

Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Computação e Automação
Engenharia de Computação

Processamento Digital De Imagens

Lista 1

Discente
Arthur Diniz Flor Torquato Fernandes

Professor: Adrião Duarte Doria Neto

Natal
03 de outubro de 2020.

Sumário

1	Lista	3
1.1	<i>Questão 1</i>	3
1.2	<i>Questão 2</i>	3
1.3	<i>Questão 3</i>	5
1.4	<i>Questão 4</i>	6
1.5	<i>Questão 5</i>	6
1.6	<i>Questão 6</i>	8
1.7	<i>Questão 7</i>	9
1.8	<i>Questão 8</i>	11
1.9	<i>Questão 9</i>	11
1.10	<i>Questão 10</i>	11
1.11	<i>Questão 11</i>	11
1.12	<i>Questão 12</i>	11
1.13	<i>Questão 13</i>	13
1.14	<i>Questão 14</i>	14
1.15	<i>Questão 15</i>	14
2	Trabalho	15

1 Lista

1.1 Questão 1

Temos abaixo a lista de probabilidades fornecidas na questão:

$$p(\text{Teste} = \text{Positivo} / \text{Doença} = \text{Verdadeira}) = 0.98$$

$$p(\text{Doença} = \text{Verdadeira}) = 1/10000$$

a partir desses valores podemos calcular a probabilidade da teste ser positivo e da doença ser verdadeira

$$p(PV) = p(P/V) * p(V) = 0.98 * 1/10000 = 0.000098$$

logo temos que o paciente possui uma probabilidade de 0.0098% de chance de ter o teste positivo e possuir a doença

Temos que a equação apresentada representa o processo de conversão de uma imagem analógica para digital e que para se retirar os elementos indesejados dessa imagem é necessário retirar os elementos indesejados(representados pelo ruído $n(x,y)$) dela por meio de filtros redutores de ruídos.

1.2 Questão 2

Primeiro calculamos o vetor media obtendo

$$\hat{\mu} = \begin{bmatrix} 4.75 \\ 4 \\ 6.5 \end{bmatrix}$$

Sabendo que os elementos da matriz de covariância são dados por:

$$S_{ij} = \frac{\sum_{l=1}^N (x_{il} - \hat{\mu}_i)(x_{jl} - \hat{\mu}_j)}{(N - 1)}$$

Como a matriz é simétrica obtemos os elementos

$$S_{11} = 5.5833...$$

$$S_{22} = 5.33$$

$$S_{33} = 7$$

$$S_{12} = -2$$

$$S_{13} = 3.17$$

$$S_{23} = 2.67$$

escrevemos então a matriz de covariância

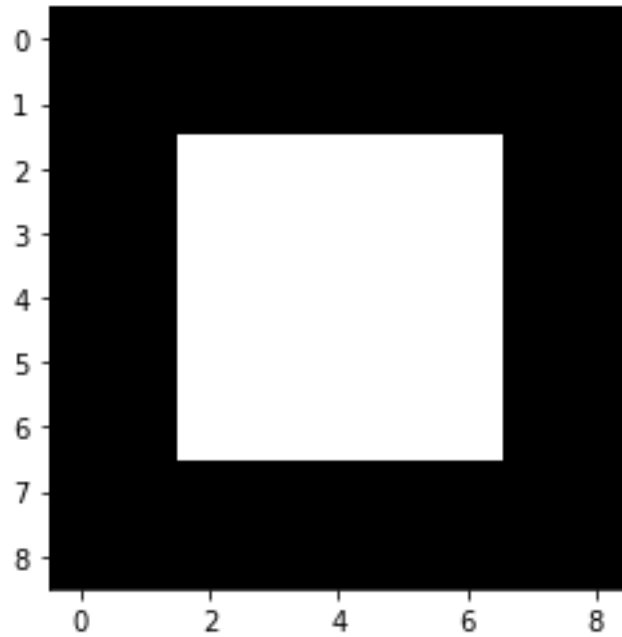
$$S = \begin{bmatrix} 5.5833 & -2 & 3.17 \\ -2 & 5.33 & 2.67 \\ 3.17 & 2.67 & 7 \end{bmatrix}$$

