

PROJETO INDIVIDUAL - PROCESSAMENTO DIGITAIS DE IMAGENS

Matheus Gois Vieira - 17/0019136

Universidade de Brasília

1. INTRODUÇÃO

O projeto individual visa prover aos alunos um espaço de desenvolvimento de senso crítico e capacidade de avaliação de problemas e suas soluções. Assuntos abordados:

- **Transformações de intensidade no domínio espacial:** Nesta tema é proposto a utilização de equalização de histograma para realizar um ajuste intensidade em uma imagem de baixo contraste.
- **Realce de imagem de baixa resolução:** Imagens boradas e com baixa nitidez, serão tratadas com filtros no domínio da frequência, utilizando filtros laplacianos.
- **Filtragem de ruído:** Eliminar ruídos de uma imagem através de filtros notchs, trabalhando no domínio da frequência.
- **Granulometria;** Neste tópico será abordado a classificação de componentes dentro de uma imagem, utilizando técnicas de morfologia matemática.

Aplicação de pseudocor por fatiamento de intensidade: Dado uma imagem com pontos de intensidade variáveis é necessário exibir um mapa de cor para cada ponto de intensidade.

Além de trabalhar com essa técnicas abordados pelo projeto, também será tratado e demostrado, possíveis melhorias nas imagens, como recuperação de detalhes e objetos.

2. AJUSTE INTENSIDADE

Um problema muito comum em vários tipos de imagens é a falta de intensidade e baixo contraste, como as imagens utilizadas como entrada do nosso algoritmo mostradas abaixo;

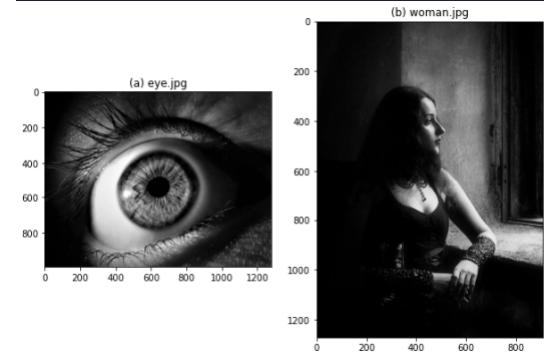


Fig. 1. (a) Imagem de um olho humano, com um falta de intensidade nas laterais da imagem. (b) Imagem de uma mulher, com um contraste muito baixo. Ambas imagens tiradas após a pesquisa no Google Imagens

Para ajustar a intensidades das imagens acima temos que primeiramente desenvolver e analisar o histograma de cada imagem, para isso foi desenvolvido a função **counter** da biblioteca **collections** onde é feito a contagem dos diferenças valores de intensidades dos pixels da imagem, com isso temos os r_k (da imagem original ($p_r(r)$)), para isso, é preciso encontrar r_k (diferentes níveis de intensidade dos pixels da imagens) e n_k (número de pixels com intensidade r_k), com isso temos os seguintes histogramas da imagem eye.jpg e woman.jpg, respectivamente:

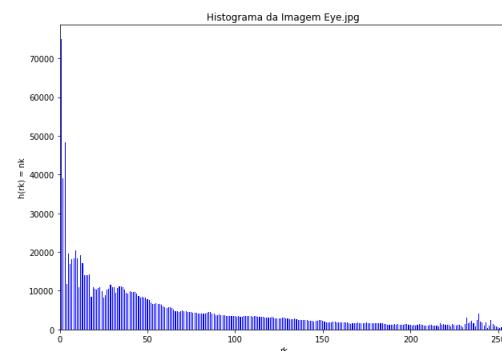


Fig. 2. Histograma da imagem eye.jpg

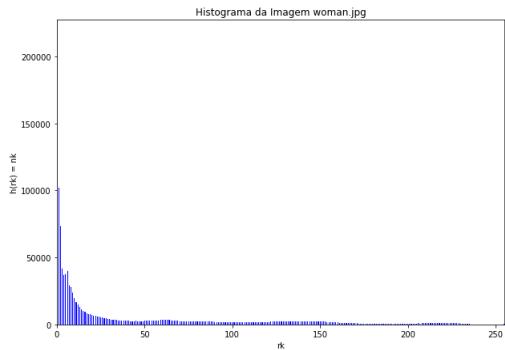


Fig. 3. Histograma da imagem woman.jpg

E para calcular os valores da função de transformação de equalização de histograma, é proposto a seguinte equação:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_k) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j,$$

