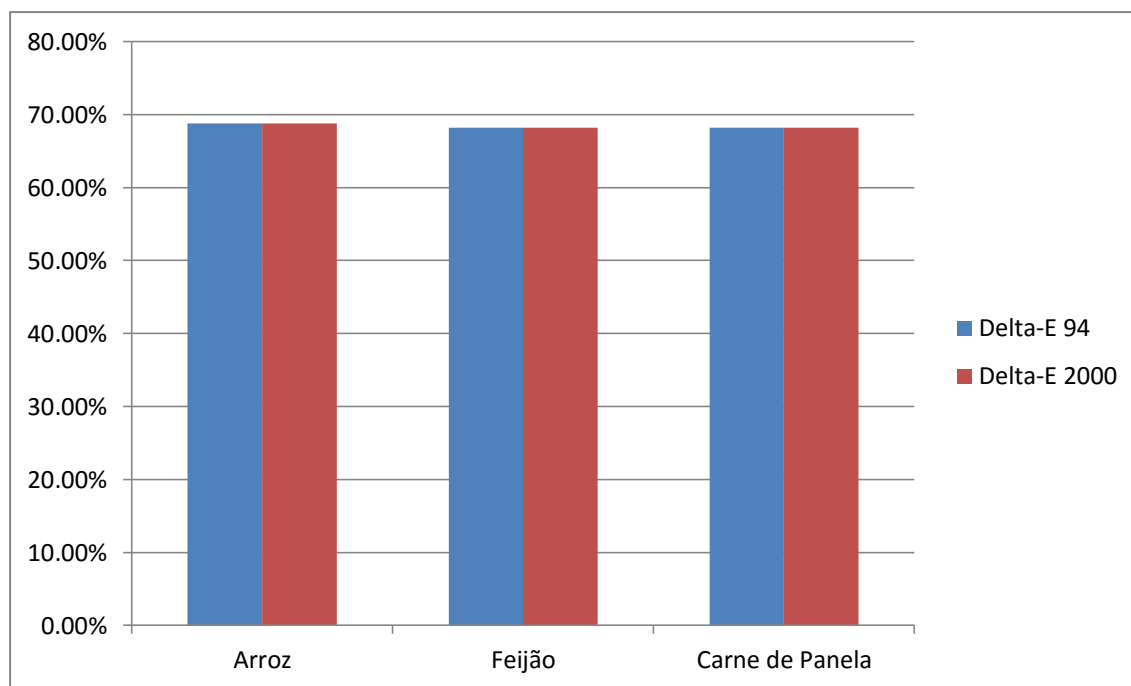


Figura 43 - Taxa de Acerto dos Segmentos Válidos de Cada uma das Classes.

5

Conclusões e Propostas Futuras

5.1 Conclusões

O presente trabalho apresentou um estudo do uso do algoritmo *k-means* na segmentação e na classificação de imagens de refeições, utilizando uma adaptação desse algoritmo para trabalhar com imagens.

De modo geral, o algoritmo se mostrou eficaz quanto à segmentação por cores, fazendo boa parte do trabalho que um bom algoritmo faria. Porém, o algoritmo se mostrou falho em imagens que possuem fragmentos com cores similares nas porções de comida. Isso mostra que a segmentação por cores por si só não alcança a precisão adequada para este tipo de segmentação.

O classificador proposto neste projeto se mostrou de modo geral eficiente, acertando mais de 75% dos alimentos que compunham a imagem quase metade dos casos.

Porém, como apresentado, alguns tipos de imagens geravam falhas de segmentação que criavam muitas vezes, centroides que não eram consistentes com a sua respectiva classe na base de conhecimento. Esse tipo de falha faz com que esses centroides não fiquem tão próximos à maioria de seu grupo na base de conhecimento e conseqüentemente, faz com que o centroide médio que representa a classe em questão seja desviado para a direção do centroide inconsistente.

Dado que o classificador simplesmente calculava a menor distância do centroide do segmento atual para o centroide médio de cada uma das classes, se um centroide médio for bruscamente desviado por vários segmentos inconsistentes, logo o resultado dessa classificação não seria confiável.