temos que a distribuição gaussiana multivariada é dada pela formula

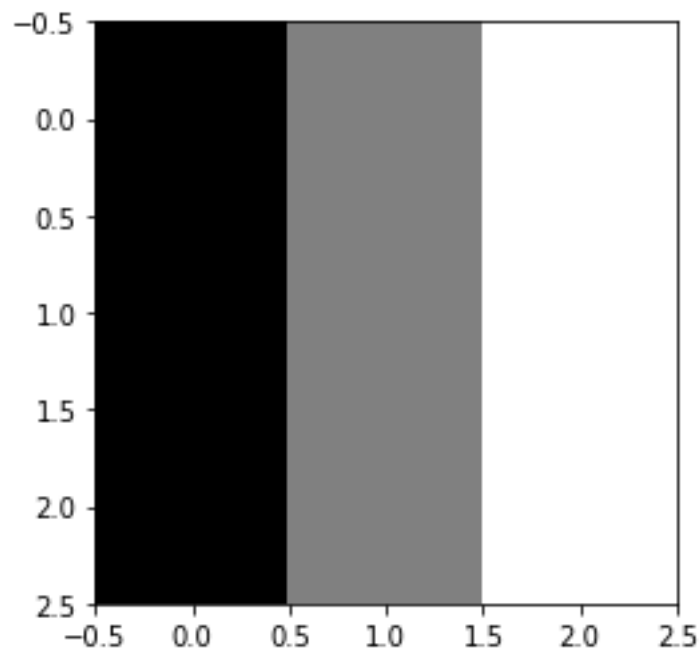
$$p(x) = \frac{1}{(2\pi^{3/2} \det(S)^{1/2})} \exp -\frac{1}{2}[(x - \hat{\mu})^t S^{-1} (x - \hat{\mu})]$$

sendo x um vetor de 3 variáveis

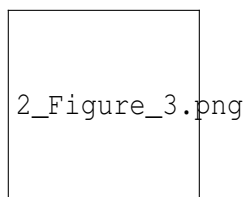
Temos que a matriz é representada pela imagem abaixo



Abaixo temos a representação da mascara



por fim temos a convolução das imagem com a mascara



1.3 *Questão 3*

Temos que para realizarmos o teste t-student primeiro precisamos calcular o valor de t dado pela formula

$$t = \frac{-\mu}{s\sqrt{n}}$$

aplicando os valores fornecidos na questão

$$\hat{\mu} = 0.06, n = 20, \hat{x} = 0.07, s = 0.008$$

obtemos

$$t = 5.59$$

A) Temos que pela tabela de distribuição t de student temos que t precisa ser maior que 2.0930 como ele o é então a hipótese nula é rejeitada e adotamos a nova hipótese

B) De forma analoga a letra A a hipotese nula também será rejeitada pois t tem um valor maior que 2,8609

a) O sistema deve percorrer as bordas da imagem e aplicar o algoritmo flood-fill sempre que encontrar um ponto preto(e preenche-lo de branco)

b) Após realizar isso deve-se percorrer por inteiro e sempre que um objeto for encontrado aplicar o algoritmo flood-fill com uma label diferente para cada objeto, e ao mesmo tempo deve-se incrementar um contador.

c) Devemos aplicar a função do openCv findContours para achar os contornos da imagem e após isso aplicar em cada contorno encontrado o método countourArea para acharmos a área de cada objeto

d) Devemos usar a função approxPolyDP para pegarmos o numero de vertices de cada contorno e a partir dai caracterizar os objetos como circulos, quadrados e triângulos

e) Enquanto aplicarmos o floodfill devemos armazenar os pontos que contem objetos em uma lista e assim teremos a posição dos objetos em cena

1.4 Questão 4

1.5 Questão 5

Para se resolver essa questão precisamos aplicar diretamente o método de naive-bayes para isso devemos calcular as probabilidades

$$p(\text{MauPagador} = \text{Sim} / (\text{CasaPropria} = \text{Não}, \text{EstadoCivil} = \text{Divorciado}, \text{Rendimentos} = \text{Alto})) = ?$$

$$p(\text{MauPagador} = \text{Sim} / (\text{CasaPropria} = \text{Não}, \text{EstadoCivil} = \text{Divorciado}, \text{Rendimentos} = \text{Alto})) = ?$$

