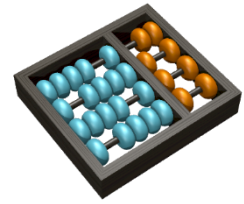




**Universidade Estadual de Campinas
Instituto de Computação**



Disciplina do 2º Semestre de 2020

IC - UNICAMP

Curso: Bacharelado em Ciência da Computação

MC920 - Introdução ao Processamento de Imagens Digitais

Trabalho 1 - Aplicação de Filtros em Imagens Monocromáticas

Alunos: Gabriel Volpato Giliotti **RA:** 197569

Professores: Hélio Pedrini

Campinas – SP
2020

Trabalho 1 - Introdução ao Processamento de Imagem Digital

Nome: Gabriel Volpato Giliotti - **RA:** 197569

1 - Introdução

O objetivo deste trabalho é realizar a aplicação de diferentes tipos de filtros/máscaras em imagens monocromáticas no formato .png, além de uma análise sobre o comportamento de cada filtro previamente definido no enunciado.

Atualmente, o processamento de imagens é um artifício muito utilizado em mídias sociais, aparelhos eletrônicos e softwares, para aplicação de filtros e outros recursos em imagens. Assim, os filtros/máscaras nos oferecem diferentes objetivos de aplicações, disponibilizando a recuperação, suavização ou destaque de pontos específicos (pixels) em imagens. Com isso, o projeto tem como objetivo introduzir a aplicação de diferentes tipos de máscaras de imagens, oferecendo base para o entendimento de como alguns filtros podem ser aproveitados no processamento de imagens monocromáticas.

Junto desse relatório será entregue o arquivo .zip ou .tgz, que possui todos os arquivos citados nesse relatório e que foram utilizados para processar as imagens dadas como entrada.

2 - Programas

O script para processamento das imagens foram implementados em Python 3.7.6 junto das bibliotecas Numpy 1.19.2 e OpenCV 4.4.0 que são voltadas para o processamento de imagens. Outras bibliotecas como Math e Sys, que auxiliam no processamento, também foram utilizadas mas não oferecem recursos específicos para processamento de imagens.

2.1 - Como Executar

Inicialmente, antes de executar o script, devemos criar uma pasta chamada FilterOutputs que corresponde ao local de saída das imagens da execução do arquivo .py referente ao trabalho 1. Para se obter as imagens de saída, você deve criar a pasta no mesmo diretório onde executará o arquivo .py.

Para o trabalho 1 foi criado um único arquivo com extensão do tipo .py que é responsável pela aplicação das máscaras definidas no enunciado. A seguir está o formato de construção para chamada do arquivo via linha de comando (Windows: CMD/ Linux: Bash):

```
C:\> python filtros.py pathImagem tipoFiltro
```

Substitua “pathImagem” pelo caminho (path) correto da imagem à qual se deseja aplicar uma máscara e “tipoFiltro” para escolha do tipo de filtro que será aplicado. Os possíveis tipos de filtros são “h1”, “h2”, “h3”, “h4”, “h5”, “h6”, “h7”, “h8”, “h9”, “h10”, “h11” e “h3h4”.

2.2 - Entradas

Como entrada para o arquivo .py é necessário o path da imagem a ser processada, além da definição do tipo de filtro a ser aplicado. Todas as imagens devem ser Monocromáticas, no formato PNG (Portable Network Graphics). As imagens utilizadas para esse projeto vieram de um repositório indicado pelo professor e estão compactadas no .zip ou .tgz junto ao arquivo .py em uma pasta chamada ImagesMono.

2.3 - Saídas

As saídas são obtidas na pasta criada previamente, antes da execução do arquivo. Todas as imagens de saída serão Monocromáticas, no formato PNG (Portable Network Graphics) e com a mesma resolução de pixels da imagem de entrada. Além disso, por causa do modo como a convolução é aplicada sobre a imagem (sem a utilização de bibliotecas que auxiliam o processamento), o tempo para gerar uma imagem de saída tem baixa velocidade, fazendo com que o tempo de aplicação dos filtros sejam de 5 a 10 segundos para cada imagem (O tempo pode variar de acordo com a resolução da imagem de entrada).

