

**UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI**  
**CENTRO DE TECNOLOGIA - CT**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGE**

**Professor:** Flávio Henrique Duarte de Araújo

**Disciplina:** Visão Computacional

**Discente:** Clésio de Araújo Gonçalves

**RELATÓRIO APRESENTANDO OS RESULTADOS DOS PROCESSAMENTOS DE**  
**TODAS AS QUESTÕES**

**Questão 1:**

O procedimento básico da função de convolução desenvolvida é somar os produtos entre os coeficientes da máscara (filtro) e os valores de cinza no local específico da imagem (recorte).

O processamento sobre uma vizinhança consiste de: definir um ponto central (x, y); executar uma operação que envolva apenas os pixels da vizinhança pré-definida sobre o ponto central; considerar o resultado da operação como sendo a resposta do processo no ponto (x, y); repetir o processo para todo o ponto da imagem.

**A função de convolução desenvolvida possui três parâmetros: imagem, filtro e borda.** O algoritmo desenvolvido pode receber **imagens coloridas ou em tons de cinza**. No caso de imagens coloridas, o processamento é realizado em cada banda de cor (Red, Green, Blue), retornando o resultado em uma única imagem.

A função de convolução pode receber qualquer tipo de **filtro com dimensões ímpares, maiores ou iguais a 3 e menores que o tamanho da imagem original (em cada dimensão)**. Para a implementação da função de convolução foi utilizado 5 tipos de filtros: laplaciano (3x3), laplaciano com diagonais (3x3), suavização (3x3), gaussiano (5x5) e outro filtro com uma janela grande (21x21).

**O parâmetro borda pode receber os seguintes valores: ignore** (a borda não será processada); **espelho** (no tratamento da borda os pixels serão espelhados); **zero** (o tratamento da borda será adicionando pixels de valor zero - zero padding); **replicar** (no tratamento da borda somente os pixels das linhas extremas serão replicados). Por padrão, na função de convolução desenvolvida, o parâmetro borda recebe o valor 'ignore'.

A Figura 1 exibe a imagem original utilizada na função de convolução:



Figura 1: imagem original utilizada nos algoritmos

Conforme apresentado, foram aplicados cinco filtros (kernels) na imagem original utilizando a função de convolução. Esses filtros foram aplicados com o objetivo de reduzir transições abruptas na intensidade; reduzir ruído; suavizar falsos contornos, resultantes de uma quantização com número insuficiente de níveis de cinza; reduzir detalhes irrelevantes na imagem (regiões menores que o tamanho da máscara).

- a) Foi aplicado o filtro laplaciano de segunda derivada para realce de imagem ( $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ ) com bordas ignoradas. O filtro Laplaciano enfatiza regiões de descontinuidade (borda) e ameniza regiões de variação lenta de níveis de intensidade (rampa gradual). Ao fazer a operação de soma houve uma melhora nos detalhes mais finos das imagens (realce), tanto na horizontal como na vertical.

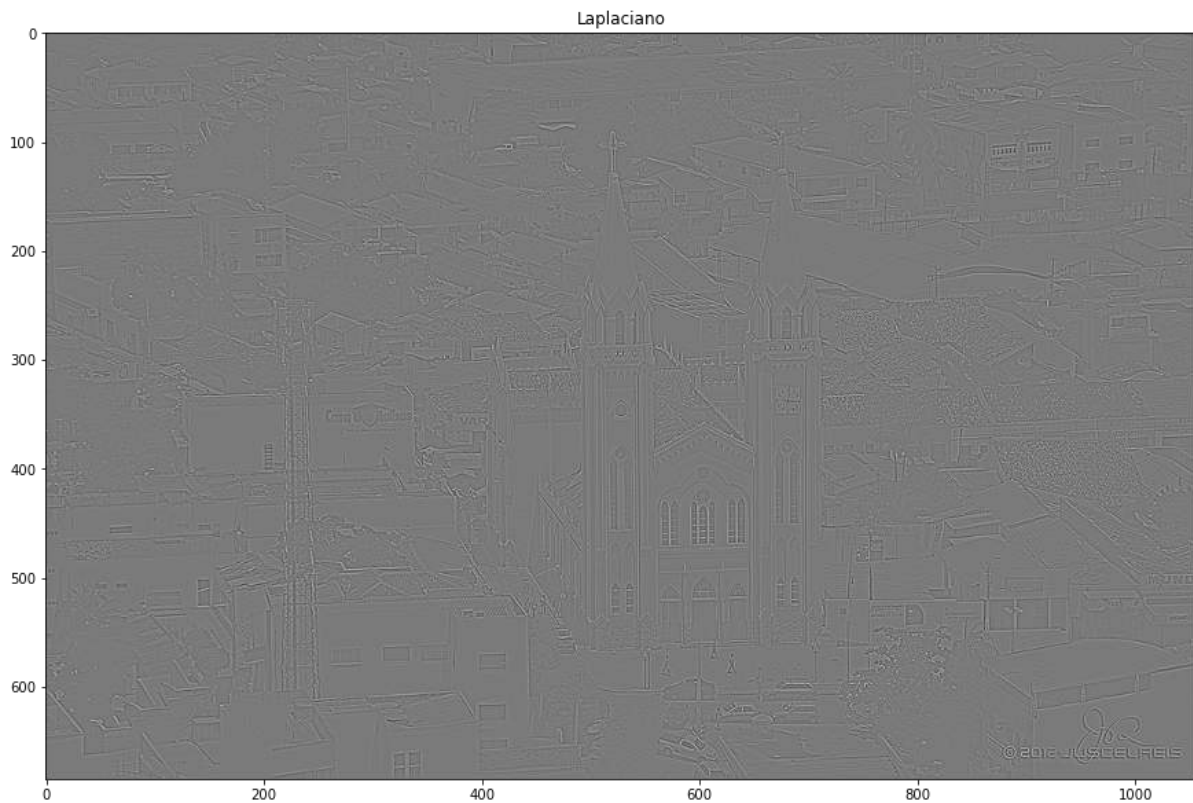


Figura 2: Imagem resultante do filtro laplaciano (realce na horizontal e vertical)