5.2 Propostas Futuras

Em trabalhos futuros outros métodos como segmentação por textura, remoção de ruídos, podem ser acrescentados a fim de aprimorar sua eficácia; tratamentos prévios na imagem podem ser feitos a fim de remover reflexos de luz, e outras nuances que podem dificultar o processo de segmentação como um todo.

A adição de um limiar de disparidade pode ser acrescentada durante o cálculo da classificação a fim de não permitir que classes que estejam mais perto do padrão a ser classificado que as demais, porém ainda muito longe do padrão, sejam escolhidas como a classe ao qual o padrão pertence.

Quando esses fatores que reduzem a qualidade do processo de segmentação e classificação das imagens das refeições forem minimizados ou removidos, métodos que segmentem o prato de comida da comida, e o prato do fundo da imagem devem ser acrescentados a fim de o algoritmo se tornar viável para usuários comuns.

6

Referências Bibliográficas

ALPAYDIN, E. **Introduction to Machine Learning**. 2^a. ed. Cambridge, Massachusetts: The MIT Press, 2009. ISBN ISBN: 978-0-262-01243-0.

BRUCE LINDBLOOM.COM. Delta E (CIE 2000). **Bruce Lindbloom.com**, 02 fev. 2009. Disponível em: <http://www.brucelindbloom.com/index.html?Eqn_DeltaE_CIE2000.html>. Acesso em: 2015 dez. 08.

BRUCE LINDBLOOM.COM. Delta E (CIE 1994). **Bruce Lindbloom.com**, 14 dez. 2011. Disponível em: <http://www.brucelindbloom.com/index.html?Eqn_DeltaE_CIE94.html>. Acesso em: 08 dez. 2015.

CHEN, ; LUO, ; PARKER, K. J. Image Segmentation via Adaptive K-Mean Clustering and Knowledge-Based Morphological Operations with Biomedical Applications. **IEEE TRANSACTIONS ON IMAGE PROCESSING**, v. 7, p. 1673-1683, dez. 1998.

DATA Mining Algorithms In R/Clustering/K-Means. **Wiki Books**, 16 Novembro 2015. Disponível em: <https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/K-Means#View>. Acesso em: 09 dez. 2015.

DIXIT, A. Adaptive kmeans Clustering for Color and Gray Image. **MathWorks**, 2014. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/45057-adaptive-kmeans-clustering-for-color-and-gray-image>>. Acesso em: 15 dez. 2015.

ENTENDENDO o Espaço de Cor L*a*b*. **Konica Minolta Brasil**. Disponível em: <<http://sensing.konicaminolta.com.br/2013/11/entendendo-o-espaco-de-cor-lab/>>. Acesso em: 1 dez. 2015.

ESTRADA, F. J.; JEPSON, A. D. Benchmarking Image Segmentation Algorithms. **International Journal of Computer Vision**, v. 85, p. 167-181, Novembro 2009. ISSN ISSN: 0920-5691.

FORD, ; ROBERTS,. Colour Space Conversions. **Charles Poynton**, 11 ago. 1998. Disponível em: <<http://www.poynton.com/PDFs/coloureq.pdf>>. Acesso em: 09 dez. 2015.

GONAL , ; KOHIR , V. Classification of Brain MR Images using Wavelets Texture Features and k-Means Classifier. **Electrical, Electronics, Signals, Communication and**

Optimization (EESCO), 2015 on International Conference, Visakhapatnam, Índia, p. 1 - 5, jan. 2015. ISSN ISBN: 978-1-4799-7676-8.

HRUSCHKA, E. R.; F. EBECKEN,. A Genetic Algorithm for Cluster Analysis. **Intelligent Data Analysis**, Amsterdã, v. 7, p. 15 - 25, jan. 2003. ISSN ISSN: 1088-467X.

JAIN, A. ; DUBES, R. C. **Algorithms for Clustering Data**. Englewood Cliffs: Prentice Hall, 1988.

JAIN, A. ; MURTY, M. ; FLYNN,. Data clustering: A Review. **ACM Computing Surveys (CSUR)**, 3 September 1999. 264-323.

JEYANTHI, P.; JAWAHAR , S. K. Image Classification by K-means Clustering. **Advances in Computational Sciences and Technology**, v. 3, p. 1-8, dez. 2009. ISSN ISSN: 0973-6107.