3 - Parâmetros Utilizados

Os parâmetros utilizados para esse projeto são o path da imagem ou imagens a serem processadas e o tipo de filtro a ser aplicado. Novamente, os possíveis tipos de filtros aplicados podem ser “h1”, “h2”, “h3”, “h4”, “h5”, “h6”, “h7”, “h8”, “h9”, “h10”, “h11” e “h3h4”.

4 - Soluções

A seguir estão as comparações e observações da aplicação de cada filtro com sua correspondente imagem original. A escolha das imagens foram aleatórias, na tentativa de melhor se apresentar os resultados obtidos. Além disso, tentamos apresentar as aplicações de forma mais abrangente possível.

4.1 - Filtro H1

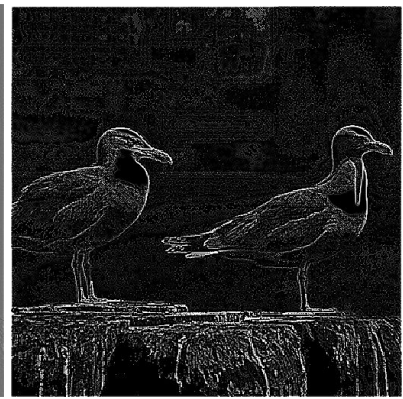
O filtro “h1” quando aplicado, ressalta áreas onde os valores de pixels tem maior contraste de na imagem original. Como consequência do alto valor do pixel central do kernel, ruídos também são intensificados. Observe a imagem mais escura, onde é melhor visível que muitos pixels no background receberam uma tonalidade branca.



seagull.png Original



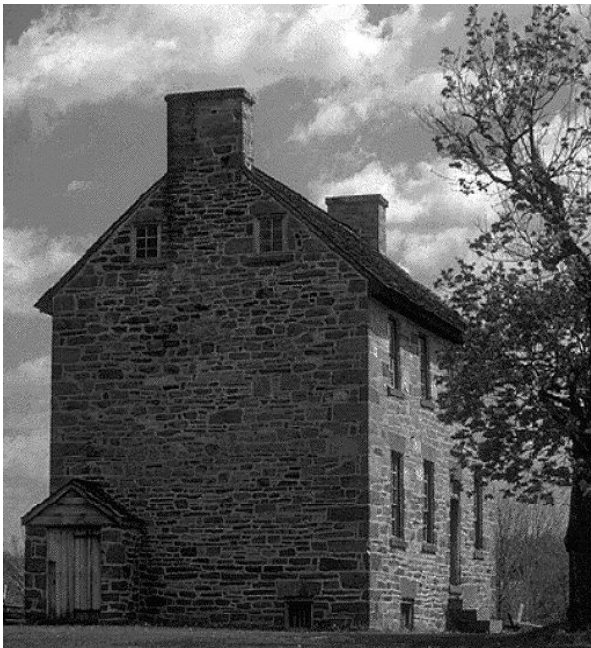
seagull.png filtro h1 [0,255]



