

GUSTAVO HENRIQUE NEVES SILOTO

**APRENDIZADO DE MÁQUINA NÃO-SUPERVISIONADO PARA DETECÇÃO DE
PADRÕES EM IMAGENS**

GUSTAVO HENRIQUE NEVES SILOTO

**APRENDIZADO DE MÁQUINA NÃO-SUPERVISIONADO PARA DETECÇÃO DE
PADRÕES EM IMAGENS**

Projeto de pesquisa apresentado ao Curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientado(a): Gustavo Henrique Neves Siloto
Orientador(a): Prof. Msc. Felipe Alexandre
Cardoso Pazinato

FICHA CATALOGRÁFICA

SILOTO, Gustavo Henrique Neves

Aprendizado de Máquina Não-Supervisionado para a Detecção de Padrões em
Imagens/Gustavo Henrique Neves Siloto. Fundação Educacional de Assis – Assis, 2016

Num paginas

Trabalho de Conclusão de Curso – Instituição Municipal de Ensino Superior de Assis

Orientador: Almir Rogério Camolesi

1. Inteligência Artificial. 2. Máquina de Aprendizado

CDD:001.6

APRENDIZADO DE MÁQUINA NÃO-SUPERVISIONADO PARA DETECÇÃO DE PADRÕES EM IMAGENS

GUSTAVO HENRIQUE NEVES SILOTO

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal
de Ensino Superior de Assis, como
requisito do Curso de Graduação,
analisado pela seguinte comissão
examinadora:

Orientator: _____

Analizador(1): _____

**Assis/SP
2016**

No final, você não é avaliado por quão você se comprometeu, mas pelo o
que você realizou.
Donald Trump
(1946-Atualmente)

Resumo

Este trabalho tem como objetivo desenvolver um sistema de reconhecimento de padrões em imagens, empregando técnicas de processamento de imagem para obtenção de informação, e a técnica de inteligência artificial, aprendizado de máquina não-supervisionado, k-média, para a categorização dos dados.

Palavras-Chave: Inteligência; Artificial; Não-Supervisionado; Aprendizado; Máquina; Processamento; Imagem; K-média.

Abstract

This work has as objective to develop a system to recognize patterns in images, employing image process techniques to obtain the image information, and the artificial intelligence technique, unsupervised machine learning, k-means, for categorizing the data;

Palavras-Chave:Artificial; Intelligence; Unsupervised; Machine; Learning; Image; Processing; K-means.

Lista de Ilustrações

Figura 1: Exemplo do funcionamento do AM.....	2
Figura 2: Imagem enviada de Londres a New York.....	5
Figura 3: Comprimento da luz visualizada pelo humano.....	5
Figura 4: Interpretação do RGB através de um cubo composto de três valores.....	6
Figura 5: Diferentes valores da escala de cinza.....	6
Figura 6: Aplicação de um filtro linear.....	7
Figura 7: Máscara de Sobel Horizontal(G_x).....	8
Figura 8: Máscara de Sobel Vertical(G_y).....	8
Figura 9: Exemplo de imagem a ser aplicado o Operador de Sobel.....	8
Figura 10: a) Imagem Original b) Imagem tratada com a máscara G_x c) Imagem tratada com a máscara G_y	9
Figura 11: Imagem gerada utilizando a fórmula 4.....	9
Figura 12: Matriz resultante do $\sigma=1$	10
Figura 13: Imagem ao ser tratado com a suavização gaussiana com os valores $\sigma=10$ e $k=5$	10
Figura 14: Exemplo de detecção de borda.....	11
Figura 15: Valores do código em cadeia e suas direções.....	11
Figura 16: Imagem de exemplo para utilizar o código em cadeia.....	12
Figura 17: Aplicado o código em cadeia a imagem 18.....	12
Figura 18: a)Imagem escura e b) seu histograma. c)Imagem clara e d) seu histograma..	13
Figura 19: Efeito do filtro threshold.....	14
Figura 20: Representação do AM.....	16
Figura 21: Demonstração de dados em um AM não-supervisionado.....	17
Figura 22: Centroids aplicados ao AM.....	17
Figura 23: Marcação dos dados aos centroids.....	18
Figura 24: Alteração da posição dos centroids e a remarcação dos dados.....	18
Figura 25: Novamente, alteração de posição e remarcação.....	19
Figura 26: Algoritmo do processamento de imagem.....	21
Figura 27: Produto final do algoritmo de processamento de imagem.....	22
Figura 28: Algoritmo de carregamento do centroid.....	22
Figura 29: Algoritmo de verificação de cluster.....	23
Figura 30: Algoritmo de categorização.....	23

Figura 31: Imagem do software.....	24
Figura 32: Lista de imagens a serem analisados.....	25
Figura 33: Todas as imagens inseridas.....	25
Figura 34: Imagens categorizadas.....	26
Figura 35: a) Imagem após filtro sobel b) Histograma c) Imagem a após aplicado threshold entre 127 e 255.....	27

Sumário

1.INTRODUÇÃO	1
1.1.OBJETIVO.....	2
1.1.1.objetivos específicos	2
1.2.HIPÓTESE.....	2
1.3.PROBLEMÁTICA.....	2
1.4.JUSTIFICATIVA.....	2
1.5.ESTRUTURA DO TRABALHO.....	3
2.REVISÃO DA LITERATURA	4
2.2.PROCESSAMENTO DE IMAGEM DIGITAL.....	4
2.2.1.rgb	5
2.2.2.escala de cinza	6
2.2.3.segmentação de imagem	7
2.2.4.filtro linear	7
2.2.4.1.operador de sobel.....	7
2.2.4.2.suavização gaussiana.....	10
2.2.1.deteccção de borda	11
2.2.1.1.código em cadeia.....	11
2.2.2.histograma	12
2.2.3.threshold	14
2.3.INTELIGÊNCIA ARTIFICIAL.....	14
2.3.1.aprendizado de máquina	15
2.3.1.treinamento supervisionado	16
2.3.2.treinamento não-supervisionado	16
2.3.2.1.clustering.....	17
2.3.2.2.k-média.....	17
2.4.C++.....	19
2.5.OPENCV.....	20
3.METODOLOGIA, DESENVOLVIMENTO, TESTES E RESULTADOS	21
3.1.METODOLOGIA.....	21
3.1.DESENVOLVIMENTO.....	21
3.2.1.processamento da imagem	21
3.2.2.k-média	22

3.2.3.software	24
3.3.TESTE.....	24
3.4.RESULTADO.....	26
4.CONCLUSÃO	28
REFERÊNCIAS	29

1. INTRODUÇÃO

Inteligência Artificial(IA) é um conceito teórico em Ciência da Computação que pode ser aplicada em diversas áreas, dentre elas sistema de inspeção e automação industriais, sistemas de diagnóstico na área médicas, área científica, dentre outras. A história da IA pode ser datada de épocas distantes, onde grandes filósofos definiam a consciência e o pensar do homem.

As discussões sobre IA começaram durante a 2ª Guerra Mundial, com o desenvolvimento das primeiras máquinas consideradas "computadores", possuindo capacidade computacional suficiente a ponto de levantar diversos questionamentos, citando como exemplo de questionamento: uma máquina um dia poderá pensar? Como saber se uma máquina se aproxima de comportamento e raciocínio humano? E o quão perigoso isso seria? Foi neste ponto que cientistas e matemáticos pensaram na ideia de uma máquina capaz de ser inteligente, capaz de solucionar problemas sem a necessidade da interferência humana. Alan Turing, cientista, durante a 2ª Guerra, criou um teste teórico chamado de Teste de Turing(1950), onde um interrogador(humano) passa questões a outro humano, que as repassa a máquina. A máquina responde ao humano, que responde ao interrogador. Se o interrogador pensasse que do outro lado, quem estava respondendo as questões fosse um humano, é dito que a AI presente em tal máquina tenha a capacidade de ser similar a um ser humano.

Com a rápida evolução tecnológica, gerando uma quantidade gigantesca de dados variados, este conjunto de dados pode conter informações pertinentes. Neste ponto a IA se torna uma ferramenta para a análise destes dados. As técnicas de *Aprendizado de Máquina(AM)*, colocam o foco em analisar os dados e transformá-los em informações a serem visualizadas, existindo dois grupos de técnicas: *Aprendizado Supervisionado* e *Aprendizado Não- Supervisionado*. Ambas as técnicas contém "semelhanças" em suas aplicações finais, mas, as diferenças principais se encontram nos dados a serem usado como treinamento. Em métodos supervisionados, os dados se encontram como tipos definidos de resultados, e esperado que novos dados se assemelhem a tais características, enquanto dados de métodos *não-supervisionados* não recebem nenhum tipo de informação pré tratamento, sendo esperado que a máquina "trate" tais dados para serem visualizados.

O termo "dados" pode ser variado no contexto do processamento de Images e Sinais, não

sendo obrigatoriamente os tipos encontrados em bancos de dados ou em arquivos de armazenamento textual, mas também mídias como sons, imagens e vídeos. É possível citar diversas aplicações que fazem a utilização de técnicas semelhantes como reconhecimento de faces, reconhecimento e caligrafia ou sistema segurança.

