

Introdução ao Processamento Digital de Imagem

Trabalho 02

Filipe Alves Sampaio
RA 249092

I. Introdução

Este trabalho tem como objetivo explicar o funcionamento, implementação e resultados do trabalho 02, desenvolvido na disciplina MO443. O trabalho tem como objetivo aplicar a transformada rápida de Fourier (do inglês, Fast Fourier Transform - FFT) em imagens digitais, explorando o processamento de imagens no domínio de frequência.

A seção II inclui exemplos de execução do programa, bem como os parâmetros necessários para sua execução e uma descrição, tanto dos parâmetros de entrada como das saídas geradas pelo script.

II. Execução

Dentro do zip referente ao trabalho, encontra-se o arquivo: *fft.py*. Esse arquivo, escrito em Python, foi desenvolvido para realizar as tarefas especificadas na descrição do trabalho. Nessa seção serão apresentados:

A. Dependências necessárias para a execução do trabalho.

É opcional, mas aconselha-se a utilizar um ambiente de desenvolvimento para isolar as bibliotecas utilizadas nesse trabalho.

Assim, como orientado pelo professor da disciplina, utilize o Anaconda ou semelhante e crie um ambiente separado.

Após isso, instale as seguintes dependências:

- **Python:** Foi utilizado a versão 3.12.2. Essa versão é opcional, dado que os scripts desenvolvidos são simples e não utilizam algoritmos sofisticados que não existam em versões anteriores do Python.
- **Numpy:** Usada para manipular vetores e otimizar o encode da esteganografia.
- **Sys:** Usada para manipular argumentos passados por linha de comando.
- **Opencv:** Biblioteca usada para manipular imagens.
- **Matplotlib:** Biblioteca usada para gerar histogramas das imagens.

B. Funcionalidades implementadas e suas ações sobre as tarefas exigidas no trabalho.

Essa seção tem como finalidade explicar brevemente o funcionamento do que foi implementado. Uma análise mais profunda dos algoritmos usados será descrita na seção III e uma análise dos resultados será discutida na seção IV.

Como descrito pelo trabalho, o objetivo é aplicar a transformada rápida de Fourier em imagens digitais para explorar o domínio da frequência em imagens. Para isso, é requisitada a aplicação de 4 filtros na imagem: passa-baixa, passa-alta, passa-faixa e rejeita faixa em imagens monocromáticas por meio do espectro de Fourier. Com a aplicação do filtro, é requisitado que um limiar no espectro seja aplicado para executar uma compressão na imagem,

removendo os valores menores que esse limiar.

C. Parâmetros.

Neste tópico, é tratado os parâmetros possíveis para executar o *script* implementado. Seguindo o que foi informado pela descrição do trabalho, foi utilizado os seguintes parâmetros:

- A. **<imagem_entrada.png>**: é o caminho até a imagem a ser usada no experimento.
- B. **threshold**: é o limiar de compressão usado na imagem.

Além desses parâmetros, é informado na seguinte lista, os arquivos que serão utilizados para execução do programa:

- A. **fft.py**: arquivo principal que irá executar o experimento de filtragem no domínio da frequência.

D. Exemplos de execução.

No diretório do programa há uma pasta, chamada: **/imgs**. Essa pasta contém as imagens de teste utilizadas na implementação do trabalho.

Ao finalizar a execução do *script* **fft.py**, serão gerados imagens (resultantes do processo de filtragem e compressão, além dos filtros utilizados). Durante a execução do programa, serão plotados alguns resultados do experimento.

O programa pode ser executado da seguinte forma:

```
python fft.py <imagem_entrada.png>
threshold
```

Para exemplificar, suponha que queira aplicar um filtro na imagem **/imgs/baboon.png**, com um limiar de 10000. Assim, basta aplicar o seguinte comando no terminal (dentro do diretório do projeto):

```
python fft.py /imgs/baboon.png 10000
```

Uma observação sobre o limiar é que em alguns casos o valor necessário para conseguir ver algum efeito de perda deverá ser bem alto, como 10000 ou

1000000. Foi cogitado utilizar porcentagem, porém não teria um controle fino de limiar exato, por isso optou-se em deixar um valor inteiro.

Ao executar esse comando acima, serão plotados as imagens: original, convertida no espectro de frequência, filtros, aplicações dos filtros, imagens filtradas com compressão e histogramas antes e depois da compressão.

III. Implementação

Nessa seção serão descritos os passos e motivações que levaram ao estado final da implementação do programa. Juntamente, será feita uma descrição do funcionamento de cada etapa.

A. Código do FFT

Abrindo o arquivo **fft.py**, será possível visualizar todas as funções implementadas para realizar a filtragem da imagem de entrada no espectro de Fourier.

Neste trabalho, seguiu-se uma ideia simples para aplicação dos filtros no espectro de Fourier. Assim, utilizou-se funções prontas da biblioteca **Numpy** para isso.

Os passos da implementação foram os requisitados pelo trabalho, conforme a lista abaixo:

- Abrir uma imagem de entrada convertida para escala de cinza;
- Aplicar a transformada rápida de Fourier;
- Centralizar o espectro de frequência;
- Criar os núcleos (máscaras) para os diferentes filtros com as mesmas dimensões das imagens, tal que os raios dos círculos definem as frequências que serão atenuadas/preservadas;

- Aplicar cada filtro por meio da multiplicação entre o espectro de frequência e a máscara do filtro;
- Aplicar a transformada inversa de Fourier para converter o espectro de frequência filtrada de volta para o domínio espacial, gerando a imagem filtrada;
- Visualizar e analisar os resultados.