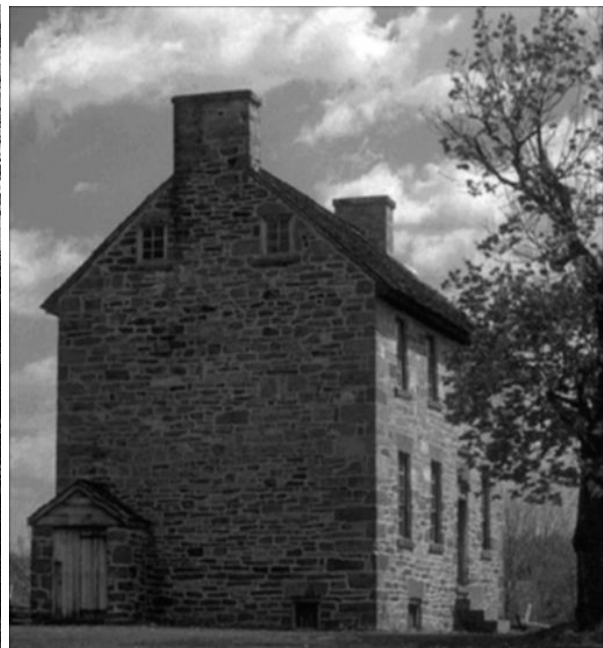
seagull.png filtro h1

4.2 - Filtro H2

O filtro “h2” apresenta uma suavização dos valores dos pixels da imagem original. Quando aproximados e comparados, os pixels da imagem original apresenta tonalidades pretas e brancas aleatoriamente, enquanto na imagem com filtro, os pixels aparentam apresentar tonalidades similares em seus entornos. Isso causa a impressão do embaçamento na imagem de saída como um todo, mas também de suavização dos contornos e uma redução dos ruídos da imagem original. Além disso, esse filtro não apresenta uma imagem escura como o primeiro, pois nos valores da matriz da máscara, não existem números negativos.



house.png Original



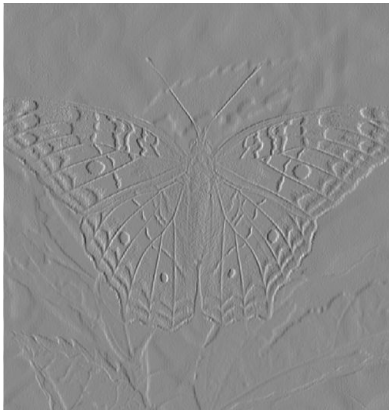
house.png filtro h2

4.3 - Filtro H3

O filtro “h3” parece fazer um realce em relevo de partes mais claras da imagem original, fazendo com que, no espectro acinzentado [0,255], as partes mais claras ganhem uma aparência em relevo com relação a partes mais escuras. Observe também que no espectro negativo (imagem preta , mais a direita), as partes escuras da imagem original não ganham um contorno em branco tão bom quanto partes escuras.



butterfly.png Original



butterfly.png filtro h3 [0,255]



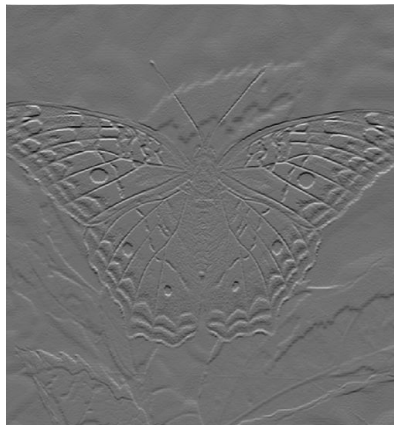
butterfly.png filtro h3

4.4 - Filtro H4

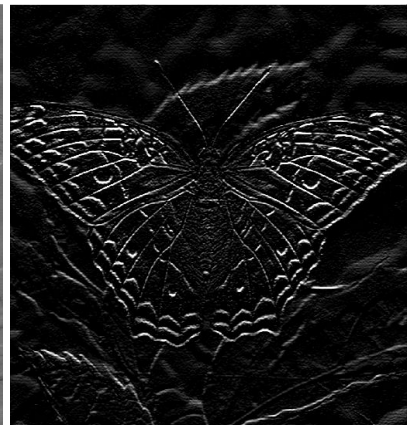
O filtro “h4” causa um realce nos contornos mais escuros da imagem, diferente do filtro “h3” que realçou áreas mais claras. Note que a diferença entre os filtros está exatamente na diferença de realces. Em “h3” o que parece ganhar relevo no espectro cinza [0,255] são áreas claras, e em “h4” o que parece ganhar relevo no espectro cinza são contornos escuros. Tal diferença é notável nas imagens mais escuras, onde os contornos brancos parecem estar diferentes.