1.1. OBJETIVO

Desenvolver um *AM não-supervisionada*, a qual seja capaz de reconhecer as diferentes formas encontradas em uma ou mais imagens, e correlacionar com imagens futuras.

1.1.1. objetivos específicos

- Codificar informações encontradas nas imagens.
- Desenvolvimento de uma AM não-supervisionado.

1.2. HIPÓTESE

A partir da codificação de uma ou mais imagens, seria possível categorizar tais informações e encontrar padrões contendo o mesmo tipo de informação?

1.3. PROBLEMÁTICA

Seguindo a hipótese definida previamente, partindo de uma *AM não-supervisionado*, recebendo dados de diferentes imagens e seus respectivos detalhes, tal *AM* teria a capacidade de reconhecer diferentes padrões em diferentes imagens, seguindo o exemplo na figura 1 abaixo.

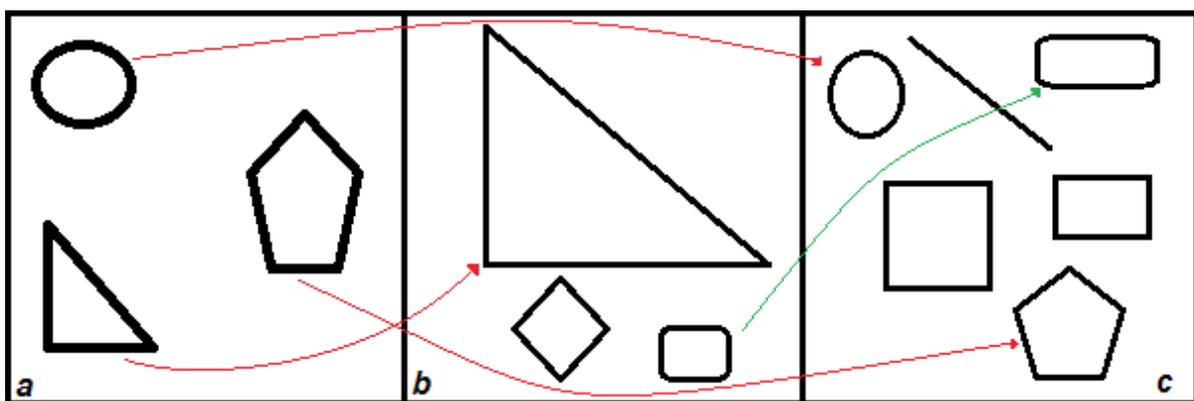


Figura 1: Exemplo do funcionamento do AM

1.4. JUSTIFICATIVA

A IA atualmente, possui métodos cada vez mais complexos, podendo ser citados como exemplos os carros auto-condutores desenvolvidos pela Google (Google Self-Driving Car Project, 2016), os quais estão próximos de serem lançados comercialmente, demonstrando uma necessidade de criação de ferramentas automáticas sem a

necessidade da interação humana.

1.5. ESTRUTURA DO TRABALHO

Este trabalho está dividido em duas principais partes, sendo a primeira parte a revisão da literatura, onde as técnicas e ferramentas utilizadas estão descritas, e a metodologia, onde é descrito o funcionamento do AM proposto pelo autor.

2. REVISÃO DA LITERATURA

Este capítulo faz a revisão do conteúdo técnico a serem desenvolvidos no trabalho.

2.1. VISÃO COMPUTACIONAL

Visão Computacional(VC), é um campo de estudo focado em desenvolver a capacidade de compreensão de uma máquina quanto à visualização de uma imagem, como Forsyth et. al. (2012) definem, a VC utiliza de modelos matemáticos, geométricos, físicos, probabilísticos e teorias de aprendizagem para a extração de informações do mundo real ao digital. Atualmente o campo da VC é aplicado a diferentes campos, como reconhecimento (reconhecimento de caracteres, objetos ou digitais), captura de movimentos (Mocap), cameras de tráfego e digitalização(modelos tridimensionais a partir de imagens).

Esta área de estudo teve seu desenvolvimento concomitante à inteligência artificial, e um dos pioneiros da área foi *Marvin Minsky* em 1966, quando pediu para um dos seus estudantes fazer um computador entender o ambiente através de uma câmera.

2.2. PROCESSAMENTO DE IMAGEM DIGITAL

Processamento de Imagem Digitais é o campo de estudo e desenvolvimento de técnicas de processamento de uma ou mais imagens digitais. Gonzales et. al. (2008) definem uma imagem sendo a seguinte função bidimensional:

$$f(x, y)$$

Fórmula 1: Função bidimensional para descrição de uma imagem digital de tamanho finito

Sendo x e y valores finitos onde representam a posição horizontal e vertical de um plano cartesiano, respectivamente. Cada par x e y que compõem a função, faz a representação de um *pixel*, uma composição de três diferentes valores vermelho(*Red*), verde(*Green*) e azul(*Blue*), no qual é uma combinação chamado de RGB.

A primeira imagem a ser processada pela humanidade, visto na figura 2 abaixo, foi feita no século passado, na década de 20, quando foi necessário enviar uma imagem entre o jornal de Londres ao jornal *New York*, ela foi transmitida por uma série de cabos que passam pelo oceano Atlântico através de telex. a imagem foi codificado para o sistema de Bartlane, e enviado para outro na America do Norte, no tempo total de três horas para a

transmissão. Novas técnicas e tecnologias cresceram de acordo com a evolução das técnicas computacionais.



Figura 2: Imagem enviada de Londres a New York

2.2.1. rgb

O ser humano comum tem a capacidade de enxergar três diferentes cores primárias (vermelho, verde e azul), as quais são ondas de luzes recebidas pelos olhos, impulsos elétricos atravessam o nervo ótico e são transmitidas ao cérebro, sendo reconhecidas e interpretadas. O reflexo da luz sobre diferentes superfícies gera ondas de luz de diferentes comprimentos. O ser humano tem a capacidade de visualizar ondas entre $400 \text{ exp } -7$ (400nm), sendo a cor azul, ao $700 \text{ exp } -7$ (700nm), sendo a cor vermelha, como é visto na figura 3 abaixo.

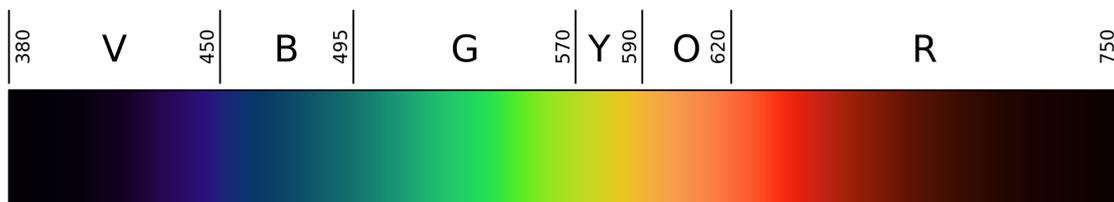


Figura 3: Comprimento da luz visualizada pelo humano

Na computação, as cores são visualizadas de forma semelhante. Monitores coloridos recebem uma combinação de três diferentes valores, vermelho(*Red*), verde(*Green*) e azul(*Blue*), respectivamente, as quais são combinadas no monitor em espaços tão pequenos(70 a 240 pontos por polegada), e por causa do espaço ínfimo entre as junções dos *pixels*, eles serão interpretados como combinação de cores. Cada valor, normalmente é composto por 8 *bits*, um total de 2^8 diferentes valores, com a combinação RGB o gera 2^{24} (16 milhões) combinações de cores. Na figura 4 abaixo é possível visualizar a

representação do espaço RGB em um cubo, e a partir de um da obtenção das posições x, y e z , obtendo diferentes cores.

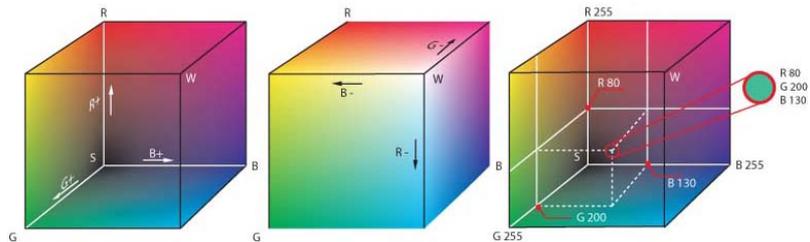


Figura 4: Interpretação do RGB através de um cubo composto de três valores

2.2.2. escala de cinza

A escala de cinza de uma imagem pode ser considerada como a intensidade de uma cor, sendo ela entre o valor 0 (preto) ao 255 (branco), é possível ver todos os valores na figura 5 abaixo.