Para conseguir esse valor aplicamos diretamente o teorema de Bayes

$$p(MP / (CP = N, EC = D, R = A)) = \frac{p(CP = N / MP) p(EC = D / MP) p(R = A / MP) p(MP)}{p(CP = N) p(EC = D) p(R = A)}$$

Calculamos esses valores

$$p(MP = S) = \frac{3}{10}$$

$$p(CP = N) = \frac{7}{10}$$

$$p(EC = D) = \frac{2}{10}$$

$$p(R = A) = \frac{3}{10}$$

$$p(CP = S / MP = S) = \frac{2}{3}$$

$$p(CP = S / MP = N) = \frac{4}{7}$$

$$p(EC = D/MP = S) = \frac{1}{3}$$

$$p(EC = D/MP = N) = \frac{1}{7}$$

$$p(R = A/MP = S) = \frac{0}{3} = 0$$

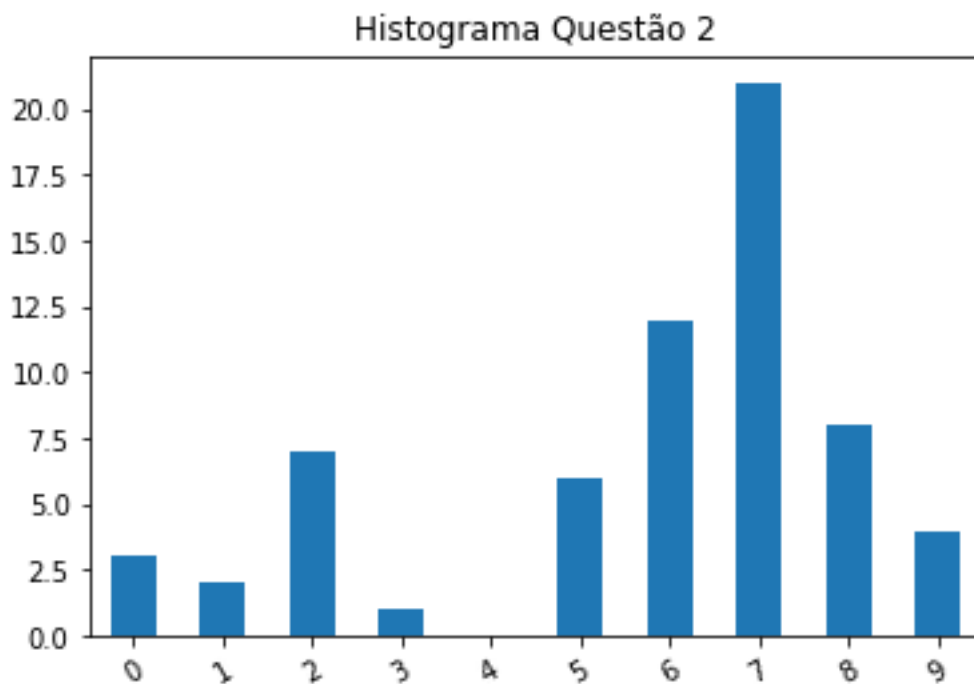
$$p(R = A/MP = N) = \frac{3}{7}$$

aplicando os valores na formula temos

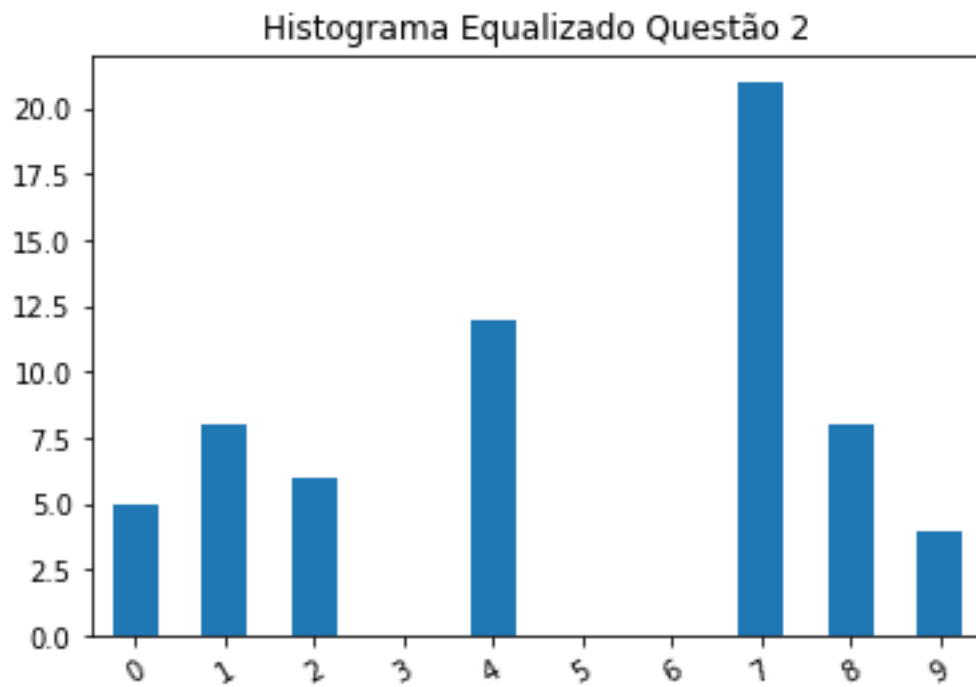
$$p(MP = S/CP = N, EC = D, R = A) = 0$$

$$p(MP = N/CP = N, EC = D, R = A) = 0.58$$

logo pela probabilidade o banco deve fornecer o emprestimo pois existe uma probabilidade maior do cliente não ser mau pagador A) Temos abaixo o histograma da imagem



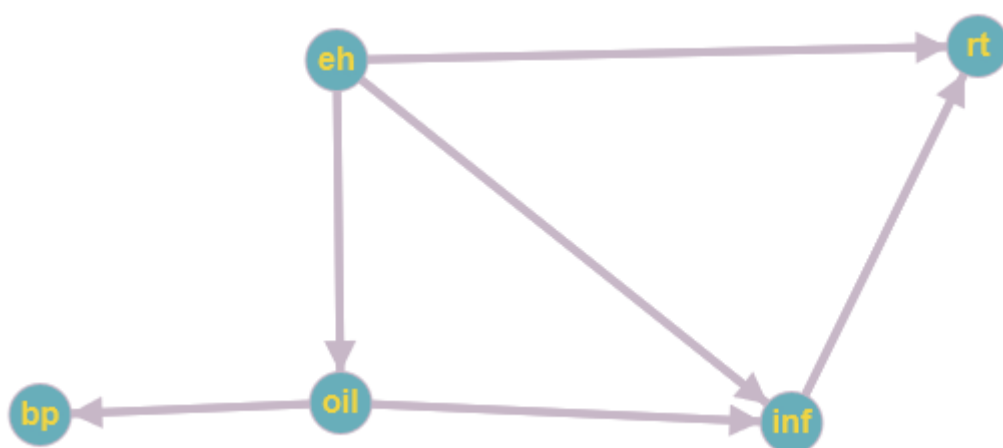
