

DETECÇÃO DE BORDA

Sobel

O detector de bordas de Sobel consiste em duas matrizes, 3x3, 5x5, 7x7, 9x9... implementadas no “sobel.m”, onde uma delas serve para a detecção de bordas verticais, e a outra para a detecção de bordas horizontais.

Primeiramente, é aplicada a máscara vertical na imagem original, logo após nós aplicamos a máscara horizontal também a imagem original. Com as duas imagens, nós aplicamos o gradiente e obtemos a imagem com os filtros unidos.

□ Operador de Sobel

▫ Máscaras

$$h_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

▫ Gradiente

$$|\nabla f(x, y)| \approx \sqrt{(f * h_1)^2 + (f * h_2)^2}$$

Para aplicarmos a máscara na imagem, utilizamos de um processo de filtragem parecido com uma operação de convolução.

□ Processo de filtragem

- Cada elemento da máscara é multiplicado pelo valor do pixel correspondente na imagem f
- A soma desses resultados é o novo valor do nível de cinza na nova imagem g
- Exemplo: w é uma janela de $n \times n = k$ pixels. O processo de filtragem para cada pixel na imagem $g(x, y)$ será dada por

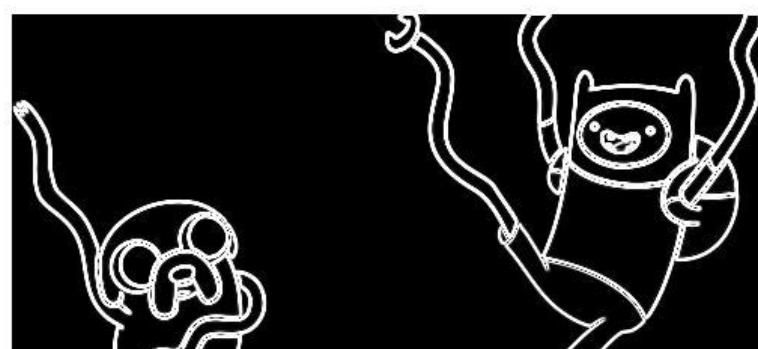
$$g(x, y) = \sum_{i=1}^k w_i \cdot f(x, y)$$

Vemos na imagem abaixo exatamente como a máscara passa na imagem. Esse processo resulta numa perca de pixels.

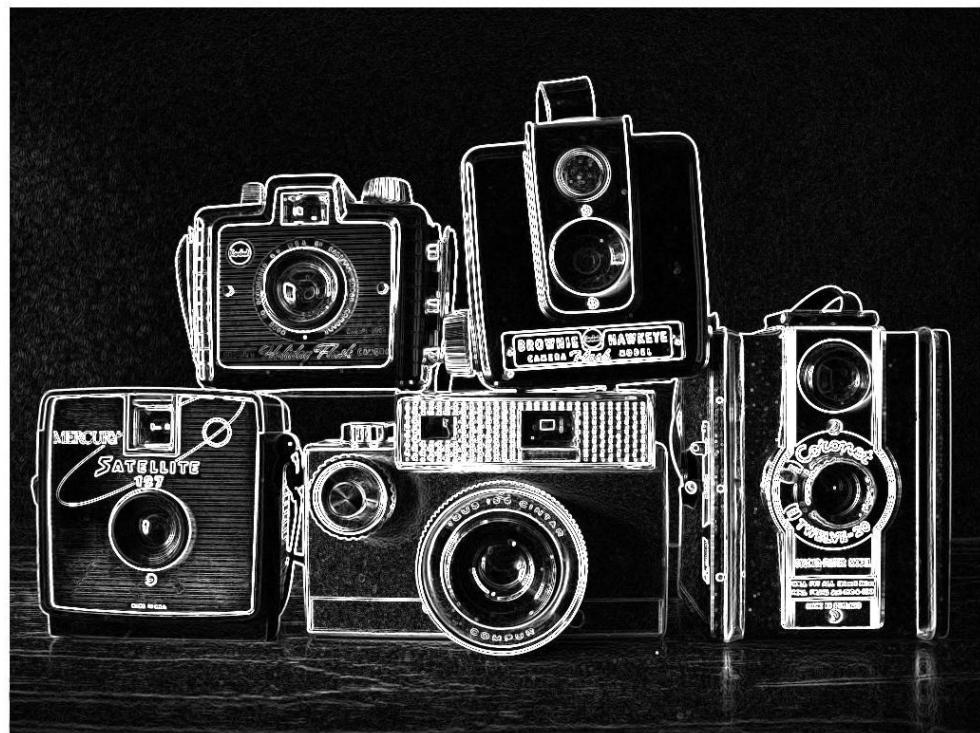
□ Seguintes conjunto de operações
Desloca, Multiplica, Soma

máscara	Imagem	Resultado																																														
<table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table>	1	0	1	0	0	1	<table border="1"> <tr><td>1</td><td>1</td><td>3</td><td>3</td><td>4</td></tr> <tr><td>1</td><td>1</td><td>4</td><td>4</td><td>3</td></tr> <tr><td>2</td><td>1</td><td>3</td><td>3</td><td>3</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>4</td><td>4</td></tr> </table>	1	1	3	3	4	1	1	4	4	3	2	1	3	3	3	1	1	1	4	4	<table border="1"> <tr><td>2</td><td>5</td><td>7</td><td>6</td><td>*</td></tr> <tr><td>2</td><td>4</td><td>7</td><td>7</td><td>*</td></tr> <tr><td>3</td><td>2</td><td>7</td><td>7</td><td>*</td></tr> <tr><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td></tr> </table>	2	5	7	6	*	2	4	7	7	*	3	2	7	7	*	*	*	*	*	*
1	0																																															
1	0																																															
0	1																																															
1	1	3	3	4																																												
1	1	4	4	3																																												
2	1	3	3	3																																												
1	1	1	4	4																																												
2	5	7	6	*																																												
2	4	7	7	*																																												
3	2	7	7	*																																												
*	*	*	*	*																																												
	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>3</td><td>3</td><td>3</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>4</td><td>4</td><td></td></tr> </table>	1	0	0	0	0	0	0	1	0	0	0	1	2	1	3	3	3		1	1	1	4	4		<p>A imagem resultado é menor do que a imagem original. Os valores marcados com * não podem ser calculados.</p>																						
1	0	0	0	0	0																																											
0	1	0	0	0	1																																											
2	1	3	3	3																																												
1	1	1	4	4																																												