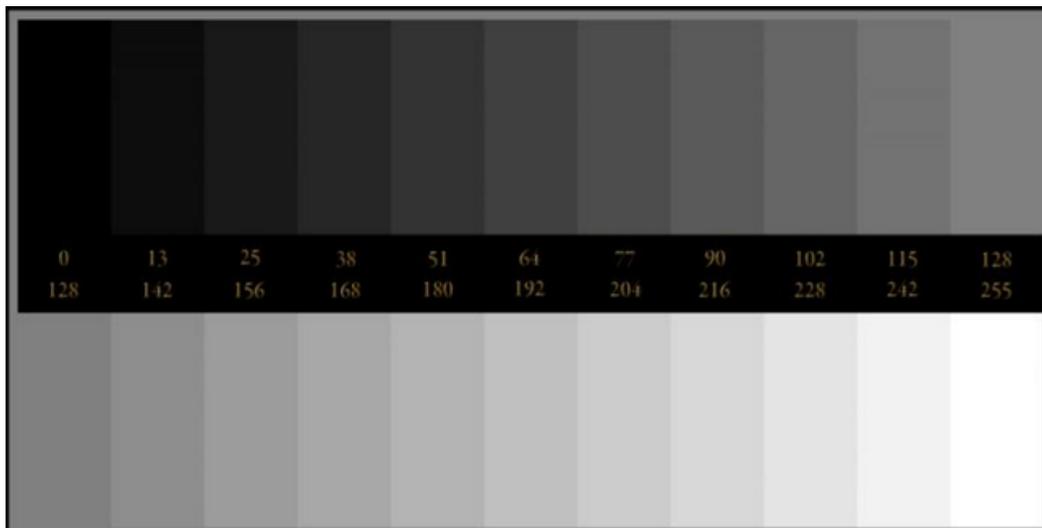


Figura 5: Diferentes valores da escala de cinza

É possível gerar uma nova imagem (P) em uma escala de cinza utilizando o seguinte fórmula abaixo, aplicado a cada *pixel* da imagem, onde R_{xy} , G_{xy} e B_{xy} são respectivamente os componentes Vermelho, Verde e Azul de cada *pixel* na imagem.

$$P(x, y) = \frac{(R_{xy} + G_{xy} + B_{xy})}{3}$$

Fórmula 2: Fórmula para calcular a intensidade de cada *pixel*

2.2.3. segmentação de imagem

Segmentação como Jain(1989) define, é a decomposição dos componentes de uma cena. É uma técnica de divisão de detalhes de uma imagem, mostrando a diferença entre os diferentes objetos nela presente, e mantendo uma certa quantidade de detalhes que mostrava.

2.2.4. filtro linear

De acordo com Szekiski(1991) filtro linear é uma técnica de processamento de imagem para a alteração de qualidade, suavização ou modificação de cores por exemplo. Este faz a utilização de um *kernel*, uma matriz de tamanho n , onde cada posição encontra-se um valor, a ser multiplicado por regiões de uma imagem.

Detalhadamente, seguindo a figura 6 abaixo, o *kernel* k será calculado com certa região da imagem, sendo cada posição do *kernel* multiplicado pela posição correspondente na imagem, após todos os valores serem multiplicados, são somados e divididos pelo tamanho do *kernel* k^2 , este valor é o valor representante da nova image na posição (x,y) .

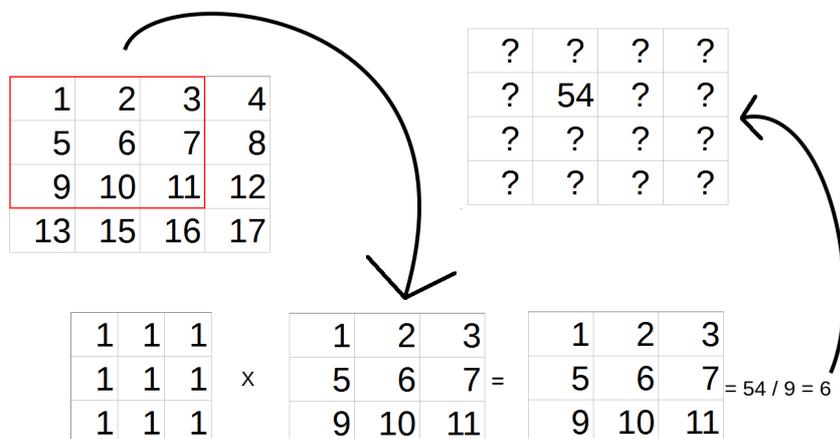


Figura 6: Aplicação de um filtro linear

2.2.4.1. operador de sobel

O *Operador de Sobel* é uma técnica sugerida por Sobel(2014) para a detecção de bordas a partir de uma imagem em escala de cinza. Ele faz a utilização de duas máscaras para calcular duas novas imagens, G_x e G_y , nas figuras 7 e 8 abaixo.

<i>i/j</i>	0	1	2
0	-1	-2	-1
1	0	0	0
2	1	2	1

Figura 7: Máscara de Sobel Horizontal(Gx)

<i>i/j</i>	0	1	2
0	-1	0	1
1	-2	0	2
2	-1	0	1

Figura 8: Máscara de Sobel Vertical(Gy)

Cada uma destas máscaras, geram uma imagem possuindo um gradiente dos *pixels* mais fortes em direções indicadas pela máscara. O calculo das novas imagens é feito a partir do seguinte método, inicialmente é separada uma parte da imagem composta de 3x3 *pixels*, como visto na imagem 9 abaixo.

<i>i/j</i>	0	1	2
0	127	127	127
1	127	0	0
2	127	0	0

Figura 9: Exemplo de imagem a ser aplicado o Operador de Sobel

O *pixel* central, $f(1, 1)$ é o *pixel* a ser calculado e aplicado a nova imagem Gx ou Gy. Os valores encontrados dentro desta matriz, são valores que variam entre 0 a 255 em uma escala de cinza. A partir da máscara e desta região, é utilizado a fórmula 3 abaixo para a geração do novo *pixel* a ser formado na nova imagem, partindo de um pressuposto que $R(x,y)$ representa a região obtida da imagem, e $M(x,y)$ representa a máscara a ser utilizada, podendo ser Gx ou Gy, e P sendo o novo *pixel* a ser colocado na imagem:

$$P = \sum_{x,y=0}^{x,y<3} (R_{xy} * M_{xy})$$

Fórmula 3: Fórmula para a geração de novos pixels utilizando as máscaras do operador de Sobel

Após a aplicação da fórmula a região selecionada, mostrada na figura 9 e a mascara Gy na figura 8, o resultado do novo *pixel* será de -381. Seguindo as figuras abaixo, a figura 10.a, imagem original, contém diferentes formas e cores, a figura 10.b, é uma imagem gerada a partir do cálculo Sobel da imagem original e a máscara Gx, a figura 10.c, é uma imagem gerada com a partir da máscara Gy.

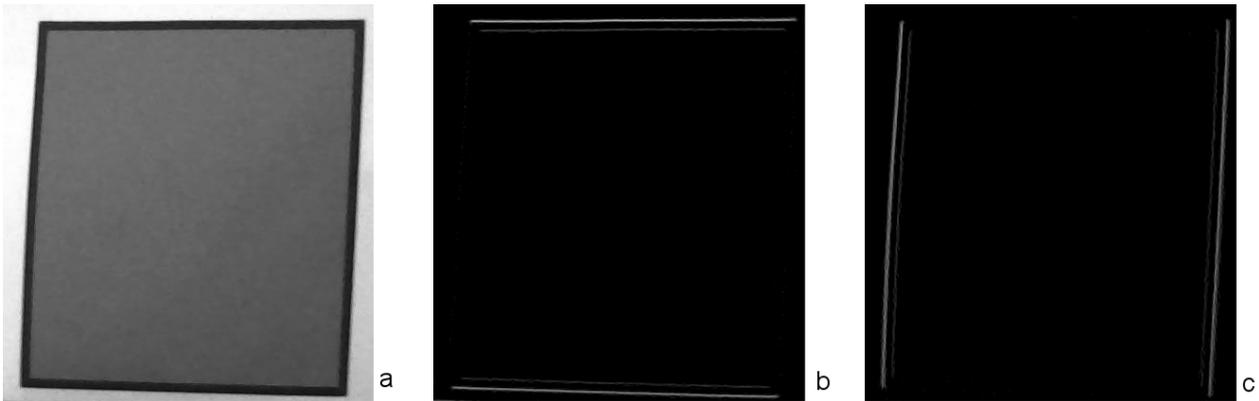


Figura 10: a) Imagem Original b) Imagem tratada com a máscara Gx c) Imagem tratada com a máscara Gy

A partir destas duas novas imagens, 10.b e 10.c, é possível obter uma nova imagem contendo o gradiente descendente de ambas, mostrando os detalhes em vertical, horizontal e diagonal. Gonzalez et. al.(2008), define a utilização da seguinte fórmula 4 para a obtenção de uma imagem final, sendo $G(x,y)$ a imagem final a ser gerada:

$$G(x, y) = |Gx| + |Gy|$$

Fórmula 4: Fórmula para a obtenção das bordas em diversas direções

Os autores afirmam que a utilização desta equação para a geração desta imagem, é interessante para uma aplicação computacional, simplificando o espaço amostral, além de manter a discrepância entre as diferentes intensidades dos *pixels*. O resultado de ambas imagens pode ser visto na figura 11 abaixo:

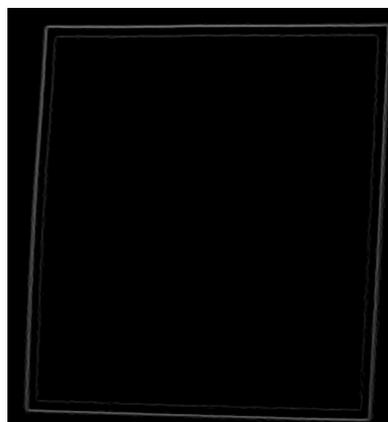


Figura 11: Imagem gerada utilizando a fórmula 4

2.2.4.2. suavização gaussiana

Suavização Gaussiana é uma técnica de filtro linear no qual tem como foco modificar valores de *pixels*, para que sejam semelhantes a seus vizinhos, gerando uma suavização na imagem final. De acordo com VERNON(ANO), suavização gaussiana faz a utilização da função gaussiana $G(x,y)$ para alimentar valores ao kernel, a fórmula 5 abaixo define função gaussiana:

$$G(x, y) = \frac{I}{2\pi\sigma^2} \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right]$$

Fórmula 5: Função Gaussiana utilizada na suavização gaussiana

O valor σ dita os valores resultantes para alimentar as posições do kernel, mostrado na figura 12 abaixo.