Na função **etapa_1()** é feita a leitura da imagem e conversão para escala de cinza. Depois essa imagem é passada para a função **etapa_2()**, que aplica a FFT na imagem monocromática, aplicando logo na sequência a função **etapa_3()**, responsável por centralizar o espectro da frequência através da função **ffshift()** do **Numpy**.

Com isso, é chamado a função **etapa_4()**, responsável por criar as máscaras para os filtros requisitados (passa-baixa, passa-alta, passa-faixa e rejeita-faixa), com as mesmas dimensões da imagem de entrada e com raios que definem o quanto será atenuada ou preservada a imagem após a aplicação dos filtros. Aqui foi definido tamanhos fixos dos raios, seguindo o tamanho aproximado do exemplo dado na descrição do trabalho.

Logo na sequência, é chamado a função **etapa_5()**, responsável por aplicar a multiplicação de cada filtro com a imagem no no espectro de frequência. A seguir, é chamado a função **etapa_6()** que aplicará a transformada inversa de Fourier para converter o espectro de frequência da imagem de volta para o domínio espacial.

Depois disso, é repetido a chamada da função **etapa_6()**, mas aplicando a função **compressao_por_remocao_de_coeficientes()**, onde nessa função é removido as magnitudes menores que o limiar

informado no parâmetro do programa. Por fim, a função **etapa_7()** realiza os *plots* dos histogramas (antes e depois do processo de compressão).

B. Validação e apresentação dos dados

Ao final da execução do programa, no diretório do trabalho, serão gerados as imagens resultantes do processo, que são a imagem original em escala de cinza, a imagem no espectro de frequência, os filtros, às imagens com filtros, às imagens com filtros com compressão e os histogramas. Durante a execução do programa, essas mesmas imagens serão plotadas na tela.

IV. Resultados

Nessa seção é apresentado alguns experimentos realizados durante a implementação.

O primeiro experimento é executando o comando:

python fft.py imgs/baboon.png 10000

Observe nas Figuras 1 e 2, respectivamente, as imagens no domínio espacial (em tons de cinza) e em espectro de frequência.

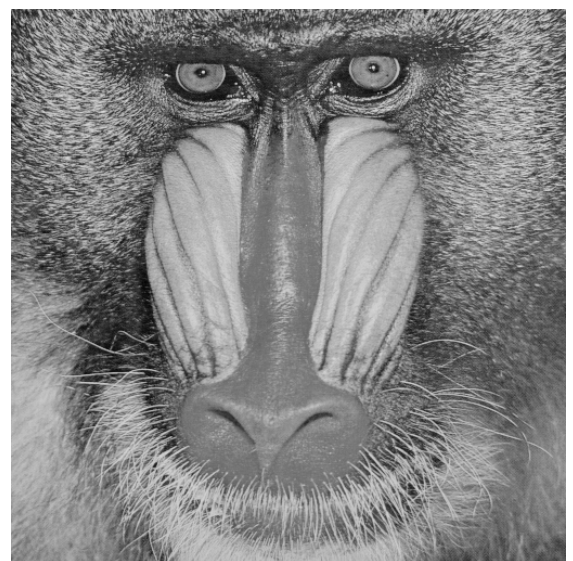


Figura 1. Imagem Original em tons de cinza.

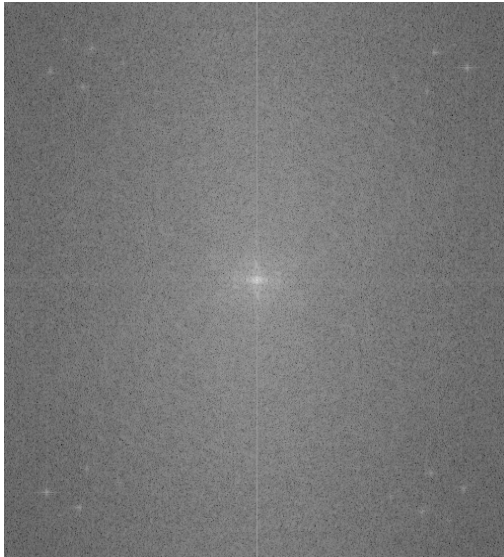


Figura 2. Imagem após FFT espectro de frequência centralizado.

A seguir, nas Figuras 3 a 6, é apresentado os respectivos filtros passa-baixa, passa-alta, passa-faixa e rejeita-faixa, construídos com raio 60, utilizado nos filtros passa-baixa e passa-alta e raio de banda 10 e 60 para os filtros passa-faixa e rejeita-faixa.

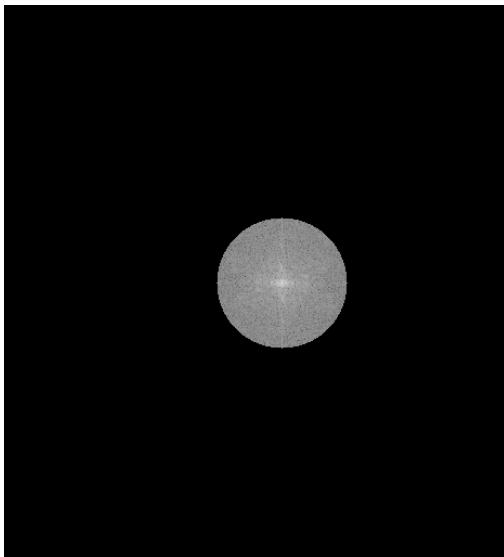


Figura 3. Filtro passa-baixa.