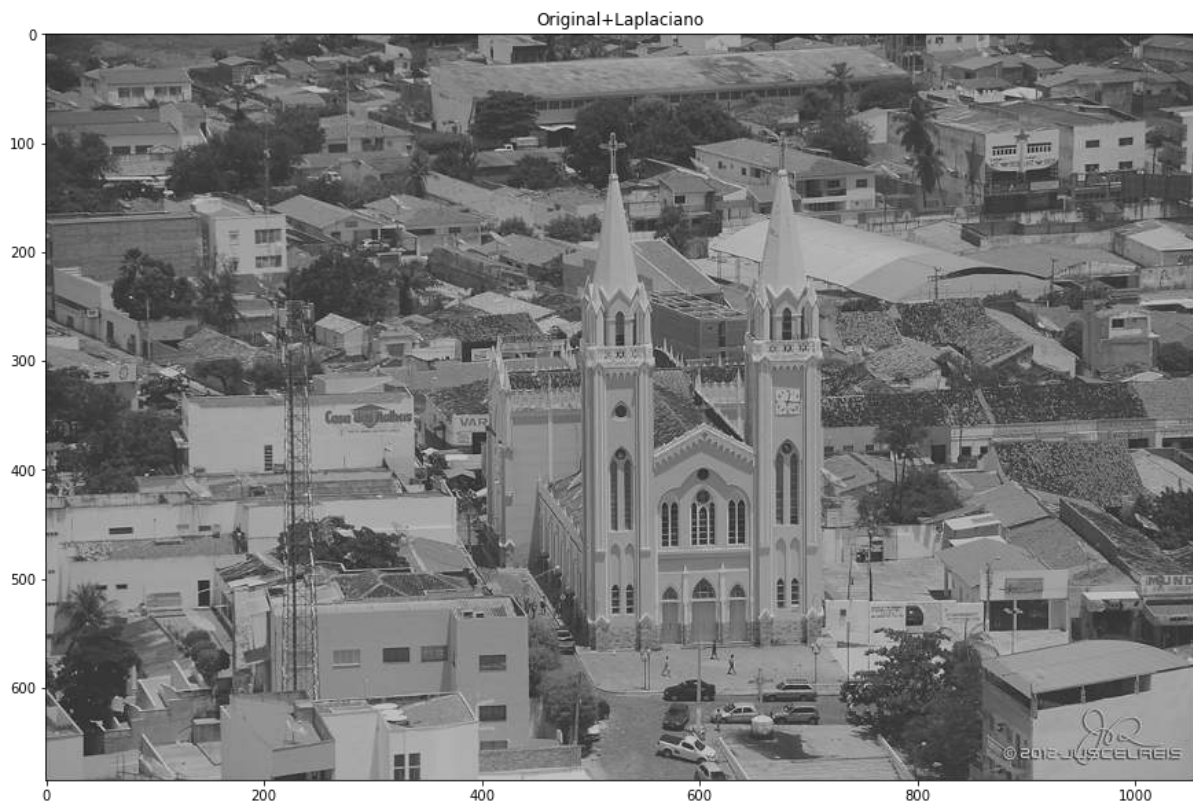


Figura 3: Imagem original somada com filtro laplaciano da Figura 2

- b) Foi aplicado o filtro laplaciano de segunda derivada para realce de imagem ( $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ ) com bordas ignoradas. Como o pixel central possui valor positivo, então o valor da constante  $c = -1$ . Ao fazer a operação de soma houve um realce nos detalhes horizontais, verticais e diagonais da imagem. A imagem resultante houve melhor realce em comparação ao filtro laplaciano anterior por acrescentar o realce no eixo diagonal da imagem.

