

PROCESSAMENTO DIGITAL DE IMAGENS - PROJETO INDIVIDUAL

Bruna Medeiros da Silva - 16/0048711

UnB - FGA

1. QUESTÃO 1 - AJUSTE DE INTENSIDADE

Para a realização do ajuste de intensidade, foi utilizada a técnica de **equalização de histograma**, que possibilita a aplicação de diversas formas diferentes para equilíbrio entre os pixels, mas também não possui muitos requisitos para funcionar, podendo ser aplicado em diversas imagens e casos diferentes.

1.1. Metodologia

O processo de especificação de histograma pode ser descrito em alguns passos:

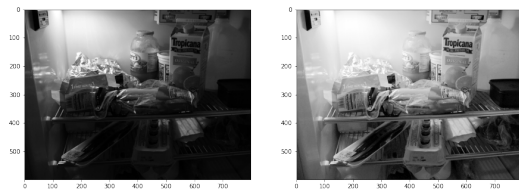
1. Encontrar o histograma da imagem, que mostra graficamente a distribuição de níveis de cinza da imagem. Os valores obtidos podem ser chamados de $p_r(r)$;
2. Encontrar os valores s_k da Função de Distribuição Acumulada (CDF - *Cumulative Distribution Function*) da imagem. Para gerar a CDF, foi utilizada a equação 1, que calcula a probabilidade de os pixels possuírem valores menores ou iguais a determinado valor dentro do intervalo de níveis de cinza da imagem. L é igual ao número de níveis de cinza disponíveis ($2^{n_{bits}}$), MN é a quantidade de pixels que a imagem possui e k é o nível de cinza para o qual estamos fazendo o cálculo e $p_r(j)$ é o valor de j no histograma;
3. Encontrar os valores z_k da CDF da função especificada. Gerar a CDF a partir de uma PDF (*Probability Density Function*) é um processo ainda mais simples, visto que nós já temos a probabilidade de cada valor, precisando apenas somá-los aos anteriores para obter a CDF equivalente (Equação 2). Onde $p_z(j)$ é a probabilidade de um pixel ter o valor j . Nesse caso, a PDF padrão utilizada na função implementada foi uma pdf que distribui igualmente os pixels em toda a escala de cinza disponível (geralmente, em *uint8*, se trata de 0 a 255);
4. Fazer a equivalência entre os valores de s para z . Para encontrar o valor $z = G^{-1}(s)$, precisamos encontrar o menor z para o qual $G(z)$ seja o mais próximo possível de s_k . Fazendo isso para todos os valores de s , teremos

o valor equivalente em z para todos os níveis de cinza presentes na imagem;

5. utilizar a nova CDF obtida para gerar uma nova imagem equalizada com os valores dos pixels s_k convertidos para z_k .
6. Depois disso, apenas aplicamos essa transformação nos pixels da imagem e, caso necessário, geramos os novos gráficos de histograma, PDF e/ou CDF para efeitos de comparação.

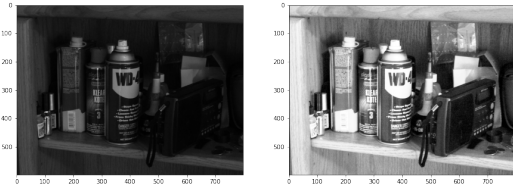
$$s_k = \frac{(L-1)}{MN} \sum_{j=0}^k p_r(j) \quad (1)$$

$$G(z) = \sum_{j=0}^z p_z(j) \quad (2)$$



Resultado questão 1 - Imagem 1

se necessário que o usuário defina apenas 2 parâmetros: a imagem e o valor D_0 , que equivale ao raio do filtro a ser aplicado na frequência da imagem. para a transformada para o domínio da frequência, utilizou-se a fft implementada na biblioteca *numpy* do Python.



Resultado questão 2 - Imagem 1

2. QUESTÃO 2 - REALCE DE IMAGEM DE BAIXA RESOLUÇÃO

Para a questão 2 foram testados 2 métodos para realce de imagens: aplicação do laplaciano e do filtro butterworth. Isso porque observou-se que o Laplaciano, em muitos casos, gera resultados bem mais significativos na obtenção de realce, resultando em uma imagem visualmente com muito mais nitidez e transmitindo uma impressão de maior qualidade para quem vê.

Em ambas as aplicações, a imagem foi percorrida e modificada por meio de um laço duplo, iterando em cada uma de suas colunas e linhas.