B) Abaixo está o histograma equalizado da imagem



C) Temos que depende do resultado desejado, se o objetivo for aumentar o contraste então a equalização do histograma normalmente produz os melhores resultados mas se for outra prioridade então a equalização não produz o melhor resultado.

1.6 Questão 6

a) Temos abaixo a representação gráfica da rede de crença do problema expressada abaixo



b) Queremos saber

$$p(inf = h / (rt = h, bp = n)) = \frac{p(inf = h, rt = h, bp = n)}{p(rt = h, bp = n)}$$

então temos

$$p = \alpha \sum_{eh} \sum_{oil}^{h,l} p(eh) p(oil/eh) p(bp = n/oil) p(inf = h/(eh, oil) p(rt = h/(inf = h, eh)))$$

$$\alpha = \frac{1}{p(rt = h, bp = n)}$$

$$p(rt = h, bp = n) = \sum_{eh} \sum_{oil}^{h,l} \sum_{inf}^{h,l} p(eh) p(oil/eh) p(bp = n/oil) p(inf/(eh, oil) p(rt = h/(inf, eh)))$$

logo temos

$$p(inf = h/(rt = h, bp = n)) = \frac{0.3099708}{0.3163869} \approx 0.98$$

a) O sistema visual humano percebe a cor através de bastonetes que captam diferentes frequências de luz