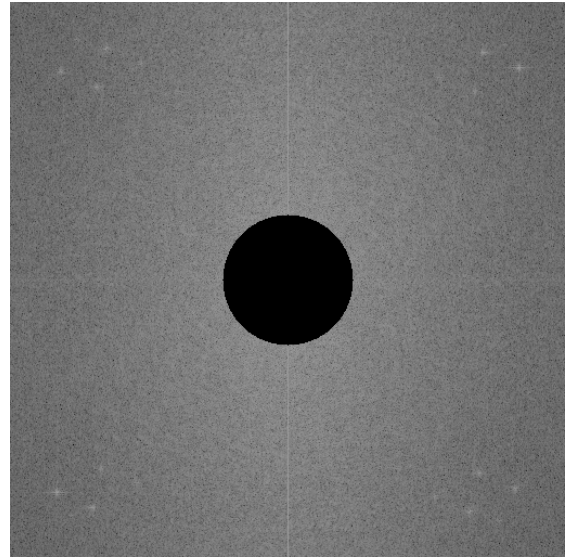


Figura 4. Filtro passa-alta.

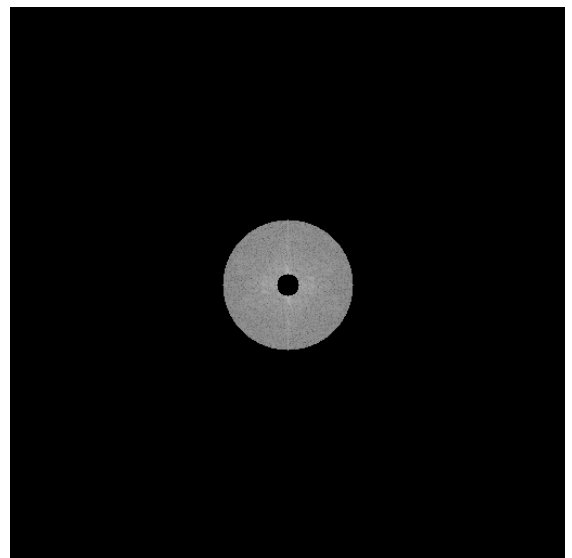


Figura 5. Filtro passa-faixa.

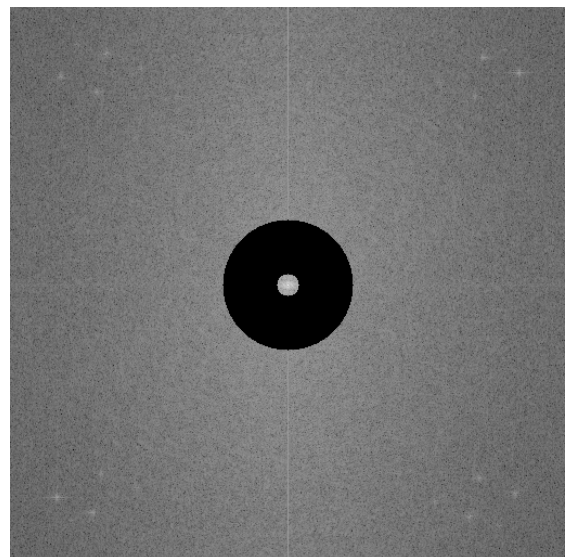


Figura 6. Filtro rejeita-faixa.

Com as máscaras em mãos, é aplicado a multiplicação do filtro com o espectro de frequência da imagem, obtendo os seguintes resultados das Figuras 7 a 10.



Figura 7. Figura com filtro passa-baixa sem compressão.

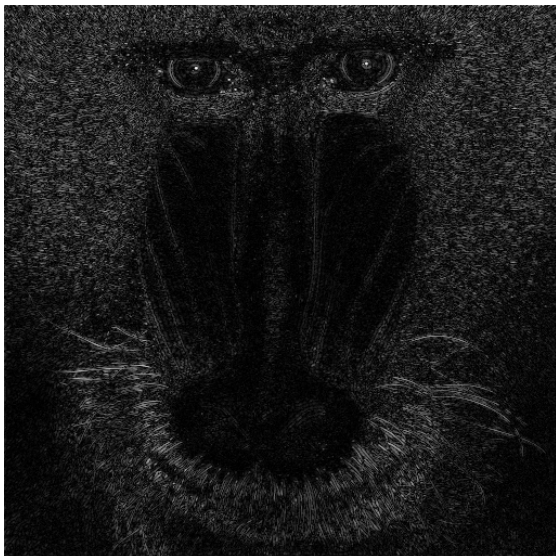


Figura 8. Figura com filtro passa-alta sem compressão.



Figura 9. Figura com filtro passa-faixa sem compressão.

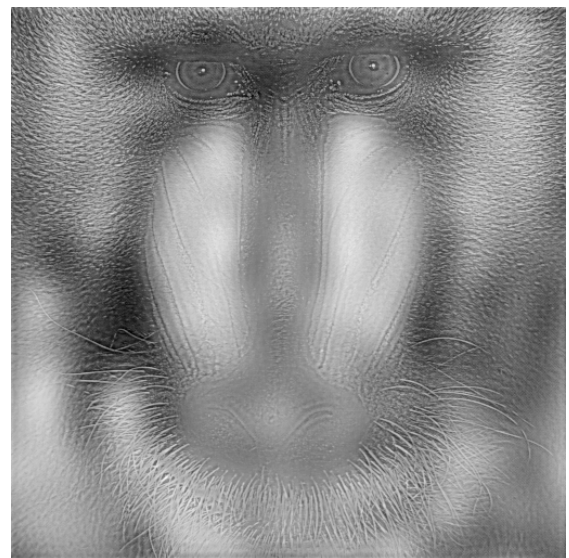


Figura 10. Figura com filtro rejeita-faixa sem compressão.