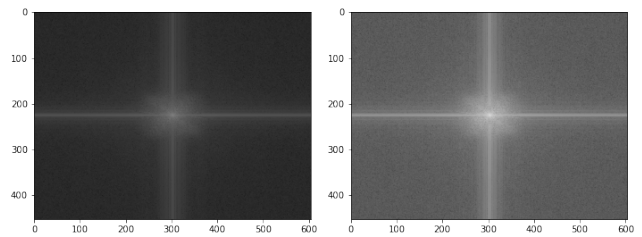
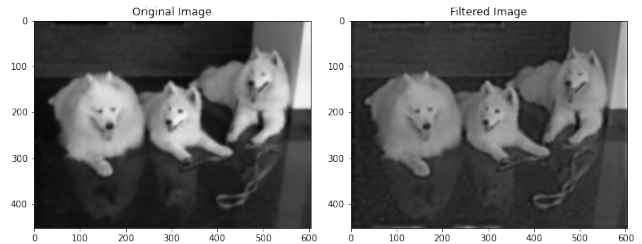
2.1. Butterworth

O filtro butterworth utilizado foi um filtro passa-altas, com o intuito de destacar apenas componentes de alta frequência da imagem (como bordas e detalhes) e depois somá-los à imagem original, gerando uma imagem com detalhes mais destacados. Outra opção também seria utilizar o filtro gaussiano, mas observou-se que os resultados com o mesmo não foram muito significativos, optando-se assim pelo butterworth 3.

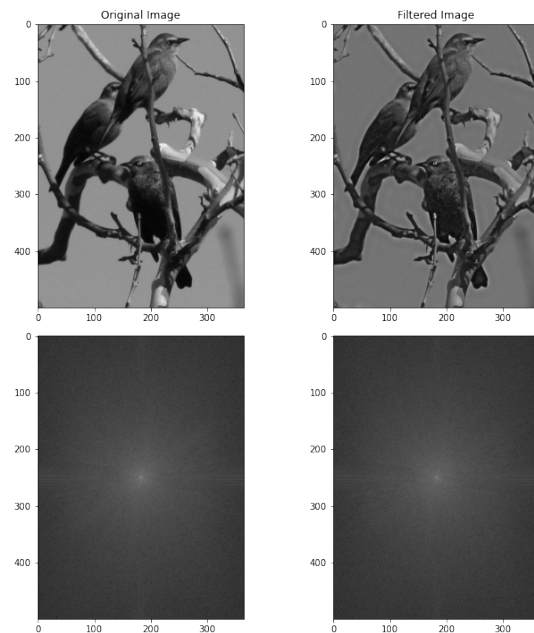
$$D = ((u - (P/2))^2 + (v - (Q/2))^2)^{1/2} \quad (3)$$

$$H[u][v] = \frac{1}{1 + (D_0/D)^{2n}} \quad (4)$$

Para obter a imagem final, utilizou-se da fórmula entregue no livro texto da disciplina. Com isso, dentro da função, faz-



Resultado questão 2 - Imagem 1 com Butterworth



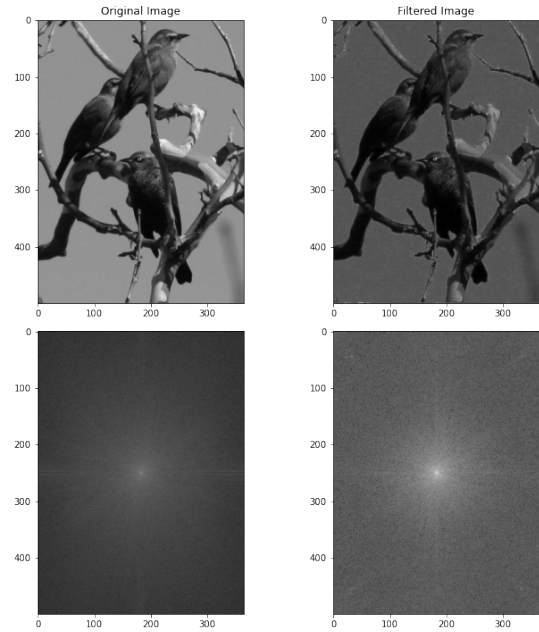
Resultado questão 2 - Imagem 2 com Butterworth