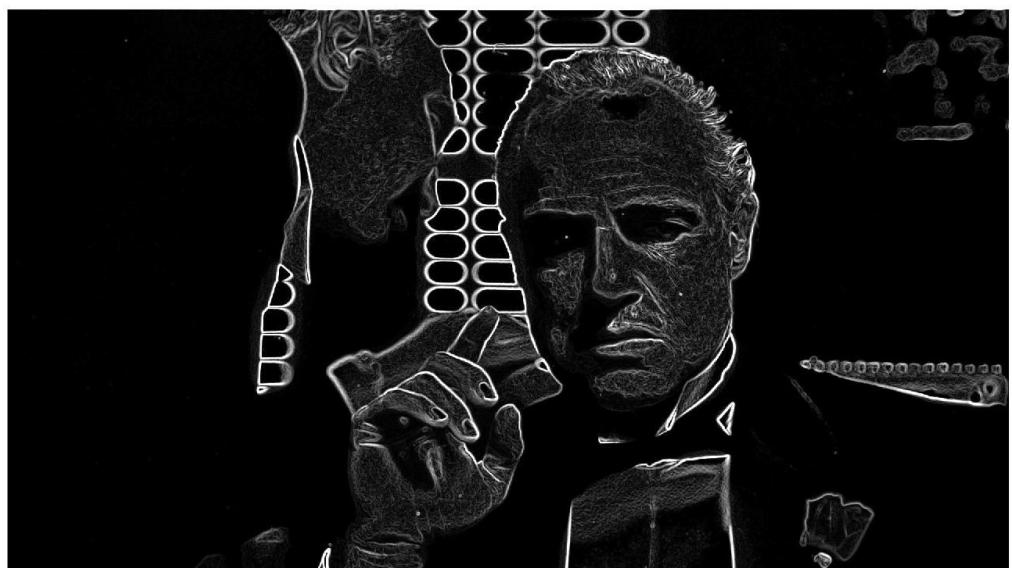
A Seguir algumas imagens que receberam a máscara de sobel.

