



Francisco Carlos Lira

Aplicação de Agrupamento Semissupervisionado para Segmentação de Imagens Coloridas

Recife

Fevereiro, 2018

Francisco Carlos Lira

Aplicação de Agrupamento Semissupervisionado para Segmentação de Imagens Coloridas

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Ciência da Computação

Orientador: Valmir Macário Filho

Recife

Fevereiro, 2018



MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

<http://www.bcc.ufrpe.br>

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Francisco Carlos Lira às 10 horas e 30 minutos do dia 07 de fevereiro de 2018, no Auditório do CEAGRI-02 – Sala 07, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado **Aplicação de Agrupamento Semissupervisionado para Segmentação de Imagens Coloridas**, orientado por Valmir Macário Filho e aprovado pela seguinte banca examinadora:

Valmir Macário Filho
DEINFO/UFRPE

André Câmara do Nascimento
DEINFO/UFRPE

Filipe Rolim Cordeiro
DEINFO/UFRPE

Dedico este trabalho à minha mãe, Lindinalva e, (in memoriam), ao meu pai, Francisco Zacarias Lira, aos meus avós, Antônio e Aurora, ao meu tio, Zuca, e ao meu eterno amigo Milu.

Agradecimentos

Agradeço primeiramente à minha mãe, por ser a minha base e sempre ter me apoiado e incentivado em todos os momentos da minha vida. Agradeço à minha namorada, Raphaela, por seu amor e companheirismo. Agradeço ao meu irmão, Antônio, e à minha irmã, Rosana, que sempre estão ao meu lado quando necessito. Agradeço aos meus amigos da faculdade, especialmente a Victor Sales, por todo auxílio que me prestou durante a realização deste e outros trabalhos, pelas caronas, e pela amizade; e a Rodolfo Ferreira, Sergio Chevtchenko e Rafaella Vale por serem sempre pessoas com quem posso contar e serem sempre grandes amigos. Agradeço ao meu felino, Java, por todas as noites de companhia enquanto fazia meus trabalhos. Agradeço a todos os meus professores, especialmente ao meu orientador, Valmir Macário, por todas as dicas, auxílio e incentivo prestados, ao professor Ricardo Souza, pela oportunidade de iniciação científica, e à professora Jeane Melo, por sua amizade e ensinamentos. Agradeço aos meus colegas de trabalho e amigos da L2 Tecnologia, onde obtive vários conhecimentos, me desenvolvi pessoalmente e profissionalmente, e também compartilhei bons momentos e boas partidas de FIFA e CS. Por fim, gostaria de agradecer à UFRPE, que foi minha segunda casa durante esses quase cinco anos, e a todos seus funcionários, em especial aos funcionários do RU.

Resumo

A segmentação de imagens é um processo que subdivide uma imagem em seus objetos ou regiões constituintes e esta é uma etapa muito importante para aplicações que necessitam de processamento de imagens, pois através dos resultados da segmentação é possível identificar regiões de interesse e objetos na cena. Uma das maneiras de obter a segmentação, é através da aplicação de algoritmos de agrupamento não-supervisionados, que utilizam dados totalmente não *pixels*; e semissupervisionados, que utilizam apenas uma parte dos dados rotulados. A segmentação de imagens através de algoritmos de agrupamento semissupervisionados é um tipo de método semi-automático que explora uma pequena quantidade de dados rotulados para guiar o processo de segmentação o que pode proporcionar melhores resultados. Porém para que os algoritmos semissupervisionados tenham bons resultados, é necessário aplicar estratégias eficientes para rotulação parcial a partir dos rótulos iniciais fornecidos pelo usuário.

Neste trabalho são apresentados e comparados os resultados do algoritmo de agrupamento *Fuzzy C-means* (FCM) e de três algoritmos de agrupamento semissupervisionados nele baseados, o algoritmo *Standard Semi-Supervised Clustering* (sSSC), o algoritmo *Entropy Regularized Semi-Supervised Clustering* (ESSC) e o algoritmo semissupervisionado proposto por Pedrycz. Os resultados indicam que o algoritmo proposto por Pedrycz possui os melhores resultados entre os algoritmos semissupervisionados, e supera o algoritmo FCM, dependendo da quantidade e da estratégia de expansão dos rótulos iniciais.

Palavras-chave: processamento de imagens, segmentação de imagens, agrupamento não-supervisionado, agrupamento semissupervisionado, *Fuzzy C-means*.

Abstract

Image segmentation is a process that subdivides an image into its objects or constituent regions and is a very important step for applications that require some sort of image processing, for through its results, it is possible to identify regions of interest and objects in the scene. One way to perform image segmentation is through unsupervised, which uses unlabeled data, and semi-supervised, which uses labeled and unlabeled data, clustering algorithms. Image segmentation through semi-supervised clustering algorithms is a kind of semi-automatic method that exploits a small amount of labeled data to guide the segmentation process, which in turn can provide better results, or complete it faster. However, for this kind of algorithm achieve good results, is necessary to apply efficient strategies for partial labeling expansion considering the the initial user provided labels.

In this work, the results of the Fuzzy C-means (FCM) clustering and algorithm the semi-supervised FCM-based algorithms Standard Semi-supervised Clustering (sSSC); Entropy Regularized Semi-supervised Clustering (ESSC); and Pedrycz algorithm are presented and compared. The initial labels are provided through user clicks over the image, where the user can label it as foreground or background. Results show that the Pedrycz algorithm achieved better results among the semi-supervised clustering algorithms, and can even overcome the standard FCM algorithm, depending on the amount of initial labels provided by the user.

Keywords: image processing, image segmentation, unsupervised clustering, semi-supervised clustering, Fuzzy C-means.

Lista de ilustrações

Figura 1 – Agrupamento <i>fuzzy</i> e agrupamento <i>hard</i>	18
Figura 2 – Aprendizados não supervisionado, semissupervisionado e supervisionado	20
Figura 3 – Representação matricial de uma imagem digital	25
Figura 4 – Vizinhanças de um pixel	26
Figura 5 – Operações lógicas com imagens binárias	28
Figura 6 – Espaço de cores RGB	29
Figura 7 – Canais RGB	29
Figura 8 – Espaço de cores CIE L*a*b	30
Figura 9 – Exemplo de Histograma	31
Figura 10 – Exemplo de histograma de imagem colorida	32
Figura 11 – <i>Region Growing</i>	33
Figura 12 – Exemplo de segmentação por agrupamento	34
Figura 13 – Exemplo de imagem do <i>dataset</i> CMU Cornell iCoseg	40
Figura 14 – Matriz de confusão	40
Figura 15 – Exemplo dos formatos de rotulação parcial de imagens	43
Figura 16 – Ferramenta de rotulação parcial de imagens	44
Figura 17 – Configuração dos experimentos	45
Figura 18 – Melhores MCC'S - Região Crescente	50
Figura 19 – Melhores segmentações - Região crescente	50
Figura 20 – Melhores segmentações - Quadrado ($l = 20$)	55
Figura 21 – Melhores MCC's - Quadrado ($l = 20$)	55
Figura 22 – Melhores segmentações - Quadrado ($l = 30$)	60
Figura 23 – Melhores MCC's - Quadrado ($l = 30$)	60
Figura 24 – Melhores MCC's por algoritmo	61

Lista de tabelas

Tabela 1 – Exemplo de Histograma.	31
Tabela 2 – Resultados do algoritmo <i>Fuzzy C-means</i>	46
Tabela 3 – Resultado - região crescente, RGB, 1 rótulo inicial (apenas sobre o objeto)	47
Tabela 4 – Resultado - região crescente, RGB, 1 rótulo inicial (apenas sobre o fundo)	47
Tabela 5 – Resultado - região crescente, RGB, 5 rótulos iniciais (fundo e objeto) .	47
Tabela 6 – Resultado - região crescente, LAB, 1 rótulo inicial (apenas sobre o objeto)	48
Tabela 7 – Resultado - região crescente, LAB, 1 rótulo inicial (apenas sobre o fundo)	48
Tabela 8 – Resultado - região crescente, LAB, 5 rótulos iniciais (fundo e objeto) . .	49
Tabela 9 – Teste de Wilcoxon dos Melhores MCC's - Região Crescente	49
Tabela 10 – Resultado - quadrado ($l = 20$), RGB, 1 rótulo inicial (apenas sobre o objeto)	51
Tabela 11 – Resultado - quadrado ($l = 20$), RGB, 1 rótulo inicial (apenas sobre o fundo)	51
Tabela 12 – Resultado - quadrado ($l = 20$), RGB, 5 rótulos iniciais (apenas sobre o objeto)	52
Tabela 13 – Resultado - quadrado ($l = 20$), RGB, 5 rótulos iniciais (fundo e objeto)	52
Tabela 14 – Resultado - quadrado ($l = 20$), LAB, 1 rótulo inicial (apenas sobre o objeto)	52
Tabela 15 – Resultado - quadrado ($l = 20$), LAB, 1 rótulo inicial (apenas sobre o fundo)	53
Tabela 16 – Resultado - quadrado ($l = 20$), LAB, 5 rótulos iniciais (apenas sobre o objeto)	53
Tabela 17 – Resultado - quadrado ($l = 20$), LAB, 5 rótulos iniciais (fundo e objeto)	53
Tabela 18 – Teste de Wilcoxon dos Melhores MCC's - Quadrado ($l = 20$)	54
Tabela 19 – Resultado - quadrado ($l = 30$), RGB, 1 rótulo inicial (apenas sobre o objeto)	56
Tabela 20 – Resultado - quadrado ($l = 30$), RGB, 1 rótulo inicial (apenas sobre o fundo)	56
Tabela 21 – Resultado - quadrado ($l = 30$), RGB, 5 rótulos iniciais (apenas sobre o objeto)	57
Tabela 22 – Resultado - quadrado ($l = 30$), RGB, 5 rótulos iniciais (fundo e objeto)	57
Tabela 23 – Resultado - quadrado ($l = 30$), LAB, 1 rótulo inicial (apenas sobre o objeto)	57
Tabela 24 – Resultado - quadrado ($l = 30$), LAB, 1 rótulo inicial (apenas sobre o fundo)	58

Tabela 25 – Resultado - quadrado ($l = 30$), LAB, 5 rótulos iniciais (apenas sobre o objeto)	58
Tabela 26 – Resultado - quadrado ($l = 30$), LAB, 5 rótulos iniciais (fundo e objeto)	58
Tabela 27 – Teste de Wilcoxon dos Melhores MCC's - Quadrado ($l = 30$)	59

Lista de Algoritmos

1	Algoritmo FCM	20
2	Algoritmo de Pedrycz	22
3	Algoritmo sSSC	23
4	Algoritmo eSSC	24

Lista de abreviaturas e siglas

ARFF	<i>Attribute-Relation File Format</i>
EM	<i>Expectation maximization</i>
ESSC	<i>Entropy Semi Supervised Clustering</i>
FCM	<i>Fuzzy C-Means</i>
FN	Falso negativo
FP	Falso positivo
MCC	<i>Matthews Correlation Coefficient</i>
RGB	<i>Red, Green, Blue</i>
sSSC	<i>Standard Semi Supervised Clustering</i>
VN	Verdadeiro negativo
VP	Verdadeiro positivo

Sumário

1	INTRODUÇÃO	14
1.1	Problema de Pesquisa	16
1.2	Objetivos	16
1.2.1	Objetivo geral	16
1.2.2	Objetivos específicos	16
1.3	Organização do trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Agrupamento (<i>Clustering</i>)	18
2.1.1	Algoritmo <i>Fuzzy C-Means</i> (FCM)	19
2.1.2	Algoritmos de Agrupamento Semissupervisionado	19
2.1.2.1	Algoritmo de Pedrycz	20
2.1.2.2	Algoritmo Semissupervisionado Padrão (sSSC)	21
2.1.2.3	Algoritmo Semissupervisionado Regularizado por Entropia (ESSC)	22
2.2	Processamento de Imagens	24
2.2.1	Relacionamentos entre Pixels	25
2.2.1.1	Vizinhança	25
2.2.1.2	Caminho	26
2.2.1.3	Adjacência	26
2.2.1.4	Conectividade	27
2.2.1.5	Regiões	27
2.2.2	Operações Lógicas e Aritméticas	27
2.2.3	Espaços de Cores	28
2.2.3.1	RGB	29
2.2.3.2	CIE L*a*b	30
2.2.4	Histograma	30
2.2.5	Segmentação de Imagens	32
2.2.5.1	Fundamentos	32
2.2.5.2	Região Crescente (<i>Region Growing</i>)	33
2.2.5.3	Segmentação de imagens por algoritmos de agrupamento	33
3	TRABALHOS RELACIONADOS	35
4	METODOLOGIA	39
4.1	Configuração experimental	39
4.1.1	Base de dados	39

4.1.2	Ferramentas e ambiente de programação	39
4.1.3	Metodologia de avaliação	39
4.2	Métodos de rotulação parcial de imagem	41
4.3	Ferramenta de rotulação	42
4.4	Experimentos Realizados	43
5	RESULTADOS E DISCUSSÃO	46
5.1	Resultados - <i>Fuzzy C-means</i>	46
5.1.1	Região Crescente	46
5.1.2	Quadrado ($l = 20$)	50
5.1.3	Quadrado ($l = 30$)	55
5.2	Análise dos resultados	60
6	CONCLUSÃO	62
6.1	Trabalhos futuros	62
	REFERÊNCIAS	64

1 Introdução

Com o passar do tempo e o avanço da tecnologia da informação, softwares que lidam com imagens vêm ganhando um amplo espaço no dia-a-dia das pessoas. Aplicações como Instagram¹ e SnapChat² registraram o marco de, respectivamente, 700 000 000 e 166 000 000 usuários ativos em Abril de 2017 (PORTAL, 2017b) (PORTAL, 2017a), e várias outras aplicações que utilizam imagens podem ser encontradas em vários campos, tais como diagnósticos por imagens, reconhecimento de textos digitalizados, restauração de imagens, etc. Muitas dessas aplicações incluem algum tipo de processamento de imagem para aprimorar os seus resultados. Dessa forma, é extremamente necessário desenvolver novas técnicas de processamento de imagens e aprimorar as técnicas já existentes.

O processamento de imagem pode ser entendido como uma transformação de uma imagem em uma versão melhorada dessa mesma imagem (VERNON, 1991). Uma das etapas do processamento de imagens é a segmentação, que é a etapa abordada neste trabalho. Segmentação é um termo utilizado para descrever um processo de agrupamento no qual os componentes de um grupo são similares, quando levado em consideração alguma característica, ou conjunto de características, e estes grupos identificam regiões da imagem que correspondem a objetos únicos e distintos (VERNON, 1991). Esta é uma etapa muito importante para aplicações que necessitam de processamento de imagens, pois através dos resultados da segmentação é possível identificar regiões de interesse e objetos na cena (QIAN; SI, 2005).

O processo de segmentação pode ser classificado em quatro categorias (SHIH; CHENG, 2005): limiarização, baseadas em bordas, baseadas em região ou abordagens híbridas. A limiarização se baseia na ideia que grupos na imagem correspondem ou ao plano de fundo da imagem, ou ao objeto de interesse, e a segmentação pode ser obtida separando esses grupos através de um limiar. Os métodos baseados em bordas assumem que as propriedades do *pixel*, tais como intensidade, cor e textura, mudam bruscamente entre regiões distintas. Métodos baseados em região assumem que *pixels* vizinhos dentro de uma mesma região devem ter características semelhantes, tais como cor e textura. E os métodos híbridos, combinam diferentes técnicas para obter uma melhor segmentação.

Uma das maneiras de aplicar a segmentação de imagens é através de algoritmos de aprendizado de máquina (MITCHELL, 1997). O aprendizado de máquina constrói modelos computacionais que aprendem a extrair conhecimento a partir da análise dos dados. Estes algoritmos podem ser classificados em duas categorias: supervisionada e não supervisionada

¹ <https://www.instagram.com/>

² <https://www.snapchat.com/>

No aprendizado supervisionado um conjunto de dados rotulados é fornecido ao algoritmo. A aprendizagem é realizada através desses padrões conhecidos, de forma que o algoritmo aprenda uma função que mapeie o padrão x_i ao rótulo h_i a partir do conjunto de treinamento que contém pares (x_i, h_i) , onde o $h_i \in H$ são chamados de rótulos dos exemplos x_i e $0 \leq i \leq n$, onde n é o número de padrões. O objetivo dessa abordagem é prever o rótulo h_i dos exemplos que não possuem seus rótulos conhecidos.

Na aprendizagem não supervisionada os dados apresentados ao algoritmo não possuem rótulos e o seu objetivo é aprender a estrutura dos dados apresentados. Esse tipo de algoritmo busca aprender a distribuição dos dados de acordo com alguma medida de similaridade, agrupando os padrões mais similares entre si no mesmo grupo.

Agrupamento (ou *Clustering*) é um método não supervisionado que separa os dados em diferentes grupos, onde os dados presentes em um grupo possuem uma maior similaridade entre si, de acordo com o contexto do problema (JAIN; DUBES, 1988). Os algoritmos de agrupamento mais utilizados são o K-means (MACQUEEN et al., 1967) e o Fuzzy C-Means (FCM) (BEZDEK; EHRlich; FULL, 1984). Entretanto, o FCM pondera a distância pelo grau de pertinência, que indica quanto um padrão, ou exemplo, pertence aos grupos, o que muitas vezes permite a obtenção de melhores resultados. O FCM é o algoritmo de agrupamento difuso mais utilizado no contexto de segmentação de imagens, pois ele possui características robustas para ambiguidade e pode reter mais informações do que outros algoritmos de agrupamento (YANG; HUANG, 2012).

Das abordagens de aprendizagem de máquina, existe ainda a aprendizagem semisupervisionada, que é intermediária entre a aprendizagem supervisionada, na qual os dados são rotulados, e a não supervisionada. Nela, apenas uma parte dos dados são rotulados (CHAPELLE; SCHOLKOPF; ZIEN, 2009).

A segmentação de imagens utilizando algoritmos semissupervisionados é um tipo de método semi automático que explora uma pequena quantidade de dados rotulados para guiar o processo de segmentação e, em algumas aplicações, esse método proporciona melhores resultados de segmentação, ou leva menos tempo para concluí-la (CHEN et al., 2012).

Neste trabalho será apresentada uma comparação entre algoritmos semissupervisionados baseados no Fuzzy C-Means sSSC (*Standard Semi-Supervised Clustering*) (YASUNORI et al., 2009), eSSC (*Entropy Regularized Semi-Supervised Clustering*) (YASUNORI et al., 2009) e o algoritmo de Pedrycz (PEDRYCZ; WALETZKY, 1997), quando aplicadas para segmentação de imagens coloridas, considerando os espaços de cores RGB e CIE L*a*b. Ainda serão consideradas três formas de rotulação parcial das imagens, as quais expandirão os rótulos iniciais fornecidos pelo usuário através de cliques na imagem. Essas três formas são: *i.* considerar uma quantidade fixa de *pixels* em um formato de quadrado, com tamanho $l = 20$ *pixels* a partir dos rótulos (cliques) iniciais; *ii.* em um

formato de quadrado, com tamanho $l = 30$ pixels; e *iii.* considerar os rótulos iniciais como sementes para uma expansão baseada em região crescente.