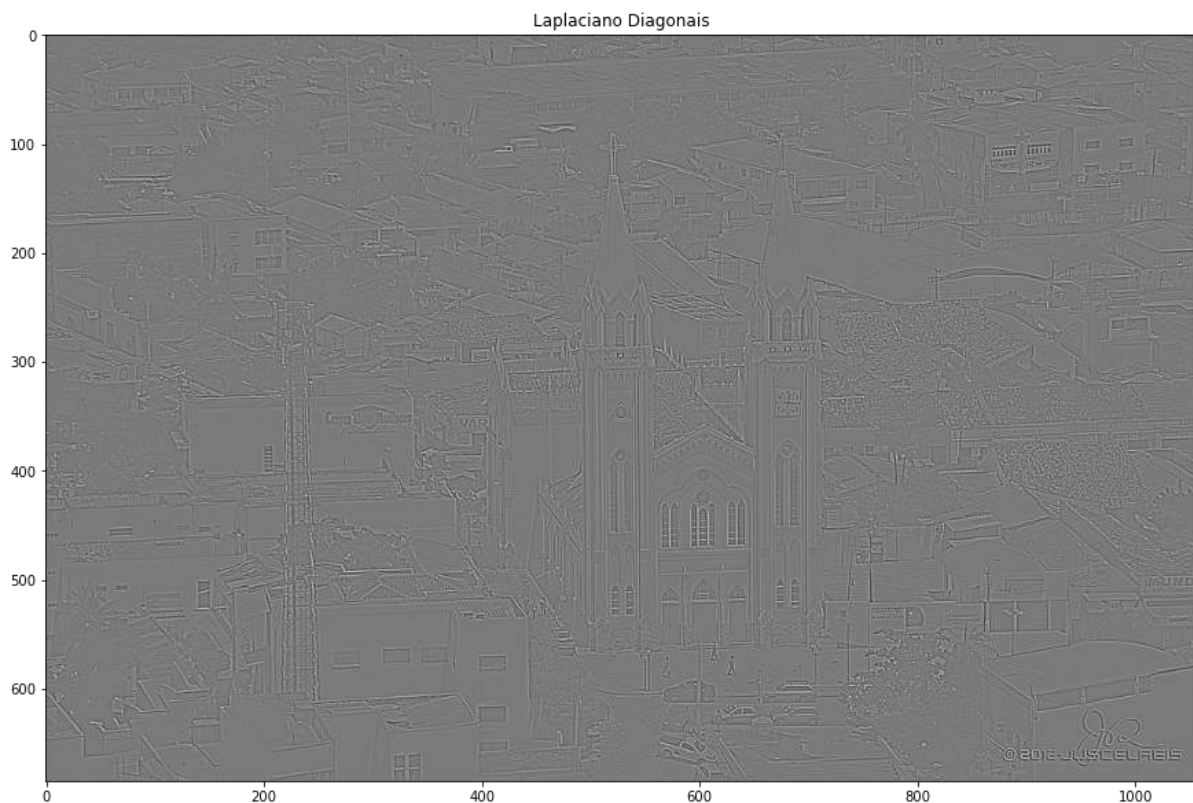


Figura 4: Imagem resultante do filtro laplaciano (realce na horizontal, vertical e diagonal)

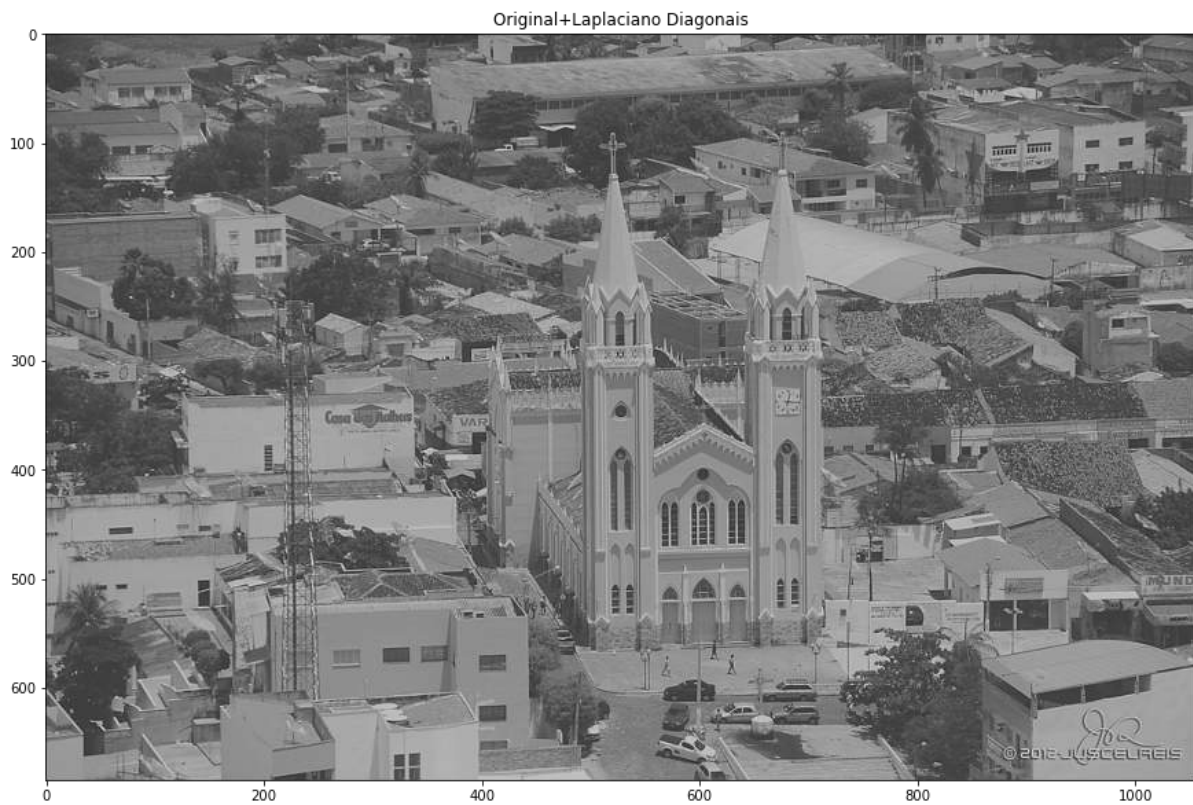


Figura 5: Imagem Original somada a laplaciana com realce horizontal, vertical e diagonal

- c) Foi aplicado o filtro de suavização  $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$  com bordas espelhadas. O filtro percorreu toda a imagem realizando a suavização dessa imagem (diminuindo a diferença dos valores de pixels de borda da imagem). Utilizou-se a borda espelhada, ou seja, as duas primeiras linhas obtiveram os valores das duas últimas linhas e as duas últimas linhas receberam os valores dos pixels das duas primeiras linhas (esse mesmo procedimento também foi realizado nas duas primeiras e duas últimas colunas).



Figura 6: Imagem suavizada

- d) Foi aplicado o filtro gaussiano  $\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$  com bordas zero. O filtro gaussiano percorreu toda a imagem realizando a suavização da imagem (diminuindo a diferença dos valores de pixels de borda da imagem). Utilizou-se a borda com valores zero, ou seja, as duas primeiras e duas últimas linhas foram preenchidas com valores zero. A imagem perde um pouco de detalhes nesse processo de convolução com filtro aplicado.



Figura 7: Imagem aplicada a convolução com filtro gaussiano



- e) Foi aplicado o filtro com janela 21x21 com valores 1 e aplicando bordas replicadas. O filtro percorreu toda a imagem realizando o realce da imagem. Utilizou-se a borda com valores replicados, ou seja, as duas primeiras e duas últimas linhas foram preenchidas com valores das duas linhas adjacentes. Por apresentar uma janela grande de filtro, a imagem perdeu detalhes nesse processo de convolução com filtro aplicado.

