



**Centro de Informática
Universidade Federal da Paraíba**

PROCESSAMENTO DIGITAL DE IMAGENS

Claudio de Souza Brito - Matrícula: 20170023696

Gabriel Teixeira Patrício - Matrícula: 20170170889

Jackson de Araujo Limeira - Matrícula: 20170017518

João Pessoa, 2020.

Sumário

Sumário	2
Introdução	3
Materiais e Métodos	4
Resultados	5
Conversão RGB-YIQ-RGB	5
Negativo	7
Correlação m x n e filtros	9
Filtros Média	9
Sobel Vertical	10
Sobel Horizontal	11
Filtro média 25x25, 25x1, 1x25	11
Filtro mediana	14
Correlação normalizada usando função de biblioteca	14
Correlação usando função implementada	15
Discussão	15
Conclusão	16
Referências	16

Introdução

A utilização de filtros de suavização em processamento digital de imagens têm como objetivo principal extrair informações e transformar a imagem, eliminando ruídos, realçando bordas, segmentando. Em geral, a aquisição de uma imagem gera erros, que são chamados "ruídos", os quais causam uma deterioração da imagem, como por exemplo, em fotografias de satélites em que a turbulência do ar altera a imagem original, em imagens médicas, etc. Torna-se preciso então, utilizar algoritmos para restaurar essas imagens. O uso de filtros e a técnica de realce de imagens são poderosas ferramentas usadas na restauração de imagens. Como o objetivo geral é processar uma imagem para obter da mesma, resultados que sejam mais apropriados que a imagem inicial, para uma aplicação específica, serão abordados os principais filtros de suavização lineares e não lineares utilizados para a eliminação de ruídos e detecção de bordas em imagens [1].

Frequentemente as imagens possuem distorções que devem ser corrigidas antes das mesmas serem usadas em algum tipo de aplicação, como por exemplo, em imagens médicas; tais como CT (transmission computed tomography), MRI (magnetic resonance imaging), MSI (magnetic source imaging), raio-X, ESI (electrical source imaging), etc; sinais obtidos via satélites; sinais obtidos do fundo do oceano; imagens obtidas por aviões para detectar alvos de guerra, entre outros. Em geral, essas correções são feitas por meio de "filtros". Os filtros podem ser classificados quanto à estrutura linear do espaço de imagens, em filtros lineares e filtros não-lineares [2].

As técnicas voltadas para a análise de dados multidimensionais, adquiridos por diversos tipos de sensores recebem o nome de processamento digital de imagens, ou seja é a manipulação de uma imagem por computador de modo onde a entrada e a saída do processo são imagens [3]. O emprego dessas técnicas permite extrair e identificar informações das imagens e melhorar a qualidade visual de certos aspectos estruturais, facilitando a percepção humana e a interpretação automática por meio de máquinas.

Baseado nos fundamentos teóricos explicados anteriormente foram implementadas as técnicas de software para manipulação de imagens com o intuito de demonstrar, através de testes, como elas funcionam na prática. Portanto, o objetivo deste relatório é relatar o processo do desenvolvimento de tais técnicas, de modo que, ao final deste documento os conhecimentos acerca dos conceitos de processamento digital de imagens estejam bem fixados.

Materiais e Métodos

O trabalho foi desenvolvido utilizando a linguagem de programação Python na versão 3.8 pelo Jupyter Notebook. Como auxílio, foram utilizadas as bibliotecas:

- Pillow* (PIL), para ler e manipular imagens
- Time*, para calcular o tempo de processamento da função
- OpenCV*, *numpy*, e *matplotlib* foram definitivas para fazer a questão número 6, onde era permitido o uso de bibliotecas com funções de processamento de imagem já prontas
- Statistics*, para calcular a média de uma lista

As técnicas que foram desenvolvidas, previamente exigidas pelo professor Leonardo Vidal, são as seguintes:

- Conversão de uma imagem RGB para YIQ e logo após para RGB: é feita a divisão da imagem em bandas R, G e B, em seguida, realiza-se o cálculo de conversão para as bandas Y, I, Q e retornando para R, G e B via cálculo de conversão, resultando em uma nova imagem, cálculos esses ensinados em sala;
- Implementação do filtro Negativo em RGB: os valores das bandas R, G e B são invertidos conforme a operação aritmética: $255 - \langle \text{valor do pixel da matriz} \rangle$;
- Implementação do filtro Negativo em Y: os valores da banda Y são invertidos conforme a operação aritmética: $255 - \langle \text{valor do pixel da matriz} \rangle$;
- Implementação do filtro média: Começamos com a extensão para 0, depois pegamos a imagem estendida, percorremos a imagem (sem contar os 0s nas bordas) com o pivô da máscara, somamos todos os valores RGB dos pixels ao redor e dividimos pela dimensão da máscara.
- Implementação do filtro Sobel 3x3: Começamos com a extensão para 0, depois pegamos a imagem estendida, percorremos a imagem (sem contar os 0s nas bordas) com o pivô da máscara, fazemos uma média ponderada entre os valores RGB dos pixels periféricos, tendo como peso os valores da máscara de sobel.
- Implementação do filtro da mediana: Começamos com a extensão para 0, depois pegamos a imagem estendida, percorremos a imagem (sem contar os 0s nas bordas) com o pivô da máscara, colocamos os valores RGB dos pixels periféricos em uma lista, depois fazemos `sort()` para colocar na ordem crescente, e usamos o valor do meio da lista.
- Implementação do filtro de correlação normalizado: Começamos com a extensão para 0, depois criamos uma lista com as médias das bandas RGB da máscara (corresponde ao w da Figura “Equação de correlação normalizada”), e depois disso tiramos a média dessa lista (corresponde ao μ_h da Figura “Equação de correlação normalizada”). Depois pegamos a imagem estendida, percorremos a imagem (sem contar os 0s nas bordas) com o pivô da máscara (sendo a máscara o

olho do babuíno), inserindo a média das bandas RGB da imagem em uma lista (corresponde ao v da Figura “Equação de correlação normalizada”) e calculando a média dessa lista (corresponde ao μ_v da Figura “Equação de correlação normalizada”). Fazemos a equação, e depois passamos o r de $[-1,1]$ para $[0, 255]$