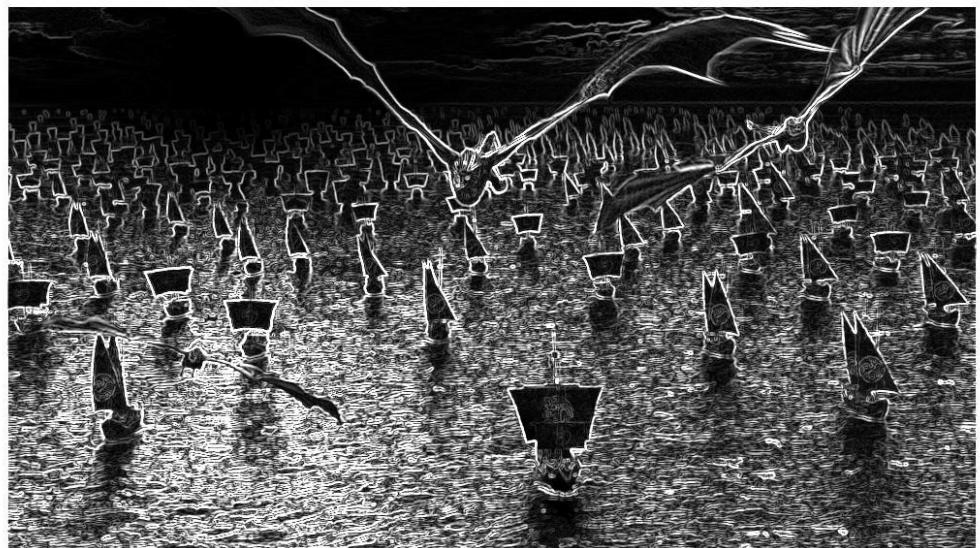


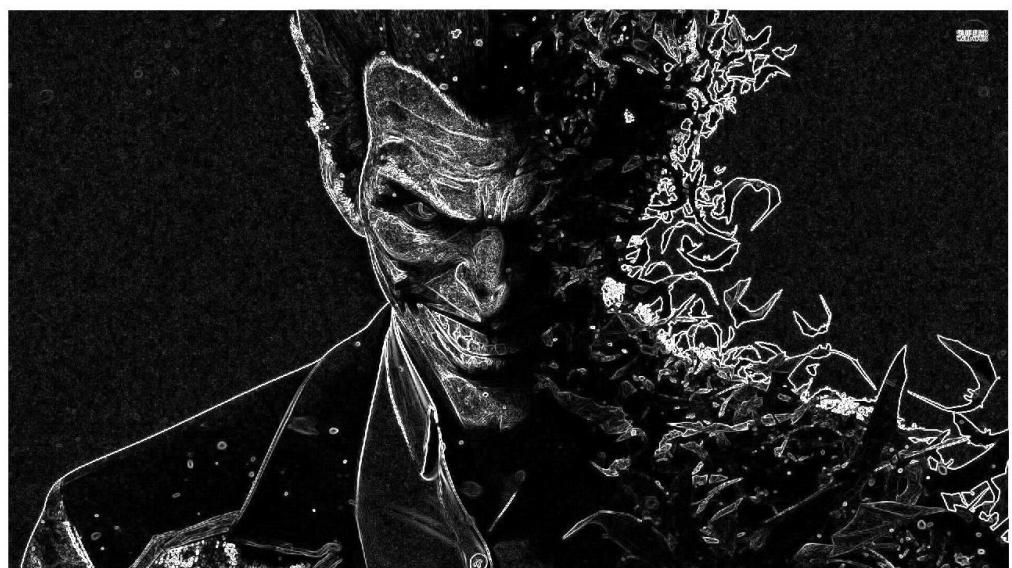




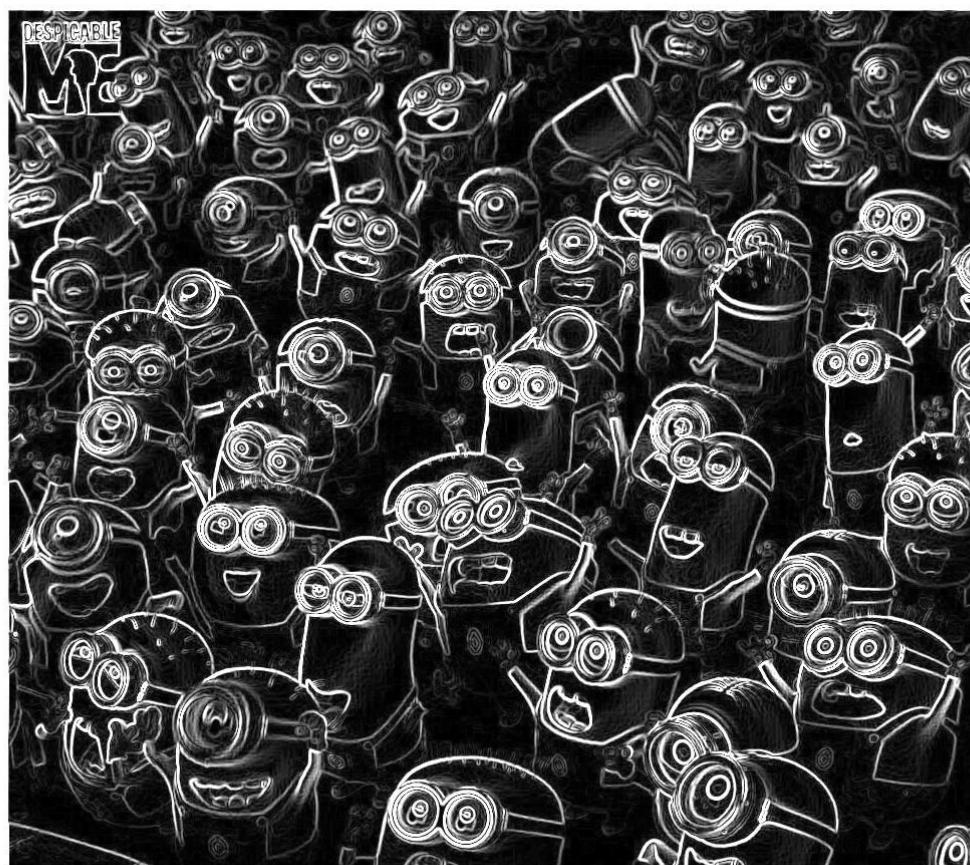
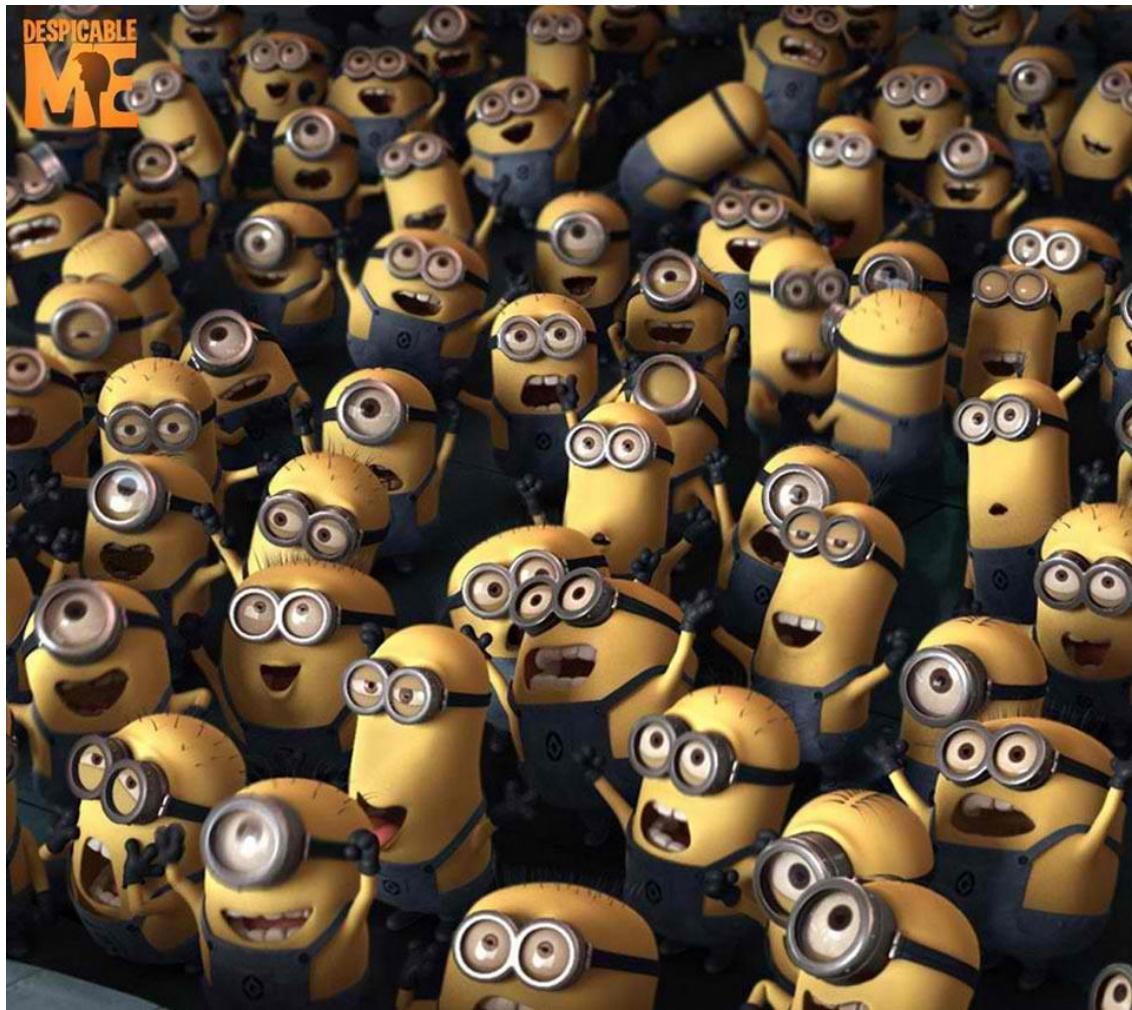