$$k = 1, 2, 3, \dots, L-1, (1)$$

Calculado os valores de s_k para imagem eye.jpg e woman.jpg, temos os histogramas equalizados na figura 4 e 5, respectivamente:

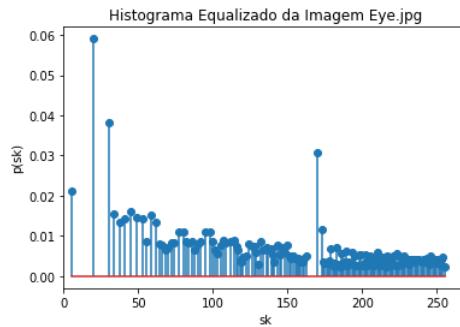


Fig. 4. Histograma Equalizado da imagem eye.jpg

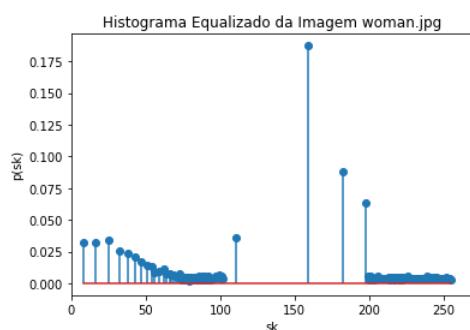


Fig. 5. Histograma Equalizado woman.jpg

Agora com os valores de s_k temos os valores dos pixels da imagem equalizada, agora apenas temos que definir a posição e construir uma nova imagem, seguindo a lógica descrito no Anexo I. Com isso temos as seguintes imagens com ajuste de intensidade:

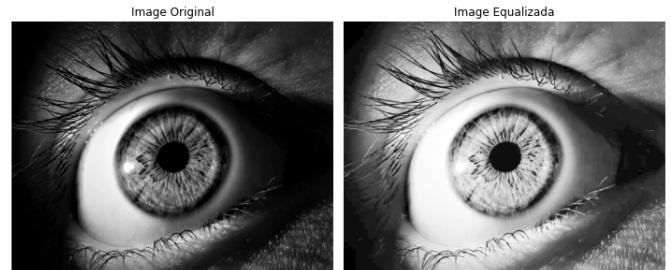


Fig. 6. Imagem equalizada eye.jpg



Fig. 7. Imagem equalizada woman.jpg

Podemos perceber que as imagens ganharam um brilho bem distribuído e recuperou algum detalhe que não estavam claros na imagem original. Na imagem 'woman.jpg' foi reconstruído o braço esquerdo e detalhes no cabelo ficaram evidentes. Portanto a Equalização de histograma se mostrou uma ferramenta muito poderosa para o ajuste de intensidade da imagem.

3. REALCE DE IMAGEM DE BAIXA RESOLUÇÃO

Para realizar o realce de baixa de resolução foi entregue as seguintes imagens como entrada para o algoritmo;



Fig. 8. Imagem dipper.jpg



Fig. 9. Imagem statue.jpg

Pode-se perceber que ambas imagens apresentam baixas resoluções devido as condições do ambiente, e alguns detalhes são escondidos pela falta de nitidez.