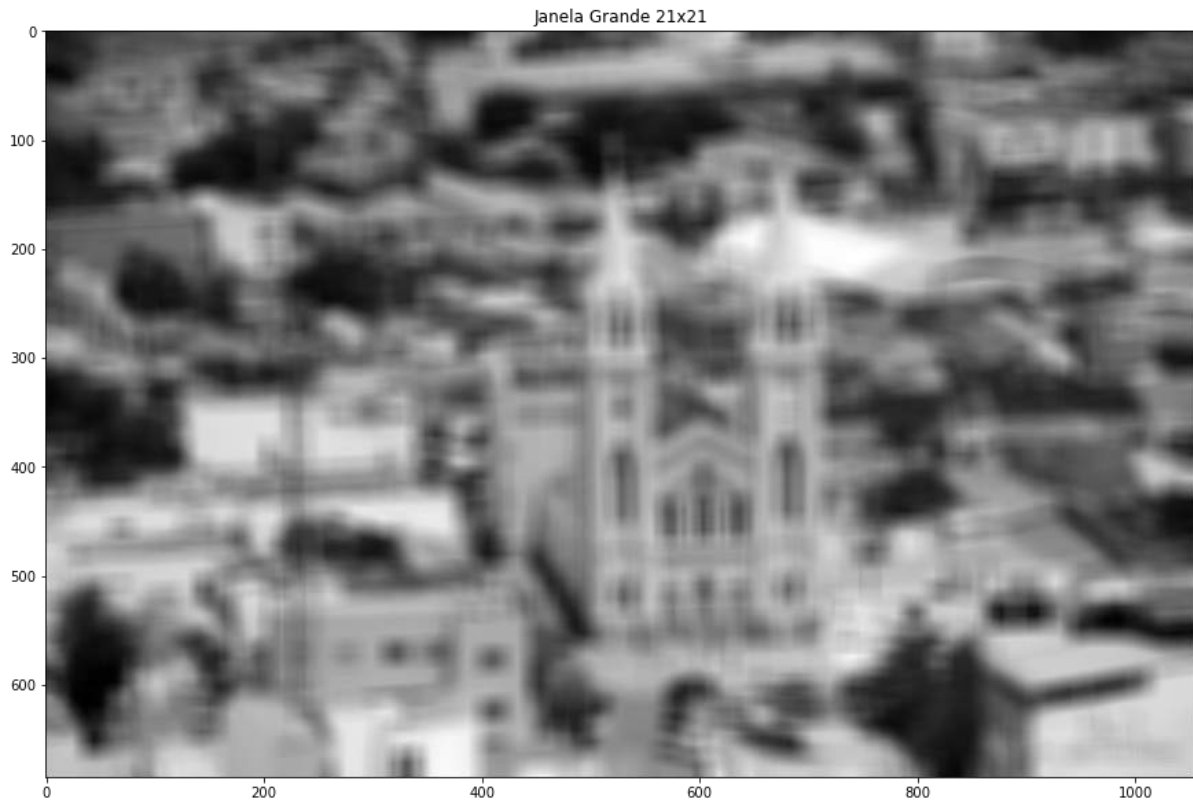


Figura 8: Imagem aplicando o filtro de janela 21x21

## Questão 2:

Filtros não lineares são baseados em operações sobre uma vizinhança. O algoritmo desenvolvido pode receber **imagens coloridas ou em tons de cinza**. No caso de imagens coloridas, o processamento é realizado em cada banda de cor (Red, Green, Blue), retornando o resultado em uma única imagem. Para os cálculos desses filtros, as bordas da imagem foram ignoradas, não sendo processadas.

### a) Mediana

O algoritmo desenvolvido substitui a intensidade de cada pixel pela mediana das intensidades na sua vizinhança baseada em uma janela pré-definida (de dimensões ímpares e menor que a imagem original). Com isso, a imagem resultante apresentou uma redução nos níveis de ruído em comparação a imagem original apresentada na Figura 1. Foi aplicada uma janela 3x3 para cálculo da mediana, em seguida o centro do recorte da imagem original teve seu valor substituído pela mediana da janela calculada.