1.1 Problema de Pesquisa

O Fuzzy C-Means é um dos algoritmos de agrupamento mais utilizados para a segmentação de imagens, pois além da sua facilidade de implementação, ele produz bons resultados e consegue extrair mais informações dos dados do que outras abordagens (RAJABY; AHADI; AGHAEINIA, 2016). Ainda assim, o FCM possui suas limitações, tal como a sensibilidade a ruídos na imagem. Além disso, uma boa segmentação pode ser subjetiva, dependendo do objetivo da aplicação. Dessa forma, algoritmos não supervisionados podem não atender às expectativas do usuário. Entretanto, através de uma abordagem semissupervisionada, o usuário pode indicar na imagem qual é a região de interesse, fornecendo informações úteis ao algoritmo, o que pode levar a melhores resultados na segmentação.

Na literatura, não há um grande volume de trabalhos que re realizam estudos comparativos acerca de quais são os melhores algoritmos semissupervisionados quando aplicados ao contexto de segmentação de imagens. Dessa forma, neste trabalho serão avaliadas diferentes variações de algoritmos semissupervisionados baseados no FCM para a segmentação de imagens coloridas, a fim de constatar qual obtém os melhores resultados, visando resolver as seguintes perguntas: ((*i.*) qual desses algoritmos possui melhores resultados; e (*ii.*) quantos rótulos iniciais seriam necessários para que eles superarem o algoritmo FCM.

1.2 Objetivos

Nesta seção estão expostos os objetivos almejados com a realização deste trabalho.

1.2.1 Objetivo geral

Avaliar o resultado de algoritmos de agrupamento semissupervisionado baseados no Fuzzy C-Means para segmentação de imagens coloridas

1.2.2 Objetivos específicos

1. Avaliar diferentes espaços de cores e combinações para saber qual é o melhor para a segmentação desses algoritmos;
2. Avaliar várias formas de estratégia de rotulação para guiar os algoritmos semissupervisionados;

3. Desenvolver uma ferramenta de segmentação de imagens coloridas para geração dos experimentos e *dataset* de arquivos de entrada para os algoritmos de agrupamento.

1.3 Organização do trabalho

Os demais capítulos deste trabalho estão dispostos da seguinte maneira: o [Capítulo 2](#) apresenta os aspectos teóricos sobre Aprendizagem de Máquina e Processamento de Imagens utilizados nesse trabalho; no [Capítulo 3](#), são apresentados trabalhos relacionados de alguma forma com este trabalho; o [Capítulo 4](#) detalha as configurações experimentais e metodologias de rotulação parcial de imagens utilizadas para gerar as entradas necessárias para os algoritmos semissupervisionados e realizar os experimentos de segmentação; o [Capítulo 5](#) apresenta e examina os resultados obtidos por cada configuração experimental apresentada no [Capítulo 5](#); o [Capítulo 6](#) expõe as considerações finais e expectativas para trabalhos futuros.

2 Fundamentação teórica

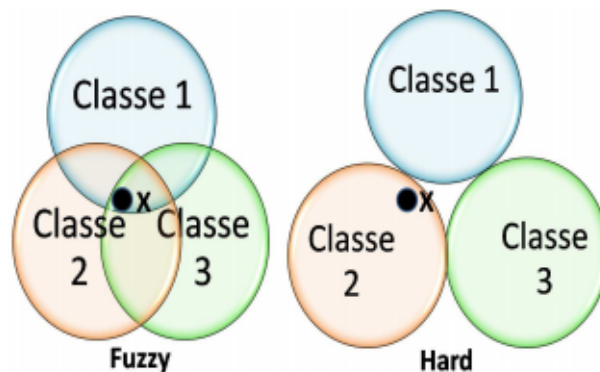
Neste capítulo são apresentados os conceitos e fundamentos necessários para a realização deste trabalho de pesquisa. A [seção 2.1](#) apresenta os conceitos dos algoritmos de agrupamento, e a [seção 2.2](#) apresenta os conceitos relacionados ao processamento de imagens digitais.

2.1 Agrupamento (*Clustering*)

Agrupamento é um tipo de tarefa não-supervisionada que, considerando alguma medida de similaridade, aglomera os objetos mais semelhantes entre si no mesmo grupo, enquanto os demais dados são agrupados em grupos distintos. Para agrupar pixels, esses algoritmos utilizam apenas as descrições destes. Estas descrições são compostas por vetores contendo vetores características descritivas de cada exemplo, tais como coordenadas do pixel, valores de diferentes sistemas de cores, entre outros.

Os algoritmos de agrupamento podem ser classificados em *hard* ou *fuzzy*. No agrupamento *hard*, o objeto só pode pertencer a um grupo (DUDA; HART; STORK, 2012). No agrupamento *fuzzy*, a condição de pertinência a um grupo é flexível. Dessa forma, um objeto x_i pode estar associado a mais de um grupo de acordo com um grau de pertinência u_{ik} no grupo v_k . O valor do grau de pertinência está contido no intervalo $[0, 1]$, onde 1 indica uma pertinência total do padrão x_i no grupo v_k , 0 indica que x_i não pertence a v_k , e caso o valor esteja contido no intervalo $0 < u_{ik} < 1$, então x_i pertence parcialmente a v_k (DUDA; HART; STORK, 2012). A figura [Figura 1](#) ilustra os dois tipos de agrupamentos.

Figura 1 – Agrupamento *fuzzy* e agrupamento *hard*



Fonte: Macário (2009)

2.1.1 Algoritmo *Fuzzy C-Means* (FCM)

O algoritmo FCM (BEZDEK; EHRLICH; FULL, 1984) é utilizado como base para todos os algoritmos de agrupamento semissupervisionados utilizados nessa pesquisa. Este algoritmo busca minimizar iterativamente uma função objetivo para agrupar os dados.

Esta função é dada pela [Equação 2.1](#).

$$J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \|x_k - v_i\|^2 \quad (2.1)$$

onde V representa o conjunto de protótipos, U é a matriz de graus de pertinência $x_c, c = 1, 2, \dots, N$ são os padrões, v_1, v_2, \dots, v_c são os vetores de protótipos do cluster v_c , $\|x_k - v_i\|^2 = d_{ik}^2$ representa uma norma de distância (tal como a Distância Euclidiana, ou Distância Mahalanobis) do dado x_k até o c -ésimo centro de cluster; e $m \in [1, \infty)$ é o parâmetro de fuzificação (ALIA; MANDAVA; AZIZ, 2011) e (PEDRYCZ; WALETZKY, 1997).

O algoritmo FCM inicia com c centros de *clusters* iniciais aleatórios e a cada iteração encontra a pertinência *fuzzy* de cada ponto para cada *cluster* através da [Equação 2.2](#)

$$u_{ik} = \frac{1}{\sum_{l=1}^c \left(\frac{\|x_i - v_k\|}{\|x_i - v_l\|} \right)^{\frac{2}{m-1}}} \quad (2.2)$$

Baseado nos valores dos graus de pertinência, os centros dos *clusters* são recalculados através da [Equação 2.3](#)

$$v_k = \frac{\sum_{i=1}^N u_{ik}^m \cdot x_i}{\sum_{i=1}^N u_{ik}^m} \quad (2.3)$$

O algoritmo encerra quando não há mais mudanças nos centros dos *clusters*. O pseudocódigo do FCM está descrito no [Algoritmo 1](#)

2.1.2 Algoritmos de Agrupamento Semissupervisionado

As abordagens de aprendizagem supervisionadas e não supervisionadas possuem algumas limitações. Por exemplo, os dados não rotulados usados na aprendizagem não supervisionada não possuem informações acerca dos padrões, o que pode tornar o processo de agrupamento uma tarefa complexa; porém, nem sempre é possível rotular uma grande quantidade de dados para utilizá-los no aprendizado supervisionado. Pelo fato de utilizar apenas uma parte dos dados rotulados, o aprendizado semissupervisionado pode obter melhores resultados com poucos dados rotulados do que se fosse utilizado uma das abordagens supracitadas, e com baixo custo (BILENKO; BASU; MOONEY, 2004) e (MACÁRIO, 2009).

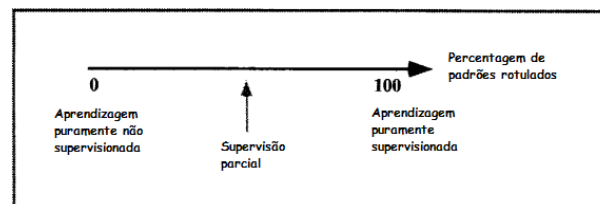
Algoritmo 1 Algoritmo FCM

```

1: Inicializa  $c$  ▷  $2 < c < n$ 
2: Inicializa  $\epsilon$  ▷ Ex.:  $\epsilon = 0.001$ 
3: Inicializa  $maxIteracoes$  ▷ Ex.:  $maxIteracoes = 100$ 
4: Seleciona métrica de distância ▷ Ex.: Distância Euclidiana
5: Inicializa  $m$  ▷  $1 < m < \infty$ 
6: Inicializa aleatoriamente os centros dos clusters  $V_0 = v_1, v_2, \dots, v_c$ 
7: para  $t \leftarrow 1$  até  $maxIteracoes$  faça
8:   Atualiza a matriz de associação  $U$  através da Equação 2.2
9:   Calcula os novos centros dos clusters  $V^t$  através da Equação 2.3
10:  Calcula a nova função objetivo  $J_m^t$  através da Equação 2.1
11:  se  $abs(J_m^t - J_m^{t-1}) < \epsilon$  então
12:    Interrompe
13:  senão
14:     $J_m^t \leftarrow J_m^{t-1}$ 
15:  fim se
16: fim para

```

Figura 2 – Aprendizados não supervisionado, semissupervisionado e supervisionado



Traduzido de: [Pedrycz e Waletzky \(1997\)](#)

Os algoritmos de agrupamento semissupervisionado serão descritos nas subseções a seguir. Estes algoritmos diferem nas equações e algumas características inerentes a função objetivo de cada um dos deles. São eles:

1. Algoritmo de Pedrycz;
2. Algoritmo Semissupervisionado Padrão (sSSC); e
3. Algoritmo Semissupervisionado Regularizado por Entropia (ESSC).

2.1.2.1 Algoritmo de Pedrycz

O método proposto por ([PEDRYCZ; WALETZKY, 1997](#)) foi um dos pioneiros entre os algoritmos semissupervisionados. Este algoritmo altera a função objetivo do algoritmo FCM, descrita na [Equação 2.1](#). O termo adicionado à função objetivo do algoritmo FCM utiliza os rótulos disponíveis dos exemplos da base de dados para aumentar os graus de pertinência atribuídos pelo algoritmo para os grupos que representam a classe à qual o exemplo pertence. Entretanto, para que a equação do cálculo da matriz de pertinência

seja formulada, o parâmetro m é definido como 2. Dessa forma, a função objetivo desse algoritmo após a modificação se torna:

$$J(U, v) = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^c \sum_{k=1}^N (u_{ik} - f_{ik} b_i)^2 d_{ik}^2 \quad (2.4)$$

onde $\alpha (\alpha \geq 0)$ é um fator de balanceamento entre os componentes supervisionados e não supervisionados. Os parâmetros que se beneficiam dos dados rotulados são o termo binário b_i e a matriz f_{ik} . O termo $b_i, i = 1, 2, \dots, N$, onde N é o número de padrões da base de dados, distingue padrões rotulados e não rotulados, de modo que:

$$b_i = \begin{cases} 1 & \text{se padrão } \mathbf{x}_i \text{ é rotulado} \\ 0 & \text{senão} \end{cases}$$

A matrix $F = [f_{ij}]$, tal que $i = 1, 2, \dots, N$ e $k = 1, 2, \dots, c$, onde c é o número de grupos formados pelo algoritmo, contém os valores dos graus de pertinência dos padrões rotulados, sendo 1 o valor de grau de pertinência para o grupo que representa a classe desse padrão e 0 para os demais grupos. U representa a matriz que contém os graus de pertinência. Cada u_{ik} representa o grau de pertinência do padrão \mathbf{x}_i no grupo \mathbf{v}_k . V representa o conjunto de protótipos \mathbf{v}_k associado a cada grupo. O parâmetro m indica o quão *fuzzy* é o agrupamento. Por último, o termo d_{ij}^2 representa a métrica de distância entre o padrão \mathbf{x}_i e o grupo \mathbf{v}_k (PEDRYCZ; WALETZKY, 1997) e (MACÁRIO, 2009).

O cálculo das matrizes de pertinência é dada pela [Equação 2.5](#)

$$u_{ik} = \frac{1}{1 + \alpha} \left\{ \frac{1 + \alpha(1 - b_i \sum_{l=1}^c f_{il})}{\sum_{l=1}^c (d_{il}^2)} + \alpha f_{ik} b_i \right\} \quad (2.5)$$

2.1.2.2 Algoritmo Semissupervisionado Padrão (sSSC)

O algoritmo desenvolvido por Yasunori et al. (2009) acrescenta uma matriz de pertinência supervisionada f_{ij} do objeto $e_i, i = 1, 2, \dots, n$ no grupo $c_j, j = 1, 2, \dots, J$ no algoritmo de agrupamento FCM padrão. Dos algoritmos semissupervisionados apresentados nesse trabalho, apenas este possui apenas um termo na função objetivo. Se o padrão for rotulado e pertencer à mesma classe representada pelo grupo $c_j, f_{ij} = 1$, dessa forma o grau de pertinência é influenciado a ter valor 1 nesse grupo. Caso isso não aconteça, a função objetivo se torna a mesma função objetivo do algoritmo FCM descrita na [Equação 2.1](#). A função objetivo deste algoritmo está descrita na [Equação 2.6](#).

$$J_m(U, v) = \sum_{i=1}^c \sum_{k=1}^N (u_{ik} - f_{ik})^m d_{ik}^2 \quad (2.6)$$

Algoritmo 2 Algoritmo de Pedrycz

-
- 1: Inicializa c ▷ $2 < c < n$
 - 2: Inicializa ϵ ▷ Ex.: $\epsilon = 0.001$
 - 3: Inicializa $maxIteracoes$ ▷ Ex.: $maxIteracoes = 100$
 - 4: Seleciona métrica de distância ▷ Ex.: Distância Euclidiana
 - 5: Inicializa m ▷ $1 < m < \infty$
 - 6: Inicializa aleatoriamente os centros dos *clusters* $V_0 = v_1, v_2, \dots, v_c$
 - 7: Inicializa matriz de protótipos
 - 8: Inicializa matriz de graus de pertinência
 - 9: **para** $t \leftarrow 1$ **até** $maxIteracoes$ **faça**
 - 10: Calcula os novos centros dos *clusters* V^t através da [Equação 2.3](#)
 - 11: Atualiza a matriz de graus de pertinência U através da [Equação 2.5](#)
 - 12: Calcula a nova função objetivo J_m^t através da [Equação 2.4](#)
 - 13: **se** $abs(J_m^t - J_m^{t-1}) < \epsilon$ **então**
 - 14: **Interrompe**
 - 15: **senão**
 - 16: $J_m^t \leftarrow J_m^{t-1}$
 - 17: **fim se**
 - 18: **fim para**
-

O sSSC é um algoritmo iterativo com duas etapas, representação e alocação, as quais são executadas iterativamente até que um critério de parada seja satisfeito. Na etapa de representação, a matriz U de graus de pertinência é mantida fixa. A função objetivo é minimizada em relação à matriz de protótipos V , onde os protótipos dos grupos $v_k (k = 1, 2, \dots, c)$ são calculados através da [Equação 2.7](#).

$$v_k = \frac{\sum_{i=1}^N (u_{ik} - f_{ik})^2 x_i}{\sum_{i=1}^N (u_{ik} - f_{ik})^2} \quad (2.7)$$

Na etapa de alocação, a matriz de protótipos V é mantida fixa e a função objetivo é minimizada em relação à matriz U de graus de pertinência. O grau de pertinência do objeto $e_i (i = 1, 2, \dots, N)$ num grupo $v_k (k = 1, 2, \dots, c)$ é calculado através da [Equação 2.8](#)

$$u_{ik} = f_{ik} + \left(1 - \sum_{l=1}^c f_{il}\right) \frac{\left(\frac{1}{d_{ik}^2}\right)^{\frac{1}{m-1}}}{\sum_{l=1}^c \left(\frac{1}{d_{il}^2}\right)^{\frac{1}{m-1}}} \quad (2.8)$$

O algoritmo sSSC está descrito no [Algoritmo 3](#)

2.1.2.3 Algoritmo Semissupervisiona Regularizado por Entropia (ESSC)

Em [Yasunori et al. \(2009\)](#) também é apresentado outro algoritmo semissupervisionado, o qual adiciona um termo supervisionado baseado na entropia à função objetivo do algoritmo FCM. Esta entropia é calculada utilizando a diferença entre o grau de pertinência de um objeto e_i ao grupo v_k e o valor de f_{ij} o qual indica se um objeto e_i pertence à

Algoritmo 3 Algoritmo sSSC

```

1: Inicializa  $c$  ▷  $2 < c < n$ 
2: Inicializa  $\epsilon$  ▷ Ex.:  $\epsilon = 0.001$ 
3: Inicializa  $maxIteracoes$  ▷ Ex.:  $maxIteracoes = 100$ 
4: Seleciona métrica de distância ▷ Ex.: Distância Euclidiana
5: Inicializa  $m$  ▷  $1 < m < \infty$ 
6: Inicializa aleatoriamente os centros dos clusters  $V_0 = v_1, v_2, \dots, v_c$ 
7: Inicializa matriz de protótipos
8: Inicializa matriz de graus de pertinência
9: para  $t \leftarrow 1$  até  $maxIteracoes$  faça
10:   Calcula os novos centros dos clusters  $V^t$  através da Equação 2.7
11:   Atualiza a matriz de graus de pertinência  $U$  através da Equação 2.8
12:   Calcula a nova função objetivo  $J_m^t$  através da Equação 2.6
13:   se  $abs(J_m^t - J_m^{t-1}) < \epsilon$  então
14:     Interrompe
15:   senão
16:      $J_m^t \leftarrow J_m^{t-1}$ 
17:   fim se
18: fim para

```

classe representada pelo grupo v_k . Se o objeto pertencer a v_k , o valor da entropia é baixo, senão, o valor é alto (MACARIO, 2015). A função objetivo deste algoritmo está descrita na [Equação 2.9](#).

$$J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m d_{ik}^2 + \alpha \sum_{i=1}^c \sum_{k=1}^N (|(u_{ik} - f_{ik})| \log |(u_{ik} - f_{ik})|) \quad (2.9)$$

O termo supervisionado desse algoritmo é otimizado da seguinte forma: caso o exemplo $e_i \in C_k$, o valor do grau de pertinência desse exemplo ao grupo c_k é forçado a ser um, caso contrário, o grau de pertinência é forçado a ser zero (MACARIO, 2015). Assim como o algoritmo sSSC, este algoritmo também é um algoritmo iterativo de duas etapas (representação e alocação). Na etapa de representação, a matriz U de graus de pertinência é mantida fixa. A função objetivo é minimizada em relação à matriz de protótipos V , onde os protótipos dos grupos $v_k (k = 1, 2, \dots, c)$ são calculados através da [Equação 2.3](#).