JOUTOU, ; YANAI,. A food Image Recognition System With Multiple Kernel Learning. **Image Processing (ICIP), 2009 16th IEEE International Conference on**, Cairo, p. 285 - 288, 7-10 Novembro 2009. ISSN ISBN: 978-1-4244-5653-6.

KONICA MINOLTA BRASIL. Entendendo o Espaço de Cor $L^*a^*b^*$. **Konica Minolta Brasil**, 2013. Disponível em: <http://sensing.konicaminolta.com.br/2013/11/entendendo-o-espaco-de-cor-lab/>. Acesso em: 1 dez. 2015.

MACQUEEN,. Some Methods for Classification and Analysis of Multivariate Observations. **5th Berkeley Symposium on Mathematical Statistics and Probability**, Berkeley, v. 1, p. 281-297, 1967. ISSN ISSN: 0097-0433.

POULADZADEH, P. et al. A Novel SVM Based Food Recognition Method for Calorie Measurement Applications. **Proc. International Workshop on Interactive Ambient Intelligence Multimedia Environments, in Proc. IEEE International Conference on Multimedia and Expo**, Melbourne, Austrália, 2012. 495-498.

POULADZADEH, P.; SHIRMOHAMMADI, S.; ALMAGHRABI, R. Measuring Calorie and Nutrition from Food Image. **Instrumentation and Measurement, IEEE Transactions**. 1947-1956.

POULADZADEH, P.; SHIRMOHAMMADI, S.; ARICI, T. Intelligent SVM Based Food Intake Measurement. **Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2013 IEEE International Conference**, 15-17 jul. 2013. 87-92.

SALGADO, et al. **Algoritmos de Agrupamento e Sistemas Especialistas Aplicados no Reconhecimento de Padrões de Escoamento de Gás/Líquido**. Simpósio Brasileiro de Pesquisa Operacional. Bento Gonçalves/RS: [s.n.]. 2010. p. 2275-2286.

SAPNA VARSHNEY, S.; RAJPAL, N.; PURWAR, R. Comparative Study of Image Segmentation Techniques and Object Matching using Segmentation. **International Conference on Methods and Models in Computer Science**, p. 1-6, dez. 2009. ISSN ISBN: 978-1-4244-5051-0.

SEMAAN, et al. Proposta de um Método Baseado em Densidade e Grade para o Problema de Agrupamento Automático. **XVII Simpósio de Pesquisa Operacional e Logística da Marinha**, v. 1, p. 153-162, ago. 2014. ISSN ISSN: 2358-5498.

SHARMA, ; WU, ; DALAL,. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. **Color Research & Applications**, p. 21–30, 2005.

STEINHAUS,. Sur la division des corps matériels en parties. **Bulletin de l'Academie Polonaise des Sciences, Classe 3**, v. 4, p. 801–804, 1967. ISSN ISSN: 0001-4095.

SUBBIAH, ; CHRISTOPHER, C. S. Image Classification Through Integrated K-Means Algorithm. **International Journal of Computer Science Issues(IJCSI)**, v. 9, n. 2, p. 518-524, mar. 2012. ISSN 1694-0814.

USMAN, B. Satellite Imagery Land Cover Classification using K-Means Clustering Algorithm Computer Vision for Environmental Information Extraction, 2013.

VEDALDI, et al. Multiple Kernels for Object Detection. **IEEE 12th International Conference on Computer Vision (ICCV)**, Kyoto, p. 606 - 613, 2009. ISSN ISBN: 978-1-4244-4420-5.

VILLALOBOS, et al. A Personal Assistive System for Nutrient Intake Monitoring. **International ACM Workshop On Ubiquitous Meta User Interfaces**, Nova Iorque, p. 17-22, 2011. ISSN ISBN: 978-1-4503-0993-6.

VILLALOBOS, G. et al. An Image Processing Approach for Calorie Intake Measurement. **Medical Measurements and Applications Proceedings (MeMeA), IEEE International Symposium**, 18-19 maio 2012. 1-5.

WORLD HEALTH ORGANIZATION. Obesity and overweight. **World Health Organization**, jan. 2015. ISSN Fact sheet N°311. Disponível em: <<http://www.who.int/mediacentre/factsheets/fs311/en/>>. Acesso em: 09 dez. 2015.

|