Como o limiar utilizado foi de 10000, não é possível visualizar degradação em nenhuma imagem, conforme mostrado nas Figuras 11 a 14.



Figura 11. Figura com filtro passa-baixa com compressão.

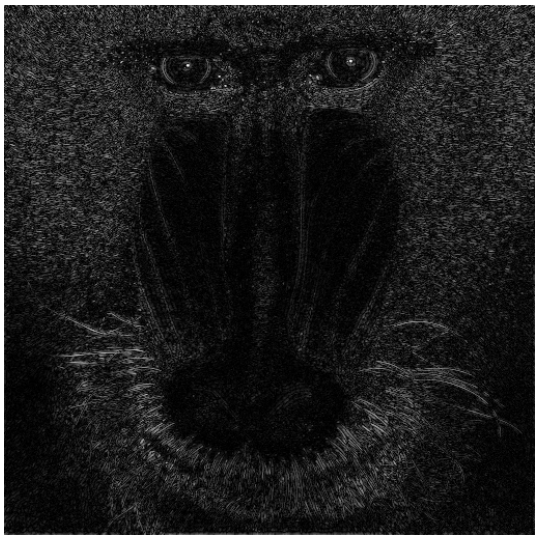


Figura 12. Figura com filtro passa-alta com compressão.



Figura 13. Figura com filtro passa-faixa com compressão.

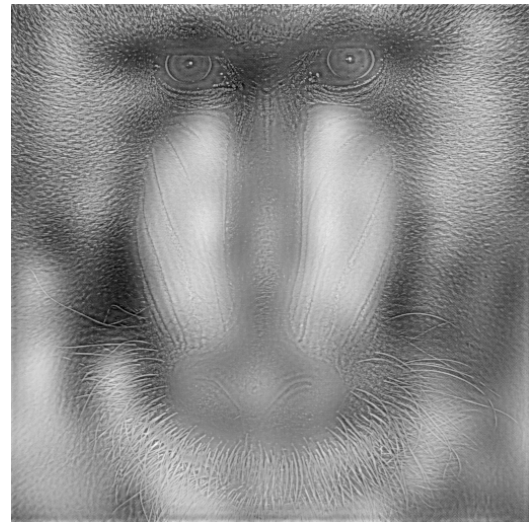


Figura 14. Figura com filtro rejeita-faixa com compressão.

Também nos histogramas não é possível visualizar nenhuma degradação por parte da compressão, conforme mostrado nas Figuras 15 a 23.

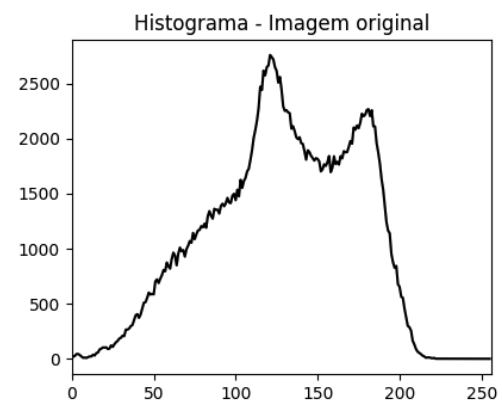


Figura 15. Histograma da imagem original em tons de cinza.

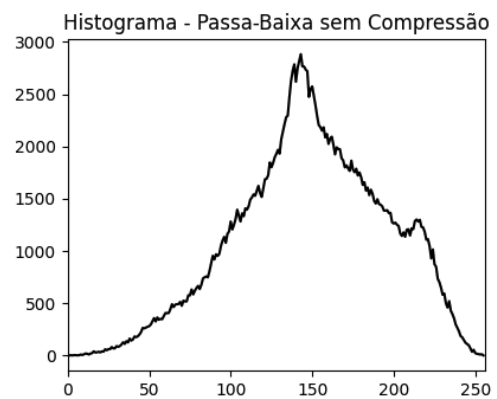


Figura 16. Histograma da imagem com filtro passa-baixa.

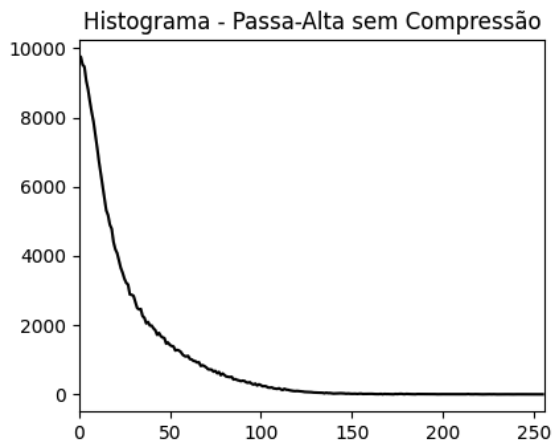


Figura 17. Histograma da imagem com filtro passa-alta.

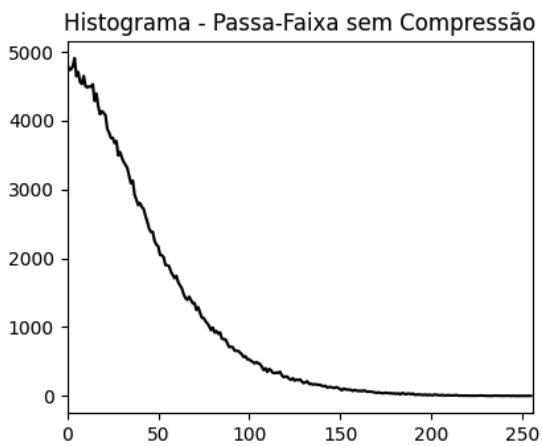


Figura 18. Histograma da imagem com filtro passa-faixa.