Na etapa de alocação, a matriz de protótipos V é mantida fixa e a função objetivo é minimizada em relação à matriz U de graus de pertinência. O grau de pertinência do objeto $e_i (i = 1, 2, \dots, N)$ num grupo $v_k (k = 1, 2, \dots, c)$ é calculado através da [Equação 2.10](#).

$$u_{ik} = f_{ik} + \frac{e^{-\alpha(d_{ik}^2)}}{\sum_{l=1}^c e^{-\alpha(d_{il}^2)}} \quad (2.10)$$

Algoritmo 4 Algoritmo eSSC

```

1: Inicializa  $c$  ▷  $2 < c < n$ 
2: Inicializa  $\epsilon$  ▷ Ex.:  $\epsilon = 0.001$ 
3: Inicializa  $maxIteracoes$  ▷ Ex.:  $maxIteracoes = 100$ 
4: Seleciona métrica de distância ▷ Ex.: Distância Euclidiana
5: Inicializa  $m$  ▷  $1 < m < \infty$ 
6: Inicializa aleatoriamente os centros dos clusters  $V_0 = v_1, v_2, \dots, v_c$ 
7: Inicializa matriz de protótipos
8: Inicializa matriz de graus de pertinência
9: para  $t \leftarrow 1$  até  $maxIteracoes$  faça
10:   Calcula os novos centros dos clusters  $V^t$  através da Equação 2.3
11:   Atualiza a matriz de graus de pertinência  $U$  através da Equação 2.10
12:   Calcula a nova função objetivo  $J_m^t$  através da Equação 2.9
13:   se  $abs(J_m^t - J_m^{t-1}) < \epsilon$  então
14:     Interrompe
15:   senão
16:      $J_m^t \leftarrow J_m^{t-1}$ 
17:   fim se
18: fim para

```

2.2 Processamento de Imagens

O processamento de imagem pode ser entendido como a aplicação de métodos e algoritmos computacionais sobre imagens, (GONZALEZ; WOODS, 2006). Esta área gera um grande interesse por estar presente em um grande número de aplicações em duas categorias distintas: *i.* o aprimoramento de informações das próprias imagens; e *ii.* a análise automática de informações extraídas de uma cena (MARQUES; VIEIRA, 1999).

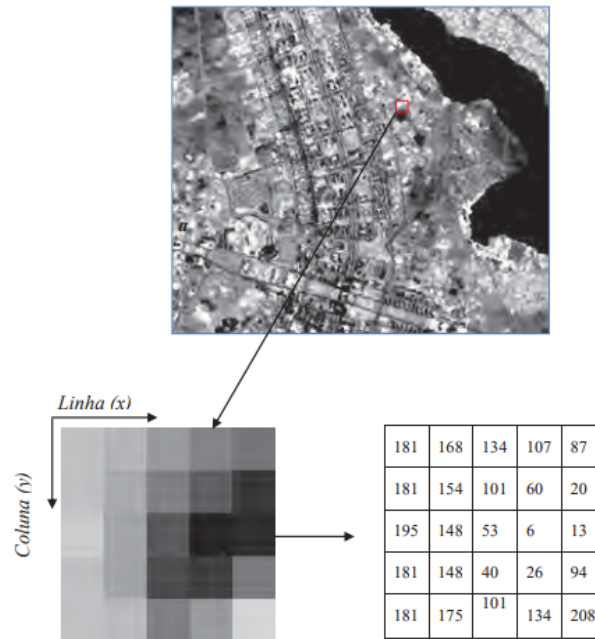
Uma imagem pode ser definida como uma função bidimensional, $f(x, y)$, onde x e y são coordenadas espaciais e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou tom de cinza naquele ponto. Quando x , y e os valores de intensidade de f são todos finitos e grandezas discretas, a imagem é chamada de imagem digital. Uma imagem digital é composta por um número finito de elementos, onde cada um possui uma localização e valor. Estes elementos são chamados de *picture elements* ou *pixels* (GONZALEZ; WOODS, 2006).

Há diversas maneiras de realizar a aquisição de imagens, porém a saída da maioria dos sensores de captação é uma forma de onda contínua. Dessa forma, para criar uma imagem digital é necessário digitalizar os dados obtidos por estes sensores, o que é feito através de dois processos: *i.* amostragem, que consiste em digitalizar os valores das coordenadas; e *ii.* quantização, que consiste em digitalizar o valor das amplitudes. Assim, é possível converter uma imagem contínua em um *array* bidimensional, ou matriz, $f(x, y)$, contendo M colunas e N linhas, onde (x, y) são coordenadas discretas chamadas de *pixels* (GONZALEZ; WOODS, 2006). A [Figura 3](#) ilustra a representação matricial de uma região

10 x 10 de uma imagem em tom de cinza, logo cada pixel está representado no intervalo $[0, 255]$, onde 0 representa a cor preta e 255, a branca.

As subseções a seguir apresentam fundamentos e teorias relevantes sobre processamento de imagens.

Figura 3 – Representação matricial de uma imagem digital



Fonte: [Meneses e Almeida \(2012\)](#)

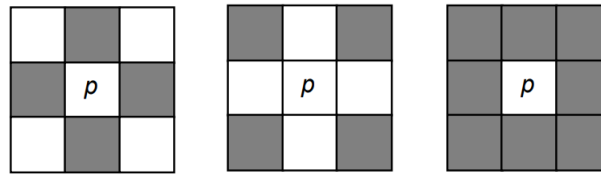
2.2.1 Relacionamentos entre Pixels

Nesta seção serão descritas as principais relações entre *pixels* em uma imagem digital.

2.2.1.1 Vizinhança

Um pixel p nas coordenadas (x, y) possui quatro vizinhos horizontais e verticais, cujas coordenadas são dadas por $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$. Este conjunto de *pixels*, chamado de 4-vizinhança, é denotado por $N_4(p)$. Cada elemento deste conjunto está a uma unidade de distância de (x, y) , e pode estar localizado fora da imagem, caso o pixel p esteja na borda da imagem ([GONZALEZ; WOODS, 2006](#)). Os quatro vizinhos diagonais de p , denotados $N_D(p)$, possuem coordenadas $(x + 1, y + 1)$, $(x + 1, y - 1)$, $(x - 1, y + 1)$, $(x - 1, y - 1)$. A união dos conjuntos $N_4(p)$ e $N_D(p)$ constitui o conjunto $N_8(p)$, chamado de 8-vizinhança ([GONZALEZ; WOODS, 2006](#)). Os conjuntos $N_4(p)$, $N_D(p)$ e $N_8(p)$ estão, respectivamente, ilustrados na [Figura 4](#)

Figura 4 – Vizinhanças de um pixel



Fonte: Marques e Vieira (1999)

2.2.1.2 Caminho

Segundo (GONZALEZ; WOODS, 2006), um caminho de um pixel p de coordenadas (x, y) a um pixel q de coordenadas (s, t) é uma sequência de pixels distintos de coordenadas: $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, onde:

- $(x_0, y_0) = (x, y)$;
- $(x_n, y_n) = (s, t)$;
- (x_i, y_i) é adjacente a (x_{i-1}, y_{i-1}) ;
- $1 \leq i \leq n$ e;
- n é chamado de comprimento do caminho.

2.2.1.3 Adjacência

O conceito de adjacência considera, além da vizinhança, os valores dos *pixels* próximos. Dessa forma, para determinar se dois *pixels* são adjacentes, é necessário levar em consideração alguma medida de similaridade. Em uma imagem binária, por exemplo, dois *pixels* podem ser 4-vizinhos, mas só serão 4-adjacentes se possuírem o mesmo valor (GONZALEZ; WOODS, 2006).

Existem três tipos de adjacência e, para os definir, levaremos em consideração um conjunto V de valores utilizados para definir a adjacência. Em imagens binárias, $V = \{1\}$ para adjacência de *pixels* com valor 1. Em imagens em tons de cinza a ideia é a mesma, porém V pode ser um subconjunto de valores entre 0 e 255 (GONZALEZ; WOODS, 2006). Dessa forma, os três tipos de adjacências são dados por:

- a. 4-adjacência: dois *pixels* p e q com valores de tom de cinza contidos em V , são 4-adjacentes se $q \in N_4(p)$.
- b. 8-adjacência: dois *pixels* p e q com valores de tom de cinza contidos em V , são 8-adjacentes se $q \in N_8(p)$.

c. m-adjacência (adjacência mista): dois *pixels* p e q com valores de tom de cinza contidos em V , são m-adjacentes se:

- (i) $q \in N_4(p)$, ou
- (ii) $q \in N_d(p)$ e $V \cap (N_4(p) \cap N_4(q)) = \emptyset$

2.2.1.4 Conectividade

A conectividade entre *pixels* é um conceito utilizado para estabelecer limites de objetos e componentes de regiões em uma imagem (MARQUES; VIEIRA, 1999). Dois *pixels* p e q são conexos em S , onde S representa um subconjunto de *pixels* da imagem, se existir um caminho entre eles constituído apenas de *pixels* contidos em S . Para qualquer *pixel* em S , o conjunto de *pixels* que estão conectados a ele em S é chamado de componente conexo, e caso exista apenas um componente conexo em S , então S é um conjunto conexo (GONZALEZ; WOODS, 2006).

2.2.1.5 Regiões

Seja R um subconjunto de *pixels* de uma imagem. R será uma região dessa imagem se ele for um conjunto conectado. Duas regiões R_i e R_j são ditas adjacentes se a união delas formar um conjunto conectado. Regiões não adjacentes são chamadas de disjuntas. Supondo que uma imagem contenha K regiões disjuntas, R_k , $k = 1, 2 \dots K$, das quais nenhuma toque na borda da imagem. Seja R_u a união de todas as K regiões e $(R_u)^c$ o seu complemento, os pontos em (R_u) são chamados de objeto (*foreground*) da imagem e $(R_u)^c$, de fundo (*background*) (GONZALEZ; WOODS, 2006).

2.2.2 Operações Lógicas e Aritméticas

As operações aritméticas entre imagens são operações vetoriais, o que implica que elas são realizadas sobre pares de pixels correspondentes. Dessa forma, é necessário que as imagens possuam o mesmo tamanho (GONZALEZ; WOODS, 2006). As quatro operações aritméticas são denotadas como:

$$s(x, y) = f(x, y) + g(x, y) \quad (2.11)$$

$$s(x, y) = f(x, y) - g(x, y) \quad (2.12)$$

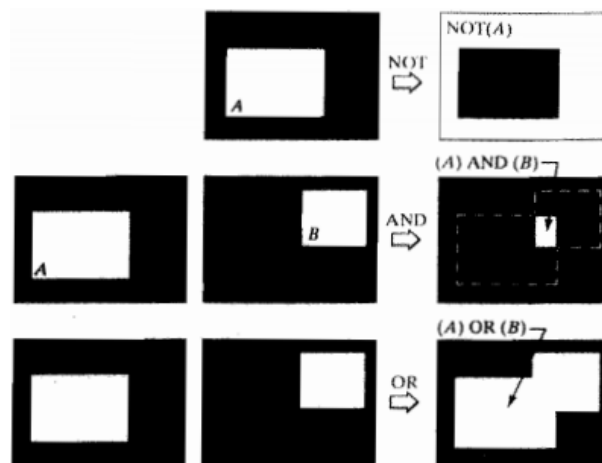
$$s(x, y) = f(x, y) \times g(x, y) \quad (2.13)$$

$$s(x, y) = f(x, y) \div g(x, y) \quad (2.14)$$

onde f e g são funções que representam as imagens.

As operações lógicas, por sua vez, podem ser aplicadas em imagens binárias, onde o 0 representa o fundo, e 1, os objetos presentes na imagem. As operações mais comuns são *OR*, *AND* e *NOT*, as quais podem ser entendidas, respectivamente, como união, intersecção e complemento da teoria dos conjuntos (GONZALEZ; WOODS, 2006). Para exemplificar essas operações, tomaremos como exemplo os conjuntos A e B da Figura 5, ambas compostas por *pixels* de objeto. A operação *OR* entre esses dois conjuntos tem como resultado um conjunto com os os *pixels* presentes em A e B. A operação *AND* resulta em um conjunto com os *pixels* comuns a A e B, e uma operação *NOT* em A, resulta em um conjunto de pixels que não estão em A.

Figura 5 – Operações lógicas com imagens binárias



Adaptado de: (GONZALEZ; WOODS, 2006)

2.2.3 Espaços de Cores

A luz visível é uma radiação eletromagnética que possui um comprimento de onda perceptível ao sistema visual humano. Este comprimento de onda determina a tonalidade da luz, enquanto a amplitude pode ser entendida como intensidade. Geralmente, a luz não é apenas uma onda, mas a combinação de todas as ondas em um espectro visível. Qualquer três ondas de luz podem ser combinadas para criar outras cores. As três cores primárias que podem ser combinadas para produzir um maior número de cores são azul, vermelho e verde (JOBLOVE; GREENBERG, 1978).

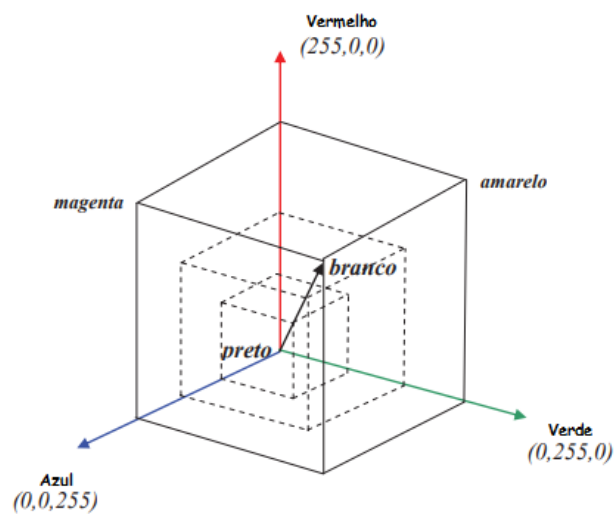
O objetivo dos espaços de cores (ou sistemas de cores) é facilitar a especificação de cores em algum padrão. Basicamente, eles são uma especificação de um sistema de coordenadas e um subespaço dentro desse espaço onde cada cor é representada por um

único ponto (GONZALEZ; WOODS, 2006). As subseções a seguir vão descrever os espaços de cores utilizados neste trabalho de pesquisa.

2.2.3.1 RGB

O sistema de cores RGB é baseado em um sistema de coordenadas cartesiano. Ele utiliza os componentes espectrais primários de vermelho, verde e azul para representar as cores. A Figura 6 ilustra o cubo cartesiano do modelo RGB.

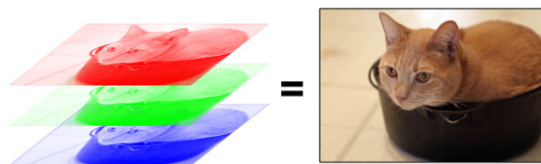
Figura 6 – Espaço de cores RGB



Adaptado de: Meneses e Almeida (2012)

Imagens representadas nesse modelo são compostas por três componentes (canais), um para cada cor primária, como ilustrado na Figura 7, e cada pixel é um trio de 8 bits, os quais são utilizados para representar as tonalidades de cada uma dessas cores (GONZALEZ; WOODS, 2006).

Figura 7 – Canais RGB

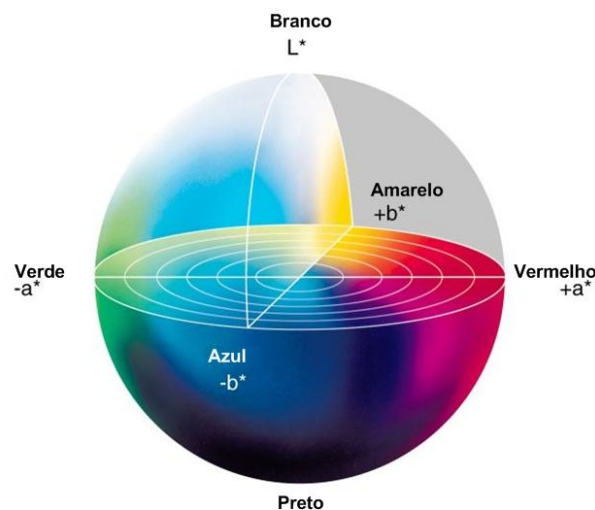


Fonte: (BATRACK, 2010)

2.2.3.2 CIE L*a*b

O espaço de cores CIE L*a*b utiliza três componentes (ou eixos) para representar as cores. O componente 'L' representa a luminosidade, o componente 'a' representa o verde/vermelho e o componente 'b', azul/amarelo (WEATHERALL; COOMBS, 1992), como ilustrado na Figura 8. Nesse modelo, os valores do eixo L são limitados entre 0 (preto) e 100 (branco), de modo que ($0 \leq L \leq 100$), porém não há limites explícitos definidos para os eixos a* e b* (HILL; ROGER; VORHAGEN, 1997). A principal vantagem deste sistema de cor é que as cores com valores (L,a,b) próximos são cores parecidas no sólido de cor, desse modo, é possível utilizar distâncias no espaço Euclidiano para verificar a similaridade entre as cores.

Figura 8 – Espaço de cores CIE L*a*b



(SANTOS et al., 2012)

2.2.4 Histograma

O histograma de uma imagem digital com níveis de intensidade contidos no intervalo $[0, L - 1]$, onde L é o número de níveis de cinza presentes na imagem, é uma função discreta a qual indica a quantidade de *pixels* que possuem um certo nível de tom de cinza (GONZALEZ; WOODS, 2006). A Equação 2.15 descreve esta função.

$$h(r_k) = n_k \quad (2.15)$$

onde:

- r_k representa o k-ésimo valor de intensidade; e
- n_k é a quantidade de *pixels* de intensidade r_k .

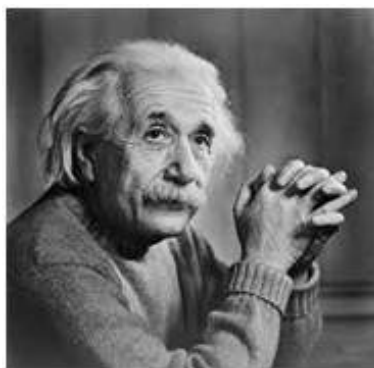
Os histogramas são a base para um grande número de técnicas de processamento de imagens. Eles podem ser utilizados, por exemplo, aprimoramento de imagens, segmentação, e ainda oferece estatísticas úteis das imagens (GONZALEZ; WOODS, 2006). A Tabela 1 mostra um exemplo de histograma. Os seus dados correspondem a uma imagem de 128 x 128 *pixels* com 8 níveis de cinza; na primeira coluna estão os níveis de cinza e na segunda, a quantidade de pixels correspondente. Histogramas também podem ser representados graficamente, como ilustrado na Figura 9, onde a Figura 9b representa o histograma da Figura 9a.

Tabela 1 – Exemplo de Histograma.