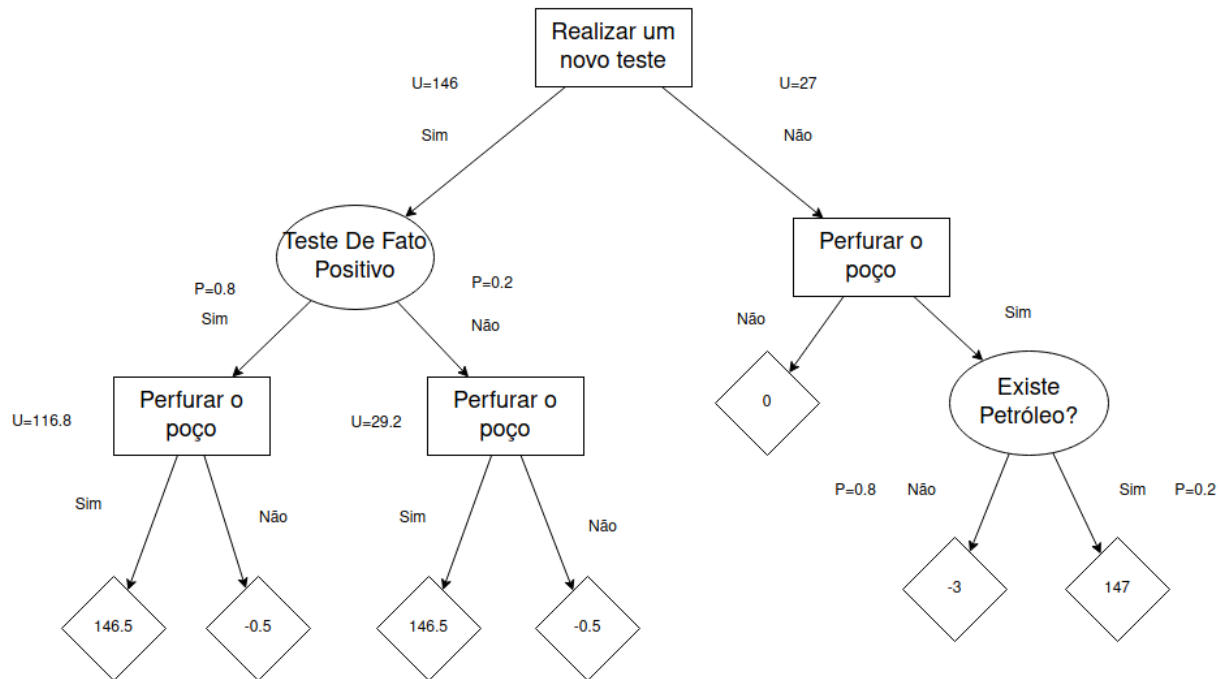
b) A teoria de composição de cores diz que a cor é feita a partir de um espectro da luz branca que é refletida por materiais que absorvem parte das cores da luz

c) A teoria da composição de cores é utilizada para capturar imagens coloridas a partir de sensores que captam a a luz vermelha e verde e a complementam com a azul Essas imagens são mostradas em monitores a partir de luzes RGB que juntas compõem os tons das cores e temos que para se imprimir imagens se utilizam-se elementos complementares do RGB para se representar cores