0.110741	0.111296	0.110741
0.111296	0.111853	0.111296
0.110741	0.111296	0.110741

Figura 12: Matriz resultante do $\sigma=1$

A figura 13 abaixo demonstra a suavização gaussiana com o valor $\sigma=10$ e $k=5$.



Figura 13: Imagem ao ser tratado com a suavização gaussiana com os valores $\sigma=10$ e $k=5$

2.2.1. detecção de borda

Jain(1989) define o método de detecção de bordas(*Edge-Detection*)(*DB*) sendo a obtenção dos limites externos de um objetos através da mudança abrupta da intensidade de um *pixel*, sendo possível visualizar um exemplo na figura 14 abaixo.

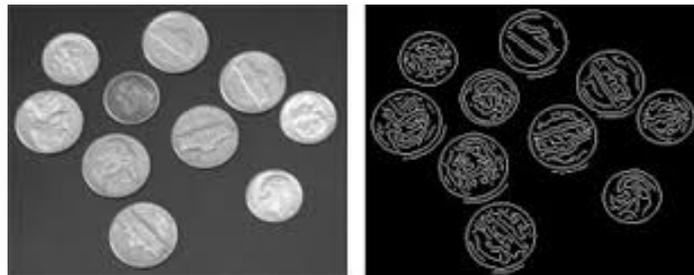


Figura 14: Exemplo de detecção de borda

2.2.1.1. código em cadeia

Gonzales et. al.(2008) define o código em cadeia como um método de representação da forma externa de um objeto através de uma sequência de códigos. Esta codificação, chamada de código em cadeia *Freeman*, é composto de oito valores, onde cada um deles representam diferentes direções, horizontal, vertical e diagonal, a figura 15 abaixo faz a representação dos valores.

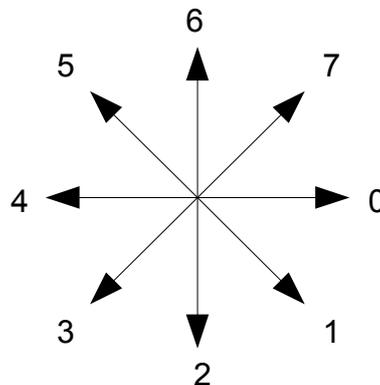


Figura 15: Valores do código em cadeia e suas direções

A utilização desta codificação se dá através da comparação de *pixels*, sendo utilizado especificamente para imagens binárias(branco e preto), onde, após encontrar um *pixel* contrastando com o anterior, começará a seguir a direção horário ou anti-horário, pelos *pixels* de cores semelhantes, gerando uma cadeia de valores. A figura 16 a abaixo representa uma imagem com a seguinte função $f(8,8)$, representando um G :

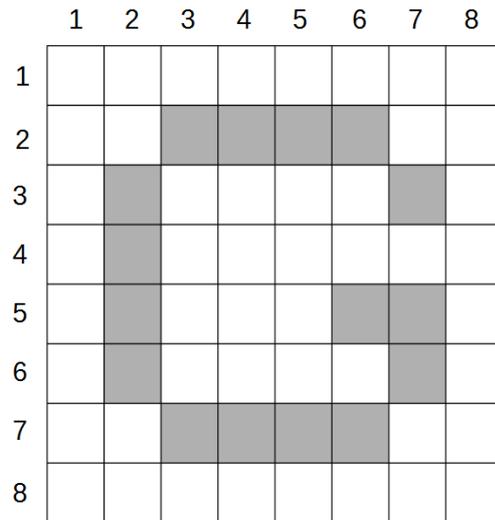


Figura 16: Imagem de exemplo para utilizar o código em cadeia

Utilizando o Código em Cadeia, é possível gerar uma cadeia de valores $C = 7\ 0\ 0\ 0\ 1\ 1\ 3\ 5\ 4\ 4\ 4\ 2\ 2\ 2\ 0\ 0\ 0\ 5\ 7\ 0\ 1\ 2\ 3\ 3\ 4\ 4\ 4\ 5\ 5\ 6\ 6\ 6$, seguindo a partir da posição $f(1,1)$ no sentido horário, onde qual geraria uma representação como a figura 17 abaixo.

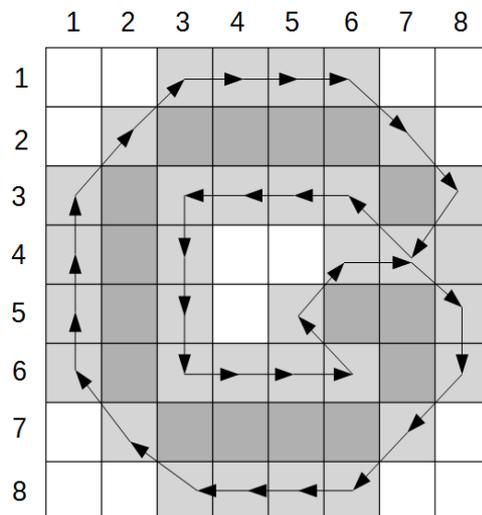


Figura 17: Aplicado o código em cadeia a imagem 18

2.2.2. histograma

Histograma é uma técnica, segundo é possível interpretar da afirmação de Gonzales e Woods, a quantidade de *pixels* os quais possuem o mesmo valor de intensidade. Sendo

possível definir matematicamente com a fórmula 6 abaixo, onde h sendo o histograma, r_k representado um valor de intensidade e n_k sendo a quantidade de *pixels* de uma imagem que tenham aquele valor:

$$h(r_k) = n_k$$

Fórmula 6: Função do histograma

A figura 18 abaixo demonstra a geração de um histograma.

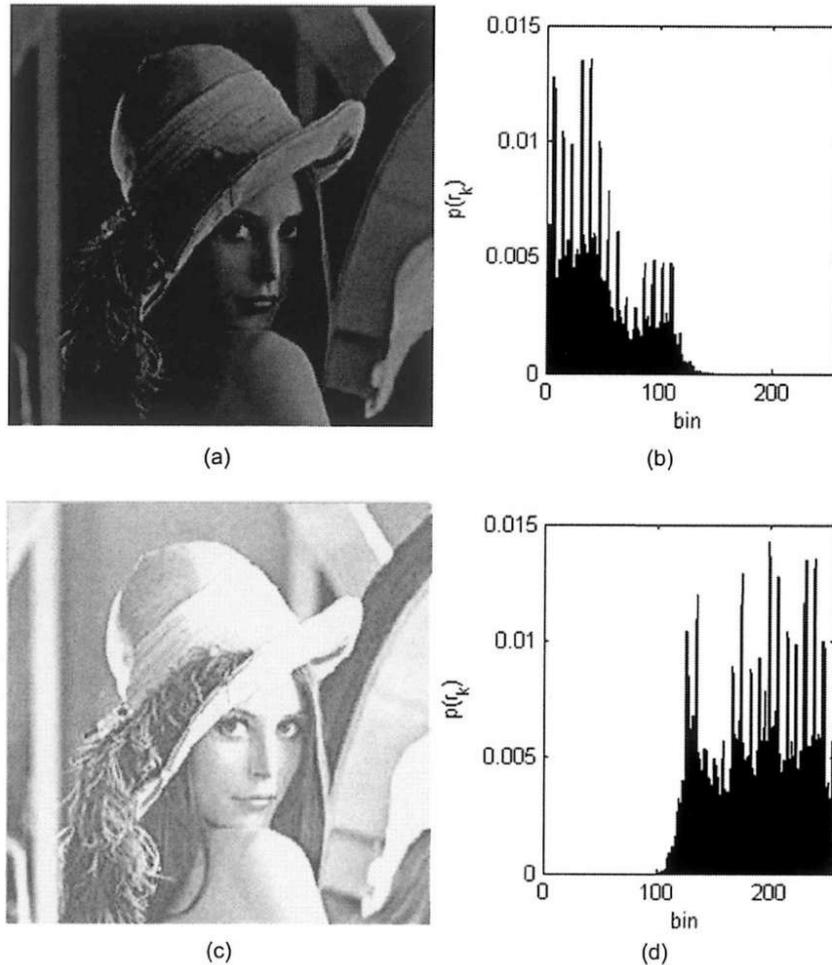


Figura 18: a) Imagem escura e b) seu histograma. c) Imagem clara e d) seu histograma.