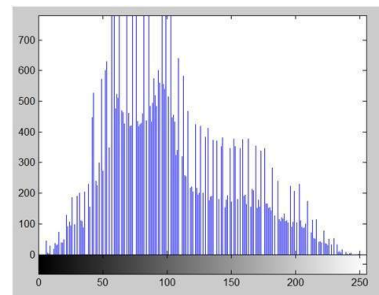
Nível de cinza (r_k)	n_k
0	1120
1/7	3214
2/7	4850
3/7	3425
4/7	1995
5/7	784
6/7	541
1	455
Total	16384

Fonte – Marques e Vieira (1999)

Figura 9 – Exemplo de Histograma



(a) Imagem em tom de cinza



(b) Histograma

(POINT, 2017)

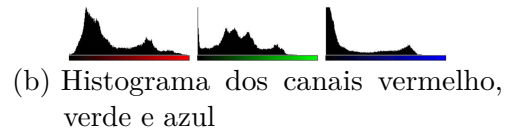
Este conceito também pode ser aplicado para imagens coloridas. Nesses casos, a imagem é decomposta de alguma forma, tal como seus componentes RGB, e um histograma

é calculado para cada componente (MARQUES; VIEIRA, 1999), como ilustrado na Figura 10.

Figura 10 – Exemplo de histograma de imagem colorida



(a) Imagem colorida



(b) Histograma dos canais vermelho, verde e azul

Fonte: Marques e Vieira (1999)

2.2.5 Segmentação de Imagens

Segmentação é um processo que subdivide uma imagem em seus objetos ou regiões constituintes. O nível de detalhe das subdivisões depende do problema em questão, ou seja, a segmentação deve acabar quando os objetos ou regiões de interesse forem detectados.

As subseções a seguir detalham os fundamentos e algumas técnicas de segmentação.

2.2.5.1 Fundamentos

Seja R uma região espacial completamente ocupada por uma imagem, a segmentação de imagens pode ser entendida como um processo que particiona R em n sub-regiões, R_1, R_2, \dots, R_n (GONZALEZ; WOODS, 2006), de modo que:

- a. $\bigcup_{i=1}^n R_i = R$.
- b. R_i é um conjunto conexo, $i = 1, 2, \dots, n$.
- c. $R_i \cap R_j = \emptyset$, para todo i e j , $i \neq j$.
- d. $Q(R_i) = VERDADEIRO$ para $i = 1, 2, \dots, n$.
- e. $Q(R_i \cup R_j) = FALSO$ para quaisquer regiões R_i e R_j adjacentes.

A condição 'a.' indica que cada pixel da imagem tem de pertencer a alguma região. A condição 'b.' requer que os pontos em uma região estejam conectados de alguma maneira predefinida, tal como 4- ou 8-conexo. A condição 'c.' requer que as regiões sejam disjuntas. A condição 'd.' trata das propriedades que devem ser satisfeitas pelos *pixels* de uma região (por exemplo, $Q(R_i) = VERDADEIRO$ se todos os *pixels* em R_i tiverem o mesmo nível

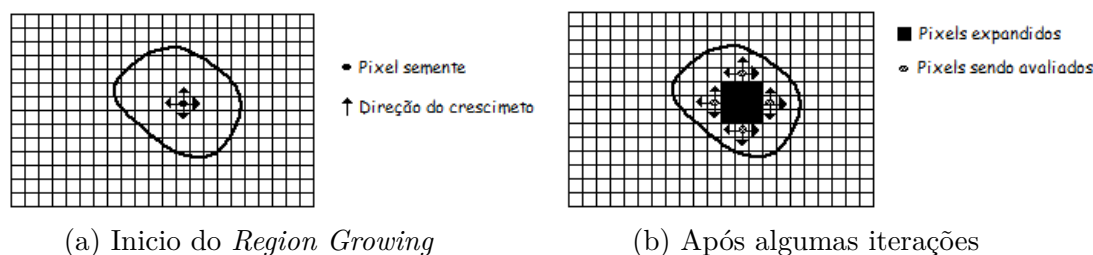
de intensidade); e a condição 'e.' indica que tuas regiões adjacentes R_i e R_j devem ser diferentes, considerando o predicado Q (GONZALEZ; WOODS, 2006).

2.2.5.2 Região Crescente (*Region Growing*)

Esta técnica, como indicado pelo nome, agrupa *pixels* ou sub-regiões em regiões maiores baseado em algum critério predefinido para o crescimento. A ideia básica desta técnica é iniciar com um conjunto de pontos iniciais, chamados de sementes, e a partir deles crescer regiões ao anexar à cada semente os *pixels* vizinhos que possuem propriedades semelhantes às da semente (tais como cor, textura, etc) (GONZALEZ; WOODS, 2006).

A seleção de um conjunto de uma ou mais sementes geralmente é baseada na natureza do problema. Quando não há uma informação prévia, o procedimento consiste em computar em cada pixel o mesmo conjunto de propriedades que será utilizado para determinar se um pixel pertence a uma região durante o processo de crescimento. Se o resultado desse procedimento formar *clusters* de valores, então as sementes podem ser os pixels cujas propriedades os coloquem mais próximos dos centroides do *clusters* formados (GONZALEZ; WOODS, 2006). O crescimento da região deve parar quando mais nenhum *pixel* satisfazer o critério de inclusão naquela região. A Figura 11 ilustra o *Region Growing*. Na Figura 11a temos o estado inicial após a seleção da semente, e na Figura 11b o processo de *Region Growing* após algumas iterações.

Figura 11 – *Region Growing*



Traduzido de: (MARSHALL, 2017)

2.2.5.3 Segmentação de imagens por algoritmos de agrupamento

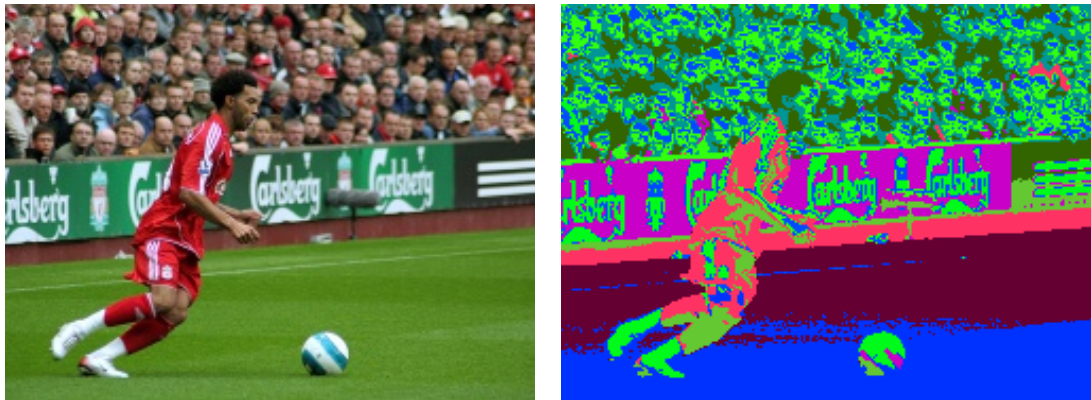
Como visto na seção 2.1, os algoritmos de agrupamento visam agrupar os dados de acordo com a descrição dos objetos considerados. Quando aplicados na segmentação de imagens, esses algoritmos podem usar características como cor, posição do pixels, textura, etc (JAIN; DUBES, 1988). A segmentação de imagens tem a mesma relação com a classificação de imagem que o aprendizado não supervisionado possui com o aprendizado supervisionado (JAIN; DUBES, 1988). Os algoritmos de agrupamento são, na sua maioria, não-supervisionados, todavia, alguma informação fornecida pelo usuário, através de cliques na imagem indicando um objeto de interesse, por exemplo, pode ser utilizada como entrada

para algoritmos de agrupamento semissupervisionado. A vantagem, neste caso, é que uma pequena interação com usuário pode promover ganhos na qualidade da segmentação.

Neste trabalho são utilizados os vetores referentes aos canais de cores, levando em consideração os espaços de cores RGB e CIE L^*a^*b , os quais foram descritos na [subseção 2.2.3](#). O objetivo do algoritmo é, então, agrupar os pixels que possuem cores mais semelhantes no mesmo grupo, enquanto *pixels* menos com cores menos semelhantes serão agrupados em outros grupos.

A [Figura 12](#) ilustra um exemplo de segmentação por agrupamento, no qual foi utilizado o algoritmo FCM ([Algoritmo 1](#)). Na [Figura 12a](#) é apresentada a imagem original, enquanto a [Figura 12b](#) ilustra o resultado, onde cada cor representa um grupo. Neste exemplo há um total de 10 grupos.

Figura 12 – Exemplo de segmentação por agrupamento



(a) Imagem original

(b) Resultado do agrupamento

Fonte: O autor

3 Trabalhos relacionados

Abaixo, estão descritos alguns trabalhos que abordam o problema de segmentação de imagens.

No trabalho de Busin *et al* (BUSIN *et al.*, 2009), é proposto um método de segmentação baseada na escolha do espaço de cor. Neste método, o histograma 3D da imagem é calculado em diferentes espaços de cores. Em seguida, é aplicado um agrupamento espectral (*spectral clustering*) nesses espaços de cores distintos. O espaço de cores que tiver melhores resultados, considerando uma medida de precisão, será selecionado. Então, os *pixels* da imagem no espaço de cor selecionado, serão rotulados para produzir a segmentação. O método proposto foi testado utilizando as imagens do banco de dados Berkeley, e utilizaram o índice Jaccard para comparar os resultados das segmentações quando utilizado o espaço de cores selecionado pelo método proposto, ou o espaço de cores RGB e constataram que, em média, o método proposto obteve melhores resultados, destacando a relevância do espaço de cores para a segmentação de imagens.

Tan e Nor (TAN; ISA, 2011), propõem um novo método de segmentação de imagens coloridas utilizando uma abordagem híbrida baseada na limiarização de histograma e *fuzzy c-means*. A etapa de limiarização do histograma é dividida em três fases, i. descoberta de pico (*peek finding*); ii. inicialização de região (*region initialization*) e iii. mesclagem (*merging*). Na primeira fase, os picos dominantes no histograma global são identificados. Na segunda fase os *pixels* são rotulados em regiões, utilizando os conjuntos de picos obtidos na primeira fase. O algoritmo dessa fase realiza a criação dos centroides, a atribuição de cada pixel da imagem para o centroide mais próximo, a criação do conjunto de pixel de cada *cluster* a remoção dos centroides que possuem uma quantidade de *pixels* a eles atribuídos menor que um limiar V e a reatribuição dos *pixels* aos seus *clusters* mais próximos. Na terceira fase, os grupos que são muito próximos são unidos para obter um melhor resultado. Após a limiarização do histograma, o algoritmo FCM é utilizado para aprimorar a compacidade dos grupos obtidos.

Zaixin, Cheng e Cheng (ZAIXIN; LIZHI; GUANGQUAN, 2013) propõem um método para segmentação baseado em uma adaptação do algoritmo FCM. Nesta adaptação, é proposto um novo critério de distância para calcular a similaridade dos *pixels*. Eles introduziram uma nova medida de peso, que leva em consideração as diferenças nas regiões da imagem e estatísticas locais, chamada de NW (*neighborhood weighted*). Comparado com a distância Euclidiana, o critério proposto fornece uma medida de distância mais robusta e exata, o que pode levar a melhores resultados, e sendo menos prejudicado por ruídos, quando comparado com outras técnicas. Neste trabalho, porém, foram consideradas

apenas imagens em tons de cinza.

Raval, Ravi e Ankit (RAVAL; SHUKLA; SHAH, 2017) fizeram um estudo de segmentação de imagens utilizando o algoritmo FCM e os espaços RGB, L*a*b, HSV e YIQ, para avaliar qual deles obtém o melhor resultado de segmentação. Segundo os testes, a segmentação utilizando o espaço de cores RGB precisa, na maioria dos casos, de um maior número de iterações, afetando o tempo de execução. Todavia, o estudo não deixa claro qual espaço de cores obtém o melhor resultado na segmentação das regiões da imagem, apenas indica com qual o FCM precisa de menos iterações.

No método proposto em (JAFARI; SAMAVI, 2015) é pedido ao usuário que declare classes para regiões previamente extraídas da imagem através de algum classificador não supervisionado baseado nos valores RGB dos pixels. Em seguida, os *pixels* que possuem menor distância para a média do grupo são encontrados e, juntamente às classes fornecidas pelo usuário, são utilizados como semente inicial para o treinamento de um classificador supervisionado. Essa semente é expandida, levando em consideração a similaridade com os *pixels* adjacentes. Esse processo de expansão pode ser realizado até que haja uma quantidade satisfatória de amostras. Então, um classificador *Support Vector Machine* (SVM) é treinado, utilizando um conjunto estendido de dados, fornecido pelos rótulos do usuário. O classificador é aplicado aos valores RGB dos *pixels* da imagem, obtendo a segmentação.

No trabalho de Qian e Si (QIAN; SI, 2005), foi apresentado um método de segmentação de imagens coloridas utilizando o algoritmo semissupervisionado *Gaussian Mixture Model* (GMM) e o algoritmo de agrupamento EM (Dempster; Laird and Rubin, 1977). Neste método, o algoritmo realiza uma quantização das cores da imagem e armazena as cores em uma tabela de cores. Em seguida, o usuário realiza cliques na imagem para indicar regiões do mesmo *cluster*, gerando, dessa forma, alguns rótulos para os dados. A vizinhança dos pontos selecionados também é adicionado aos conjuntos de coordenadas. Essa vizinhança é selecionada de acordo com uma janela retangular de tamanho fixo. O algoritmo realiza a segmentação aplicando o EM com o GMM na tabela de cores, dividindo-a em M grupos, que são as regiões da imagem.

O trabalho de Sankaei e Chandrasekar (SANKARI; CHANDRASEKAR, 2012) apresenta um método de segmentação de imagens coloridas que utiliza agrupamento semissupervisionado, com foco na extração do objeto de interesse (OOI) de uma imagem, onde o usuário realiza marcações do objeto de interesse e dos objetos de não interesse para fornecer dados. Em seguida, é utilizada fórmula de Mahalanobis para calcular a distância dos *pixels* das marcações para os seus vizinhos e determinar se eles pertencem à mesma região. Então, o algoritmo EM é utilizado, juntamente com restrições *must-link*, para agrupar a tabela de cores. Essas restrições indicam que pares de cores determinadas devem terminar no mesmo grupo.

Em outro trabalho, Jia e Zang (JIA; ZHANG, 2008) propuseram um método de aprendizado de uma métrica distância para encontrar os melhores parâmetros para segmentação semissupervisionada baseada em grafos, visando solucionar o problema de seleção de parâmetros existentes nesses métodos. Neste trabalho, a rotulação dos dados é feita através do usuário, que realiza marcas na imagem para indicar o objeto e o fundo da imagem. Esse método obteve melhores resultado do que as abordagens que utilizam parâmetros heurísticos. Contudo, não foram considerados outros espaços de cores nessa pesquisa.

Wang et al (WANG et al., 2006) propuseram um novo método de segmentação semissupervisionado utilizando propagação linear de vizinhança (LNP - *Linear Neighborhood Propagation*). Esse método se baseia na afirmação de que os rótulos de cada dado podem ser linearmente reconstruídos através dos rótulos dos seus vizinhos. O seu objetivo é aprender os pesos de construção linear e prever os rótulos de dados não rotulados através de dados rotulados. Os rótulos são fornecidos através de marcações na imagem. O processo principal se resume a duas etapas: computar a geometria da vizinhança local através dos rótulos fornecidos; utilizar essa geometria para prever os rótulos dos dados não rotulados. Este método possui bons resultados, porém, para tanto, requer muitas rotulações.

Noordman, Broek e Buydens (NOORDAM; BROEK; BUYDENS, 2000), propõem um algoritmo FCM guiado geometricamente, o GGFCM (*Geometrically Guided FCM*), baseado em uma técnica FCM semissupervisionada para segmentação de imagens multivariadas. Neste trabalho, todos os objetos são considerados rotulados, e a condição de cada pixel é determinada pelos valores associados aos *pixels* vizinhos, e então o *pixel* é adicionado ou removido do *cluster*. Entretanto, os autores só ilustram dois resultados. No primeiro, é utilizada uma imagem sintética com a presença de ruído Gaussiano, e o segundo utiliza a imagem com ruído de pimenta. No primeiro experimento, o algoritmo proposto obteve um resultado muito superior comparado ao FCM clássico, pois conseguiu segmentar quase perfeitamente, enquanto o FCM foi prejudicado pelo ruído. No segundo experimento, o FCM obteve uma maior presença de bordas falsas quando comparado ao GGFCM.

O primeiro trabalho que aplica uma versão adaptada do FCM para segmentação de imagens foi apresentado em Bensaid et al. (1996). Neste trabalho, foi proposto o algoritmo de agrupamento semissupervisionado ssFCM, o qual utiliza dados rotulados e não-rotulados em conjunto com distância ponderada para calcular o grau de pertinência para o agrupamento e segmentação de imagens de ressonância magnética. Recentemente, (SON; TUAN, 2016) utilizaram o algoritmo de agrupamento semissupervisionado baseado em entropia (ESSC) para segmentar imagens de raio-X dentárias; e (LI et al., 2017) propõe um método para segmentação de imagens de rostos 3D que utiliza um mapeamento

de detector de características Harris como entrada para um algoritmo de agrupamento semissupervisionado proposto.

Como visto nos trabalhos relacionados acima, há várias técnicas de segmentação de imagens presentes na literatura, e vários trabalhos relatam a eficiência do FCM para segmentação de imagens e alguns deles utilizam a aprendizagem semissupervisionada como algoritmo de segmentação. Entretanto, não há uma abordagem melhor que outras e vários processos relacionados à segmentação com o uso desses algoritmos influenciam no resultado da segmentação. Por exemplo, qual sistema de cor utilizar e quais informações o usuário deve rotular para guiar o algoritmo semissupervisionado.

4 Metodologia

Neste capítulo são apresentadas as configurações dos experimentos, a medida de avaliação para análise dos resultados e as ferramentas utilizadas para o desenvolvimento deste trabalho.

4.1 Configuração experimental

As subseções a seguir descrevem a base de dados de imagens utilizada para avaliação dos algoritmos de agrupamento e as ferramentas de implementação e avaliação dos resultados.

4.1.1 Base de dados

A base de dados utilizada nesse projeto, a CMU-Cornell iCoseg Dataset ([BATRA et al., 2011](#)), é composta por imagens de 38 objetos diferentes com cerca de 17 imagens cada, onde cada imagem possui um *groundtruth* binário (marcação feita por humano que corresponde ao(s) objeto(s) de interesse da imagem) para avaliação dos resultados. A [Figura 13](#) apresenta uma imagem dessa base de dados e seu respectivo *groundtruth*.