d) Temos que os 3 atributos de cores são matiz, saturação e contraste

1.7 Questão 7

Abaixo temos uma figura representando a arvore com as utilidades de cada no



- a) Conforme figura acima
- b) Conforme figura acima
- c) temos que a melhor decisão é realizar um novo teste pois esse apresenta a maior utilidade máxima
- d) 146.5 milhões de reais
- e)

```

1     def checar_arvore(no_atual):
2         if no_atual.tipo_de_no == losangulo:
3             return no_atual.valor
4         else:
5             if no_atual.tipo == retangulo:
6                 return checar_arvore(no_atual.proximo_no_sim) +
7 checar_arcore(no_atual.proximo_no_nao)
8             if no_atual.tipo == elipse:
9                 return prob_sim*checar_arvore(no_atual.proximo_no_sim)
10 + *prob_naochecar_arcore(no_atual.proximo_no_nao)
11
12     '''Main'''
13     U1 =checar_arvore(no_atual.no_sim)
14     U2 = checar_arvore(no_atual.no_nao)
15     if U1 > U2:
16         print("Escolha U1")
17     elif U1 <U2:
18         print("Escolha U2")

```

```

16     else:
17         print("Escolha Indiferente")
18

```

Primeiro é preciso se aplicar um filtro de canny para fazer uma detecção de bordas, após isso deve-se aplicar a transformada de hough e checar os ângulos na faixa desejada para serem exibidos

1.8 *Questão 8*

Assumindo que se teria um ruído branco na imagem, se aplica um filtro gaussiano para eliminar os ruídos, após isso se equaliza o histograma da imagem e por fim aplicar uma falsa cor nele

1.9 *Questão 9*

O algoritmo deve primeiro eliminar as bolhas das bordas da imagem e após isso percorrer a imagem calculando a área de todas as bolhas e utilizando o floodfill para eliminá-las de uma cópia da imagem, as áreas de cada bolha devem ser guardadas. após isso paga-se o valor da menor área e elimina-se todas as bolhas com valores maiores que essas bolhas por meio do algoritmo floodfill

1.10 *Questão 10*

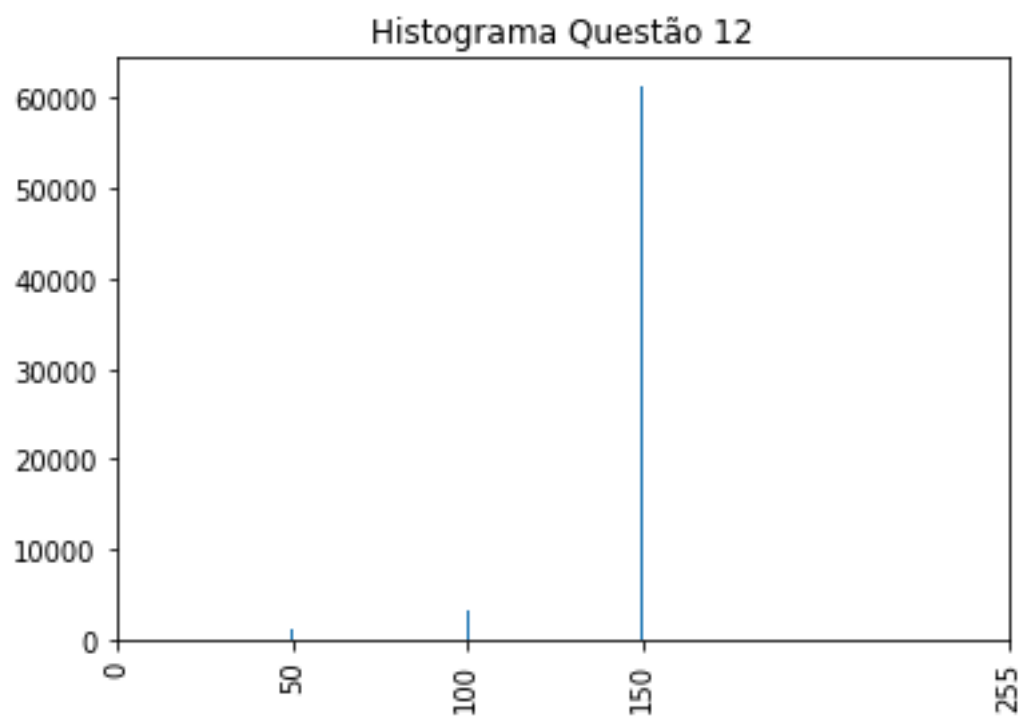
a) o efeito no histograma seria que só existiriam valores pares representados neles e devido a isso o histograma aumentaria na vertical b) o histograma seria encurtado e todos os seus valores seriam divididos por 2