2.2.3. threshold

Threshold é uma técnica para transformar os valores de uma imagem para binário, 0(preto) e 1(branco), ou 0 e 255, baseado na intensidade encontrado no *pixel*. Gonzalez e Woods definem a seguinte fórmula, mostrado na Fórmula 6 para a conversão, onde $g(x,y)$ representa a nova imagem e a constante T um valor entre 0 e 255 a ser comparado, sendo maior igual a este valor branco, menor preto.

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) \geq T \\ 0 & \text{se } f(x, y) < T \end{cases}$$

Fórmula 7: Fórmula de execução do *threshold*

Utilizando a figura 19, e aplicando o *threshold* com $T=127$, é possível obter o seguinte resultado abaixo.



Figura 19: Efeito do filtro *threshold*

2.3. INTELIGÊNCIA ARTIFICIAL

Inteligência Artificial (IA) é um campo de pesquisa da área da *Ciência da Computação* no qual foca no desenvolvimento de entidades automatadas, no qual podem se assemelhar a capacidade cognitiva e comunicativa humana, ou desenvolvidos diretamente para a automação de processos de forma eficiente.

Norvig et al. (2009) definem a *IA* em dois grupos, *IA* humana e racional, entidades

automatas com o principal foco de mimetizar ou assemelhar certas características humanas de forma parcial ou integral, podendo ser capacidades cognitivas, comunicativas ou interativas. IA racional, entidades automatas com foco na eficiência de soluções de problemas, funcionam a partir de regras "ensinadas" . IA humana e IA racional podem, também serem definidas como *Strong AI*(IA Forte) e *Weak AI*(IA Fraca) respectivamente.

Alguns autores como Norvig et al. (2009) e Jones(2007), definem o nascimento da campo de IA entre as décadas de 40(1940) e as décadas de 50(1950), o primeiro trabalho no campo foi desenvolvido por Warren McCulloch e Walter Pitts onde propuseram um modelo de neurônios artificiais, cada neurônio tendo a capacidade de ligar e desligar, sendo que caso um neurônio estivesse ligado, seria uma resposta a um estímulo aplicado a ele.

A partir desta data até a metade da década de 60 e um pouco da década de 70, o campo de IA começou a sofrer uma certa queda de interesse pelas dificuldades técnicas aplicadas a ela e a grande excitação dos cientistas no qual levaram a fazer previsões extravagantes ao tempo que levaria o desenvolvimento de IAs capazes de se aproximar ao ser humana, este espaço de tempo foi chamado de Inverso da Inteligência Artificial(AI Winter).

Após esta época o campo da IA começou a ser trabalhado com métodos científicos e testes empíricos, e a geração de novas teorias, assim como um maior investimento e interesse pela indústria.

2.3.1. aprendizado de máquina

Machine learning ou *Aprendizado de Máquina*(AM), pode ser definido, de acordo com Rostamizadeh et. al.(2012), como métodos de melhoramento de performance e previsões a partir de experiências, ou como Alpaydin(2014) define, programas de computadores para melhoramento e otimização de buscas através de análise de dados. Seguindo estas duas definições, AM, um campo de estudo da área de IA, é uma combinação de metodologias probabilísticas, classificatórias e redução, para a geração de novos dados e informações.

AM possui diversas aplicações, sendo possível citar algumas como a análise de dados de compra de uma loja, reconhecimento facial ou caligrafia. Todas estas aplicações contêm uma característica semelhante, um sistema de aprendizagem ou treinamento.

Autores fazem a representação de técnicas de *AM*, utilizando um espaço euclidiano, onde as linhas verticais e horizontais representam parâmetros de informações, podendo variar de cores, sabores, ou valores numérico. A quantidade de parâmetros refere-se a quantidade de dimensões a ser visualizada pelo no gráfico. É possível visualizar um exemplo na figura 20 abaixo.

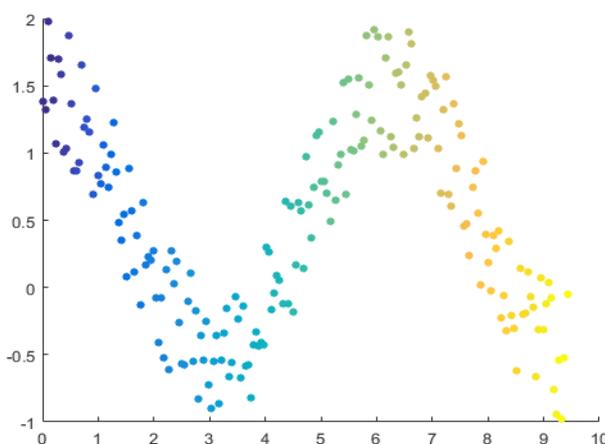


Figura 20: Representação do AM

2.3.1. treinamento supervisionado

Treinamento supervisionado é definido por certas características, a principal é a amostra de treinamento, uma série de dados no qual recebem rótulos específicos, estes rótulos podem representa qualquer tipo de informação, no qual o *AM* deverá aprender. Outra característica é a amostra de validação, contendo dados a serem testados pelo *AM* já treinada.

2.3.2. treinamento não-supervisionado

Treinamento não supervisionado, diferente de métodos supervisionados, tem o foco em prever o que um dado estará representando através de cálculos baseado em características encontradas. Todos os dados neste tipo de técnica, não recebem nenhum rótulo que possa indicar qualquer tipo de informação, a figura 21 abaixo demonstra como os dados são representados.

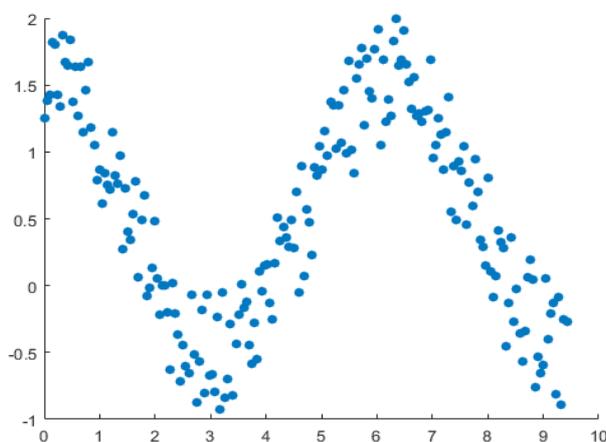


Figura 21: Demonstração de dados em um AM não-supervisionado

2.3.2.1. clustering

Wu(2012) define *cluster* como um método de análise de dados através da organização a partir de características semelhantes. Este tipo de técnica é dicado a gerar *clusters*, “grupos” de dados contendo valores semelhantes. *Clusters*, se organizam a partir de uma série de diferentes cálculos, como probabilidade, distância de valores, etc.

2.3.2.2. k-média

De acordo com Wu(2012), k-média é um algoritmo de particionamento simples de clusters, esta técnica funciona inicialmente a partir da geração de uma quantidade pré definida e fixa de *clusters*, exemplificado na figura 22 abaixo.

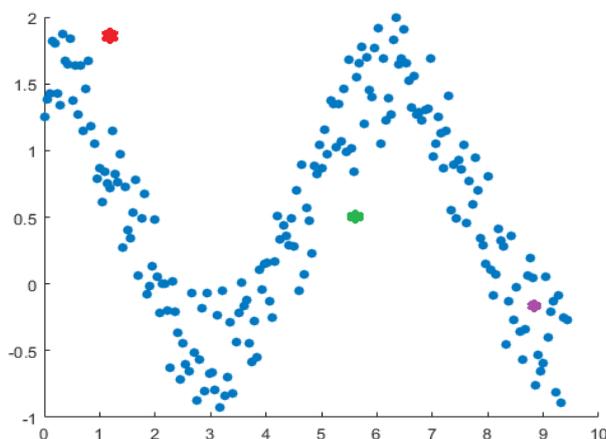


Figura 22: Centroids aplicados ao AM

Esta técnica trabalha atribuindo marcação aos dados mais próximos aos centroides, fazendo uma comparação de distância entre os múltiplos centroides, exemplificado na figura 23 abaixo.

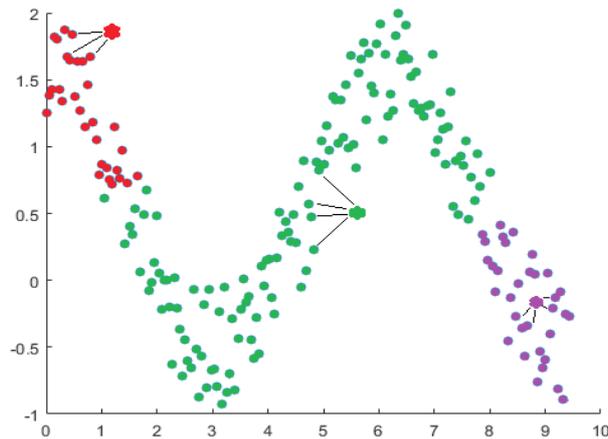


Figura 23: Marcação dos dados aos centroides

Após este passo, é realizado a alteração de posição de todos os *centroids*, assim como a remarcação de todos os dados baseados nos novos *centroids*, é possível citar que existe a possibilidade de *clusters* serem removidos caso nenhum dado tenha sua marcação, as figura 24 e 25 abaixo, representam o mesmo processo executado duas vezes.