4.1.2 Ferramentas e ambiente de programação

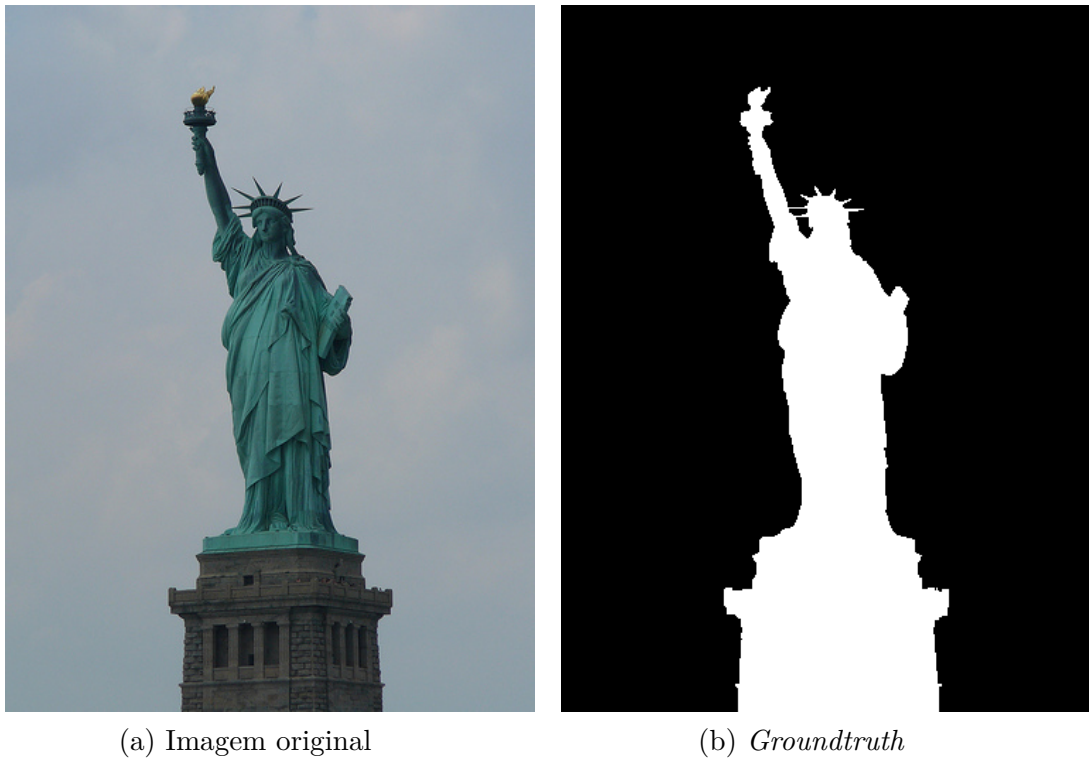
Os algoritmos de agrupamento, processamento de imagens e a ferramenta de rotulação parcial de imagens foram desenvolvidos na linguagem de programação Java¹ versão 8, com auxílio da biblioteca de computação visual OpenCV² versão 3.3.

4.1.3 Metodologia de avaliação

Em uma imagem de teste na qual foi aplicada a segmentação, qualquer *pixel* que for agrupado como objeto e que também seja objeto no seu respectivo *groundtruth* será chamado de VP (verdadeiro positivo), pois foi corretamente classificado como objeto. *Pixels* que foram agrupados como fundo e que também sejam fundo no respectivo *groundtruth* serão chamados de VN (verdadeiro negativo), pois foram corretamente classificados como fundo. *Pixels* classificados como objeto, porém que correspondem ao fundo no *groundtruth* são chamados de FP (falso positivo), pois foram classificados erroneamente como objeto, da mesma forma que *pixels* classificados como fundo, porém que são objeto no *ground*

¹ <https://www.java.com>

² <https://opencv.org/>

Figura 13 – Exemplo de imagem do *dataset* CMU Cornell iCoseg

Disponível em: <http://chenlab.ece.cornell.edu/projects/touch-coseg/>

truth são chamados de FN (falso negativo), pois o *pixel* foi erroneamente classificado como fundo, como mostra a matriz de confusão ilustrada na figura [Figura 14](#).

Figura 14 – Matriz de confusão

		Valor Verdadeiro (confirmado por análise)	
		positivos	negativos
Valor Previsto (predito pelo teste)	positivos	VP Verdadeiro Positivo	FP Falso Positivo
	negativos	FN Falso Negativo	VN Verdadeiro Negativo

Traduzido de: ([MANTHIRI, 2017](#))

Considerando as colocações acima, as medidas de avaliação utilizadas foram especificidade, sensibilidade e MCC (Coeficiente de Correlação de Matthews). Essas medidas

estão descritas nas equações (4.2), (4.1) e (4.3), respectivamente.

$$\text{Especificidade} = \frac{VN}{VN + FP} \quad (4.1)$$

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \quad (4.2)$$

$$\text{MCC} = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \quad (4.3)$$

A especificidade indica o quão bem os algoritmos conseguem agrupar os *pixels* referentes ao fundo, enquanto a sensibilidade indica o quão bem os algoritmos conseguem agrupar os *pixels* referentes ao objeto. Os valores da sensibilidade e especificidade estão entre 0 e 1, onde valores maiores indicam melhores resultados, enquanto o valor do MCC está entre -1 e 1, onde -1 indica desconformidade total entre o resultado da segmentação e o *groundtruth*, 0 indica aleatoriedade, e 1 indica conformidade total, ou seja, uma segmentação perfeita.

Para validação dos resultados, foi realizado o teste estatístico de Wilcoxon (WILCOXON, 1945) para os melhores resultados que cada algoritmo obteve. Esse é um teste pareado não paramétrico que permite avaliar o desempenho entre duas amostras relacionadas (no contexto desse trabalho, os resultados dos algoritmos) para avaliar se elas diferem. Ele realiza a classificação de acordo com a diferença absoluta entre os resultados de cada algoritmo para cada base de dados, ordenando as diferenças em ordem crescente. Em caso de empate, é calculada a média da classificação dos casos que empataram. Os valores dos MCC's foram os utilizados neste teste, pois eles apresentam uma métrica mais precisa, uma vez que levam em consideração todos os valores das matrizes de confusão.

4.2 Métodos de rotação parcial de imagem

Como citado na seção 1.1, este trabalho visa avaliar diferentes formas de rotação parcial de imagem. A partir dos rótulos iniciais fornecidos pelo usuário (através de cliques na imagem), foram testadas três formas de rotação parcial, são elas: rotação fixa em formato quadrado de lado 20 *pixels*, rotação fixa em formato quadrado de lado 30 *pixels*, e rotação dinâmica baseada em Região Crescente.

Na rotação fixa em formato quadrado, um quadrado de lado l (determinado pelo usuário) é formado a partir dos rótulos iniciais, e todos os *pixels* dentro desses quadrados são rotulados como objeto ou fundo, dependendo da escolha do usuário. Na rotação baseada em Região Crescente, os rótulos iniciais são expandidos considerando

alguma medida de similaridade em torno da 4-vizinhança de um *pixel*, como visto na subseção 2.2.5.2.

A medida de similaridade utilizada foi a Distância Euclidiana entre os vetores (RGB ou CIE L*a*b) de cores normalizados dos *pixels*, onde um *pixel* será considerado adjacente a outro, caso a distância seja menor que um limiar T . Este método possui uma restrição a qual indica que cada expansão deve rotular pelo menos 100 *pixels* para cada rótulo inicial. Caso essa condição não seja satisfeita para algum rótulo, então esse rótulo será expandido novamente com um limiar $T = T + 0.01$ até que essa restrição seja satisfeita. Essa abordagem pode ser tida como válida, pois, uma vez que os rótulos iniciais são fornecidos por usuários, então essas rótulos iniciais estão corretos. Uma vez que nem todos os dados precisam estar rotulados, essa expansão tem um limite de 100 iterações (definido empiricamente).

As estratégias de rotulação estão ilustradas na Figura 15. A Figura 15a apresenta os rótulos iniciais fornecidos pelos usuários, onde os pontos vermelhos representam as marcações do objeto e os pontos pretos as marcações do fundo da imagem. As figuras 15b, 15c e 15d apresentam as imagens após as expansões através de região crescente, formato de quadrado ($l = 20$) e formato de quadrado ($l = 30$), respectivamente.

Para que essa etapa de rotulação fosse concluída, foi desenvolvida uma ferramenta de software, a qual está descrita na seção 4.3.

4.3 Ferramenta de rotulação

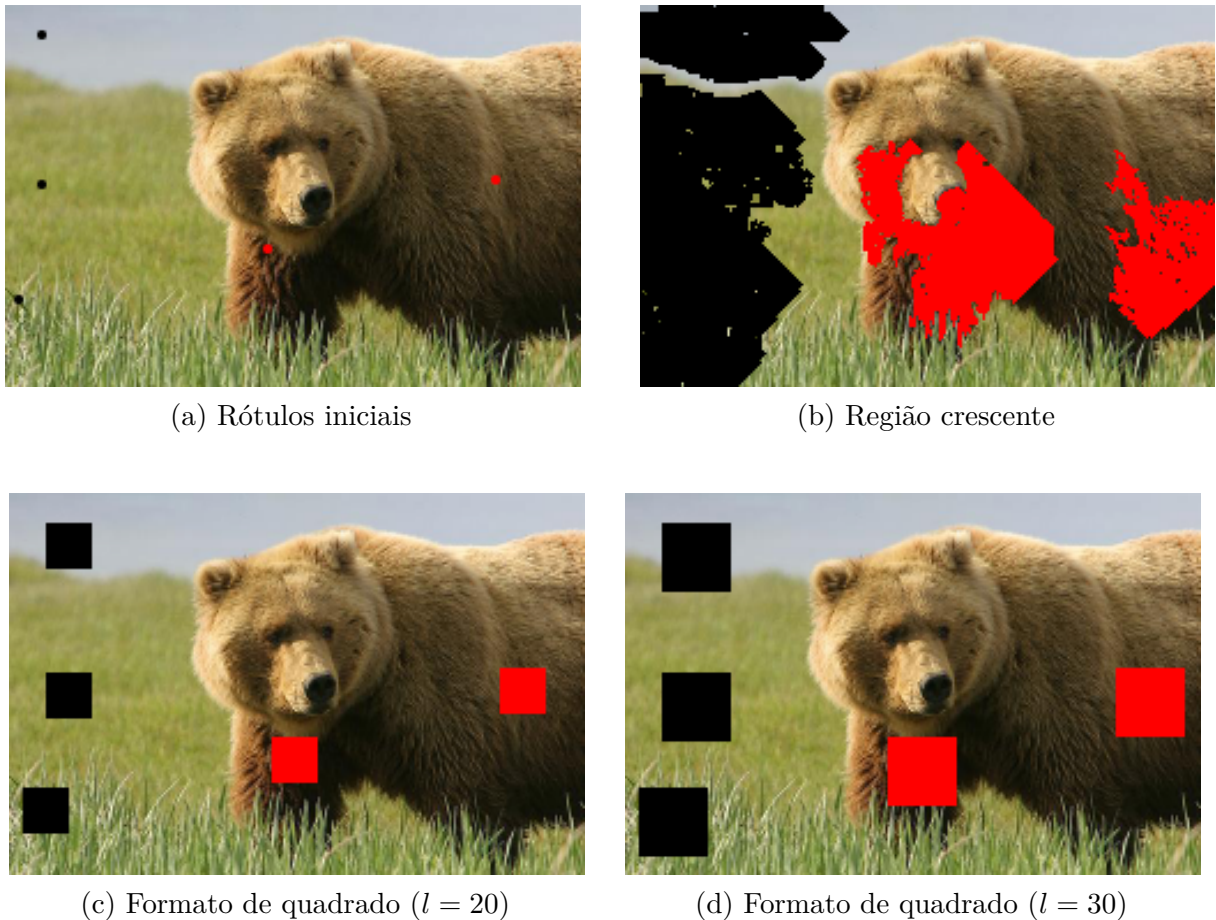
Esta ferramenta é responsável por gerar os arquivos ARFF³ (*Attribute-Relation File Format*) que são as entradas para os algoritmos de agrupamento. Cada arquivo ARFF gerado contém a identificação do padrão, que neste caso é a coordenada (x, y) do *pixel*, os valores do vetor de cores (RGB ou CIE L*a*b), e na última coluna, a classe a qual o padrão (x, y) pertence, caso esta coordenada tenha sido rotulada.

A ferramenta de rotulação parcial está ilustrada na Figura 16. As opções e funcionalidades principais da ferramenta estão destacadas e enumeradas e serão explicadas abaixo:

1. Esta opção define qual é o tipo do rótulo: fundo ou objeto.
2. Esta opção define a estratégia de expansão de rótulos: quadrado (*square*) ou região crescente (*region growing*).
3. Esta opção define o espaço de cores utilizado para gerar o arquivo ARFF: RGB ou CIE L*a*b.

³ <https://www.cs.waikato.ac.nz/ml/weka/arff.html>

Figura 15 – Exemplo dos formatos de rotulação parcial de imagens



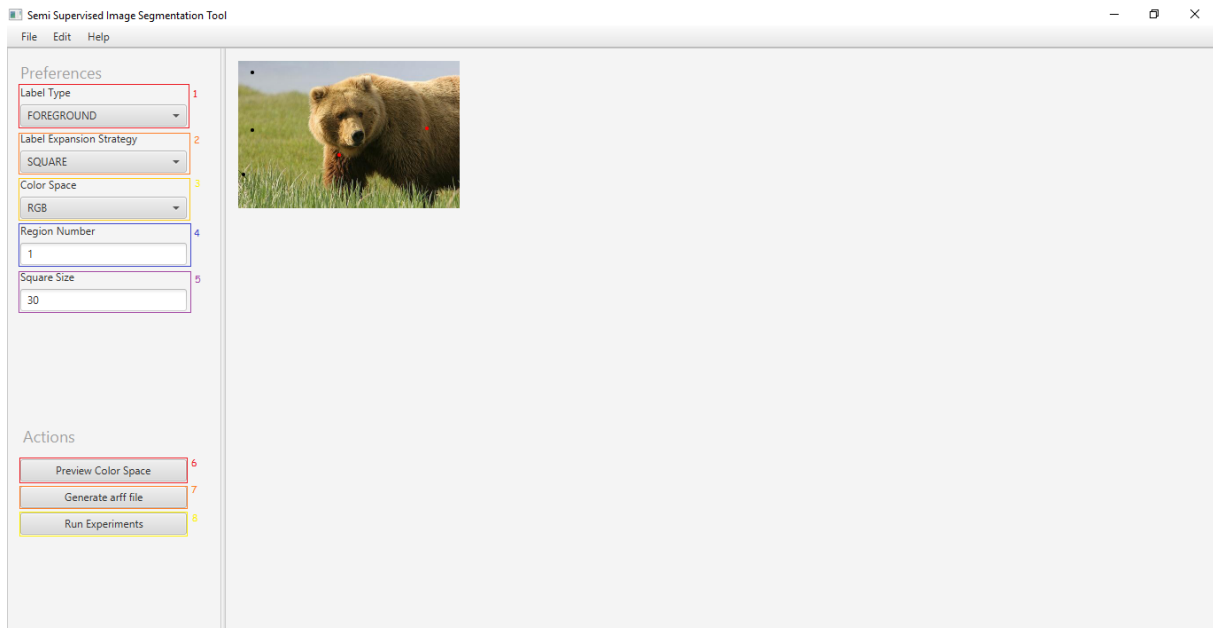
Fonte: O autor

4. Esta opção define a classe do rótulo
5. Esta opção define o tamanho do lado l do quadrado.
6. Este botão serve para converter e exibir a imagem carregada para o espaço de cores selecionado.
7. Este botão serve para realizar a expansão de rótulos, seguindo as opções selecionadas anteriormente, e gerar um arquivo ARFF
8. Este botão serve para gerar automaticamente vários arquivos ARFF com várias configurações que são utilizados para realizar os experimentos.

4.4 Experimentos Realizados

Os algoritmos de agrupamento semissupervisionados foram executados para cada uma das estratégias de expansão de rótulos supracitadas, variando a quantidade e tipo dos

Figura 16 – Ferramenta de rotulação parcial de imagens

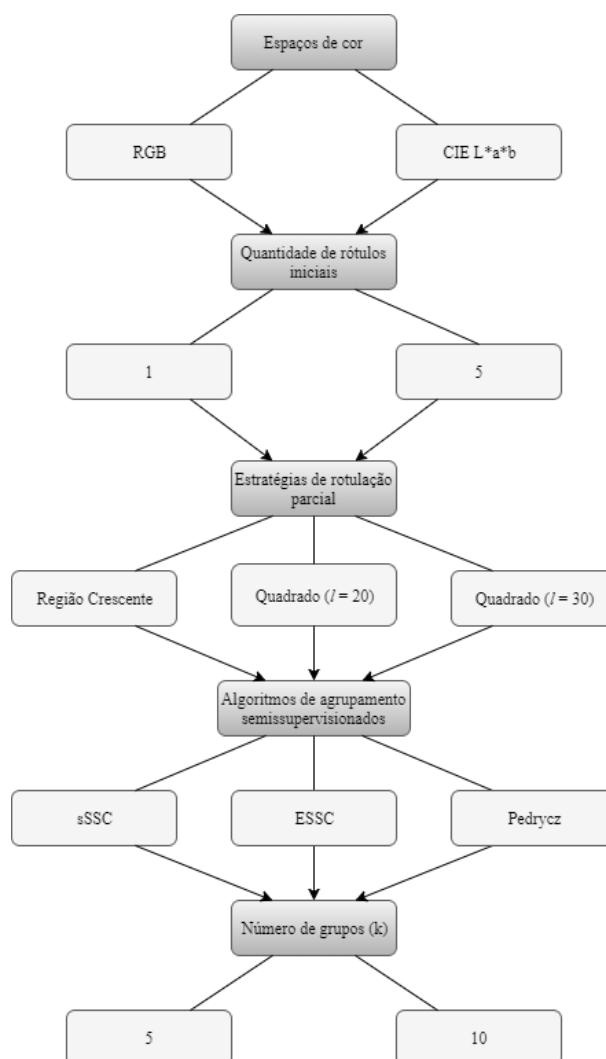


Fonte: O autor

rótulos, o espaço de cores (RGB e CIE L*a*b), o número k de grupos, onde $k \in \{5, 10\}$, e mantendo fixo o valor do parâmetro α em 1. Para o algoritmo FCM, foram variados o espaço de cor e número k de grupos, da mesma maneira que os algoritmos semissupervisionados. Para todos os algoritmos, foi utilizada a distância euclidiana e 30 iterações. A quantidade de rótulos para a rotulação em formato de quadrado foi de 1 rótulo sobre o objeto, 1 rótulo sobre o fundo, 5 rótulos sobre o objeto e 5 rótulos, 2 sobre o objeto e 3 sobre o fundo. Para a rotulação de Região crescente, foram utilizados 1 rótulo sobre o objeto, 1 rótulo sobre o fundo e 5 rótulos, 2 sobre o objeto e 3 sobre o fundo. Dessa forma, foram executados 136 experimentos por imagens. Foram segmentadas 30 imagens, totalizando 4080 experimentos. Os resultados desses experimentos estão descritos na [Capítulo 5](#).

A [Figura 17](#) ilustra a configuração dos experimentos realizados para os algoritmos semissupervisionados.