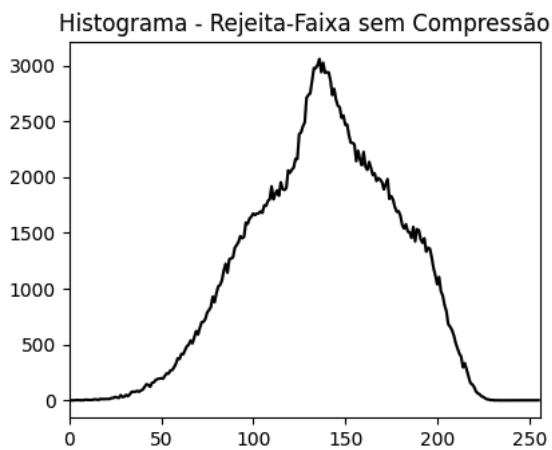


Figura 19. Histograma da imagem com filtro rejeita-faixa.

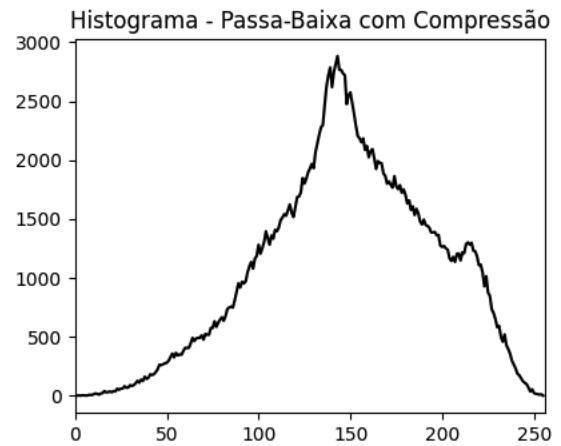


Figura 20. Histograma da imagem com filtro passa-baixa e com compressão.

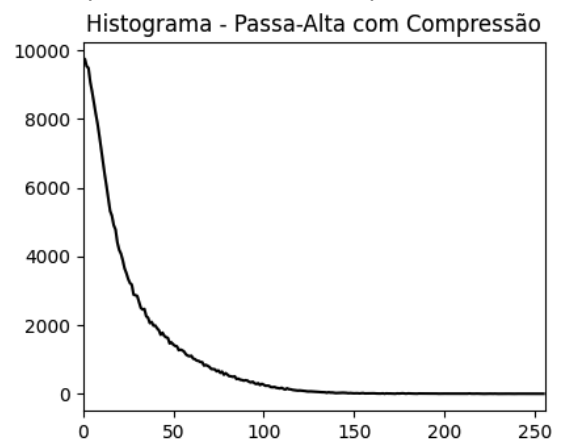


Figura 21. Histograma da imagem com filtro passa-alta e com compressão.

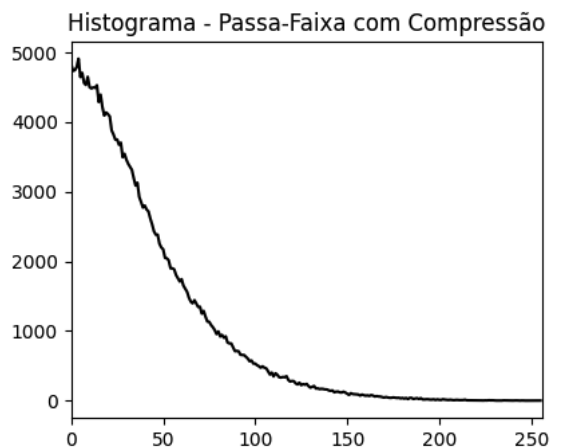


Figura 22. Histograma da imagem com filtro passa-faixa e com compressão.

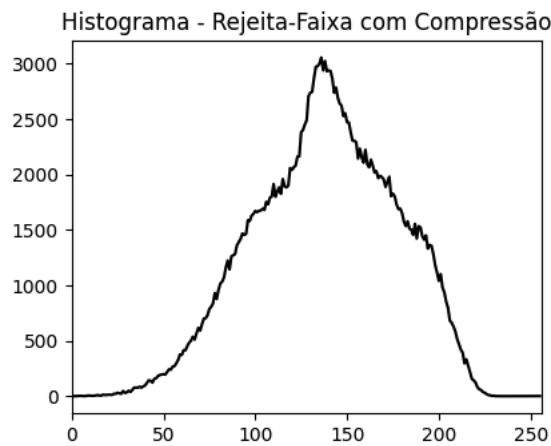


Figura 23. Histograma da imagem com filtro rejeita-faixa e com compressão.

Porém, ao adicionar um limiar de 5000000, começamos a ver degradação nas imagens com filtros, principalmente as com altas frequências, onde com esse limiar informado, uma parte da magnitude das frequências altas são afetadas.

Esse efeito é mais visível quando se aplica filtros em faixas altas de frequência, como filtro passa-alta, passa-faixa e rejeita-faixa. Veja nas Figuras 23 a 27.

Vemos também efeito na imagem com filtro passa-baixa uma leve degradação, porém no histograma isso não é tão visível se comparado com os histogramas de passa-alta e rejeita-faixa.



Figura 24. Imagem com filtro passa-baixa e com compressão.