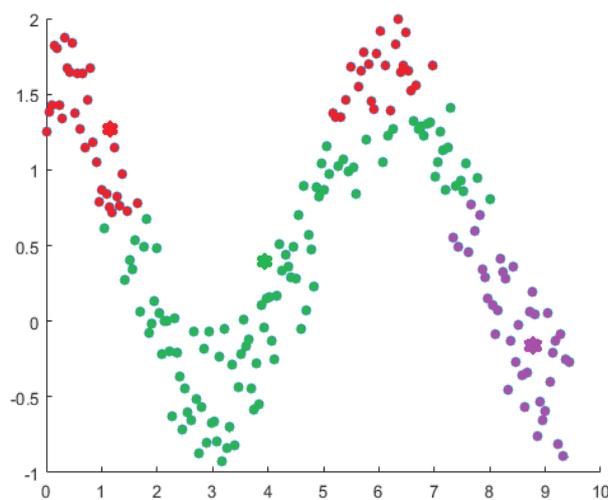


Figura 24: Alteração da posição dos centroides e a remarcação dos dados

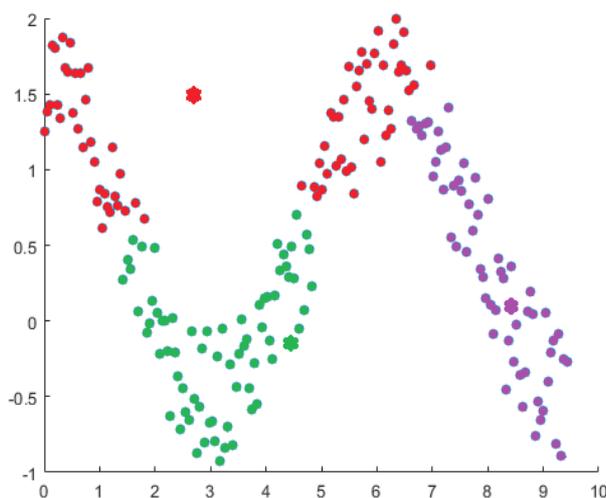


Figura 25: Novamente, alteração de posição e remarcação

O cálculo de obtenção da distância dos centroides e clusters é definido pela função de distância euclidiana, definido na fórmula 8 abaixo, onde p representa as posições do centroid e q representa as posições do cluster, ou vice-versa.

$$d = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

Fórmula 8: Função da distância euclidiana

2.4. C++

C++ é uma linguagem de programação de computadores no qual faz a utilização do paradigma de Programação Orientado a Objetos (POO). Desenvolvida por Bjarne Stroustrup em 1983, Stroustrup(2013) define que o motivo de criar esta linguagem, foi pela necessidade de escrever e ler de forma mais simples e prazerosa, comparado com as soluções encontrada na época. C++, recebeu este nome pelo fato de manter as características da linguagem de programação C, e adicionar características do paradigma POO. C++ é mantido pela Standart C++ Foundation e pela ISO, grupo com foco em organizar e gerar novos padrões para a linguagem. Atualmente o C++ se encontra na versão C++14.

2.5. OPENCV

OpenCV definido por Bradski(2008) é uma biblioteca grátis de visão computacional, este é desenvolvido em C e em C++, além de outras linguagens como Python e Java. Atualmente ela se encontra na versão 3.1 e esta sobre a licença BSD, definindo ela como gratuita para utilização comercial ou acadêmica.

3. METODOLOGIA, DESENVOLVIMENTO, TESTES E RESULTADOS

Este capítulo, demonstra a metodologia de pesquisa utilizada durante o processo de desenvolvimento, a definição do algoritmo de *AM* e do processamento de imagem, além de testes executados, e a interpretação dos resultados obtidos.

3.1. METODOLOGIA

Para a elaboração deste documento e de seu conteúdo teórico, foi realizada pesquisa em livros, artigos e páginas da web.

3.2. DESENVOLVIMENTO

O software, composto por duas partes, o processamento de imagem para a obtenção de dados referentes, e o *k*-média para o processamento destes dados, foi desenvolvido utilizando a linguagem C++, junto com a biblioteca de processamento de imagem OpenCV.

3.2.1. processamento da imagem

O algoritmo de processamento de imagem tem como foco obter o código da cadeia de *Freeman*, este código é utilizado para posicionar a imagem como um *centroid* no *k*-média, o diagrama do algoritmo é definido na imagem 26 abaixo.

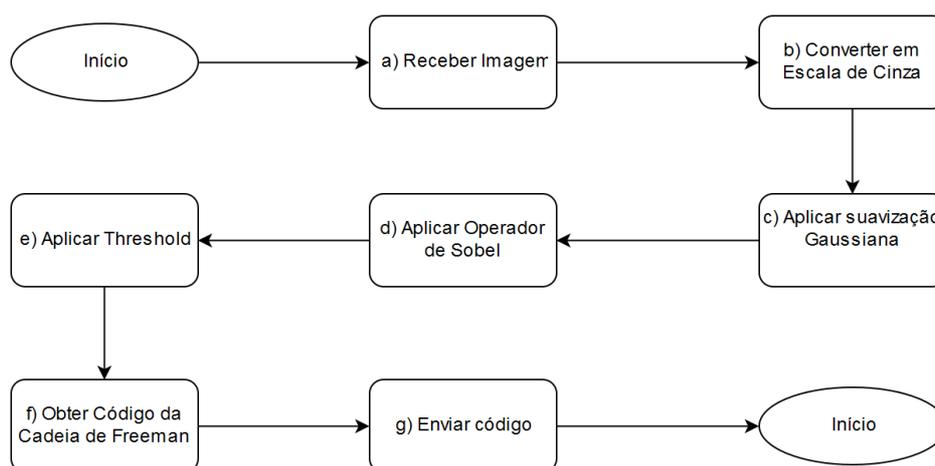


Figura 26: Algoritmo do processamento de imagem

O algoritmo inicia recebendo a imagem a ser processada(a), seguido pela conversão dela para a escala de cinza, em seguida aplicada a suavização gaussiana, próximo passo é aplicado o operador *Sobel* para fazer a obtenção das arestas, seguido pela aplicação do método de *threshold* entre os valores 127 e 255. Finalmente é obtido o código da cadeia

de *Freeman* dos *pixels* externos da imagem processada e enviado ao *k-média*. O resultado final do algoritmo é demonstrado na imagem 27 abaixo.

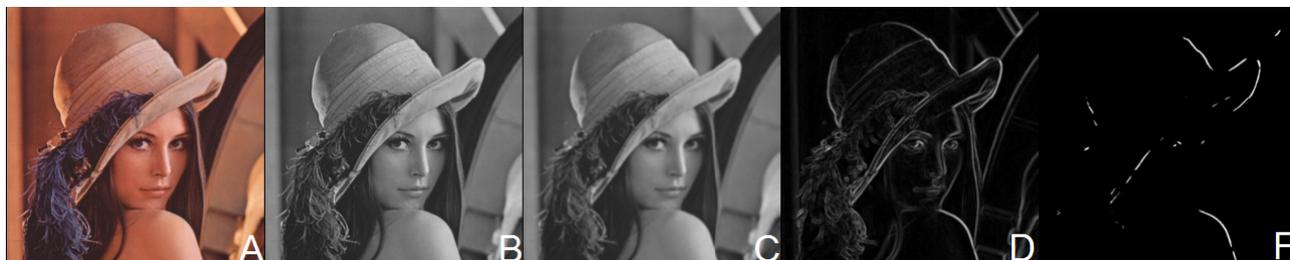


Figura 27: Produto final do algoritmo de processamento de imagem

3.2.2. *k-média*

O algoritmo do *k-média* é separado em duas partes, criação e posicionamento dos *centroids* e *clusters* (A), e categorização dos *centroids* e reposicionamento dos *clusters* (B). A figura 28 mostra o diagrama do algoritmo para a geração *centroids* e *clusters* (A).

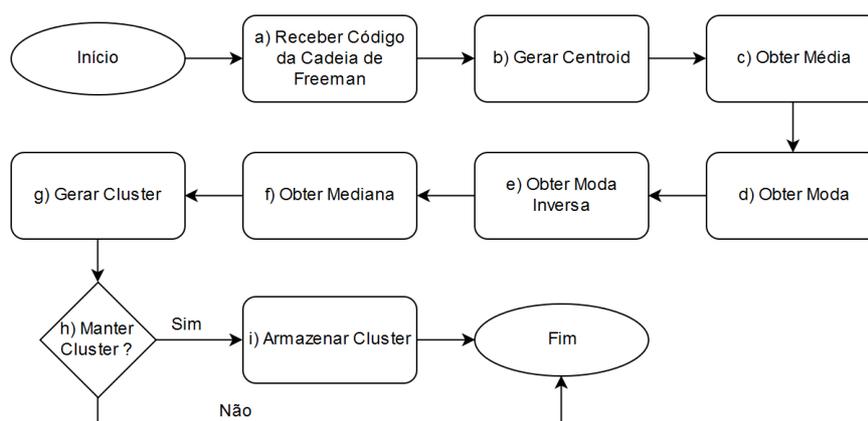


Figura 28: Algoritmo de carregamento do centroid

Os passos b ao f definem a criação do centroid e o seu posicionamento, os valores da média, moda, moda inversa e mediana são obtidos dos valores da cadeia organizado em ordem crescente, a moda inversa é definido como valor de menor aparição maior que 0. O passo g define a criação do cluster, este recebe o valor do centroid criado anteriormente, mas, pode ser removido no passo h caso outro *cluster* se encontre em uma distância x , os passos para a remoção é definido na figura 31 no diagrama abaixo.