Figura 17 – Configuração dos experimentos



Fonte: o autor

5 Resultados e discussão

Neste Capítulo serão apresentados os resultados obtidos nos experimentos de segmentação de imagens com algoritmos de agrupamento semissupervisionado. Os resultados foram divididos em subseções. A [seção 5.1](#) apresenta os resultados do algoritmo FCM. As subseções (5.1.1), (5.1.2) e (5.1.3) apresentam os resultados obtidos pelos algoritmos de agrupamento semissupervisionados para os tipos de rotulação com o algoritmo de região crescente, pelo formato de quadrado com tamanho ($l = 20$) e formato de quadrado com tamanho ($l = 30$), respectivamente. Em todas as tabelas de resultados desse capítulo são expostos os valores médios da especificidade, sensibilidade e MCC a partir de 30 imagens e, em parênteses, seus respectivos desvios padrão.

5.1 Resultados - *Fuzzy C-means*

Os resultados dos experimentos do algoritmo FCM estão descritos na [Tabela 2](#). É possível observar que o espaço de cor RGB possui melhores resultados que o espaço de cores CIE L^*a^*b , e que a quantidade de grupos influencia principalmente na taxa de verdadeiros-positivos (TP), de modo que 10 grupos obtiveram resultados consideravelmente superiores a 5 grupos.

Tabela 2 – Resultados do algoritmo *Fuzzy C-means*

Espaço de cor e número de grupos	Especificidade	Sensibilidade	MCC
RGB (k = 5)	0.977 (0.036)	0.617 (0.33)	0.631 (0.312)
RGB (k = 10)	0.980 (0.021)	0.758 (0.205)	0.783 (0.158)
LAB (k = 5)	0.980 (0.031)	0.583 (0.328)	0.607 (0.324)
LAB (k = 10)	0.982 (0.024)	0.722 (0.208)	0.77 (0.157)

Fonte – o autor

5.1.1 Região Crescente

Nas tabelas (3) e (4), estão descritos os resultados na situação que o usuário clica apenas uma vez na imagem, sendo o caso que clica uma vez no objeto e outra situação em que clica uma vez no fundo, respectivamente, para o espaço de cores RGB. É possível notar que o algoritmo de Pedrycz obtém melhores resultados em relação ao MCC, sendo que com apenas sobre o fundo os resultados são melhores para esse algoritmo.

Tabela 3 – Resultado - região crescente, RGB, 1 rótulo inicial (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.974 (0.039)	0.587 (0.308)	0.606 (0.296)
sSSC (k = 10)	0.972 (0.049)	0.712 (0.246)	0.733 (0.201)
ESSC (k = 5)	0.978 (0.12)	0.065 (0.204)	0.077 (0.166)
ESSC (k = 10)	0.976 (0.054)	0.325 (0.421)	0.321 (0.346)
Pedrycz (k = 5)	0.983 (0.027)	0.57 (0.355)	0.59 (0.346)
Pedrycz (k = 10)	0.983 (0.02)	0.722 (0.246)	0.759 (0.201)

Fonte – o autor

Tabela 4 – Resultado - região crescente, RGB, 1 rótulo inicial (apenas sobre o fundo)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.972 (0.045)	0.626 (0.333)	0.632 (0.312)
sSSC (k = 10)	0.98 (0.027)	0.721 (0.256)	0.751 (0.214)
ESSC (k = 5)	0.998 (0.006)	0.055 (0.119)	0.099 (0.181)
ESSC (k = 10)	0.953 (0.109)	0.296 (0.401)	0.273 (0.311)
Pedrycz (k = 5)	0.975 (0.037)	0.635 (0.333)	0.641 (0.313)
Pedrycz (k = 10)	0.98 (0.022)	0.758 (0.202)	0.782 (0.158)

Fonte – o autor

Na [Tabela 5](#) estão expostos os resultados para 5 rótulos iniciais e o espaço de cores RGB. Esse experimento obteve melhores resultados (considerando MCC) do que os que utilizaram apenas um rótulo, salientando a importância da quantidade de rótulos para a obtenção de melhores resultados. O algoritmo de Pedrycz obteve melhores resultados para este experimento.

Tabela 5 – Resultado - região crescente, RGB, 5 rótulos iniciais (fundo e objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.975 (0.039)	0.559 (0.337)	0.578 (0.323)
sSSC (k = 10)	0.974 (0.03)	0.719 (0.234)	0.739 (0.173)
ESSC (k = 5)	0.966 (0.126)	0.127 (0.306)	0.121 (0.225)
ESSC (k = 10)	0.96 (0.09)	0.305 (0.414)	0.288 (0.305)
Pedrycz (k = 5)	0.966 (0.046)	0.707 (0.293)	0.69 (0.261)
Pedrycz (k = 10)	0.977 (0.025)	0.778 (0.194)	0.793 (0.146)

Fonte – o autor

Nas tabelas (6) e (7) estão descritos os resultados para 1 rótulo sobre o objeto e 1 rótulo sobre o fundo, respectivamente, para o espaço de cores LAB. É possível notar que o algoritmo de Pedrycz obtém melhores resultados em relação ao MCC, sendo que com apenas sobre o objeto os resultados são melhores para esse algoritmo.

Tabela 6 – Resultado - região crescente, LAB, 1 rótulo inicial (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.977 (0.034)	0.512 (0.31)	0.538 (0.317)
sSSC (k = 10)	0.98 (0.027)	0.631 (0.231)	0.695 (0.185)
ESSC (k = 5)	0.973 (0.097)	0.149 (0.277)	0.174 (0.244)
ESSC (k = 10)	0.977 (0.05)	0.388 (0.431)	0.394 (0.367)
Pedrycz (k = 5)	0.98 (0.037)	0.549 (0.335)	0.575 (0.336)
Pedrycz (k = 10)	0.984 (0.019)	0.719 (0.203)	0.768 (0.152)

Fonte – o autor

Tabela 7 – Resultado - região crescente, LAB, 1 rótulo inicial (apenas sobre o fundo)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.972 (0.042)	0.595 (0.318)	0.612 (0.303)
sSSC (k = 10)	0.98 (0.025)	0.7 (0.238)	0.735 (0.211)
ESSC (k = 5)	0.998 (0.005)	0.078 (0.155)	0.132 (0.208)
ESSC (k = 10)	0.969 (0.051)	0.359 (0.418)	0.336 (0.34)
Pedrycz (k = 5)	0.977 (0.036)	0.595 (0.332)	0.609 (0.32)
Pedrycz (k = 10)	0.983 (0.024)	0.695 (0.239)	0.74 (0.209)

Fonte – o autor

Na [Tabela 8](#) estão expostos os resultados para 5 rótulos iniciais e o espaço de cores LAB. Para o algoritmo de Pedrycz e ESSC, esse experimento obteve melhores resultados do que com apenas um rótulo, porém o algoritmo sSSC obteve os melhores resultados com apenas 1 rótulo inicial sobre o fundo. Para esse experimento, o algoritmo de Pedrycz obteve os melhores resultados em relação ao MCC.

Tabela 8 – Resultado - região crescente, LAB, 5 rótulos iniciais (fundo e objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.963 (0.048)	0.603 (0.28)	0.606 (0.246)
sSSC (k = 10)	0.976 (0.03)	0.653 (0.25)	0.694 (0.201)
ESSC (k = 5)	0.958 (0.123)	0.222 (0.364)	0.221 (0.268)
ESSC (k = 10)	0.961 (0.064)	0.393 (0.438)	0.374 (0.337)
Pedrycz (k = 5)	0.974 (0.041)	0.643 (0.329)	0.649 (0.314)
Pedrycz (k = 10)	0.977 (0.03)	0.755 (0.181)	0.781 (0.133)

Fonte – o autor

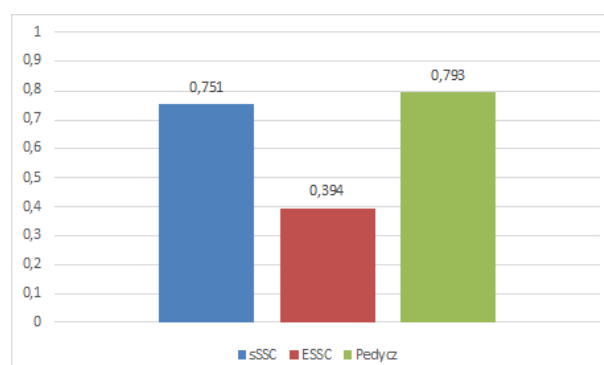
Analisando os resultados dessa subseção, observa-se que o algoritmo de Pedrycz obtém os melhores resultados em relação ao MCC, sendo que o algoritmo sSSC possui resultados bastante próximos, e que para todos os algoritmos a sensibilidade é muito influenciada pelo número de grupos, onde 10 grupos obtém melhores resultados que 5 grupos. Também é possível observar que os algoritmos ESSC e de Pedrycz obtém melhores resultados quando os rótulos são apenas sobre o objeto, enquanto o algoritmo sSSC possui melhores resultados quando os rótulos são apenas sobre fundo. Em relação ao espaço de cores, os algoritmos sSSC e de Pedrycz obtém melhores resultados através do RGB, enquanto o algoritmo ESSC, através do CIE L*a*b. O gráfico ilustrado na [Figura 18](#) exibe os melhores resultados de cada algoritmo presentes nesta subseção e a [Figura 19](#) ilustra a segmentação dos melhores resultados que cada algoritmo obteve nesta subseção para a imagem [Figura 13a](#). Na [Tabela 9](#) estão expostos os testes estatísticos aplicados para os melhores resultados dessa subseção e do algoritmo FCM. Nesta tabela, o símbolo '<' indica que o resultado foi pior do que ao qual ele foi comparado, e o símbolo '>' indica que o resultado foi melhor. Segundo os testes, os algoritmos FCM, sSSC e de Pedrycz obtiveram resultados iguais, enquanto o algoritmo ESSC foi pior que todos eles.

Tabela 9 – Teste de Wilcoxon dos Melhores MCC's - Região Crescente

	FCM	sSSC	ESSC	Pedrycz
FCM	-	=	>	=
sSSC	=	-	>	=
ESSC	<	<	-	<
Pedrycz	=	=	>	-

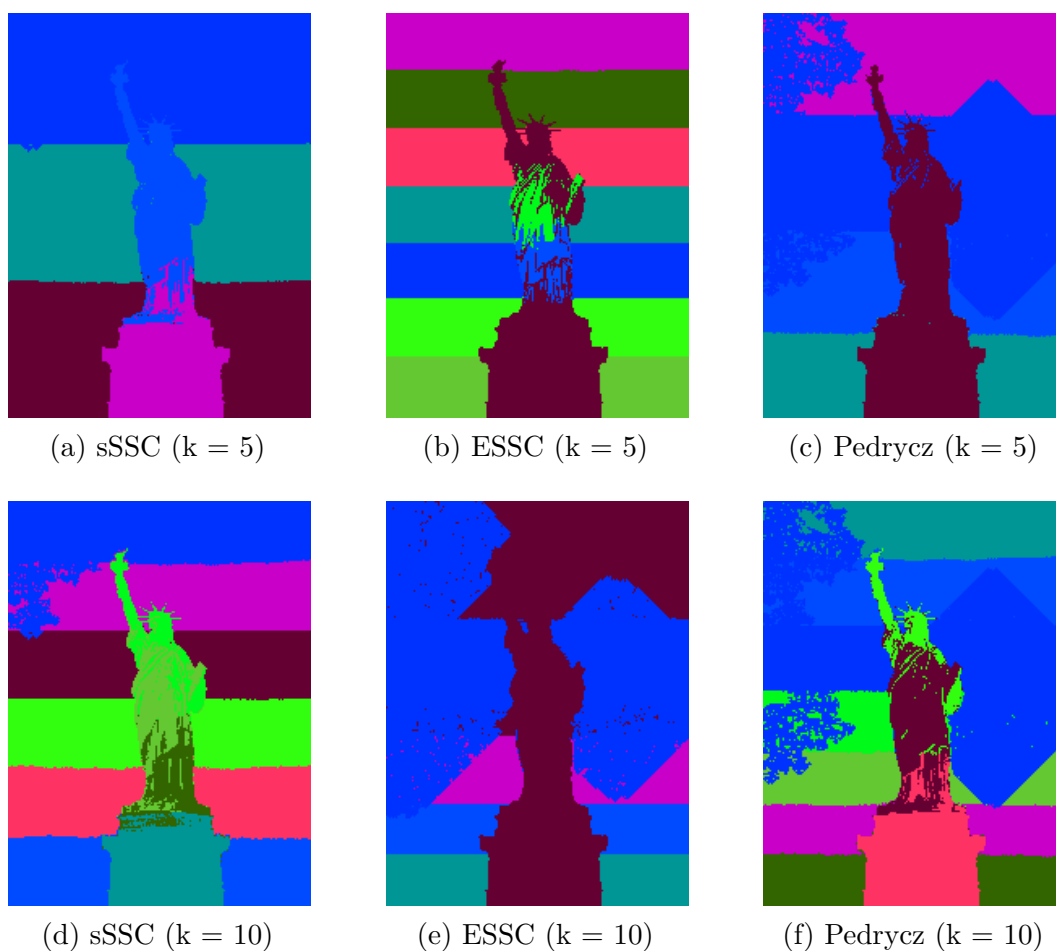
Fonte – o autor

Figura 18 – Melhores MCC'S - Região Crescente



Fonte: o autor

Figura 19 – Melhores segmentações - Região crescente



5.1.2 Quadrado ($l = 20$)

Nas tabelas (10) e (11), estão descritos os resultados para 1 rótulo sobre o objeto e 1 rótulo sobre o fundo, respectivamente, para o espaço de cores RGB. É possível notar que

o algoritmo de Pedrycz obtém melhores resultados em relação ao MCC, sendo que com apenas um clique no fundo, os resultados são melhores para esse algoritmo.

Tabela 10 – Resultado - quadrado ($l = 20$), RGB, 1 rótulo inicial (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.977 (0.035)	0.588 (0.312)	0.611 (0.302)
sSSC (k = 10)	0.98 (0.024)	0.708 (0.229)	0.745 (0.197)
ESSC (k = 5)	0.977 (0.12)	0.083 (0.209)	0.108 (0.184)
ESSC (k = 10)	0.961 (0.108)	0.297 (0.418)	0.282 (0.338)
Pedrycz (k = 5)	0.978 (0.035)	0.581 (0.337)	0.596 (0.324)
Pedrycz (k = 10)	0.98 (0.025)	0.728 (0.241)	0.757 (0.204)

Fonte – o autor

Tabela 11 – Resultado - quadrado ($l = 20$), RGB, 1 rótulo inicial (apenas sobre o fundo)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.975 (0.035)	0.618 (0.329)	0.627 (0.311)
sSSC (k = 10)	0.98 (0.022)	0.737 (0.24)	0.76 (0.203)
ESSC (k = 5)	0.971 (0.121)	0.118 (0.26)	0.144 (0.202)
ESSC (k = 10)	0.97 (0.069)	0.264 (0.379)	0.263 (0.305)
Pedrycz (k = 5)	0.976 (0.035)	0.619 (0.33)	0.631 (0.312)
Pedrycz (k = 10)	0.981 (0.024)	0.737 (0.242)	0.763 (0.204)

Fonte – o autor

Nas tabelas (12) e (13), são exibidos os resultados para 5 rótulos iniciais apenas sobre o objeto e 5 rótulos iniciais sobre o objeto e o fundo, respectivamente, ambos para o espaço de cores RGB. É possível notar que o algoritmo de Pedrycz obteve os melhores resultados para essa quantidade de rótulos, e que a rotulação apenas com cliques no objeto proporcionou melhores resultados para esse algoritmo, e o algoritmo ESSC, enquanto o algoritmo sSSC obteve melhores resultados com os dois tipos de rótulos.

Tabela 12 – Resultado - quadrado ($l = 20$), RGB, 5 rótulos iniciais (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.975 (0.04)	0.476 (0.335)	0.498 (0.335)
sSSC (k = 10)	0.979 (0.028)	0.576 (0.253)	0.636 (0.223)
ESSC (k = 5)	0.977 (0.12)	0.072 (0.193)	0.095 (0.162)
ESSC (k = 10)	0.944 (0.119)	0.367 (0.46)	0.318 (0.347)
Pedrycz (k = 5)	0.973 (0.04)	0.607 (0.357)	0.599 (0.345)
Pedrycz (k = 10)	0.977 (0.027)	0.795 (0.181)	0.802 (0.158)

Fonte – o autor

Tabela 13 – Resultado - quadrado ($l = 20$), RGB, 5 rótulos iniciais (fundo e objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.976 (0.039)	0.567 (0.33)	0.587 (0.32)
sSSC (k = 10)	0.976 (0.029)	0.662 (0.265)	0.691 (0.238)
ESSC (k = 5)	0.978 (0.118)	0.066 (0.193)	0.081 (0.161)
ESSC (k = 10)	0.944 (0.117)	0.321 (0.433)	0.281 (0.315)
Pedrycz (k = 5)	0.979 (0.036)	0.545 (0.367)	0.558 (0.356)
Pedrycz (k = 10)	0.984 (0.02)	0.71 (0.267)	0.748 (0.222)

Fonte – o autor

As tabelas (14) e (15) exibem os resultados para apenas um clique do usuário no objeto e também resultados quando o clique é sobre o fundo, respectivamente, ambos para o espaço de cores CIE L^*a^*b . É possível notar que o algoritmo de Pedrycz alcançou melhores resultados em ambos os casos, com o algoritmo sSSC obtendo resultados bastante próximos. Para esses dois algoritmos, pixels rotulados apenas no fundo obteve melhores resultados, enquanto que para o algoritmo ESSC, a rotulação apenas do objeto proporcionou melhores resultados.

Tabela 14 – Resultado - quadrado ($l = 20$), LAB, 1 rótulo inicial (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.98 (0.031)	0.542 (0.326)	0.566 (0.33)
sSSC (k = 10)	0.982 (0.025)	0.663 (0.238)	0.718 (0.207)
ESSC (k = 5)	0.979 (0.097)	0.134 (0.259)	0.171 (0.235)
ESSC (k = 10)	0.981 (0.039)	0.405 (0.419)	0.424 (0.366)
Pedrycz (k = 5)	0.977 (0.036)	0.57 (0.32)	0.591 (0.321)
Pedrycz (k = 10)	0.978 (0.027)	0.727 (0.197)	0.759 (0.156)

Fonte – o autor

Tabela 15 – Resultado - quadrado ($l = 20$), LAB, 1 rótulo inicial (apenas sobre o fundo)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.98 (0.031)	0.574 (0.323)	0.6 (0.32)
sSSC (k = 10)	0.983 (0.022)	0.705 (0.21)	0.759 (0.16)
ESSC (k = 5)	0.968 (0.103)	0.166 (0.306)	0.186 (0.245)
ESSC (k = 10)	0.969 (0.046)	0.411 (0.418)	0.395 (0.332)
Pedrycz (k = 5)	0.981 (0.031)	0.575 (0.323)	0.601 (0.32)
Pedrycz (k = 10)	0.982 (0.022)	0.717 (0.208)	0.761 (0.166)

Fonte – o autor

Nas tabelas (16) e (17) estão expostos os resultados para 5 rótulos iniciais com clique do usuário apenas no objeto e 5 rótulos iniciais com cliques no objeto e fundo, respectivamente. É possível notar que o algoritmo de Pedrycz obtém os melhores resultados para ambos experimentos, sendo que quando os rótulos são apenas no objeto, proporcionaram melhores resultados para esse algoritmo, enquanto o algoritmo sSSC possui melhores resultados para os rótulos sobre o objeto e fundo.