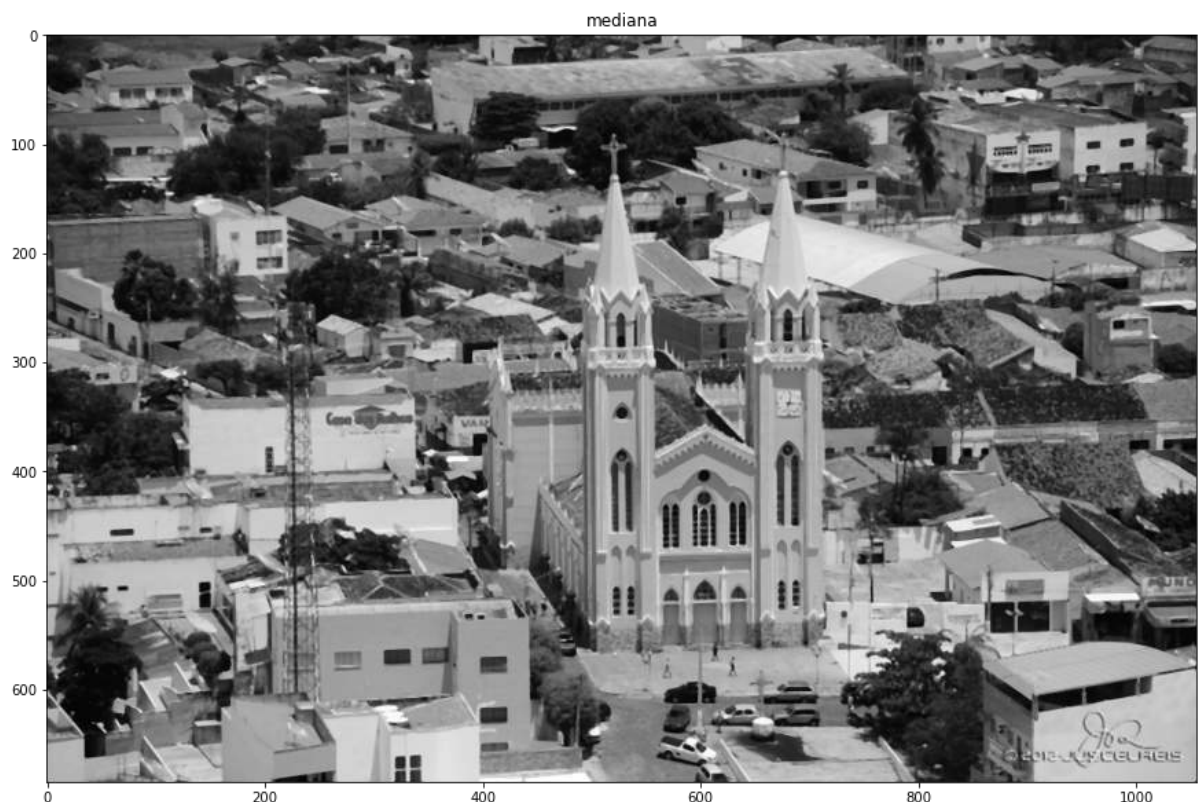


Figura 9: Imagem resultante da aplicação do filtro da mediana

## b) Moda

O algoritmo desenvolvido substitui a intensidade de cada pixel pela intensidade que ocorre com maior frequência na sua vizinhança. Com isso, a imagem resultante apresentou uma distorção das bordas em comparação a imagem original apresentada na Figura 1. Foi aplicada uma janela 3x3 para cálculo da moda, em seguida o centro do recorte da imagem original teve seu valor substituído pela moda da janela calculada.



Figura 10: Imagem resultante da aplicação do filtro da Moda

### c) Máximo

O algoritmo desenvolvido substitui a intensidade de cada pixel pela maior intensidade na sua vizinhança. Com isso, a imagem resultante apresentou aumento das regiões claras, dominando as regiões escuras em comparação a imagem original apresentada na Figura 1. Foi aplicada uma janela 3x3 para cálculo do valor máximo, em seguida o centro do recorte da imagem original teve seu valor substituído pelo valor máximo da janela calculada.

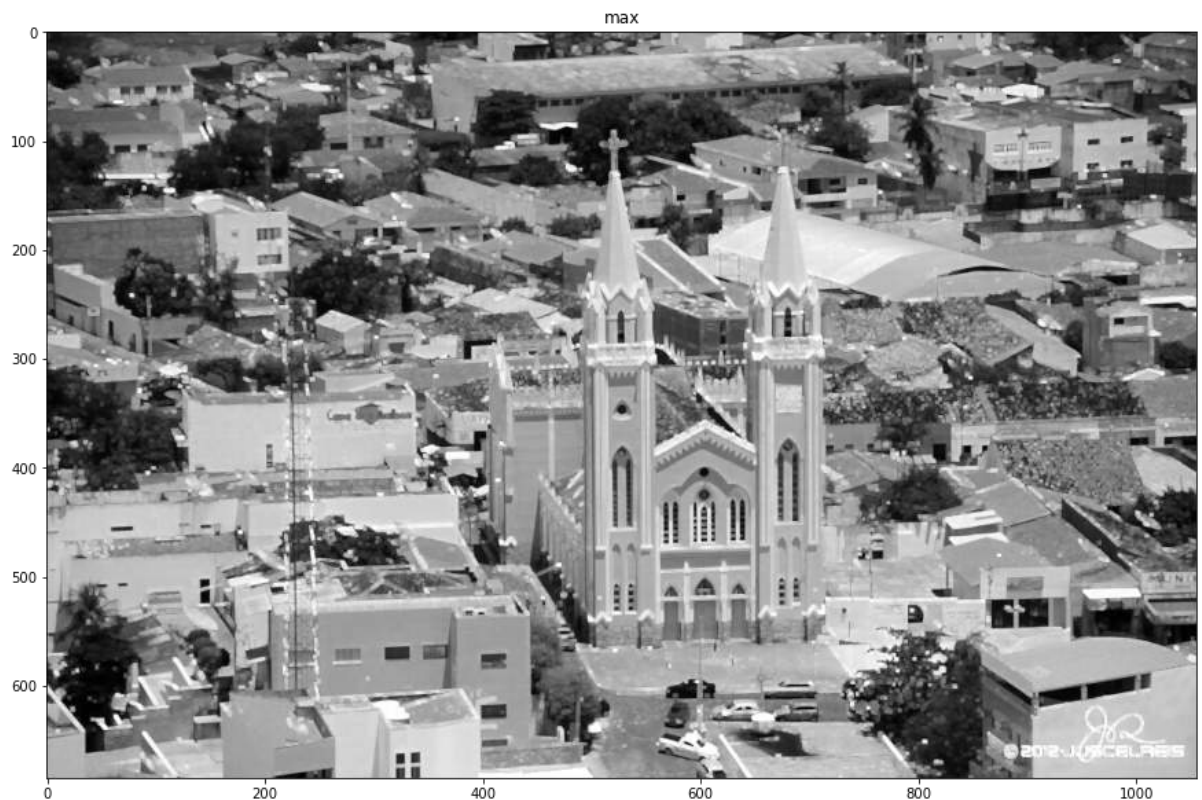


Figura 11: Imagem resultante da aplicação do filtro da janela máxima

#### d) Mínimo

O algoritmo desenvolvido substitui a intensidade de cada pixel pela menor intensidade na sua vizinhança. Com isso, a imagem resultante apresentou aumento das regiões escuras, dominando as regiões claras em comparação a imagem original apresentada na Figura 1. Foi aplicada uma janela 3x3 para cálculo do valor mínimo, em seguida o centro do recorte da imagem original teve seu valor substituído pelo valor mínimo da janela calculada.