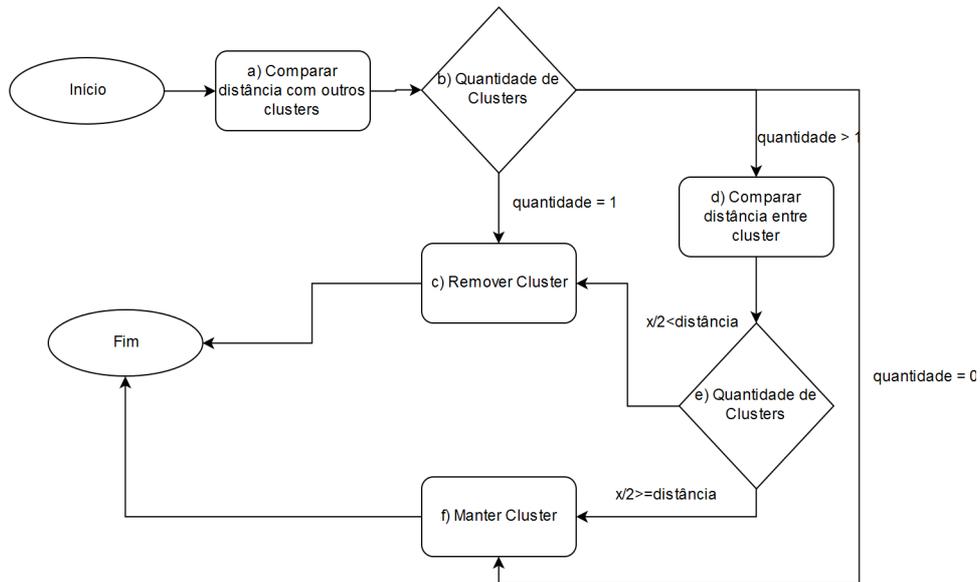


Figura 29: Algoritmo de verificação de cluster

O processo de seleção para manter este novo *cluster* é definido pela distância entre outros *cluster* já presentes, inicialmente é obtida a distância de todos os *clusters*, caso não exista nenhum cluster se encontra dentro de uma distância x , ele é mantido, caso exista um cluster, este é removido. Caso exista mais de um cluster, é novamente comparado com $x/2$, caso a distância seja menor, este cluster é removido, se não, mantido. Este processo é necessário para definir a possível quantidade de cluster para a categorização dos centroides, pois, um cluster é gerado na mesma posição de um centroid. A parte final do k-média é definido na figura 30 abaixo, este faz a categorização dos centroides aos clusters.

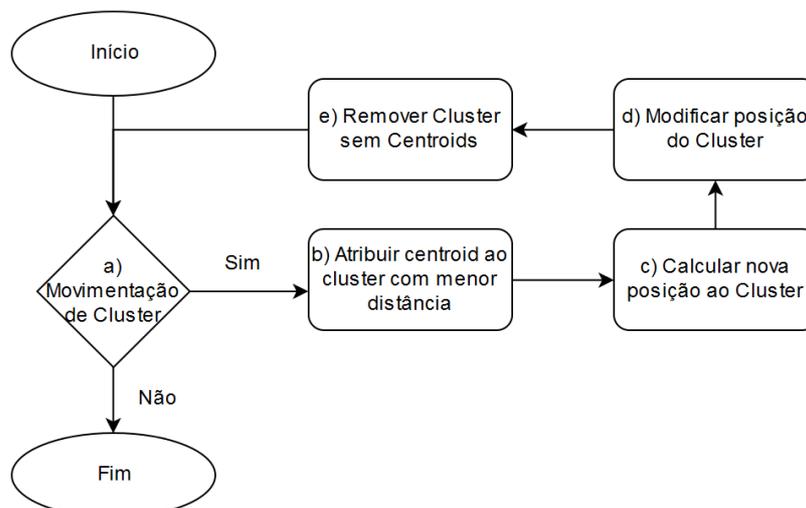


Figura 30: Algoritmo de categorização

Esta parte inicia-se em atribuir centroids aos cluster com a menor distância possível, após a esta atribuição é definido a nova posição aos clusters, caso algum cluster não possua nenhum centroid atribuído a ele, é removido do k-média.

3.2.3. software

A interface visual do software é dividido em duas 3 partes, um visualizador de imagens ao centro, um visualizados de dados dos centroids e clusters a direita, e uma lista em árvores, onde cada novo cluster gerado no qual contenha um ou mais centroids é inserido na lista como um item raiz, e os centroids correspondentes como folhas, a figura 31 abaixo mostra a interface gráfica do programa.

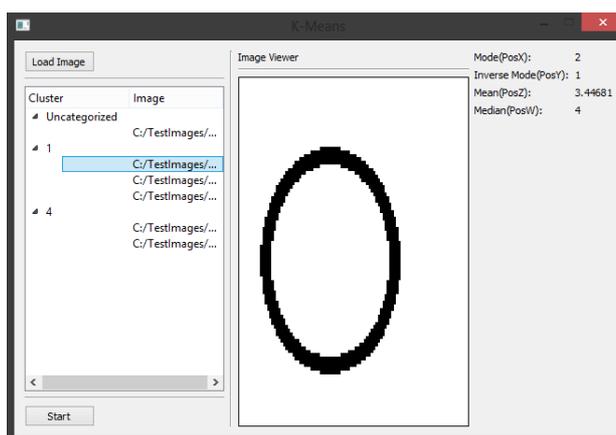


Figura 31: Imagem do software

3.3. TESTE

Os testes foram aplicados na lista de imagens na figura 34 abaixo, sendo um total de 20 imagens no formato *bmp*, compostos de 4 diferentes formas, cada uma possuindo tamanhos diversos; o *kernel* da suavização gaussiana definido como $k=3$, e o valor $\sigma=1$; a distância x para a comparação de *clusters* em suas respectivas criações, como o valor $x=1$.

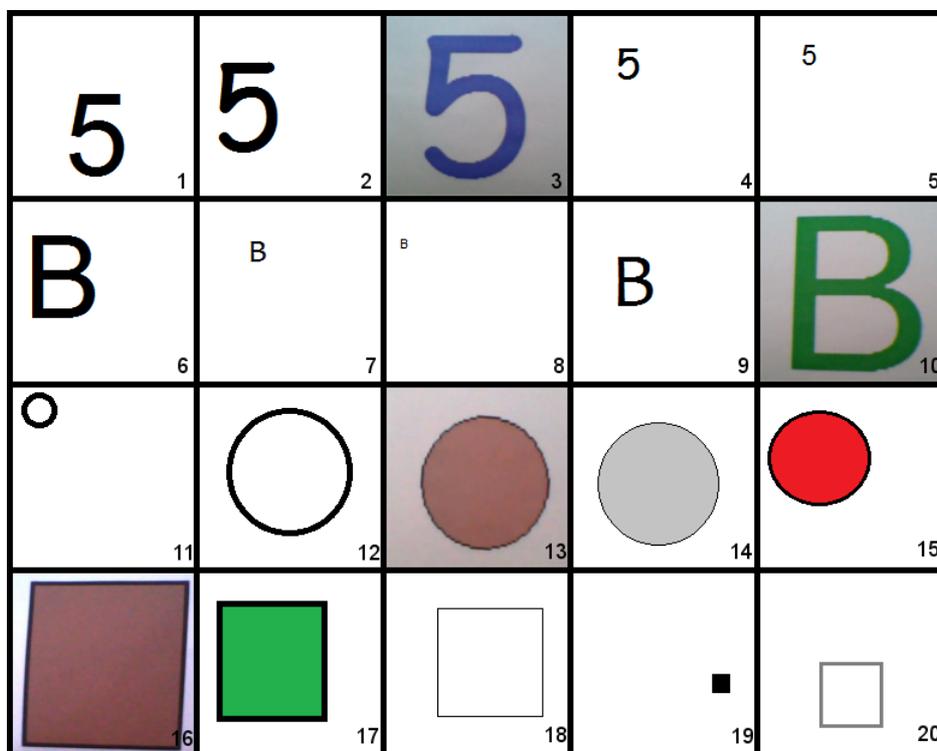


Figura 32: Lista de imagens a serem analisados

A figura 33 abaixo mostra como elas foram carregadas ao *k-média*, elas se encontram inicialmente sem serem categorizadas.