2.2. Laplaciano

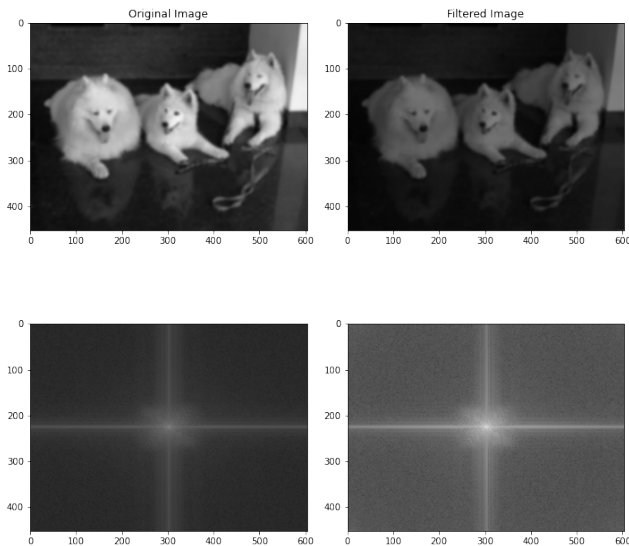
Diferentemente do filtro Butterworth, no Laplaciano o usuário não define os valores do filtro. Todos os valores estão definidos dentro da equação e dependem apenas de valores extraídos de forma subentendida durante o envio da imagem como parâmetro da transformação (como tamanho/*shape* da imagem, por exemplo).

$$D = ((u - (P/2))^2 + (v - (Q/2))^2)^{1/2} \quad (5)$$

$$H[u][v] = -4(\pi^2)D^2 \quad (6)$$



Resultado questão 2 - Imagem 2 com Laplaciano

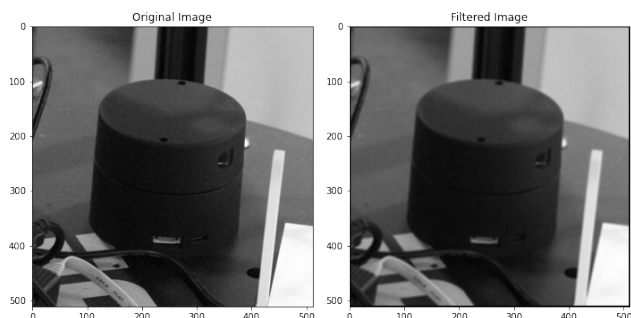


Resultado questão 2 - Imagem 1 com Laplaciano

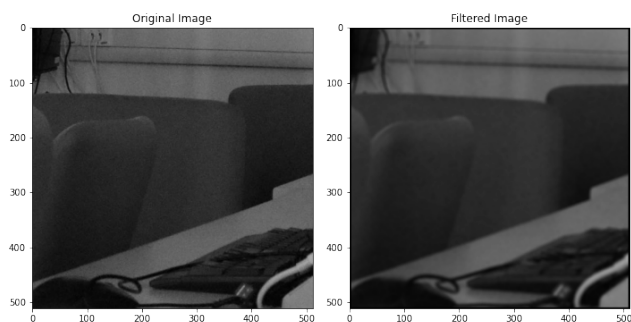
3. QUESTÃO 3 - FILTRAGEM DE RUÍDO

Para a filtragem de alguns ruídos, foi utilizado um **filtro de média retangular**. Com essa aplicação, o usuário tem a liberdade de definir o tamanho da janela utilizada para fazer a média, o que definirá o quanto a imagem será suavizada. O uso de janela excessivamente grandes para o tamanho e conteúdo da imagem fazem com que a imagem passe a ficar "borrada" e comece a perder informações e detalhes. Esse efeito de "borramento" possui uma relação diretamente proporcional ao tamanho da janela utilizada na média.

Para evitar extrapolar as fronteiras da imagem durante o processo, realiza-se a filtragem apenas em pixels nos quais pode-se colocar toda a janela sem que nenhum de seus pixels fique "fora" das bordas da imagem original.



Resultado questão 3 - Imagem 1



Resultado questão 3 - Imagem 2

4. QUESTÃO 4 - GRANULOMETRIA

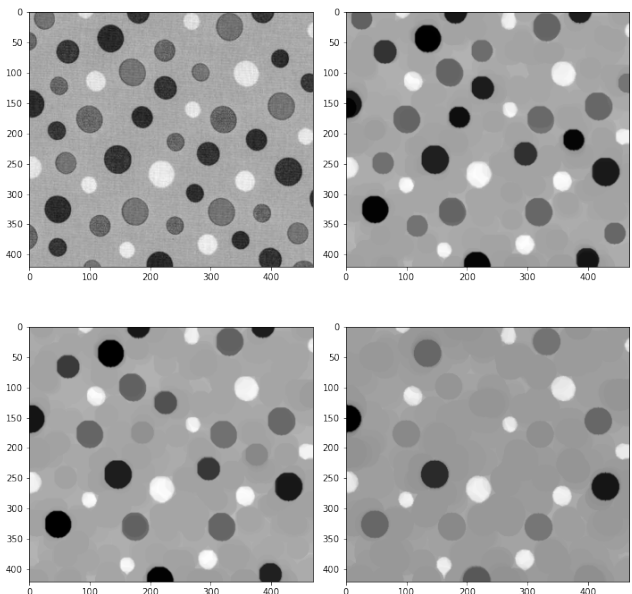
Para separar determinados componentes de uma imagem e possibilitar a remoção de alguns outros, de acordo com seu tamanho e formato, pode-se realizar a técnica de **segmentação por textura**, aplicando assim o estudo da granulometria dentro do tratamento de imagens.