Figura 12: Imagem resultante da aplicação do filtro da janela mínima

### Questão 3:

Houve um realce da imagem, melhorando os detalhes mais finos da imagem, tanto na horizontal como na vertical (por conta do filtro aplicado:  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ ), conforme Figura 15. O filtro Laplaciano enfatiza regiões de descontinuidade (borda) e ameniza regiões de variação lenta de níveis de intensidade (rampa gradual). Como o pixel central do filtro aplicado possui valor positivo, então o valor da constante  $c = 1$ , conforme equação abaixo.

$$g(x, y) = f(x, y) + c \left[ \nabla^2 f(x, y) \right]$$

Equação 1: Filtro Espacial de Aguçamento (Laplaciano)

onde  $f(x,y)$  representa a imagem original e a derivada segunda de  $f(x,y)$  representa o filtro laplaciano (neste exemplo o filtro  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ )

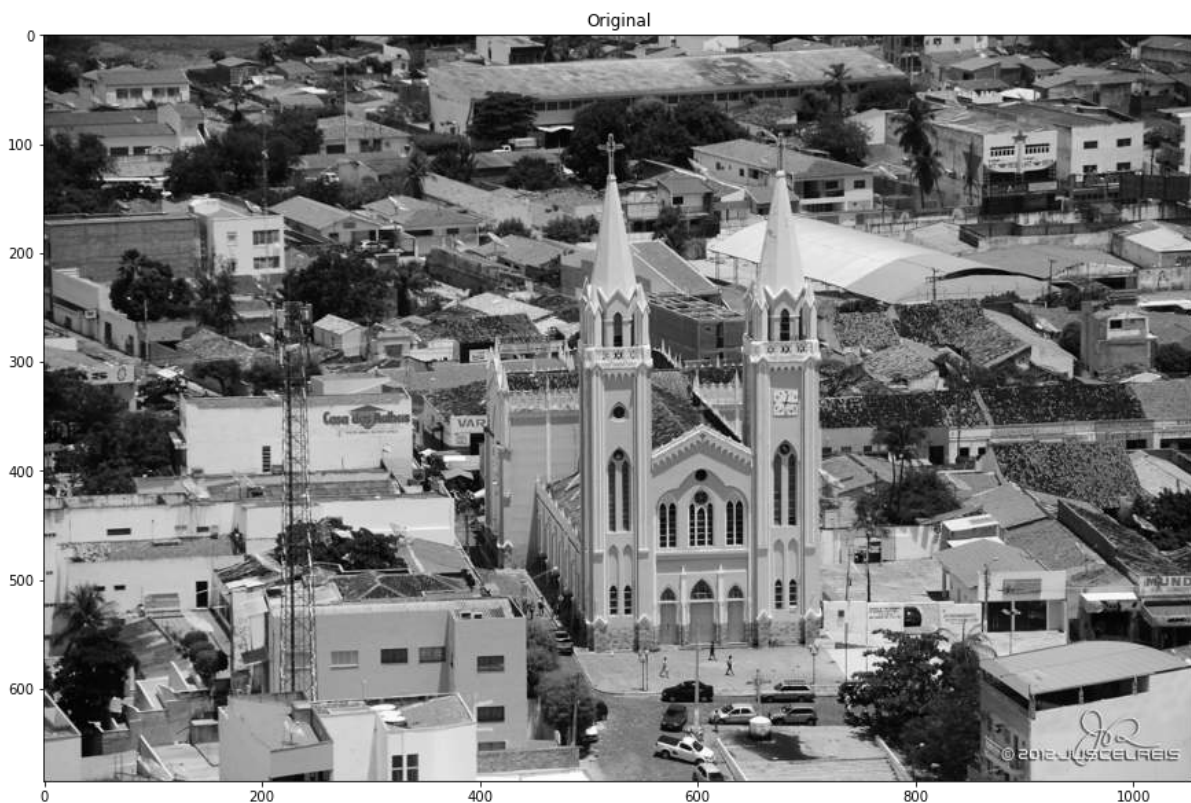


Figura 13: Imagem original



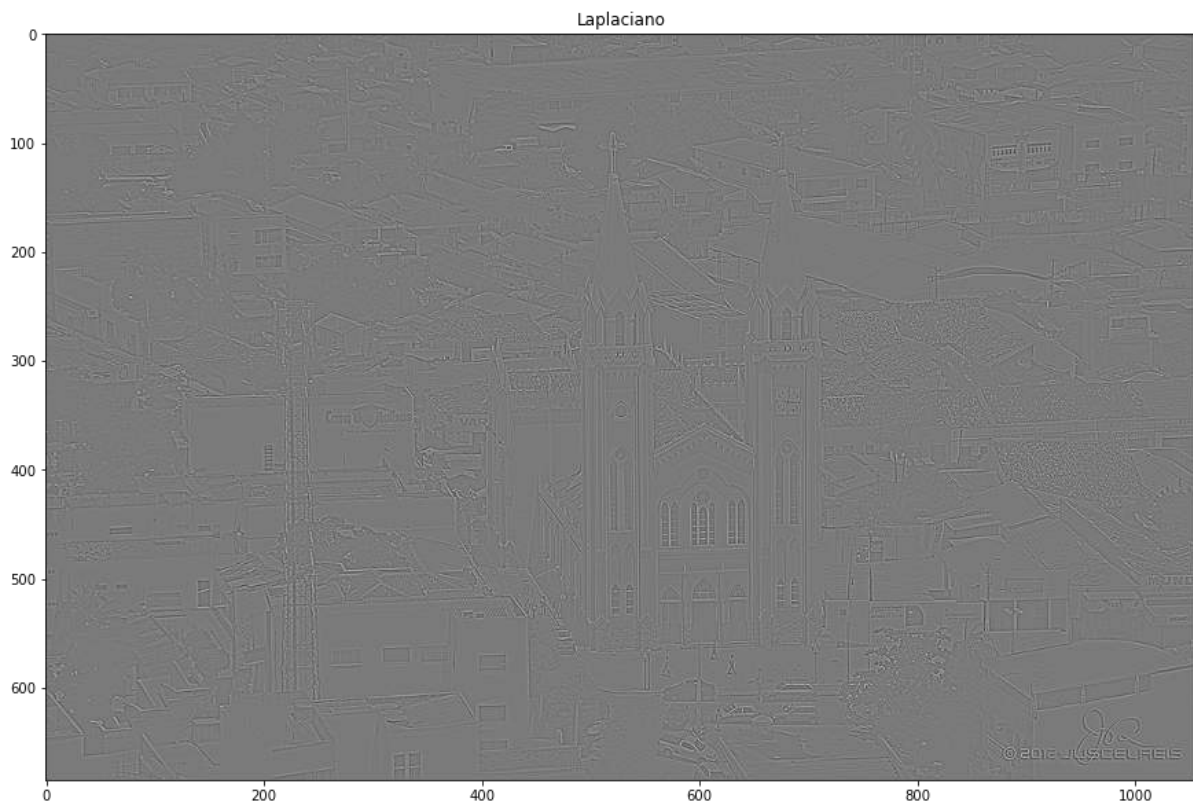


Figura 14: Filtro laplaciano 3x3