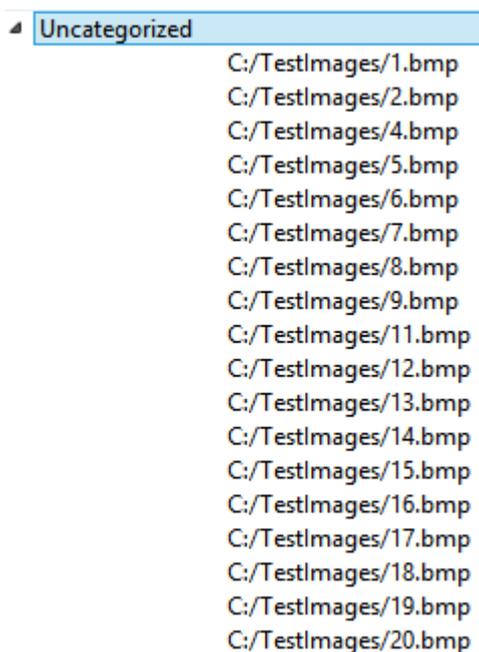


Figura 33: Todas as imagens inseridas

3.4. RESULTADO

A imagem 34 abaixo mostra como foi categorizado as imagens, onde foi gerado um total de 7 clusters.

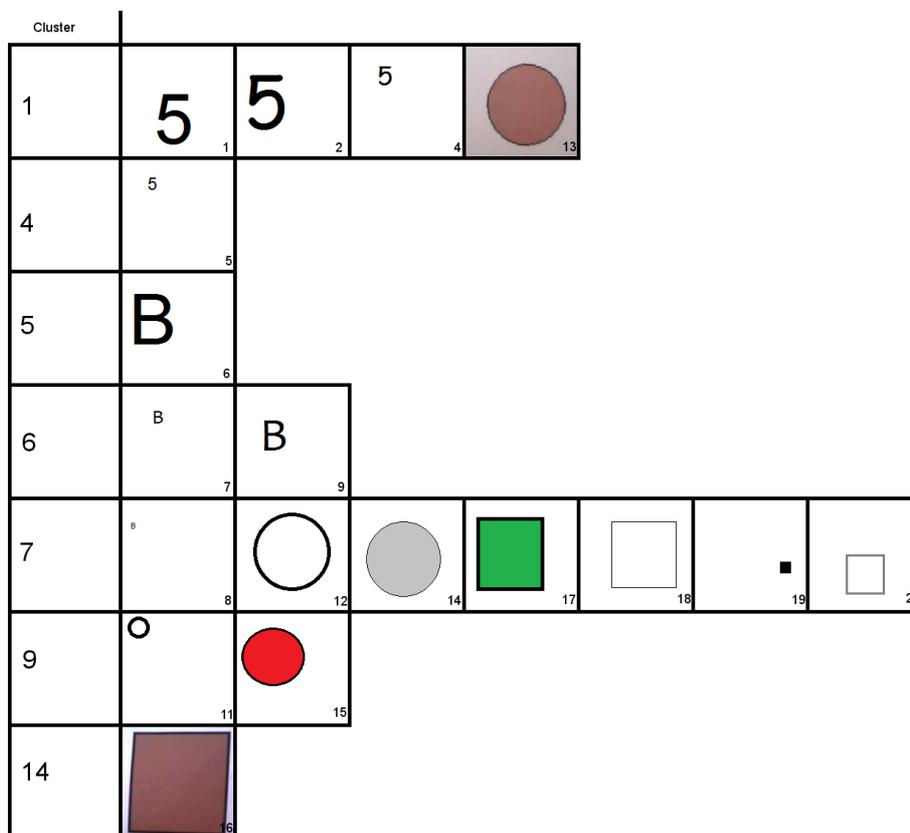


Figura 34: Imagens categorizadas

Inicialmente é possível perceber que o quinto cluster, representado pelo valor 7, contém 7 imagens com um total de 4 formas diferentes, isso ocorre pelo fato dos valores definidos para os centroides durante o processo de criação serem semelhantes, e durante o processo de categorização a distância mínima também irão se assemelhar, isso não significando que a quantidade de *pixels* encontrado em cada uma delas sejam idênticas, a tabela 1 abaixo contém os valores dos centroides, e a tabela 2 contém os valores dos clusters.

Cluster	Posição A	Posição B	Posição C	Posição D
1	4	5	3.19527	4
4	0	5	3.15	4
5	6	5	3.26549	3
6	6	5	3.36523	4
7	2	1	3.23599	4
9	0	1	3.40906	4
14	0	7	2.76998	3

Tabela 1: Tabela de Clusters

Centroid	Moda	Moda Inversa	Média	Mediana
1	4	5	3.16875	4
2	4	5	3.19373	4
3				
4	4	5	3.31858	4
5	0	5	3.15	4
6	6	5	3.26549	3
7	6	5	3.37931	4
8	2	1	3.46667	4
9	6	5	3.35115	4
10				
11	0	1	3.43478	4
12	2	1	3.43934	4
13	4	5	3.1	4
14	2	1	3.45205	4
15	0	1	3.38333	4
16	0	7	2.76998	3
17	2	1	3.06667	4
18	2	1	3.03867	4
19	2	1	3.15152	4
20	2	1	3.03704	4

Tabela 2: Tabela de centroides

Outro ponto a ser notado é o fato de dois centroides, 3 e 10, não serem encontrados em nenhum cluster, isto é causado pelo valor do *threshold* aplicado na imagem, como sendo um método binário e utilizando os valores de comparação 127 e 255, qualquer valor abaixo disso será considerado um *pixel* preto, a figura 35 abaixo demonstra esta ocorrência com o histograma.



Figura 35: a) Imagem após filtro sobel b) Histograma c) Imagem a após aplicado threshold entre 127 e 255

Também é importante notar o fator dos clusters e à distância de verificação, sendo o valor aplicado $x=1$, definiu a geração de um total de 7 clusters, que foram mantidos mesmo após a categorização dos valores.

4. CONCLUSÃO

Este trabalho teve como foco desenvolver um sistema de reconhecimento genérico de imagem utilizando técnicas de *AM* e processamento de imagem.

Pode-se concluir que o método não supervisionado de aprendizado de máquina aplicado, *k*-média, utilizando os algoritmos definidos no trabalho, requer um estudo maior no posicionamento e a quantidade de clusters a serem definidos, além de uma redefinição dos valores obtidos a partir do código em cadeia a serem aplicados nos clusters e centroids. A partir dos testes obtidos foi possível afirmar que a quantidade e a variação de valores, visto na mediana, podem alterar o resultado final do *AM*.

O Processamento de imagem também requer um melhor estudo na definição dos valores aplicados em suas técnicas; o *threshold* com o valor definido de 127 foi ineficaz para a geração de uma imagem que representa-se a silhueta de uma objeto, pois os valores encontrados na imagem processada com o operador sobel, podem ter uma variação entre 0 e 126, sendo estes ignorados pelo *threshold* e a imagem final não representando a imagem original.

Para futuros trabalhos, além de um melhor estudo sobre o tema definido, existe a possibilidade de mudança do atual *AM* para outro com um suporte de técnicas mais complexas, alteração das técnicas de processamento de imagem e também a alteração do escopo geral para uma simplicidade como a identificação de um único tipo de objeto ou, para uma complexidade maior, por exemplo, obtenção de detalhes internos, obtenção de texturas e cores que compõem a imagem.

REFERÊNCIAS

- ALPAYDIN, Ethem. **Introduction to Machine Learning**, 3. ed. Cambridge: The MIT Press, 2014.
- BRADSKI, Gary; KAHLER, Adrian. **Learning OpenCV**, 1. ed. California: O' Reilly Media, 2008.
- FORSYTH, David A.; PONCE, Jean. **Computer Vision – A Modern Approach**, 2. ed. New Jersey: Pearson Education, 2012.
- GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing**. 3. ed. New Jersey: Pearson Education, 2008.
- Google. **Google Self-Driving Car Project**. Disponível em <<https://www.google.com/selfdrivingcar/>>. Acesso em: 18 mar. 2016.
- JAIN, Anil K. **Fundamentals of Digital Image Processing**. 1. ed. New Jersey: Pearson Education, 1989.
- JONES, M. Tim. **Artificial Intelligence – A System Approach**, 1. ed. Hingham: Infinity Science Press LLC, 2008.
- MOHRI, Mehryar; ROSTAMIZADEH, Afshin; TALWALKAR, Ameet. **Foundations of Machine Learning**, 1. ed. Cambridge: The MIT Press, 2012.
- RUSSEL, Stuart J.; NORVIG, Peter. **Artificial Intelligence – Modern Approach**, 3. ed. New Jersey: Pearson Education, 2010.
- SOBEL, Irwiw. **History and Definition of the so-called "Sobel Operator", more appropriately named the Sobel-Feldman Operator**. Disponível em <https://www.researchgate.net/publication/281104656_An_Isotropic_3x3_Image_Gradient_Operator>. Acesso em: 19 mar. 2016.
- VERNON, David. **Machine Vision**, 1. ed. New Jersey: Prentice Hall, 1991.
- WU, Junjie. **Advances in K-means Clustering – A Data Mining Thinking**, 1. ed. New York: Springer, 2012.