Tabela 16 – Resultado - quadrado ($l = 20$), LAB, 5 rótulos iniciais (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.98 (0.034)	0.42 (0.338)	0.448 (0.338)
sSSC (k = 10)	0.979 (0.028)	0.566 (0.252)	0.63 (0.228)
ESSC (k = 5)	0.98 (0.094)	0.125 (0.257)	0.155 (0.235)
ESSC (k = 10)	0.961 (0.072)	0.478 (0.464)	0.43 (0.395)
Pedrycz (k = 5)	0.976 (0.038)	0.617 (0.358)	0.608 (0.356)
Pedrycz (k = 10)	0.974 (0.035)	0.776 (0.244)	0.769 (0.237)

Fonte – o autor

Tabela 17 – Resultado - quadrado ($l = 20$), LAB, 5 rótulos iniciais (fundo e objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.978 (0.035)	0.508 (0.336)	0.529 (0.341)
sSSC (k = 10)	0.981 (0.025)	0.64 (0.242)	0.694 (0.223)
ESSC (k = 5)	0.982 (0.091)	0.119 (0.251)	0.154 (0.236)
ESSC (k = 10)	0.964 (0.059)	0.45 (0.444)	0.43 (0.345)
Pedrycz (k = 5)	0.975 (0.041)	0.584 (0.329)	0.597 (0.325)
Pedrycz (k = 10)	0.981 (0.024)	0.664 (0.289)	0.696 (0.272)

Fonte – o autor

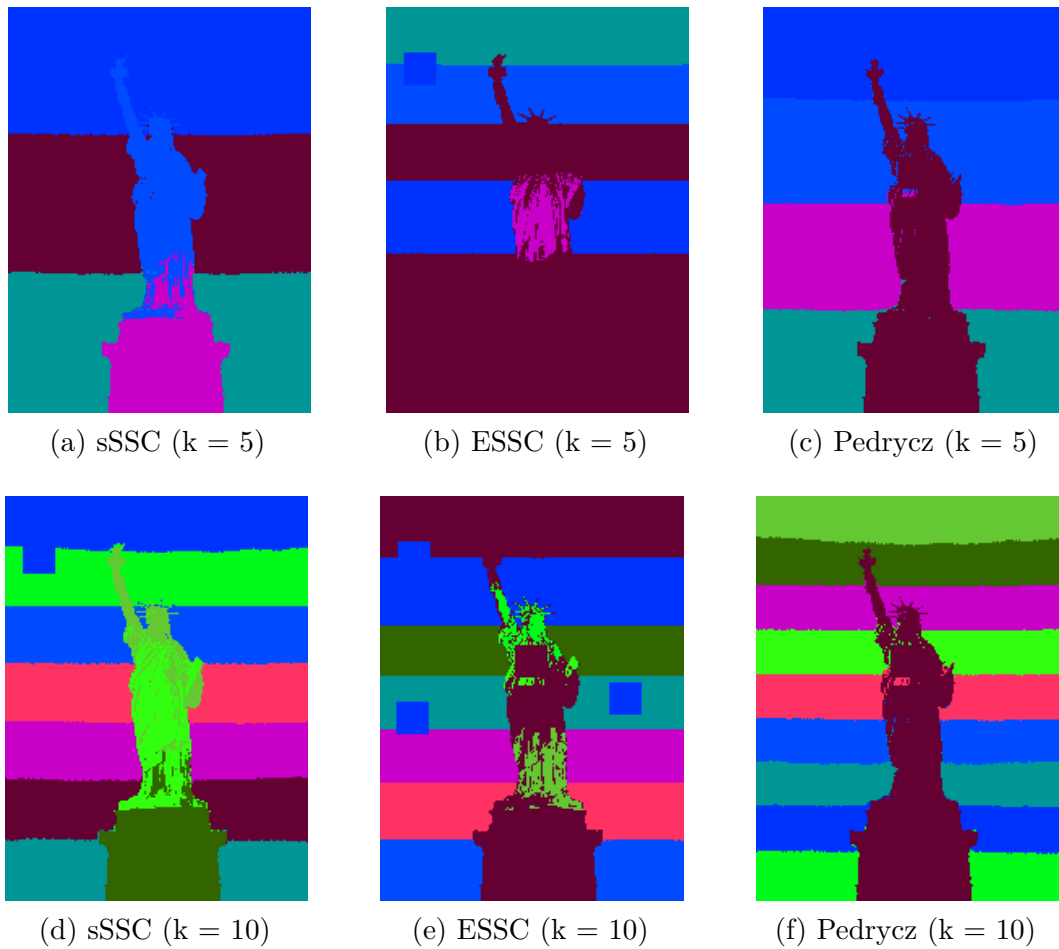
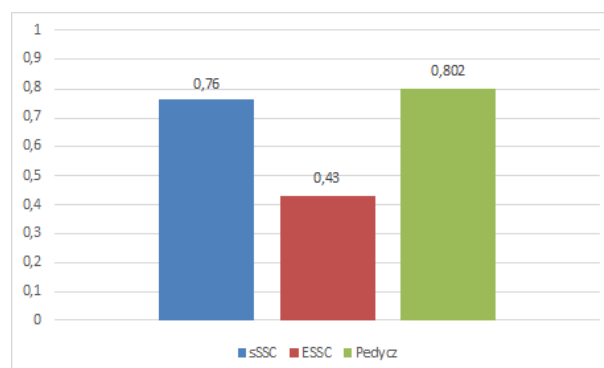
Analisando os resultados dessa subseção, observa-se que o algoritmo de Pedrycz obtém os melhores resultados em relação ao MCC, sendo que o algoritmo sSSC possui resultados bastante próximos quando há mais rótulos sobre o fundo. Também é possível observar que para todos os algoritmos, a sensibilidade é muito influenciada pelo número de grupos, onde 10 grupos obtém melhores resultados que 5 grupos. Enquanto os valores da especificidade são próximos quando muda a quantidade de cliques, espaços de cores, algoritmos e quantidade de grupos. Em relação ao espaço de cores, os algoritmos sSSC e de Pedrycz obtém melhores resultados através do RGB, enquanto o algoritmo ESSC, através do CIE L*a*b. Na [Tabela 18](#) estão expostos os testes estatísticos aplicados para os melhores resultados dessa subseção e do algoritmo FCM. Segundo os testes, os algoritmos FCM e de Pedrycz obtiveram resultados iguais, enquanto o algoritmo sSSC e ESSC obtiveram resultados inferiores a ambos.

Resumindo os resultados dessa subseção, o algoritmo sSSC obteve melhores resultados através da rotulação apenas sobre o fundo e espaço de cores RGB, o algoritmo ESSC através cinco rótulos (igual para apenas sobre o objeto e objeto e fundo) CIE L*a*b, e o algoritmo de Pedrycz, através da rotulação apenas sobre o objeto com 5 rótulos iniciais e espaço de cores RGB. O gráfico ilustrado na [Figura 21](#) exhibe os melhores resultados de cada algoritmo presentes nesta subseção. e a [Figura 20](#) ilustra a segmentação dos melhores resultados que cada algoritmo obteve nesta subseção para a imagem [Figura 13a](#).

Tabela 18 – Teste de Wilcoxon dos Melhores MCC's - Quadrado ($l = 20$)

	FCM	sSSC	ESSC	Pedrycz
FCM	-	>	>	=
sSSC	<	-	>	<
ESSC	<	<	-	<
Pedrycz	=	>	>	-

Fonte – o autor

Figura 20 – Melhores segmentações - Quadrado ($l = 20$)Figura 21 – Melhores MCC's - Quadrado ($l = 20$)

Fonte: o autor

5.1.3 Quadrado ($l = 30$)

Nas tabelas (19) e (20), estão descritos os resultados para rótulos de apenas um clique no objeto e um clique sobre o fundo, respectivamente, para o espaço de cores RGB. É possível notar que o algoritmo de Pedrycz obtém melhores resultados em relação ao

MCC, sendo que com apenas rótulos no fundo, os resultados são ligeiramente melhores para esse algoritmo. O algoritmo sSSC possui resultados próximos do algoritmo de Pedrycz quando os rótulos são apenas sobre o fundo, porém o algoritmo de Pedrycz tem uma vantagem considerável quando os rótulos são apenas no objeto.

Tabela 19 – Resultado - quadrado ($l = 30$), RGB, 1 rótulo inicial (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.975 (0.038)	0.574 (0.306)	0.595 (0.299)
sSSC (k = 10)	0.979 (0.025)	0.646 (0.248)	0.689 (0.229)
ESSC (k = 5)	0.999 (0.003)	0.05 (0.098)	0.108 (0.164)
ESSC (k = 10)	0.947 (0.121)	0.347 (0.451)	0.301 (0.351)
Pedrycz (k = 5)	0.977 (0.036)	0.608 (0.319)	0.626 (0.306)
Pedrycz (k = 10)	0.982 (0.024)	0.729 (0.239)	0.762 (0.205)

Fonte – o autor

Tabela 20 – Resultado - quadrado ($l = 30$), RGB, 1 rótulo inicial (apenas sobre o fundo)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.973 (0.038)	0.629 (0.314)	0.637 (0.294)
sSSC (k = 10)	0.979 (0.023)	0.741 (0.242)	0.761 (0.205)
ESSC (k = 5)	0.98 (0.103)	0.073 (0.208)	0.084 (0.181)
ESSC (k = 10)	0.972 (0.052)	0.335 (0.421)	0.32 (0.35)
Pedrycz (k = 5)	0.976 (0.038)	0.619 (0.33)	0.629 (0.312)
Pedrycz (k = 10)	0.982 (0.022)	0.737 (0.241)	0.766 (0.205)

Fonte – o autor

Nas tabelas (21) e (22) são exibidos os resultados para rótulos iniciais disponíveis com 5 cliques do usuário apenas no objeto e 5 cliques no objeto e no fundo, respectivamente, ambos para o espaço de cores RGB. É possível notar que o algoritmo de Pedrycz obteve os melhores resultados para essa quantidade de rótulos, e que a rotulação apenas com cliques no objeto proporcionou melhores resultados para esse algoritmo, enquanto os algoritmos sSSC e ESSC obtiveram os melhores resultados com os dois tipos de rótulos.

Tabela 21 – Resultado - quadrado ($l = 30$), RGB, 5 rótulos iniciais (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.977 (0.041)	0.343 (0.341)	0.363 (0.327)
sSSC (k = 10)	0.966 (0.047)	0.539 (0.337)	0.55 (0.293)
ESSC (k = 5)	0.978 (0.12)	0.052 (0.185)	0.067 (0.13)
ESSC (k = 10)	0.955 (0.121)	0.269 (0.42)	0.245 (0.332)
Pedrycz (k = 5)	0.98 (0.031)	0.623 (0.381)	0.614 (0.371)
Pedrycz (k = 10)	0.976 (0.032)	0.846 (0.143)	0.839 (0.133)

Fonte – o autor

Tabela 22 – Resultado - quadrado ($l = 30$), RGB, 5 rótulos iniciais (fundo e objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.978 (0.033)	0.478 (0.276)	0.521 (0.292)
sSSC (k = 10)	0.984 (0.019)	0.568 (0.268)	0.638 (0.236)
ESSC (k = 5)	0.999 (0.003)	0.035 (0.091)	0.071 (0.146)
ESSC (k = 10)	0.957 (0.103)	0.307 (0.414)	0.286 (0.317)
Pedrycz (k = 5)	0.976 (0.039)	0.536 (0.371)	0.54 (0.358)
Pedrycz (k = 10)	0.985 (0.018)	0.725 (0.261)	0.761 (0.227)

Fonte – o autor

As tabelas (23) e (24) exibem os resultados para rótulos disponíveis quando houve um clique do usuário no objeto e um clique no fundo, respectivamente, ambos para o espaço de cores CIE L^*a^*b . É possível notar que o algoritmo de Pedrycz alcançou melhores resultados em ambos os casos. O algoritmo sSSC obteve resultados bastante próximos, quando os rótulos são apenas sobre o fundo. Para esses dois algoritmos, a rotulação apenas no fundo obteve melhores resultados, enquanto que para o algoritmo ESSC, a rotulação apenas sobre o objeto proporcionou melhores resultados.

Tabela 23 – Resultado - quadrado ($l = 30$), LAB, 1 rótulo inicial (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.979 (0.032)	0.481 (0.321)	0.509 (0.328)
sSSC (k = 10)	0.984 (0.021)	0.601 (0.244)	0.668 (0.232)
ESSC (k = 5)	0.969 (0.102)	0.22 (0.363)	0.226 (0.297)
ESSC (k = 10)	0.979 (0.039)	0.451 (0.436)	0.448 (0.387)
Pedrycz (k = 5)	0.976 (0.037)	0.581 (0.321)	0.595 (0.32)
Pedrycz (k = 10)	0.98 (0.022)	0.724 (0.197)	0.758 (0.157)

Fonte – o autor

Tabela 24 – Resultado - quadrado ($l = 30$), LAB, 1 rótulo inicial (apenas sobre o fundo)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.98 (0.03)	0.574 (0.322)	0.599 (0.321)
sSSC (k = 10)	0.982 (0.022)	0.711 (0.215)	0.76 (0.162)
ESSC (k = 5)	0.974 (0.106)	0.161 (0.289)	0.189 (0.259)
ESSC (k = 10)	0.96 (0.059)	0.419 (0.438)	0.379 (0.346)
Pedrycz (k = 5)	0.981 (0.031)	0.574 (0.323)	0.601 (0.32)
Pedrycz (k = 10)	0.981 (0.026)	0.724 (0.21)	0.767 (0.16)

Fonte – o autor

Nas tabelas (25) e (26) estão expostos os resultados para rótulos gerados a partir de 5 cliques no objeto e 5 cliques no objeto e no fundo, respectivamente. É possível notar que o algoritmo de Pedrycz obtém os melhores resultados para ambos experimentos, sendo que para rótulos apenas no objeto, proporcionaram melhores resultados para esse algoritmo e para o algoritmo ESSC. Enquanto o algoritmo sSSC possui melhores resultados para os rótulos sobre o objeto e fundo.

Tabela 25 – Resultado - quadrado ($l = 30$), LAB, 5 rótulos iniciais (apenas sobre o objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.978 (0.044)	0.346 (0.379)	0.359 (0.361)
sSSC (k = 10)	0.956 (0.052)	0.551 (0.32)	0.553 (0.254)
ESSC (k = 5)	0.963 (0.108)	0.201 (0.365)	0.193 (0.275)
ESSC (k = 10)	0.956 (0.1)	0.478 (0.48)	0.435 (0.409)
Pedrycz (k = 5)	0.964 (0.053)	0.683 (0.337)	0.652 (0.325)
Pedrycz (k = 10)	0.975 (0.036)	0.825 (0.182)	0.813 (0.187)

Fonte – o autor

Tabela 26 – Resultado - quadrado ($l = 30$), LAB, 5 rótulos iniciais (fundo e objeto)

Algoritmo e número de grupos	Especificidade	Sensibilidade	MCC
sSSC (k = 5)	0.965 (0.062)	0.463 (0.344)	0.469 (0.335)
sSSC (k = 10)	0.978 (0.025)	0.616 (0.25)	0.661 (0.231)
ESSC (k = 5)	0.982 (0.091)	0.078 (0.198)	0.111 (0.17)
ESSC (k = 10)	0.97 (0.072)	0.347 (0.425)	0.344 (0.361)
Pedrycz (k = 5)	0.971 (0.043)	0.576 (0.362)	0.57 (0.345)
Pedrycz (k = 10)	0.98 (0.026)	0.701 (0.267)	0.728 (0.255)

Fonte – o autor

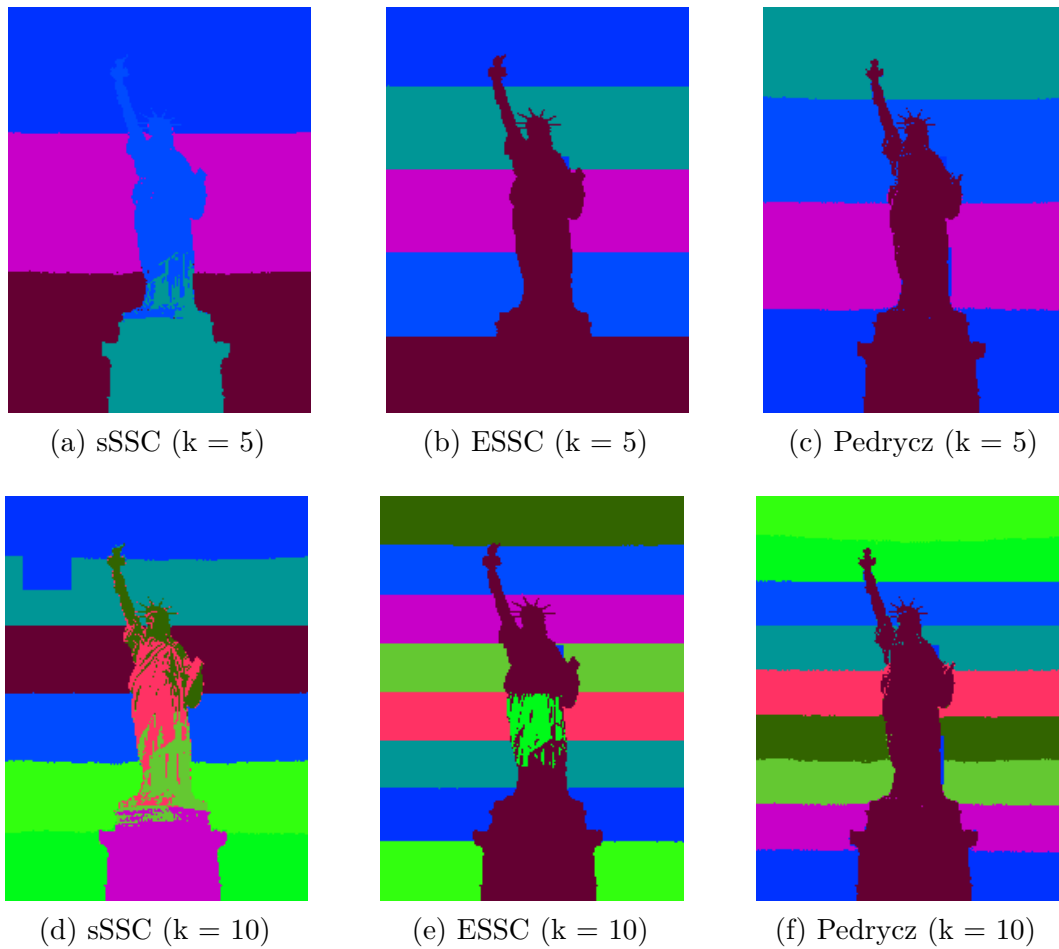
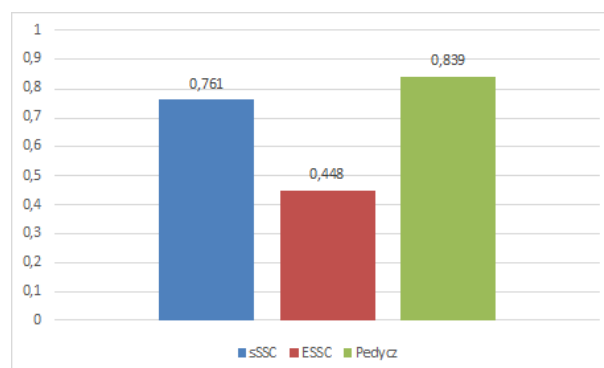
Analisando os resultados dessa subseção, observa-se que o algoritmo de Pedrycz obtém os melhores resultados em relação ao MCC, e que o algoritmo sSSC possui resultados melhores quando os rótulos são apenas sobre o fundo. Também é possível observar que para todos os algoritmos, a sensibilidade é muito influenciada pelo número de grupos, onde 10 grupos obtém melhores resultados. Enquanto a especificidade possui valores próximos mesmo quando há mudanças na quantidade de rótulos, espaços de cores, algoritmos e quantidade de grupos. Em relação ao espaço de cores, os algoritmos sSSC e de Pedrycz obtém melhores resultados através do RGB, enquanto o algoritmo ESSC, através do CIE L*a*b. Na [Tabela 27](#) estão expostos os testes estatísticos aplicados para os melhores resultados dessa subseção e do algoritmo FCM. Segundo os testes estatísticos, o algoritmo de Pedrycz obteve resultados que todos os demais algoritmos, tendo superado o FCM.