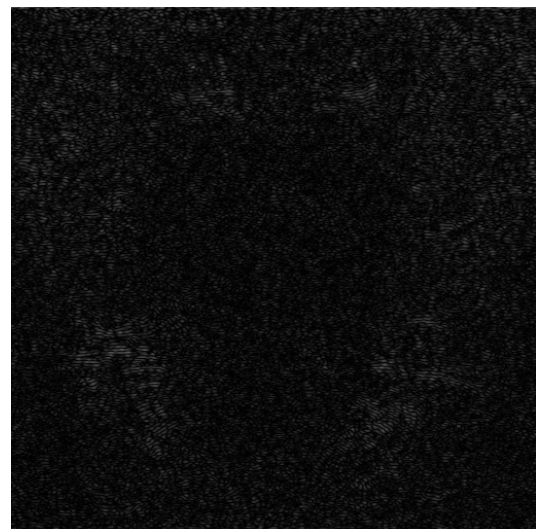


Figura 25. Imagem com filtro passa-alta e com compressão.



Figura 26. Imagem com filtro passa-faixa e com compressão.

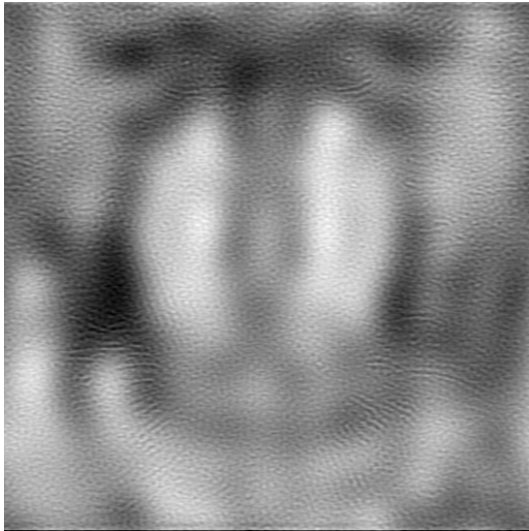


Figura 27. Imagem com filtro rejeita-faixa e com compressão.

Com o limiar de 5000000 é possível ver também nos histogramas de antes e depois da compressão, os efeitos da degradação na imagem, embora ainda sutil, Veja, conforme as Figuras 28 a 35.

Vemos nos histogramas de passa-baixa uma alteração de suavização nos picos, enquanto no rejeita-faixa, acentuação de linhas nos picos. Isso sugere que, no passa-baixa, foi eliminado uma pequena parte de frequências altas remanescentes, fazendo com que o gráfico ficasse com menos protuberâncias nos picos altos. Já no rejeita-faixa, como há uma faixa rejeitada, o limiar acaba interferindo na faixa rejeitada, trazendo para dentro da faixa pontos que eram ignorados, o que acaba mostrando mais pontas nos picos altos, conforme o gráfico mostra.

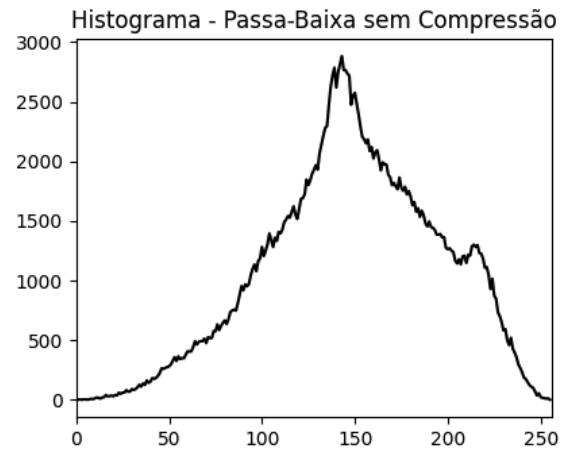


Figura 28. Histograma com filtro passa-baixa e sem compressão.

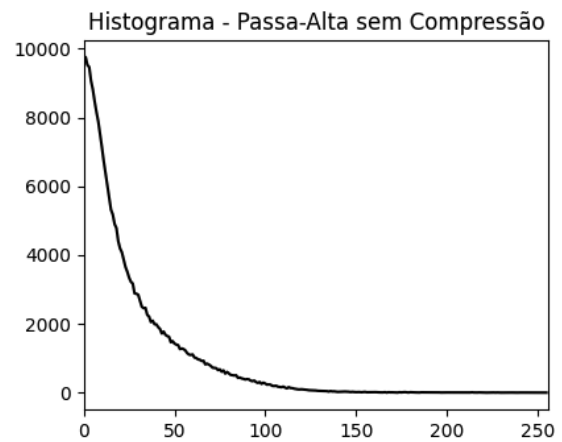


Figura 29. Histograma com filtro passa-alta e sem compressão.

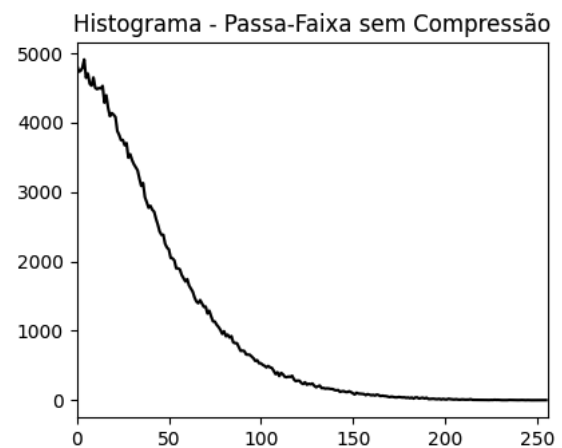


Figura 30. Histograma com filtro passa-faixa e sem compressão.