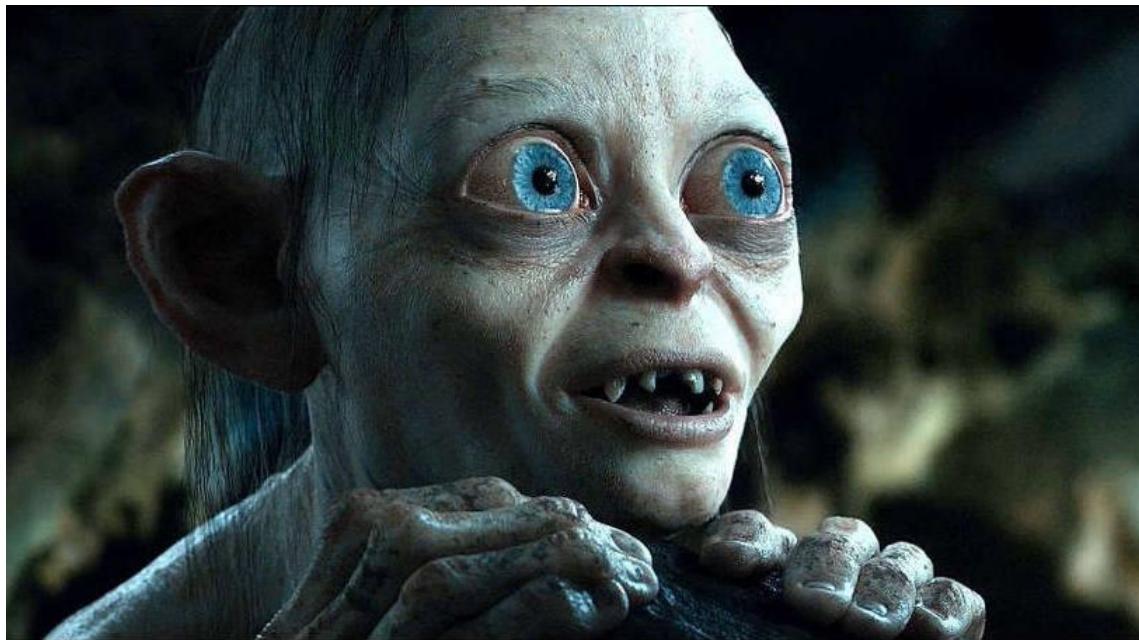








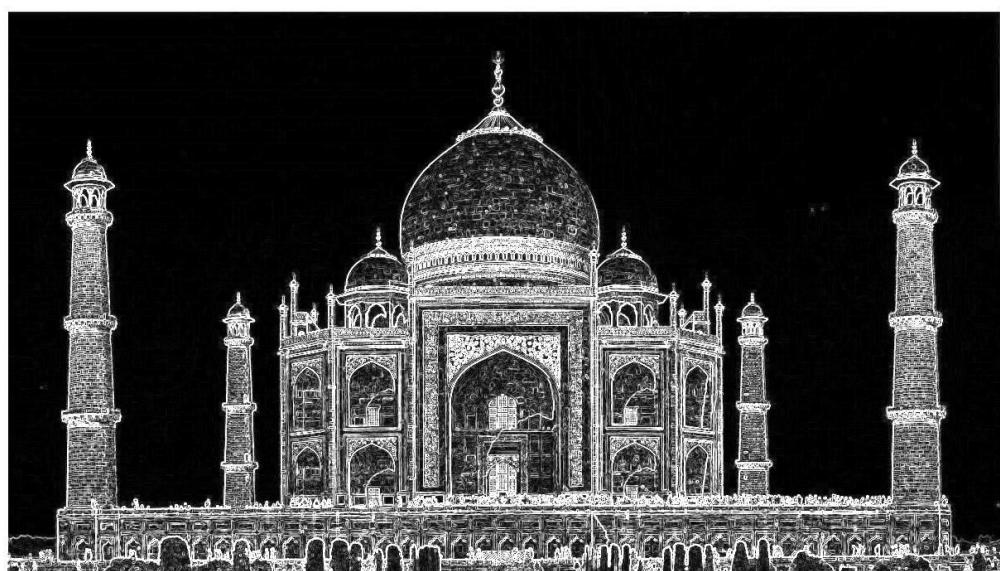
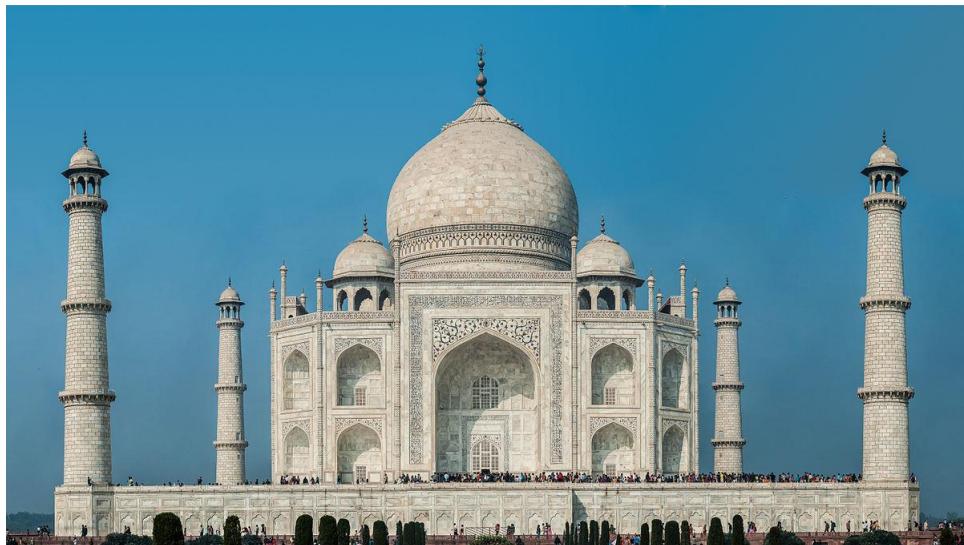