1.11 *Questão 11*

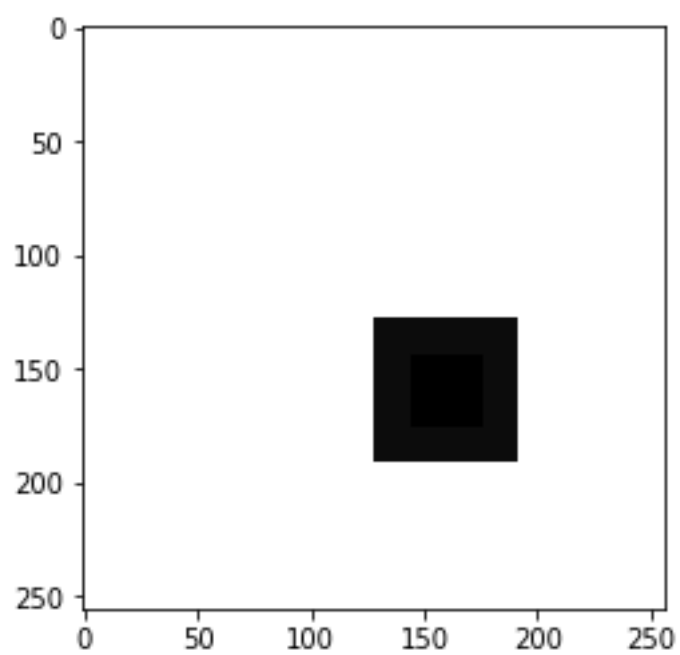
Máximo Acima Do Limiar: Seta 1 Equalização: Seta 5 Laplaciano: Seta 2 Filtro De Média: Seta 6 Filtro Da Mediana: Seta 4 Detector de Bordas de Sobel: Seta 7 Filtro de aguçamento: Seta 3 Transformação de brilho: Seta 8

1.12 *Questão 12*

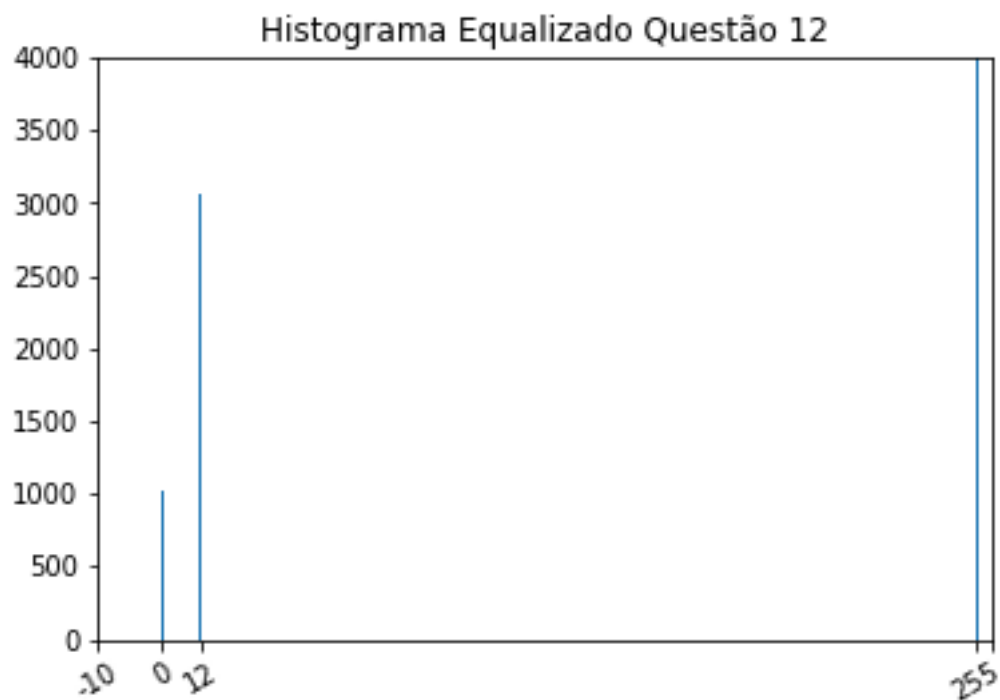
a)