$$r = \frac{(v - \mu_v)}{|v - \mu_v|} \cdot \frac{(w - \mu_h)}{|w - \mu_h|}$$

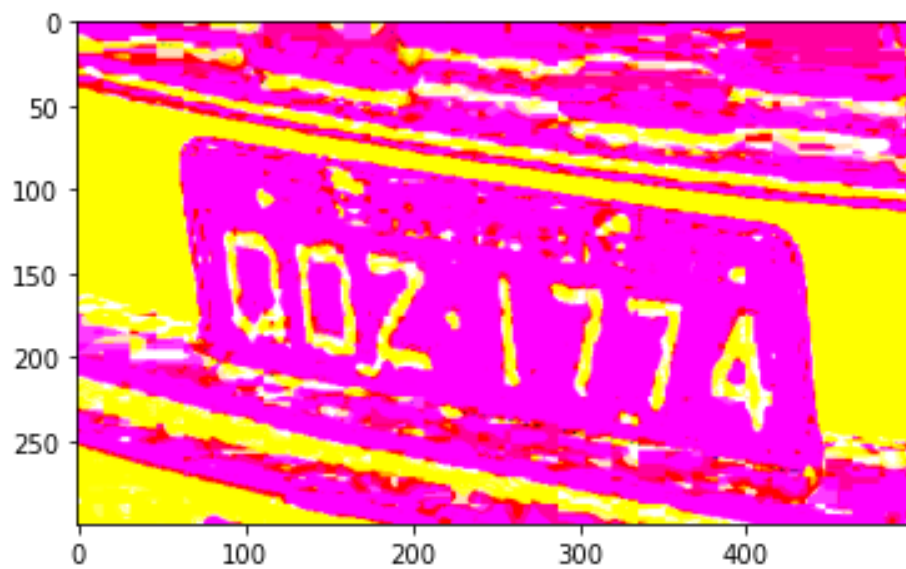
Equação de correlação normalizada

Resultados

1. Conversão RGB-YIQ-RGB



Foto original



Negativo RGB

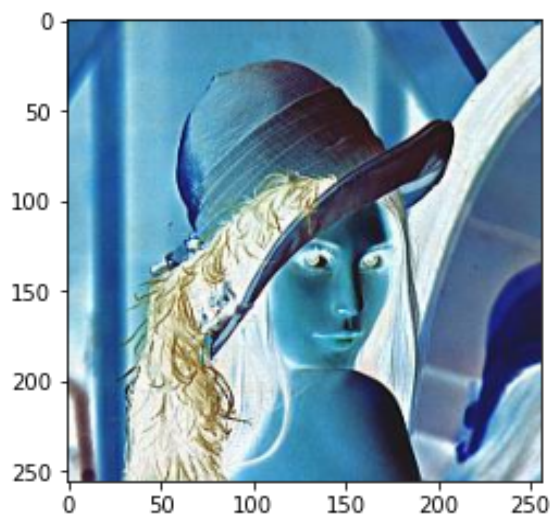


De volta para RGB

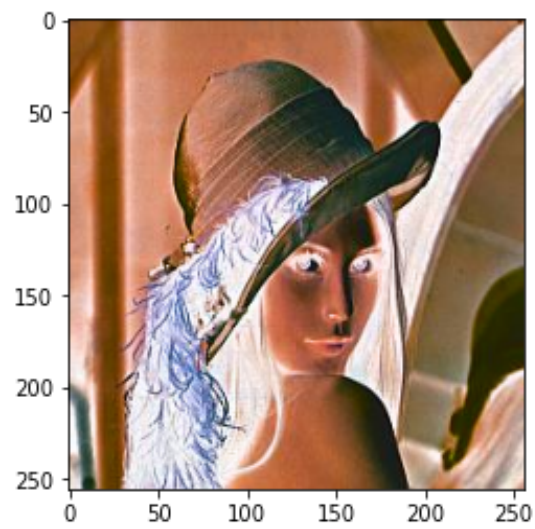
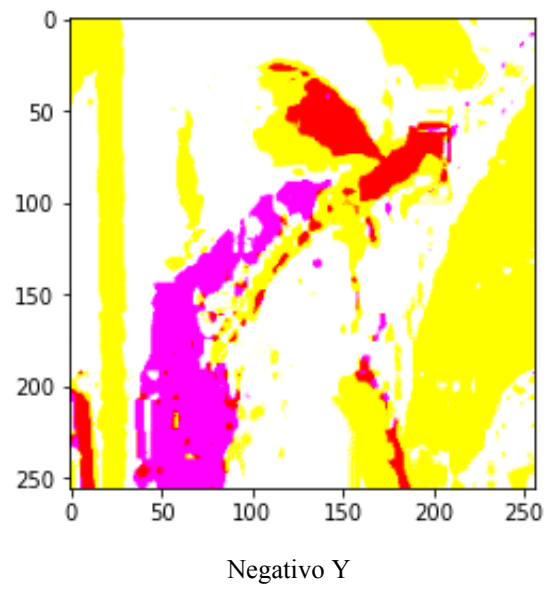
2. Negativo