Laplacian

O detector de bordas de Laplacian consiste em duas matrizes, 3x3, 5x5, 7x7, 9x9... implementadas no “laplacian.m”, no nosso caso nós usamos o filtro laplaciano com centro positivo (remove bordas interiores); É um filtro que utiliza derivadas de segunda ordem e é isotrópico.

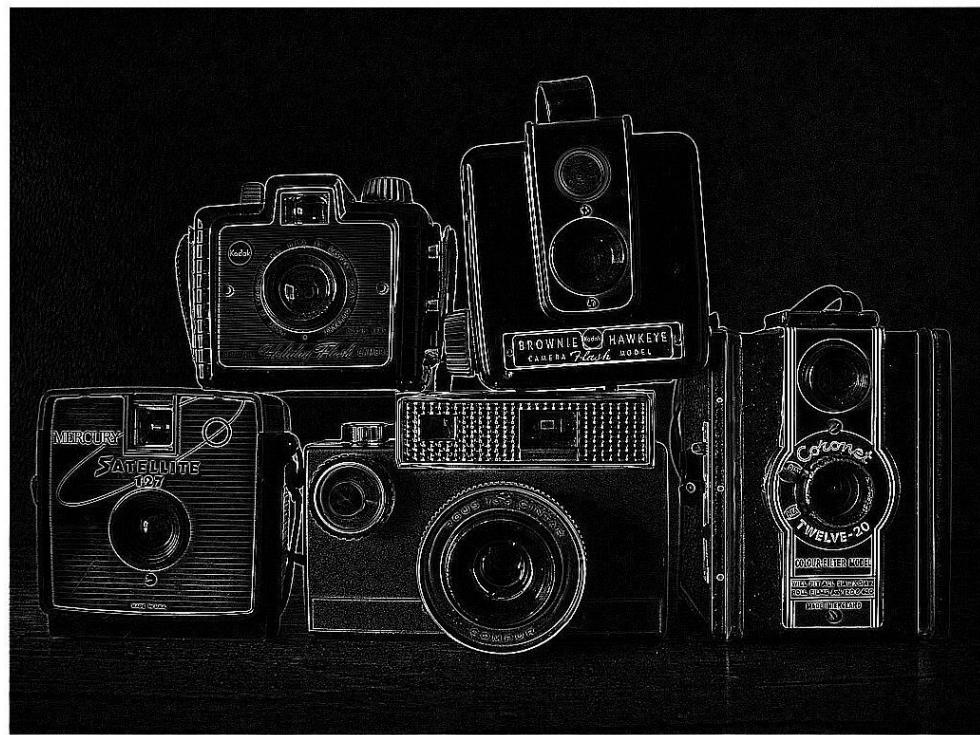
□ Filtro Laplaciano

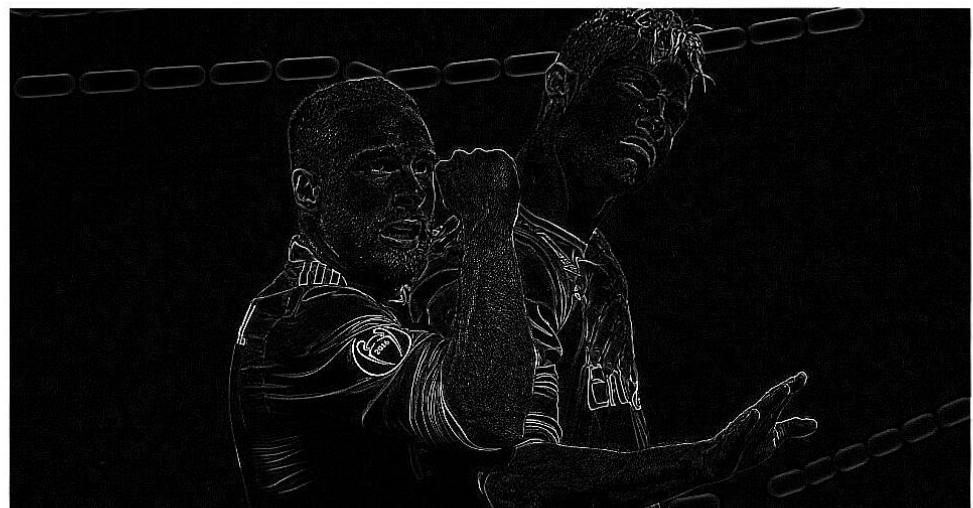
- Utiliza derivadas de segunda ordem
 - Resposta mais acentuada a detalhes finos como pontos isolados e linhas
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$
.
- É um filtro isotrópico
 - A resposta é independente da direção da descontinuidade na imagem em que o filtro é aplicado (invariante à rotação);