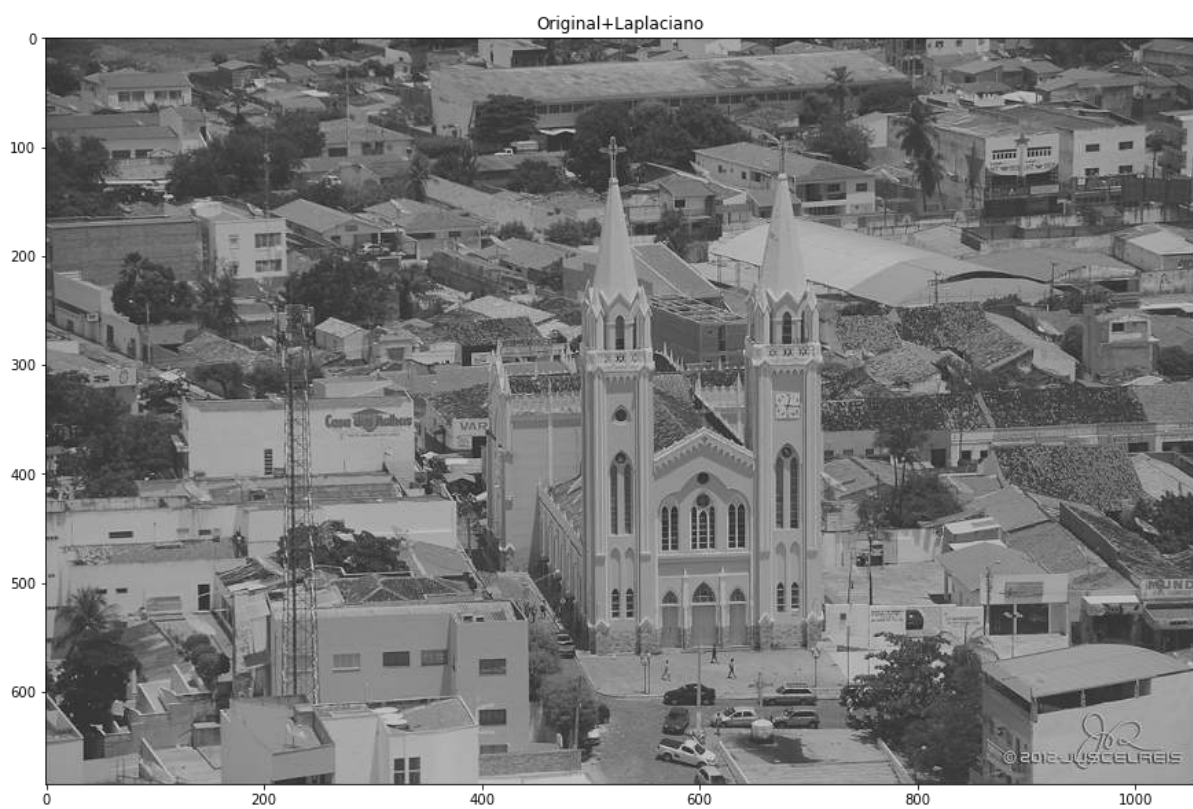


Figura 15: Imagem original somada com o filtro laplaciano

#### Questão 4:

A Máscara de Nitidez e Filtragem high-boost realça a imagem ao subtrair uma versão suavizada de uma imagem da sua original. O algoritmo desenvolvido consiste nos seguintes passos:

1. aplicar o filtro  $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$  no algoritmo de convolução da questão 1 - Figura 17;
2. subtrair a imagem de convolução da original (resultado é chamado de máscara) - Figura 18;
3. somar a máscara à imagem original, variando o parâmetro  $k$  ( $\text{high\_boost} = \text{img} + k * \text{máscara}$ ).
  - a. para  $k = 1$ , a imagem é somada a imagem original - Figura 20
  - b. para  $k < 1$ , reduz a contribuição da máscara - Figura 19
  - c. para  $k > 1$ , o processo é conhecido como filtragem high-boost (filtro com ênfase) - Figura 21

Assim, com o parâmetro  $k$  é possível determinar o peso da máscara de nitidez no realce da imagem, utilizando a filtragem high-boost.