butterfly.png Original



butterfly.png filtro h4 [0,255]



butterfly.png filtro h4

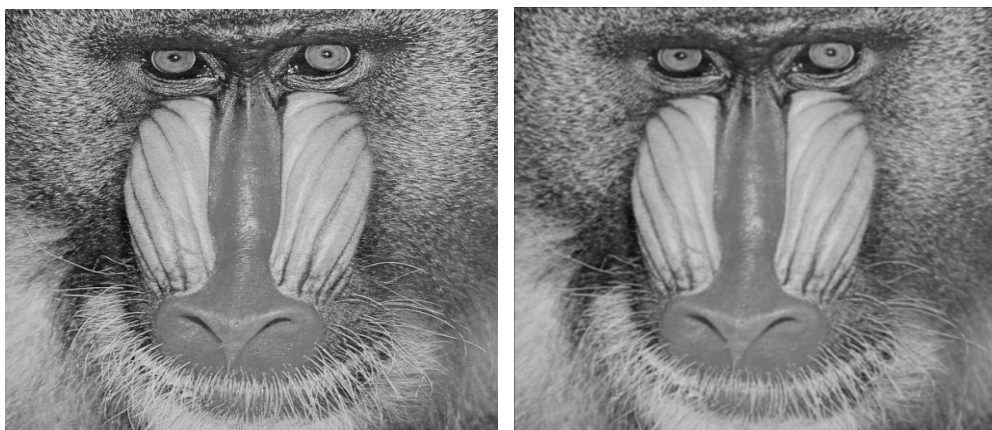
4.5 - Filtro H5

O filtro “h5” causa o realce de contornos (áreas com alto contraste) nas imagens, o que configura redução do borramento. No espectro cinza [0, 255] é necessário aproximar a imagem para melhor visualizar o branqueamento dos pequenos detalhes, mas observe que na imagem escura, como linhas e pontos são convertidos para uma coloração branca, sendo possível enxergar até mesmo as telhas das casas (o que não é possível tão facilmente na imagem original).



4.6 - Filtro H6

O filtro “h6” trabalha de forma similar ao filtro “h2”, fazendo com que ocorra uma suavização dos pixels dos entornos. Diferente de “h2” que aparenta apresentar certo embaçamento, o filtro “h6” causa um efeito de redução de reflexo da imagem original, além de uma redução de contornos (coloração dos pixels de linhas finas parecem estar dispostos de forma mais uniforme). Vale ressaltar que os valores dos pixels da máscara “h6” não apresentam uma distribuição gaussiana, mas sim uniforme, o que causa uma suavização de contornos mais leve que “h2”, mas também um embaçamento menor. Além disso, O filtro “h6” não apresenta uma imagem escura como nos filtros anteriores, pois nos valores da matriz da máscara, não existem números negativos.



baboon.png Original

baboon.png filtro h6

4.7 - Filtros H7 e H8

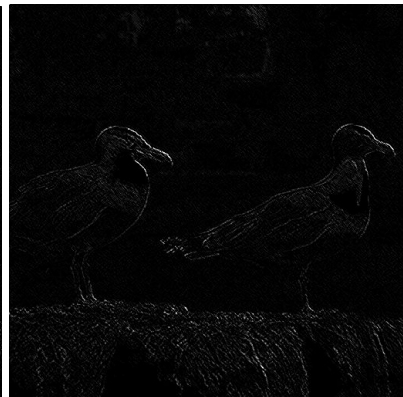
Os filtros “h7” e “h8” apresentam os mesmos valores nas matrizes do filtro, mudando apenas a orientação dos valores mais altos, isto é, em “h7” os valores mais altos aparecem na diagonal secundária do filtro e em “h8” os valores mais altos aparecem na diagonal principal do filtro. Além disso, essas máscaras causam um realce em locais de maior contraste de cores da imagem original e também apresentam um pouco de ruído.