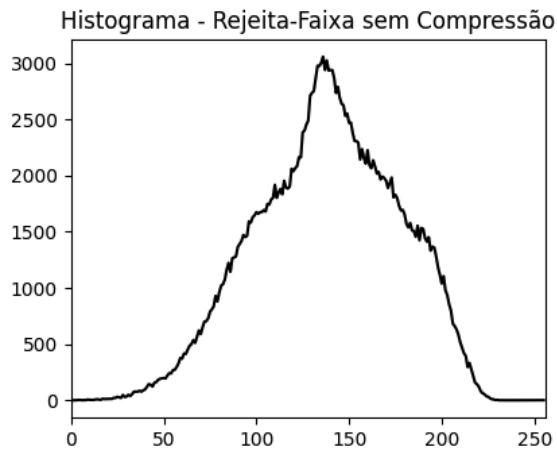


Figura 31. Histograma com filtro passa-alta e sem compressão.

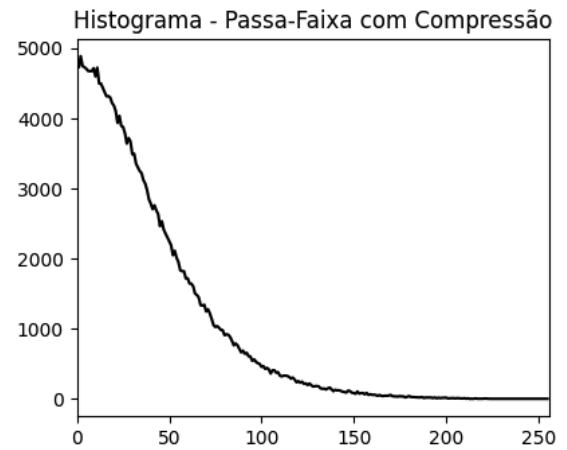


Figura 34. Histograma com filtro passa-faixa e com compressão.

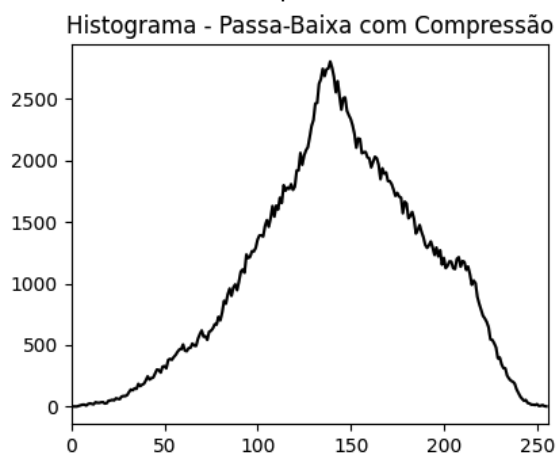


Figura 32. Histograma com filtro passa-baixa e com compressão.

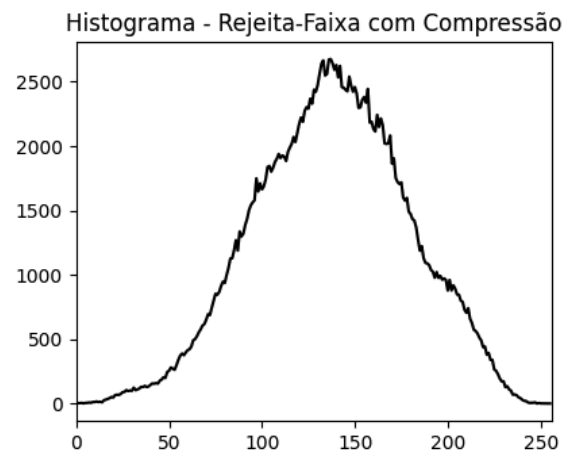


Figura 35. Histograma com filtro rejeita-faixa e com compressão.

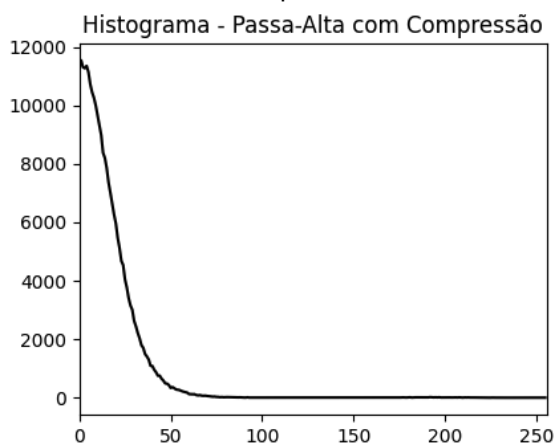


Figura 33. Histograma com filtro passa-alta e com compressão.

Para um limiar bem alto nessa imagem, como exemplo 50000000, é possível ver uma degradação considerável em todos os filtros aplicados. Veja nas Figuras 36 a 39 o efeito de degradação na imagem com esse limiar aplicado.



Figura 36. Imagem com filtro passa-baixa e com compressão.



Figura 37. Imagem com filtro passa-alta e com compressão.

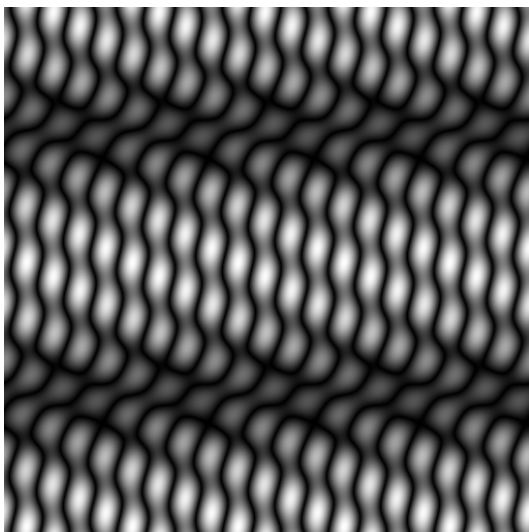


Figura 38. Imagem com filtro passa-faixa e com compressão.