Foto original



Negativo RGB



Transformação do Negativo Y para RGB

3. Correlação $m \times n$ e filtros

a. Filtros Média



Foto original



Aplicação do filtro de média

b. Sobel Vertical

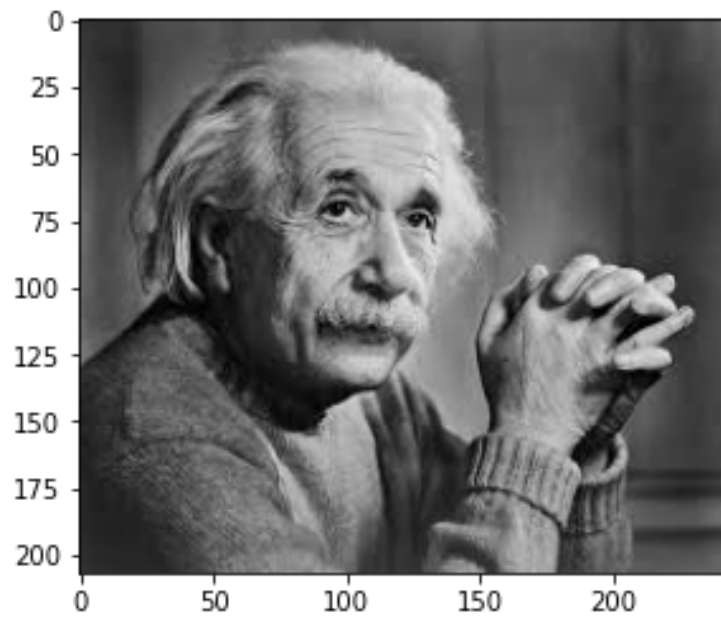
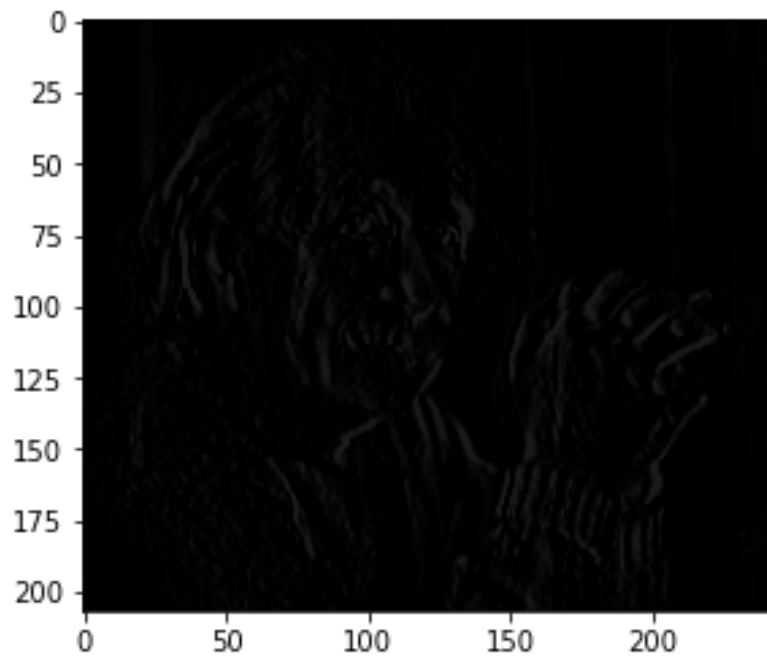
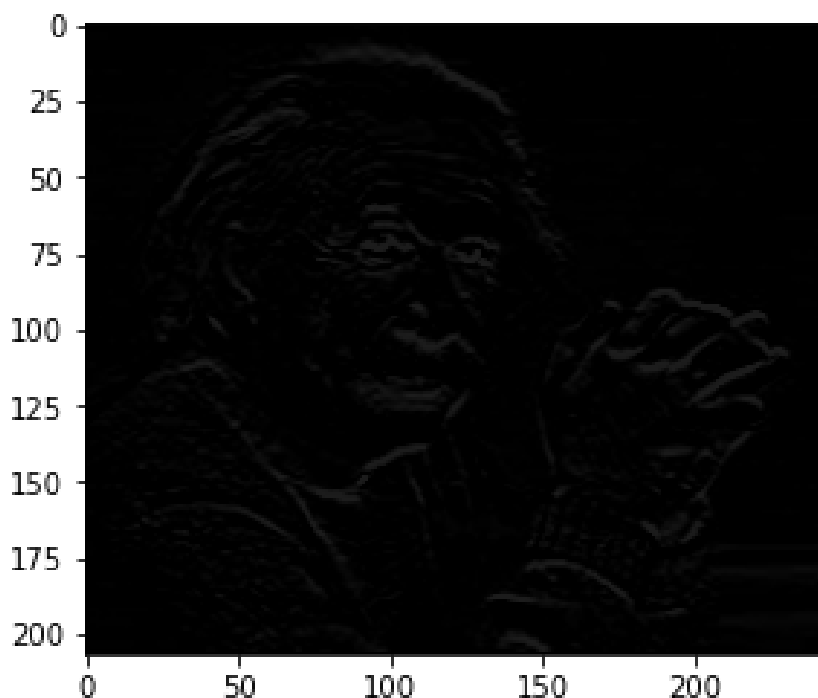


Foto original



Aplicação do filtro de sobel na Vertical

c. Sobel Horizontal



Aplicação do filtro de sobel na Horizontal

4. Filtro média 25x25, 25x1, 1x25

Podemos perceber que a imagem pós filtro de máscara 25x25 ficou completamente borrada, e foi o filtro que demorou mais tempo para processar, cerca de 33 segundos. Já na imagem onde foi aplicado a máscara 25x1, temos a impressão de uma imagem borrada horizontalmente, e o tempo de processamento baixou para 3 segundos. E onde foi utilizado a máscara 1x25, o tempo caiu para 1 segundo, e a impressão é de um resultado com a imagem borrada verticalmente.