A solução encontrada para solução do problema apresentado nas imagens é realizar filtragem no domínio da frequência, utilizando o filtro homomórfico. Inicialmente é apresentado o espectro das imagem, para isso é necessário aplicar a transformada de Fourier, fazer o deslocamento e encontrar a magnitude, depois desses processos temos os seguintes espectros:

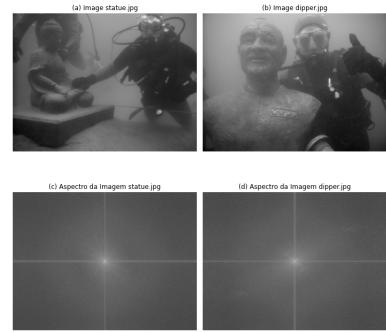


Fig. 10. Espectros das imagens de entrada

É possível perceber através dos espectros em frequência que as imagens apresentam baixa luminosidade.

Para aplicar a filtragem homomórfica temos os seguinte passos ilustrados abaixo:

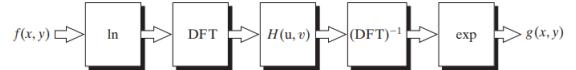


Fig. 11. Resumo dos passos na filtragem homomórfica

Com as imagens devidamente transformadas, temos agora que calcular o filtro homomórfico de acordo com as seguintes equações:

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2} \quad (2)$$

sendo D_0 é uma constante positiva, e $D(u, v)$ é a distância entre um ponto (u, v) no domínio da frequência e o centro do retângulo de frequência.

$$H(u, v) = (H - \gamma_L)[1 - e^{-c[D^2(u, v)/D_0^2]}] + \gamma_L \quad (3)$$

sendo c é a constante que controla a inclinação da função à medida que ela realiza a transição entre γ_L e γ_H .

Calculado o filtro para cada imagens temos os seguintes filtros no domínio da frequência:

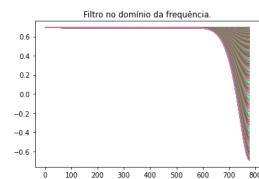


Fig. 12. Filtro no domínio da frequência para as imagens de entrada

Finalizado os filtros devemos aplica-lo a imagem, depois aplicar a transformada inversa de fourier de acordo com o

passo 3 descrita anterior. Com isso, temos a seguintes imagem processadas:



Fig. 13. Comparação da imagem dipper.jpg original a esquerda com a filtrada a direita



Fig. 14. Comparação da imagem statue.jpg original a esquerda com a filtrada a direita

Comparando os espectros:

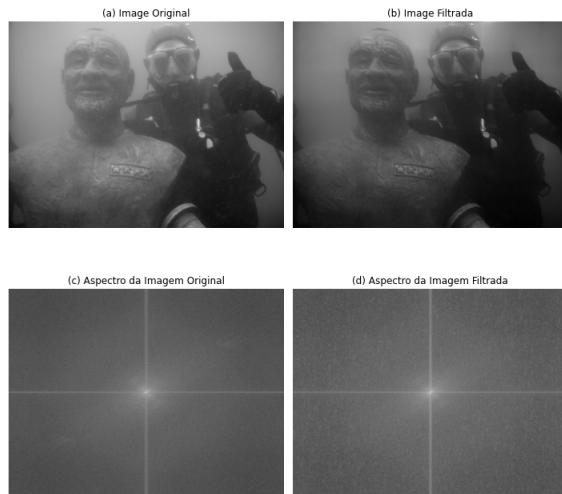


Fig. 15. Comparação dos espectro em frequência da imagem dipper.jpg original a esquerda com a filtrada a direita

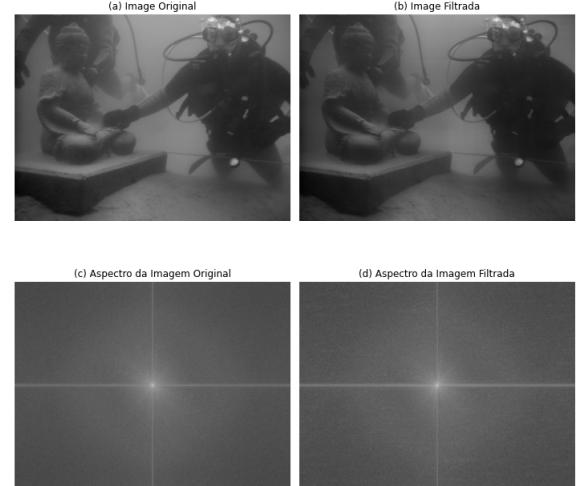


Fig. 16. Comparação dos espectro em frequência da imagem statue.jpg original a esquerda com a filtrada a direita

Após a filtragem da imagem é possível um ganho de nitidez e um realce da resolução da imagem.