seagull.png Original



seagull.png filtro h7



seagull.png filtro h8



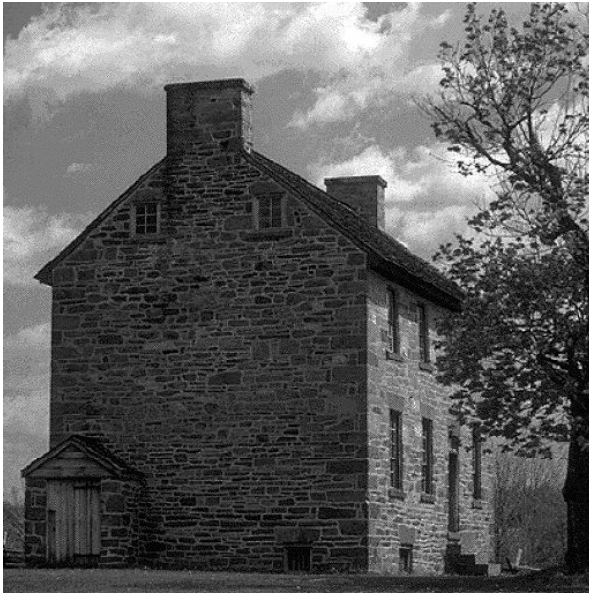
seagull.png filtro h7 [0,255]



seagull.png filtro h8 [0,255]

4.8 - Filtro H9

O filtro “h9” faz uma transformação dos pixels, de forma a dispô-los na saída com valores similares na diagonal, isto é, aproximando a imagem é possível verificar que os pixels nas diferentes diagonais da foto apresentam uma tonalidade similar, o que também causa o embaçamento na imagem. Além disso, “h9” não apresenta imagem escura como em alguns filtros anteriores, pois nos valores da matriz da máscara, não existem números negativos.



house.png Original



house.png filtro h9

4.9 - Filtro H10

O filtro “h10” tem uma distribuição gaussiana no centro do kernel, com valores negativos no entorno e, além disso, está sendo normalizado. Tal configuração permite que o filtro, normalizado ou não apresente uma imagem com coloração clara, realçando locais com maior contraste, uma melhor definição de contornos e redução do borramento. Observe também que aplicação da normalização causa um escurecimento (em tons de cinza) na imagem com filtro aplicado.



city.png Original



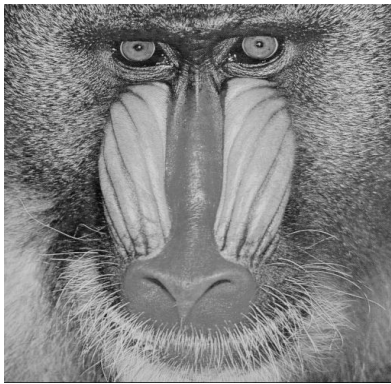
city.png filtro h10



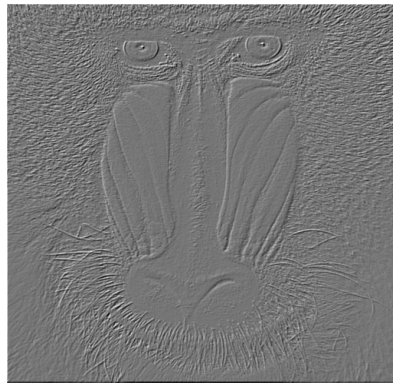
city.png filtro h10 [0,255]

4.10 - Filtro H11

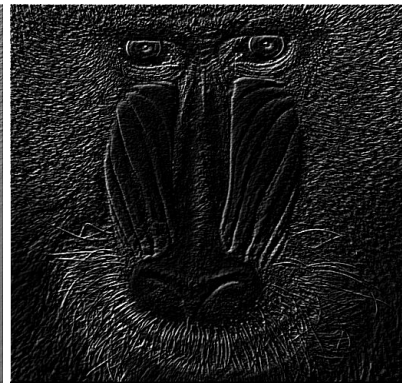
O filtro “h11” permite o realçamento de contornos da imagem original. Observe que em pontos onde ocorre reflexão de luz (regiões claras na imagem original) também são considerados locais a serem realçados.



baboon.png Original



baboon.png filtro h11 [0,255]