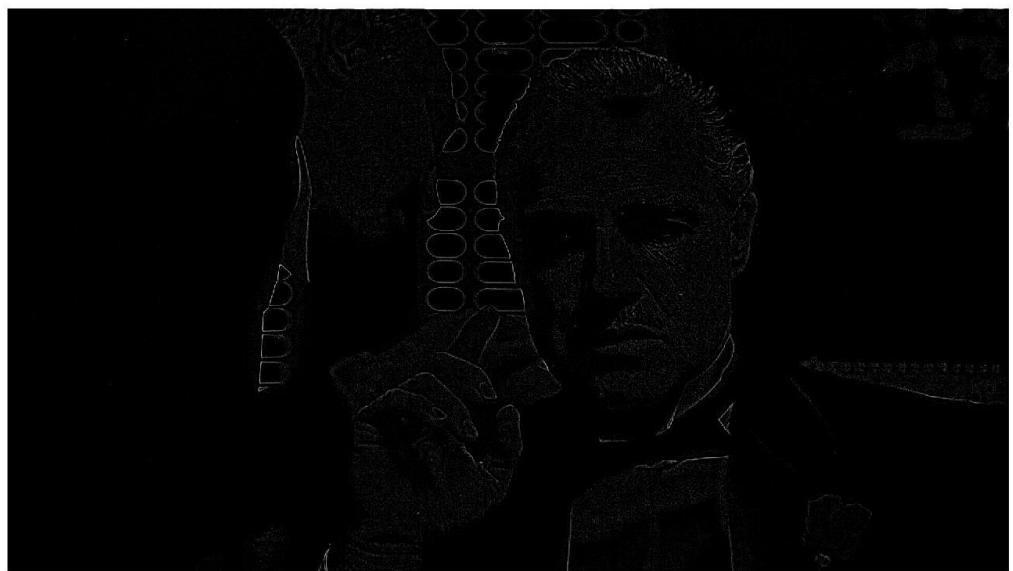
A aplicação dele é feita de forma bastante similar ao filtro de Sobel. A diferença no algoritmo está no fato de que o filtro laplaciano em “laplacian.m” é calculado da seguinte forma: a função recebe a imagem e o tamanho da matriz filtro (filtros “pequenos” apresentam melhores resultados em imagens “pequenas”), a matriz é calculada colando o valor $(tamanho*tamanho)-1$ na posição $\text{ceil}(tamanho/2)$ e o resto da matriz recebe a constante -1. Filtro feito, aplicamos na imagem com a mesma função de filtro do sobel.

Vamos observar algumas imagens criadas com o filtro Laplaciano:

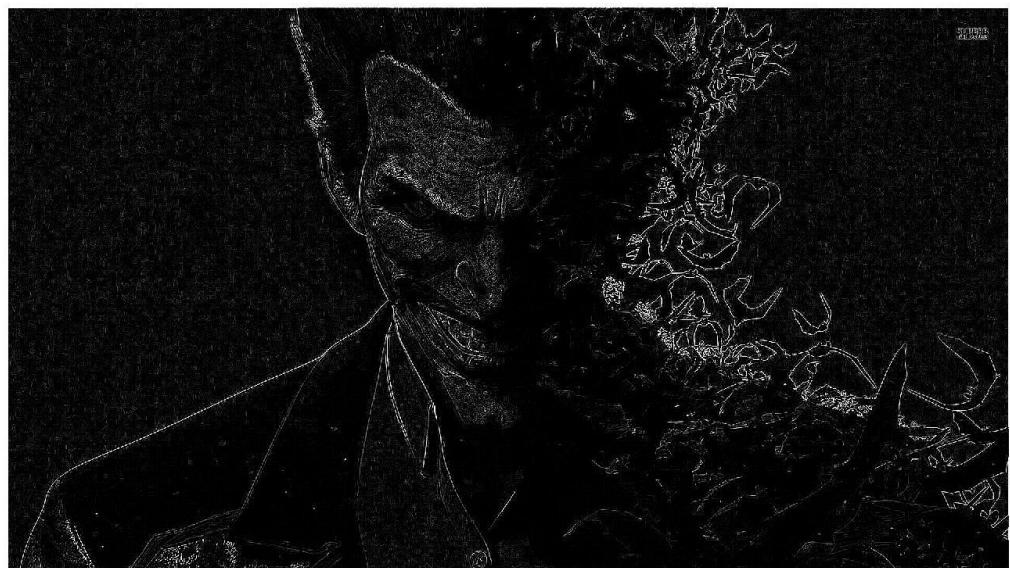


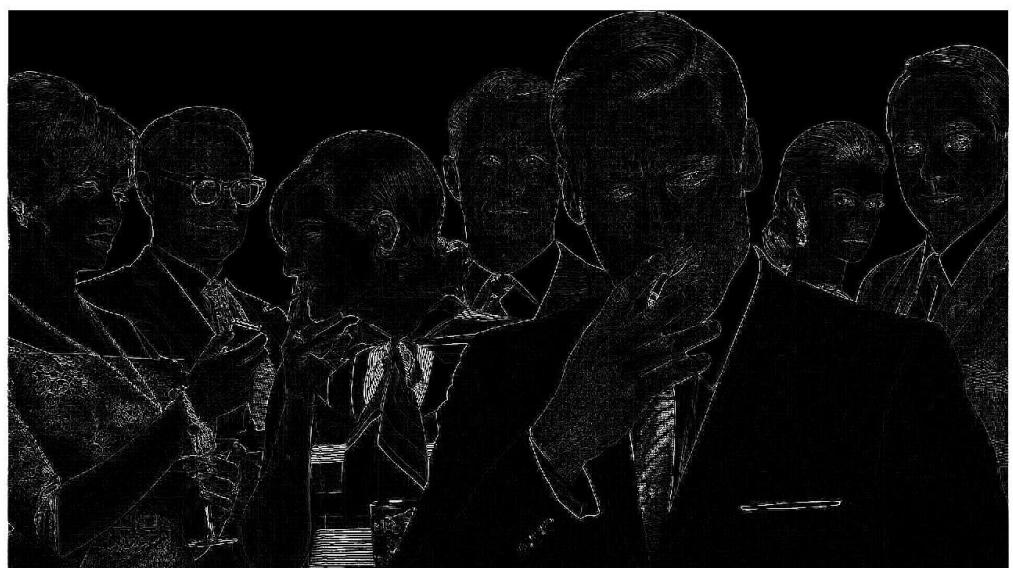


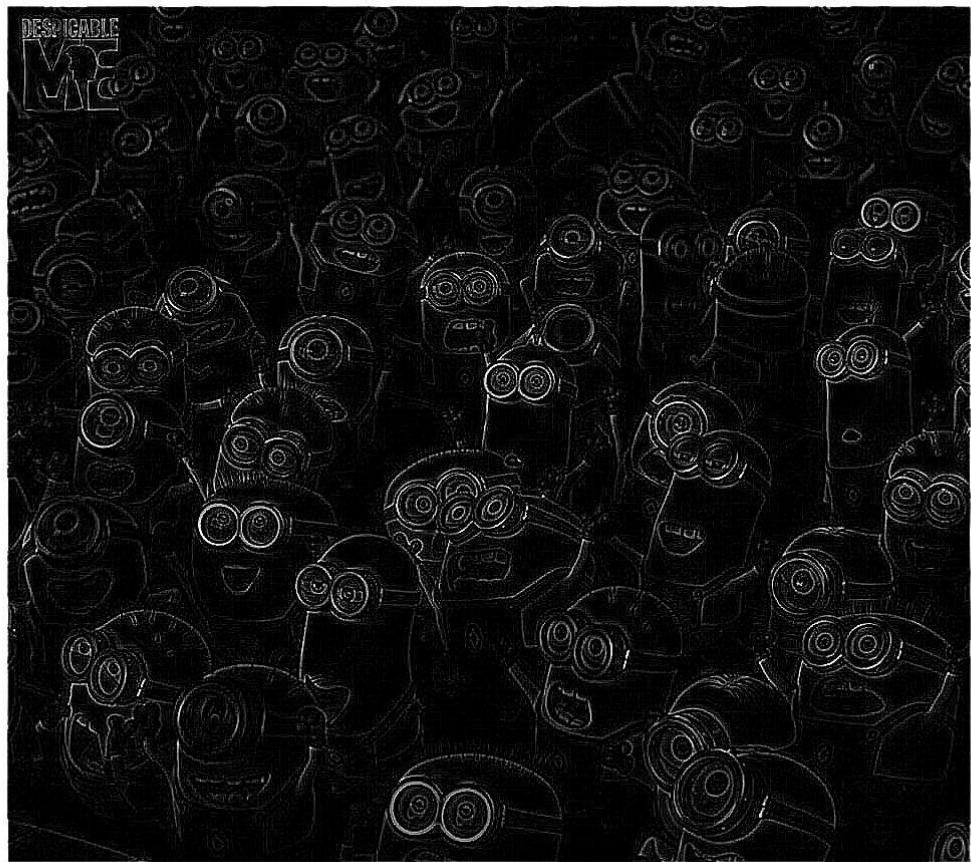
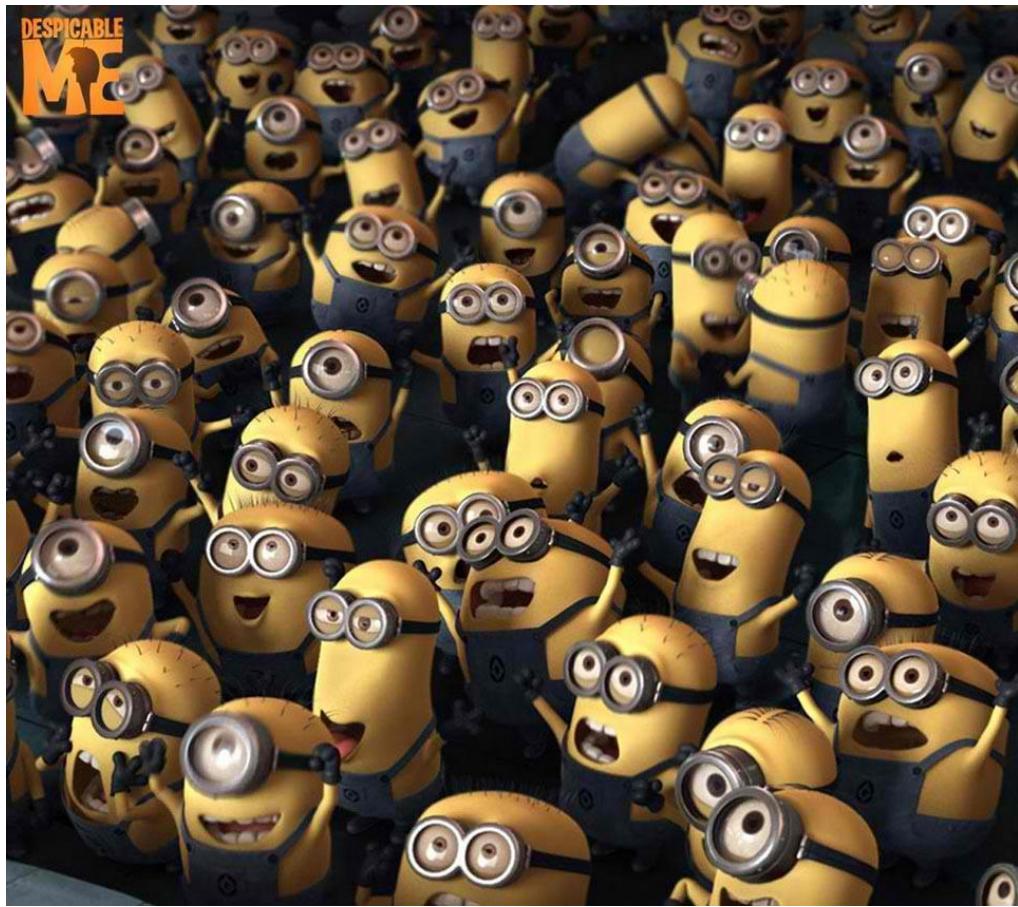