4. FILTRAGEM DE RUÍDO

Um problema em imagens que não foram bem tratadas são ruídos que prejudica a visão e o entendimento das informações passada pela imagem, como as imagens utilizadas como entrada do nosso algoritmo mostradas abaixo:

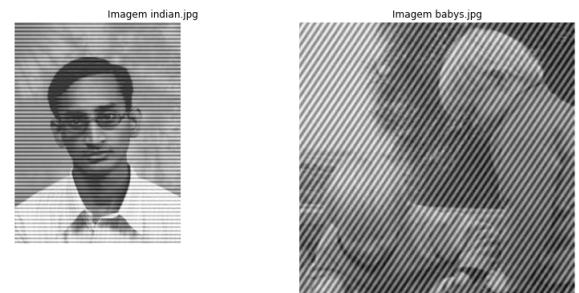


Fig. 17. A imagem a esquerda é indian.png e a direita é babys.png

É possível perceber o ruído em ambas imagens e para retirar os ruídos é necessário análise o espectro em frequência e aplicar um filtro notch.

Para aplicar o filtro notch e possuindo as imagens no domínio da frequência, deve-se com desenvolver os filtros de acordo com as seguintes equações:

$$D_k(u, v) = [(u - M/2 - u_k)^2 + (v - N/2 - v_k)^2]^{1/2} \quad (4)$$

e,

$$D_{-k}(u, v) = [(u - M/2 + u_k)^2 + (v - N/2 + v_k)^2]^{1/2} \quad (5)$$

sendo que (u_k, v_k) são os centros do filtro notch.

$$H_{NR}(u, v) = \prod_{k=1}^3 \left[\frac{1}{1 + [D_{0k}/D_k(u, v)]^{2n}} \frac{1}{1 + [D_{0k}/D_{-k}(u, v)]^{2n}} \right] \quad (6)$$

sendo que a constante D_{0k} é a mesma para cada par de notches mas pode ser diferente para diferentes pares.

Com os filtros calculas temos os espectros das imagens propostas e seu determinado filtro, considerando $D0 = 10$ e $n = 4$:

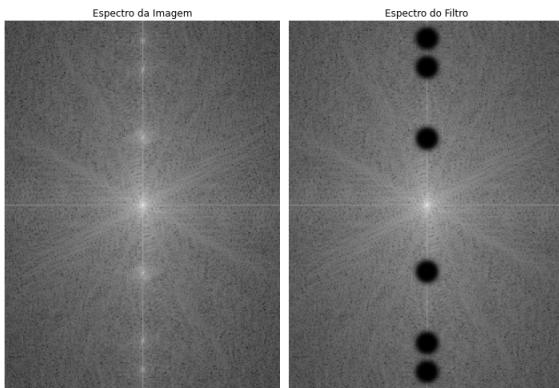


Fig. 18. Espectro e Filtro da imagem indian.png



Fig. 20. Indian.png processada.



Fig. 21. Babys.png processada.

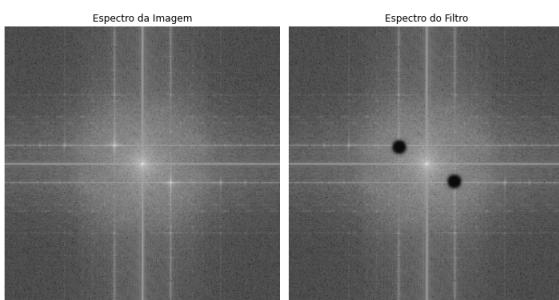


Fig. 19. Espectro e Filtro da imagem babys.png

Finalizado os filtros devemos aplicá-lo a imagem, depois aplicar a transformada inversa de fourier. Com isso, temos a seguintes imagem processadas:

Obteve êxito nas imagens filtradas, pois foi retirado os ruídos presente em cada imagem. Uma dificuldade encontrada foi escolher as imagens com ruídos para aplicar o filtro notch.