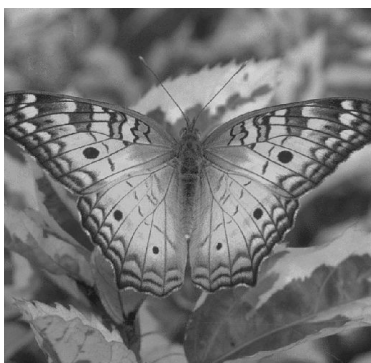
baboon.png filtro h11

4.11 - Filtro H3H4

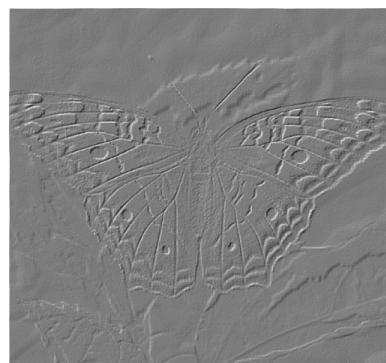
Aqui, os filtros “h3” e “h4” são aplicados de forma conjunta sobre uma imagem de entrada, de acordo com método especificado no enunciado. Sendo uma outra forma de detectar bordas, essa aplicação é chamada algoritmo de Sobel e segue o seguinte procedimento:

$$\begin{aligned} \text{ImgFiltroh3} &= \text{ImgOriginal} \times \text{Filtroh3} \\ \text{ImgFiltroh4} &= \text{ImgOriginal} \times \text{Filtroh4} \\ \text{ImgSaida} &= \sqrt{\text{ImgFiltroh3}^2 + \text{ImgFiltroh4}^2} \end{aligned}$$

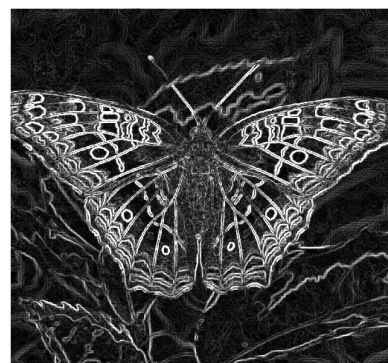
Observe como os contornos são melhores definidos com a aplicação desse algoritmo. A imagem normalizada apresenta um alto relevo para regiões de contorno e, na imagem não normalizada, também é possível verificar que os contornos estão melhores definidos.



butterfly.png Original



butterfly.png filtro h3h4 [0,255]



butterfly.png filtro h3h4

4.12 - Limitações no Projeto

Como limitações pertinentes ao projeto, temos inicialmente a falta de experiência com a utilização da linguagem python, além do não conhecimento do repertório oferecido pelas bibliotecas utilizadas e disponíveis. Além disso, foi recomendado a não utilização de laços de repetição para processar as imagens, mas nesse caso tal recomendação não foi seguida pois houve dificuldade em encontrar um algoritmo que fizesse a convolução de forma correta sobre as imagens. Assim, todos os algoritmos de convolução disponíveis em bibliotecas, encontrados e aplicados diretamente, apresentaram resultados não esperados, nos levando a optar por uma saída menos eficiente com relação ao tempo, mas que resultasse nos valores corretos da aplicação da convolução. Além disso, houve certa incerteza sobre como os filtros iriam resultar, logo apresentamos as imagens (com os filtros aplicados) com e sem a normalização dos valores dos pixels (pixels com valores no intervalo $[0,255]$). Outro ponto importante é que a todo momento, os entornos das imagens foram tratados com valor 0, isso é, uma moldura de 0's foi construída antes da aplicação dos filtros e isso não é uma regra. Existem outras formas de tratamento além de valores 0's nos entornos, como por exemplo fazer uma reflexão dos pixels próximos às bordas, ou pré-definir um valor diferente de 0 e aplicá-lo. A moldura de 0's foi tomada como a escolha mais simples.

4.13 - Conclusão

Através trabalho 1, pudemos obter um primeiro contato com o Processamento de Imagens no que diz respeito à aplicação de filtros em imagens, sendo possível observar diferentes dificuldades que existem na aplicação de máscaras em imagens monocromáticas, com relação a realização de convolução sobre os pixels, tipos de tratamentos e abordagens que podem ser tomadas para valores negativos aplicados e para valores fora dos limites da imagem. Além disso, pudemos ter um melhor entendimento do objetivo da aplicação de cada tipo de filtro disponibilizado.