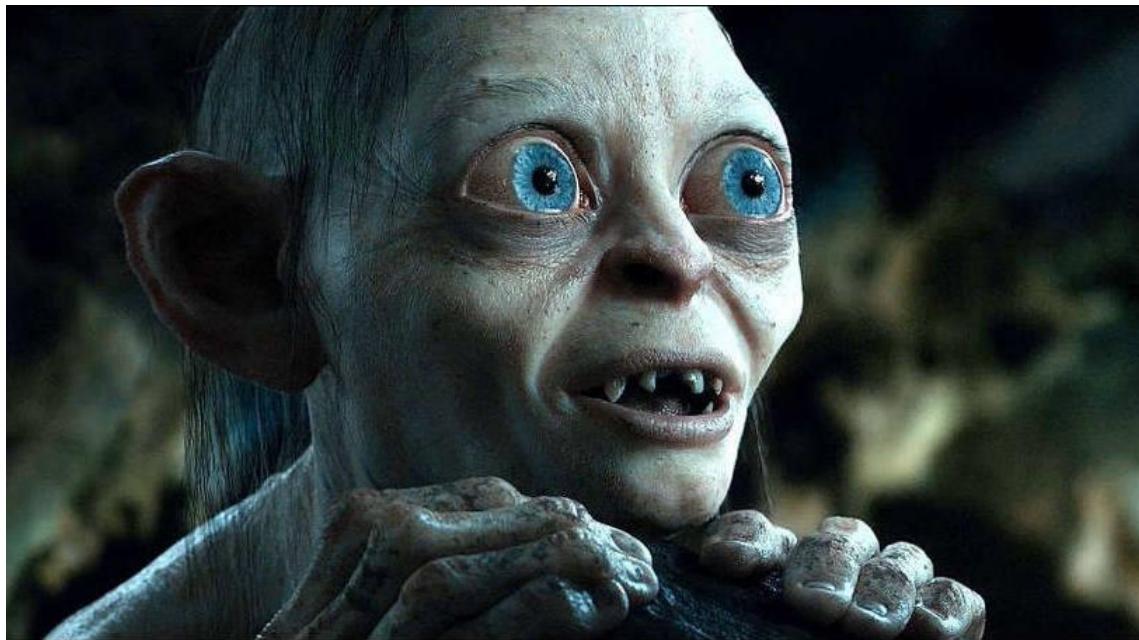












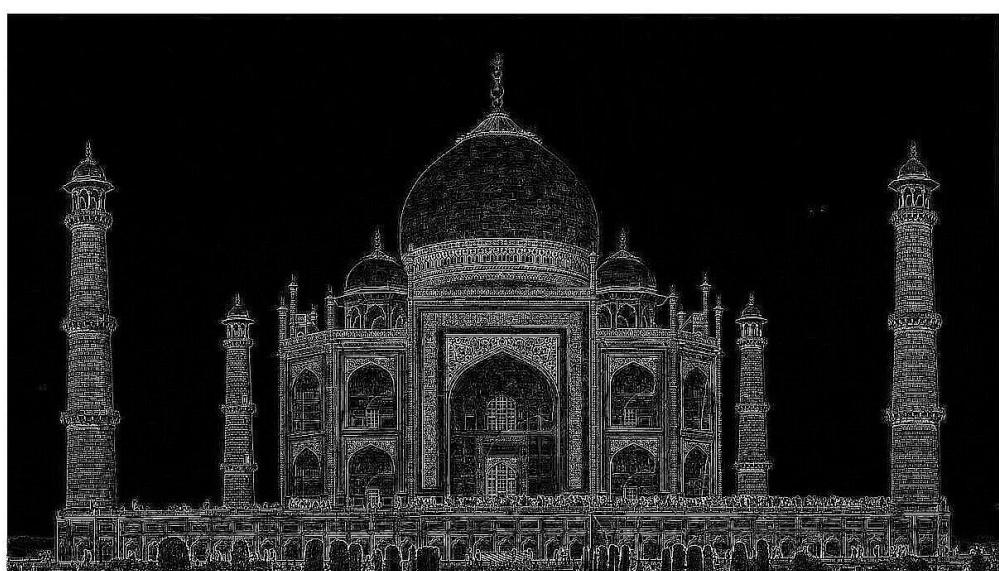








d



Canny

O Canny é um operador para detecção de contornos. Ele assegura a detecção de contornos com erros mínimos e pouco ruído. Mais, ele optimiza a detecção de contornos através da maximização do ratio de sinal/ruído do gradiente.

Começamos a implementar o canny, porém alguns problemas aconteceram e ele veio a ficar incompleto. Tentamos seguir alguns passos, e chegamos a implementar alguns desses. São:

1. Aplicar o filtro Gaussiano para suavizar a imagem e remover o ruído. Este filtro foi implementado no arquivo “canny.m”.
2. Encontrar os gradientes de intensidade da imagem. Este passo também foi implementado no arquivo “canny.m”.
3. Aplicar a “Non-Maximum suppression” para se livrar de falsas respostas a detecção de bordas.
4. Aplicar um “double threshold” para determinar bordas potenciais.
5. Finalizar a detecção de bordas, suprimindo todas as outras arestas fracas e não conectadas a arestas fortes.(Track edge by Hysteresis)