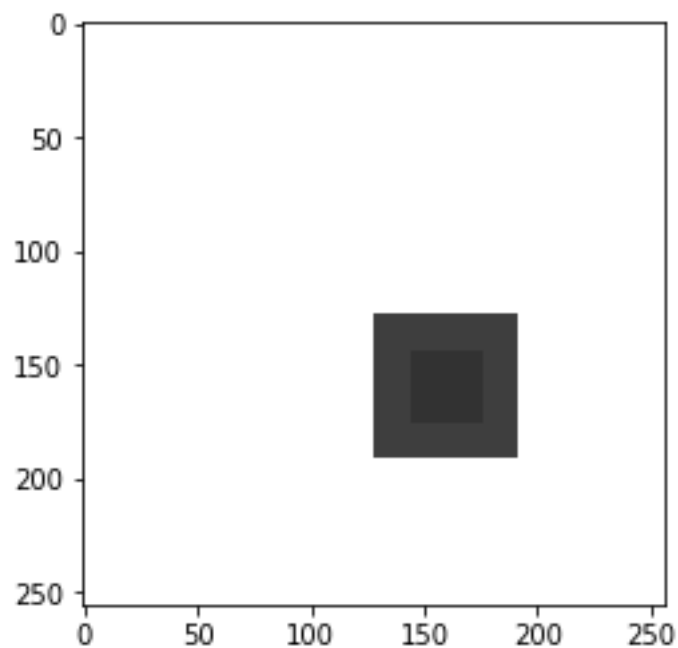
b)



c)



d)



1.13 *Questão 13*

Temos que o histograma da imagem A tera os seus valores elementos concentrados nas bordas pois ao se fazer a média dos valores temos que como a imagem apresenta um padrão bastante dividido

Já na segunda imagem ocorre o contrario devido as cores estarem dispostas em um padrão

de tabuleiro de xadrez, logo ao se aplicar o filtro da média ocorrerá uma homogeneização do histograma em direção ao centro

1.14 *Questão 14*

As condições que devem ser consideradas são iluminação semelhante, posicionamento semelhante e poucos ruídos da imagem

1.15 *Questão 15*

a) Matriz do filtro laplaciano que detecta todas as bordas de uma imagem

$$S = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

b) Matriz de sobel para diagonais pois pode-se observar que somente anti-diagonais não são detectadas na imagem

$$S = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

c) Mesmo caso da letra b entretanto com uma matriz de sobel para anti-diagonais

$$S = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

d) Mesmo caso que o anterior mas com uma matriz de sobel horizontal

$$S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2 Trabalho

Abaixo se encontra o código fonte utilizado para realizar o filtro de spam para a solução adotada

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 from sklearn.naive_bayes import MultinomialNB
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import accuracy_score
5 import pandas as pd
6
7 '''Leitura do DataSet'''
8 df = pd.read_csv("spam_or_not_spam.csv")
9 X = df.email
10 y = df.label
11
12 '''Separar Dataset em conjunto de dados e conjunto de testes'''
13 X_train, X_test, y_train, y_test = train_test_split(X,y)
14
15 '''Vetorizar os dados e criar variavel que contem numero de repeticao de
    palavras'''
16 vectorizer = CountVectorizer()
17 counts = vectorizer.fit_transform(X_train.values.astype('U'))
18
19 '''Criar Classificador do tipo MultinomialNB() e treinamento desse
    classificador'''
20 classifier = MultinomialNB()
21 targets = y_train.values
22 classifier.fit(counts,targets)
23
24 '''Converter dados para teste'''
25 vetor_x_teste = CountVectorizer(vocabulary=vectorizer.vocabulary_)
26 vetor_x_teste = vetor_x_teste.transform(X_test.values.astype('U'))
27 y_test = y_test.values
28 y_test_resultado=classifier.predict(vetor_x_teste)
29
30 '''Obter o nivel de acuracia do classificador'''
31 score=accuracy_score(y_test, y_test_resultado)
```

Temos que o código teve como saída o valor de 0.9933333... logo teve uma acurácia bastante expressiva