5. GRANULOMETRIA

Dentro da granulometria, tópico trabalho em morfolia matemática, é possível rotular componentes com o propósito de identificá-los e retirar informações para diversas aplicações. As imagens escolhidas para serem trabalhadas é moedas brasileiras e americanas, com o propósito de identificá-las pelo tamanho de cada componente na imagem. As imagens foram utilizadas como entrada do nosso algoritmo mostradas abaixo;



Fig. 22. Imagem que demonstra várias moedas do Brasil



Fig. 23. Imagem que demonstra várias moedas dos Estados Unidos, esta imagem, foi encontrada no Google Imagens

A imagem da figura 23 possui uma marca da empresa responsável pela foto, isso pode ser um desafio para esta questão trabalhar com esse contratempo.

Para a rotulação desses componentes foi utilizado a função **connectedComponents** que conta a quantidade de componentes e rotula os componentes. Porém como previsto a imagem quando transformada para np.uint8 apresentou algumas falhas, com isso foi utilizado a **dilate** do openCV que permite dilatar com componentes da imagem, feito isso as falhas se conectaram e a função **connectedComponents** conseguiu rotular os componentes apresentados nas imagens. Os passos para estão ilustrados nas imagens abaixo:

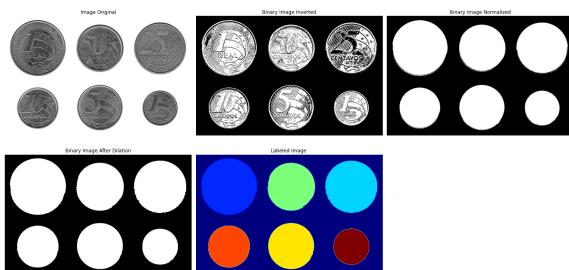


Fig. 24. Passos do algoritmo para a rotulação dos componentes da imagem que demonstra várias moedas do Brasil

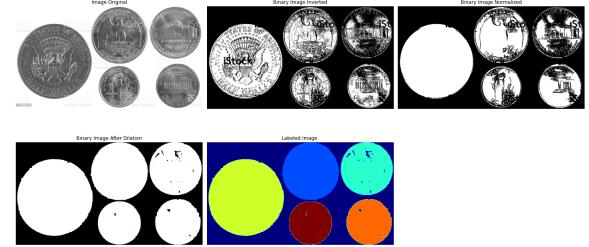


Fig. 25. Passos do algoritmo para a rotulação dos componentes da imagem que demonstra várias moedas dos Estados Unidos

Rotulado os componentes, é possível utilizando a função **Counter** da biblioteca collections, onde é calculado o número de pixels que possuem diferentes intensidade após a rotulação da imagem, ou seja, conta a quantos pixels são ocupados pelas determinadas moedas. Com isso e fazendo um cálculo de porcentagem, temos valor de quanto cada moeda ocupa na imagem e quais são as maiores e menores.

6. APLICAÇÃO DE PSEUDOCOR POR FATIAMENTO DE INTENSIDADE

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

7. REFERENCES

- [1]. GONZALEZ, Rafael C., WOODS, Richard E. **Processamento Digital de Imagens**. 3^a Edição. PEARSON.

ANEXO I

```
# Trecho do código intensity_adjustment_hist.py
# Matheus Gois Vieira - 170019136

# Vetor de zeros com o tamanho da imagem original
new_img = np.zeros(img.shape);
w, h = new_img.shape; # Dimensões da imagem original
for i in range(w):
    for j in range(h):
        new_img[i][j] = s[img[i][j]] # Recriando a nova imagem
```