Figura 39. Imagem com filtro rejeita-faixa e com compressão.

Em especial, no filtro passa-alta, a imagem ficou totalmente preto devido a remoção total da magnitude que resultava nos contornos e bordas.

A seguir, nas Figuras 40 a 43, vemos os histogramas resultantes. Percebe-se com mais clareza, alterações nos pixels com um limiar bem alto, em comparação com os outros experimentos.

Devido ao alto limiar, os picos dos histogramas são bastante afetados de praticamente todas as imagens. O filtro passa-alta foi totalmente linearizado, conforme a Figura 41, onde temos apenas intensidade 0, resultando uma imagem toda preta.

Observa-se também bastante ruído nos picos em comparação com os outros histogramas.

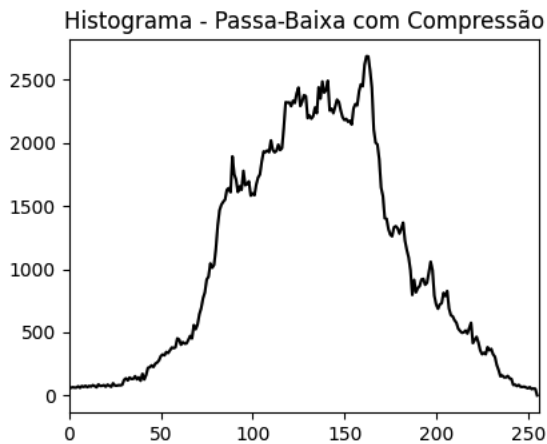


Figura 40. Histograma com filtro passa-baixa e com compressão.

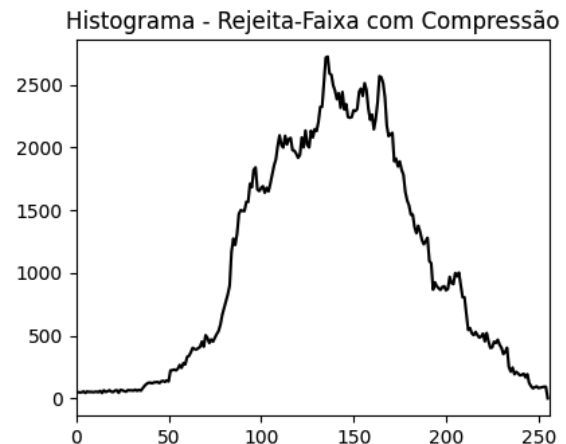


Figura 43. Histograma com filtro rejeita-faixa e com compressão.

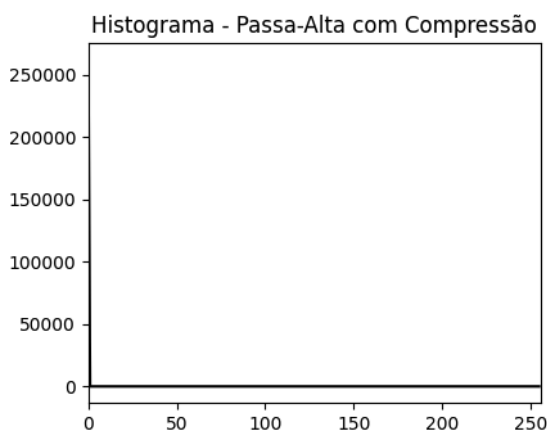


Figura 41. Histograma com filtro passa-alta e com compressão.

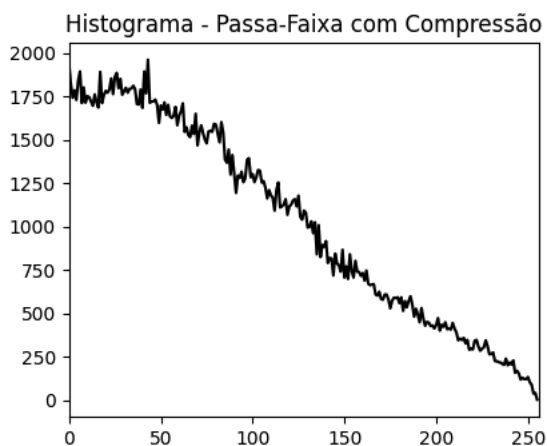


Figura 42. Histograma com filtro passa-faixa e com compressão.

V. Conclusões

Neste trabalho foi possível explorar o processamento de imagens no domínio de frequência. Foi aplicado os filtros passa-baixa, passa-alta, passa-faixa e rejeita-faixa em imagens monocromáticas por meio do espectro de Fourier. Nesse relatório é apresentado em apenas uma imagem, porém os mesmos efeitos foram observados em outras imagens.

Foi decidido manter os raios dos filtros e variar apenas o limiar para verificar os efeitos de degradação na imagem conforme eram removidas as magnitudes.

Nos experimentos percebe-se que a manipulação da magnitude afeta a remoção de nitidez da imagem, afetando altas frequências conforme aumenta o limiar. Isso é observado nas Figuras de passa-alta, onde os pixels de primeiro plano (pixels claros) passam para o plano de fundo (pixels escuros). Vê-se também, em filtros passa-baixa, um efeito de embaçamento com ruído na imagem.