Foto original

tempo de processamento 25x25: 33.484375



Filtro de média 25x25

tempo de processamento 25x1: 3.09375



Filtro de média 25x1

tempo de processamento 1x25: 1.78125



Filtro de média 1x25

5. Filtro mediana



Filtro de mediana

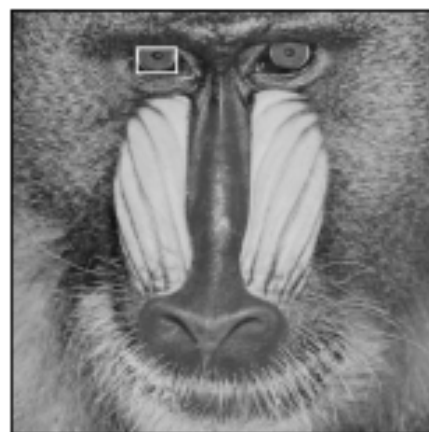
6. Correlação normalizada usando função de biblioteca

Correlação normalizada

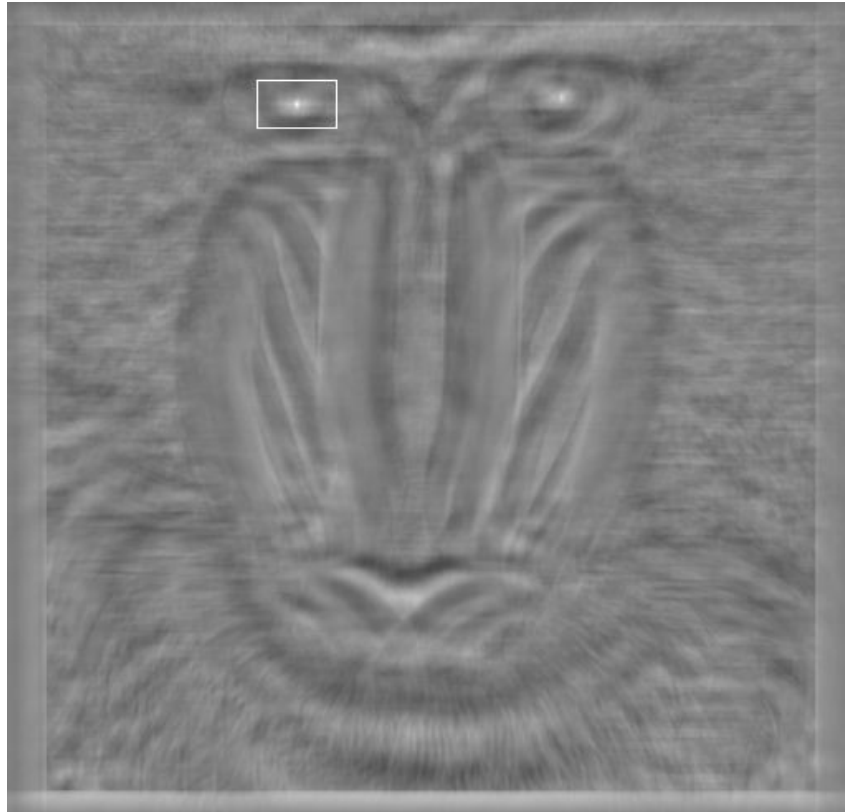
Matching Result



Detected Point



7. Correlação usando função implementada



Discussão

Apesar das aulas e material fornecido, a limitação de não poder usar bibliotecas e funções prontas, trouxe para o trabalho uma dificuldade a mais em construir do zero as funções para cada filtro, em especial tudo o que envolvia utilizar a função “for” para percorrer as matrizes, a confusão de ter até quatro dessas funções uma dentro da outra e ter que pensar nos intervalos de cada um deles sem receber o seguinte erro “out of index”, que acontece quando se tenta acessar espaço de memória que não existe, foi enorme.

Outro quesito muito questionado pelo grupo foi a validação de cada filtro para ver se o resultado era o esperado, que foi resolvido com os exemplos dados no slide oferecido pelo professor da disciplina.

Conclusão

Ao fim do trabalho pode-se concluir que os conhecimentos acerca do processamento digital de imagem foi posto em prática, pois baseando-se nos conceitos teóricos tornou-se possível a implementação das técnicas de manipulação, o que por sua vez ajudou a fixar os conteúdos estudados na disciplina. Os filtros foram bem aplicados para todas as imagens testadas, e podem ser utilizados para aplicar em qualquer imagem desejada. Além disso, houve um aprofundamento nos conhecimentos da linguagem Python e na utilização da ferramenta OpenCV.

Referências

- [1] <https://projetos.extras.ufg.br/conpeex/2004/pibic/exatas/Vinicius.html>
- [2] GOMES, J.; VELHO, L. Computação Gráfica: Imagem. Rio de Janeiro: IMPA/SBM, 1994.
- [3] Teoria: Processamento de Imagens. Disponível em:
<http://www.dpi.inpe.br/spring/teoria/realce/realce.htm>. Acesso em: 12/11/2020.