Resumindo os resultados dessa subseção, o algoritmo sSSC obteve melhores resultados através da rotulação apenas sobre o fundo e espaço de cores RGB. O algoritmo ESSC obteve o melhor desempenho quando havia rótulos disponíveis de apenas um clique do usuário no fundo e espaço de cores CIE L*a*b. Enquanto, o algoritmo de Pedrycz, obteve os melhores desempenhos através de um clique sobre o objeto e para 5 cliques e espaço de cores RGB. O gráfico ilustrado na [Figura 23](#) exibe os melhores resultados de cada algoritmo presentes nesta subseção e a [Figura 22](#) ilustra a segmentação dos melhores resultados que cada algoritmo obteve nesta subseção para a imagem [Figura 13a](#).

Tabela 27 – Teste de Wilcoxon dos Melhores MCC's - Quadrado ($l = 30$)

	FCM	sSSC	ESSC	Pedrycz
FCM	-	>	>	<
sSSC	<	-	>	<
ESSC	<	<	-	<
Pedrycz	>	>	>	-

Fonte – o autor

Figura 22 – Melhores segmentações - Quadrado ($l = 30$)Figura 23 – Melhores MCC's - Quadrado ($l = 30$)

Fonte: o autor

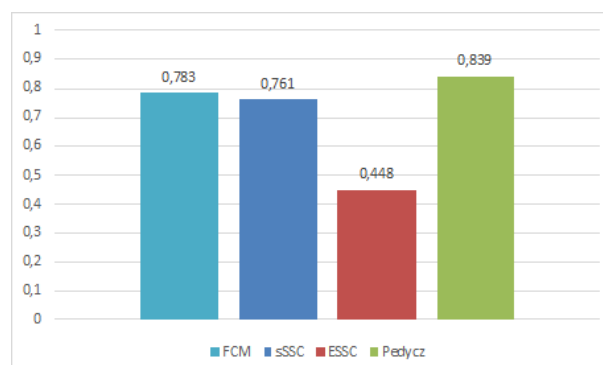
5.2 Análise dos resultados

Analisando os resultados das subseções anteriores, é possível definir que o algoritmo de Pedrycz obteve o melhor desempenho dentre os algoritmos semissupervisionados avaliados nesse trabalho, independentemente da quantidade de rótulos iniciais, do tipo

desses rótulos, da estratégia de rotulação, do espaço de cores ou da quantidade de grupos. Porém, ele apresenta melhores resultados para uma maior quantidade de rótulos sobre o objeto e para o espaço de cores RGB. Quando são realizados cinco cliques sobre o objeto da imagem, o algoritmo de Pedrycz apresenta melhores resultados que o FCM, dependendo de quantos cliques foram realizados e sobre qual área para rotular a imagem, tendo ambos obtido os melhores resultados para o espaço de cores RGB. O algoritmo sSSC obteve melhores resultados quando a rotulação era apenas sobre o objeto e o espaço de cores RGB. Ao passo que o algoritmo ESSC foi o único algoritmo que obteve melhores resultados com o espaço de cores CIE L^*a^*b e com rótulos gerados de 5 cliques do usuário sobre o objeto e fundo. Nenhum desses dois algoritmos superou o desempenho do FCM.

Em relação às estratégias de rotulação, a rotulação em formato de quadrado com $l = 30$ foi a que proporcionou os melhores resultados para todos os algoritmos. O gráfico ilustrado na Figura 24 exibe os melhores resultados que cada algoritmo obteve considerando todos os experimentos realizados. Os resultados também mostram que o número de grupos impacta principalmente sobre a sensibilidade, indicando que a maior dificuldade dos algoritmos é agrupar corretamente os *pixels* do objeto das imagens.

Figura 24 – Melhores MCC's por algoritmo



Fonte: o autor

6 Conclusão

Neste trabalho foi apresentado um estudo comparativo entre o algoritmo de agrupamento não-supervisionado *Fuzzy C-Means* e três outros algoritmos de agrupamento semissupervisionado nele baseados, o algoritmo sSSC, ESSC e de Pedrycz, quando aplicados no contexto de segmentação de imagens coloridas. Porém, para que os algoritmos semissupervisionados pudessem ter bons resultados, foi desenvolvida uma ferramenta de software capaz de gerar, através de cliques na imagem, três formas de rotulação parcial, onde o usuário seleciona o tipo do rótulo (objeto ou fundo). Para todos os algoritmos, também foi variado o número de grupos (5 ou 10) e os espaços de cores (RGB ou CIE L^*a^*b).

Os resultados indicaram que o algoritmo de Pedrycz obteve os melhores resultados entre os algoritmos semissupervisionados, e, com 5 rótulos sobre o objeto, melhores que o FCM, e que o espaço de cores RGB é a melhor opção para os algoritmos FCM, sSSC e de Pedrycz, enquanto o algoritmo ESSC obteve melhores resultados com o espaço de cores CIE L^*a^*b . Em suma, as principais contribuições desse trabalho foram:

- Identificação do melhor entre os três algoritmos semissupervisionados baseados no FCM para segmentação de imagens coloridas;
- Desenvolvimento de uma ferramenta de software para rotulação parcial de imagens, a qual será disponibilizada na *web*;
- Avaliação do melhor espaço de cor para cada um dos algoritmos testados;
- Desenvolvimento de duas formas de rotulação parcial de imagens, em forma de quadrado de um tamanho pré-fixado e o algoritmo de região de crescimento;
- Avaliação do melhor tipo de rótulo (fundo ou objeto) para cada algoritmo semissupervisionado;
- Desenvolvimento de um dataset de arquivos arff para cada experimento realizado neste trabalho, o qual será disponibilizado na *web*;
- Análise do impacto causado pela quantidade de grupos para os algoritmos de agrupamento.

6.1 Trabalhos futuros

Os trabalhos futuros podem seguir várias direções. Uma delas é na fase de rotulação parcial, tal como aprimorar a rotulação parcial baseada em região crescente, pois este tipo

de rotulação não obteve os resultados esperados, devido ao fato de nem sempre conseguir rotular corretamente as imagens. Além disso, há diversas pesquisas sobre região crescente com semeadura automática, tais como os trabalhos apresentados em (FAN et al., 2001), (SHIH; CHENG, 2005) e (GÓMEZ; GONZÁLEZ; MORALES, 2007), que poderiam ser adaptados para realizar a rotulação parcial, o que obteria uma grande quantidade de rótulos e dispensaria a necessidade de iteração com usuários. Outra possibilidade é incluir outros formatos, além do quadrado, para a rotulação parcial, e também testar outras quantidades de rótulos iniciais.

Outro caminho a seguir, seria incluir outros algoritmos de agrupamento semissupervisionados baseados no *Fuzzy C-means*, tais como o Algoritmo de Bouchachia (BOUCHACHIA; PEDRYCZ, 2006) e o algoritmo Nebfuzz (MACARIO; CARVALHO, 2012). Por fim, outra possibilidade seria testar outros espaços de cores.

Referências

- ALIA, O. M.; MANDAVA, R.; AZIZ, M. E. A hybrid harmony search algorithm for mri brain segmentation. *Evolutionary Intelligence*, Springer, v. 4, n. 1, p. 31–49, 2011. Citado na página 19.
- BATRA, D.; KOWDLE, A.; PARIKH, D.; LUO, J.; CHEN, T. Interactively co-segmentating topically related images with intelligent scribble guidance. *International journal of computer vision*, Springer, v. 93, n. 3, p. 273–292, 2011. Citado na página 39.
- BATRACK, C. *Create a Retro CRT Distortion Effect Using RGB Shifting*. 2010. <<https://code.tutsplus.com/tutorials/create-a-retro-crt-distortion-effect-using-rgb-shifting--active-3359>>. [Online; acesso em 10 de Setembro de 2017]. Citado na página 29.
- BENSAID, A. M.; HALL, L. O.; BEZDEK, J. C.; CLARKE, L. P. Partially supervised clustering for image segmentation. *Pattern recognition*, Elsevier, v. 29, n. 5, p. 859–871, 1996. Citado na página 37.
- BEZDEK, J. C.; EHRLICH, R.; FULL, W. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, Elsevier, v. 10, n. 2-3, p. 191–203, 1984. Citado 2 vezes nas páginas 15 e 19.
- BILENKO, M.; BASU, S.; MOONEY, R. J. Integrating constraints and metric learning in semi-supervised clustering. In: ACM. *Proceedings of the twenty-first international conference on Machine learning*. [S.l.], 2004. p. 11. Citado na página 19.
- BOUCHACHIA, A.; PEDRYCZ, W. Data clustering with partial supervision. *Data Mining and Knowledge Discovery*, Springer, v. 12, n. 1, p. 47–78, 2006. Citado na página 63.
- BUSIN, L.; SHI, J.; VANDENBROUCKE, N.; MACAIRE, L. Color space selection for color image segmentation by spectral clustering. In: IEEE. *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*. [S.l.], 2009. p. 262–267. Citado na página 35.
- CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning (chappelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542–542, 2009. Citado na página 15.
- CHEN, S.; SUN, Z.; ZHOU, J.; LI, Y. Semi-supervised image segmentation combining ssfcm and random walks. In: IEEE. *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*. [S.l.], 2012. v. 1, p. 185–190. Citado na página 15.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2012. Citado na página 18.
- FAN, J.; YAU, D. K.; ELMAGARMID, A. K.; AREF, W. G. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE*

- transactions on image processing*, IEEE, v. 10, n. 10, p. 1454–1466, 2001. Citado na página 63.
- GÓMEZ, O.; GONZÁLEZ, J. A.; MORALES, E. F. Image segmentation using automatic seeded region growing and instance-based learning. In: SPRINGER. *Iberoamerican Congress on Pattern Recognition*. [S.l.], 2007. p. 192–201. Citado na página 63.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X. Citado 10 vezes nas páginas 24, 25, 26, 27, 28, 29, 30, 31, 32 e 33.
- HILL, B.; ROGER, T.; VORHAGEN, F. W. Comparative analysis of the quantization of color spaces on the basis of the cielab color-difference formula. *ACM Transactions on Graphics (TOG)*, ACM, v. 16, n. 2, p. 109–154, 1997. Citado na página 30.
- JAFARI, M. H.; SAMAVI, S. Iterative semi-supervised learning approach for color image segmentation. In: IEEE. *Machine Vision and Image Processing (MVIP), 2015 9th Iranian Conference on*. [S.l.], 2015. p. 76–79. Citado na página 36.
- JAIN, A. K.; DUBES, R. C. *Algorithms for clustering data*. [S.l.]: Prentice-Hall, Inc., 1988. Citado 2 vezes nas páginas 15 e 33.
- JIA, Y.; ZHANG, C. Learning distance metric for semi-supervised image segmentation. In: IEEE. *Image Processing, 2008. IICIP 2008. 15th IEEE International Conference on*. [S.l.], 2008. p. 3204–3207. Citado na página 37.
- JOBLOVE, G. H.; GREENBERG, D. Color spaces for computer graphics. In: ACM. *ACM siggraph computer graphics*. [S.l.], 1978. v. 12, n. 3, p. 20–25. Citado na página 28.
- LI, C.; TAN, Y.; WANG, D.; MA, P. Research on 3d face recognition method in cloud environment based on semi supervised clustering algorithm. *Multimedia Tools and Applications*, Springer, v. 76, n. 16, p. 17055–17073, 2017. Citado na página 37.
- MACÁRIO, V. Um novo algoritmo de agrupamento semisupervisionado baseado no fuzzy c-means. Universidade Federal de Pernambuco, 2009. Citado 3 vezes nas páginas 18, 19 e 21.
- MACARIO, V. Algoritmos particionais semissupervisionados com ponderação automática de variáveis. UNIVERSIDADE FEDERAL DE PERNAMBUCO, 2015. Citado na página 23.
- MACARIO, V.; CARVALHO, F. d. A. de. An adaptive semi-supervised fuzzy clustering algorithm based on objective function optimization. In: IEEE. *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*. [S.l.], 2012. p. 1–8. Citado na página 63.
- MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 15.
- MANTHIRI, A. *Multi Class Confusion Matrix*. 2017. <https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html>. [Online; acesso em 12 de Setembro de 2017]. Citado na página 40.

- MARQUES, O. F.; VIEIRA, H. N. *Processamento digital de imagens*. [S.l.]: Brasport, 1999. Citado 5 vezes nas páginas 24, 26, 27, 31 e 32.
- MARSHALL, D. *Region Growing*. 2017. <https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html>. [Online; acesso em 12 de Setembro de 2017]. Citado na página 33.
- MENESES, P. R.; ALMEIDA, T. d. Introdução ao processamento de imagens de sensoriamento remoto. *Embrapa Cerrados-Livros técnicos (INFOTECA-E)*, Brasília, DF: UnB, 2012., 2012. Citado 2 vezes nas páginas 25 e 29.
- MITCHELL, T. M. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, 1997. Citado na página 14.
- NOORDAM, J.; BROEK, W. V. D.; BUYDENS, L. M. Geometrically guided fuzzy c-means clustering for multivariate image segmentation. In: IEEE. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. [S.l.], 2000. v. 1, p. 462–465. Citado na página 37.
- PEDRYCZ, W.; WALETZKY, J. Fuzzy clustering with partial supervision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 27, n. 5, p. 787–795, 1997. Citado 4 vezes nas páginas 15, 19, 20 e 21.
- POINT, T. *Histograms Introduction*. 2017. <https://www.tutorialspoint.com/dip/histograms_introduction.htm>. [Online; acesso em 10 de Setembro de 2017]. Citado na página 31.
- PORTAL, T. S. *The Statistics Portal*, “Number of daily active Snapchat users from 1st quarter 2014 to 1st quarter 2017 (in millions)”. 2017. <<https://www.statista.com/statistics/545967/snapchat-app-dau>> [Online; acesso em 18 de Junho de 2017]. Citado na página 14.
- PORTAL, T. S. “Number of monthly active Instagram users from January 2013 to April 2017 (in millions)”. 2017. <<https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users>>. [Online; acesso em 18 de Junho de 2017]. Citado na página 14.
- QIAN, Y.; SI, W. A semi-supervised color image segmentation method. In: IEEE. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. [S.l.], 2005. v. 2, p. II–1194. Citado 2 vezes nas páginas 14 e 36.
- RAJABY, E.; AHADI, S. M.; AGHAEINIA, H. Robust color image segmentation using fuzzy c-means with weighted hue and intensity. *Digital Signal Processing*, Elsevier, v. 51, p. 170–183, 2016. Citado na página 16.
- RAVAL, K.; SHUKLA, R.; SHAH, A. K. Color image segmentation using fcm clustering technique in rgb, l* a* b, hsv, yiq color spaces. *European Journal of Advances in Engineering and Technology*, v. 4, n. 3, p. 194–200, 2017. Citado na página 36.
- SANKARI, L.; CHANDRASEKAR, C. A new enhanced semi supervised image segmentation using marker as prior information. *International Journal of Image, Graphics and Signal Processing*, Modern Education and Computer Science Press, v. 4, n. 1, p. 51, 2012. Citado na página 36.

- SANTOS, C. A. M. d. et al. Análise de características para detecção de nudez em imagens. Universidade Federal do Amazonas, 2012. Citado na página 30.
- SHIH, F. Y.; CHENG, S. Automatic seeded region growing for color image segmentation. *Image and vision computing*, Elsevier, v. 23, n. 10, p. 877–886, 2005. Citado 2 vezes nas páginas 14 e 63.
- SON, L. H.; TUAN, T. M. A cooperative semi-supervised fuzzy clustering framework for dental x-ray image segmentation. *Expert Systems with Applications*, Elsevier, v. 46, p. 380–393, 2016. Citado na página 37.
- TAN, K. S.; ISA, N. A. M. Color image segmentation using histogram thresholding–fuzzy c-means hybrid approach. *Pattern Recognition*, Elsevier, v. 44, n. 1, p. 1–15, 2011. Citado na página 35.
- VERNON, D. Machine vision-automated visual inspection and robot vision. *NASA STI/Recon Technical Report A*, v. 92, 1991. Citado na página 14.
- WANG, F.; ZHANG, C.; SHEN, H. C.; WANG, J. Semi-supervised classification using linear neighborhood propagation. In: IEEE. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. [S.l.], 2006. v. 1, p. 160–167. Citado na página 37.
- WEATHERALL, I. L.; COOMBS, B. D. Skin color measurements in terms of cielab color space values. *Journal of investigative dermatology*, Elsevier, v. 99, n. 4, p. 468–473, 1992. Citado na página 30.
- WILCOXON, F. Individual comparisons by ranking methods. *Biometrics bulletin*, JSTOR, v. 1, n. 6, p. 80–83, 1945. Citado na página 41.
- YANG, Y.; HUANG, S. Image segmentation by fuzzy c-means clustering algorithm with a novel penalty term. *Computing and Informatics*, v. 26, n. 1, p. 17–31, 2012. Citado na página 15.
- YASUNORI, E.; YUKIHIRO, H.; MAKITO, Y.; SADAANKI, M. On semi-supervised fuzzy c-means clustering. In: IEEE. *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*. [S.l.], 2009. p. 1119–1124. Citado 3 vezes nas páginas 15, 21 e 22.
- ZAXIN, Z.; LIZHI, C.; GUANGQUAN, C. Neighbourhood weighted fuzzy c-means clustering algorithm for image segmentation. *IET Image processing*, IET, v. 8, n. 3, p. 150–161, 2013. Citado na página 35.