Dentro do projeto, foram utilizadas imagens com "bolhas" e componentes circulares de diversos tamanhos. Para a identificação desses componentes, realiza-se a diferença entre a soma de todos os pixels da imagem após o processo de abertura com diferentes tamanhos de kernel. Realizando essa comparação e plotando seus resultados, torna-se possível identificar o tamanho dessas componentes, que equivale aos picos do gráfico.

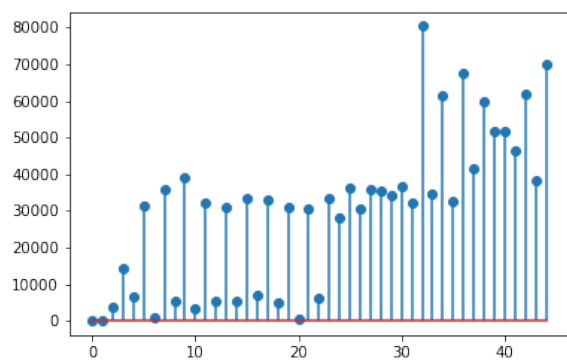
A técnica foi utilizada para separar círculos de diferentes tamanhos, podendo gerar imagens diferentes com cada um deles, se necessário.

É importante ressaltar que antes da aplicação também é possível e recomendado que seja realizado o processo de suavização da imagem, que consiste em realizar a abertura e o fechamento da imagem, respectivamente.

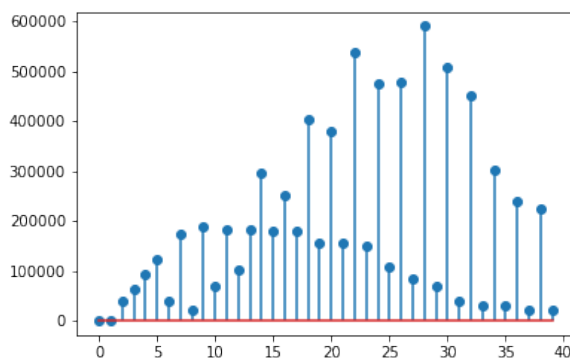
Essa técnica é estabelecida para casos em que esses componentes são mais claros que os valores de fundo. Na tentativa de adaptá-los, é possível passar parâmetros para a função de suavização que realizam o processo de forma inversa (primeiro o fechamento e depois a abertura da imagem).



Resultado questão 4 - Imagem 1

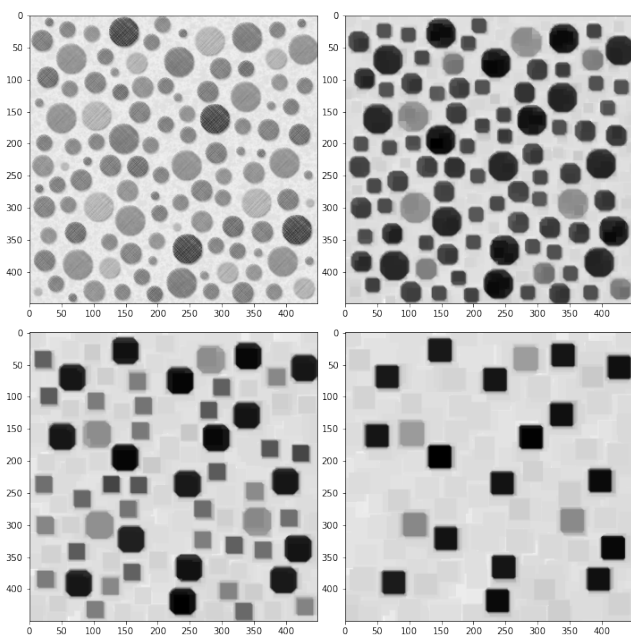


Resultado questão 4 - Gráfico da Imagem 1



Resultado questão 4 - Gráfico da Imagem 2

Obs.: Os códigos-fonte estão em anexo, todos compactados em um arquivo .zip.



Resultado questão 4 - Imagem 2