Figura 16: Imagem original





Figura 17: Imagem aplicando o filtro de convolução da questão 1

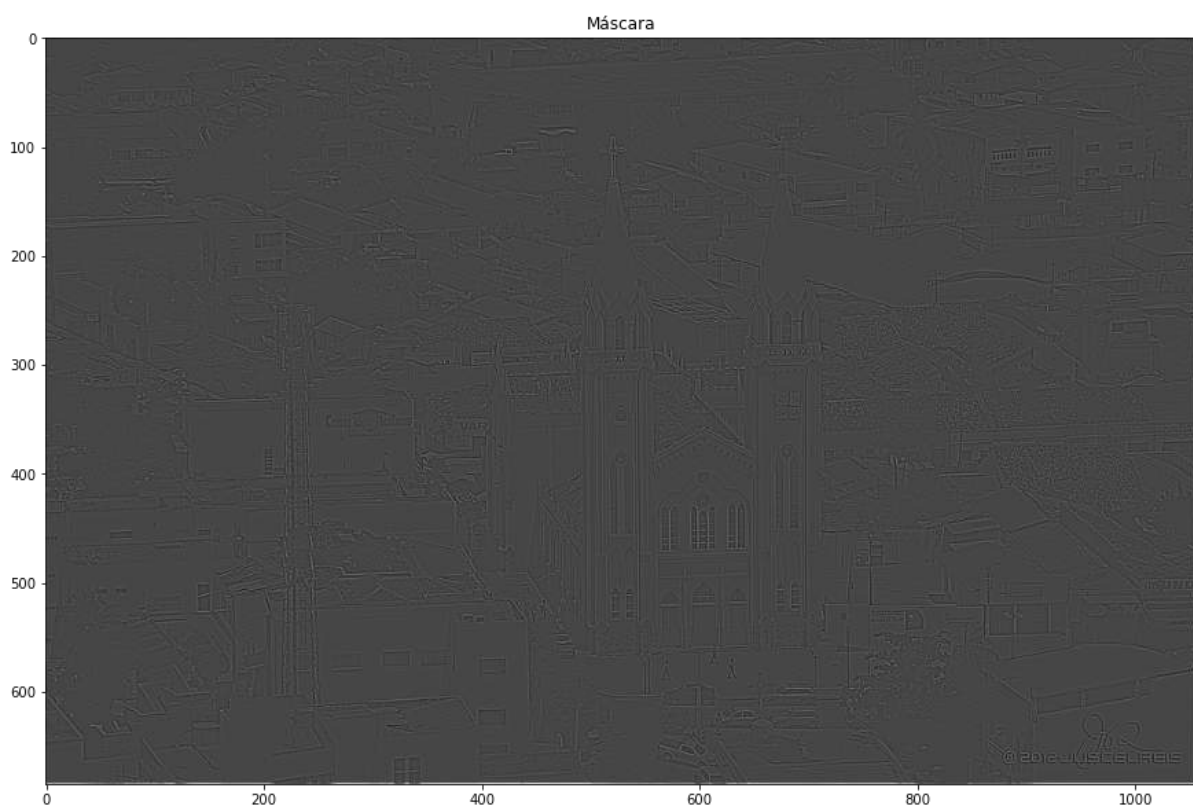


Figura 18: Imagem aplicando a máscara de nitidez

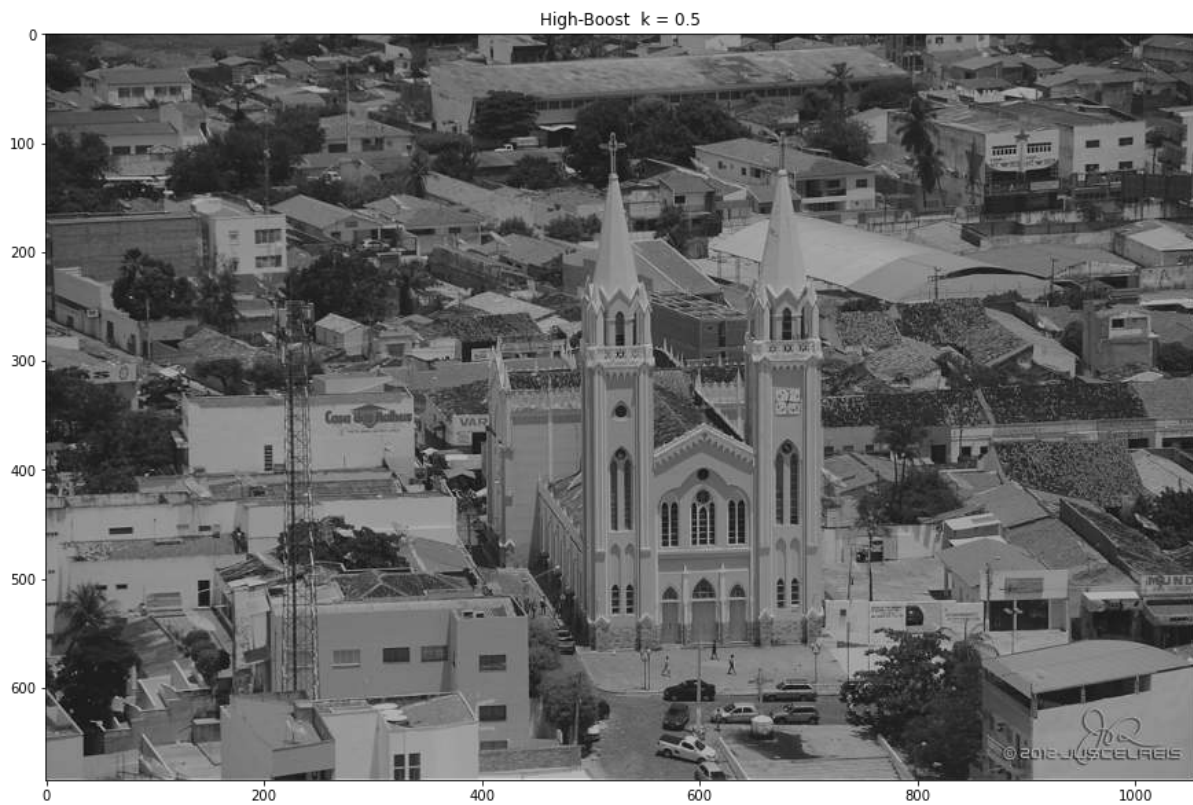


Figura 19: Imagem aplicada a filtragem high-boost com  $k = 0.5$

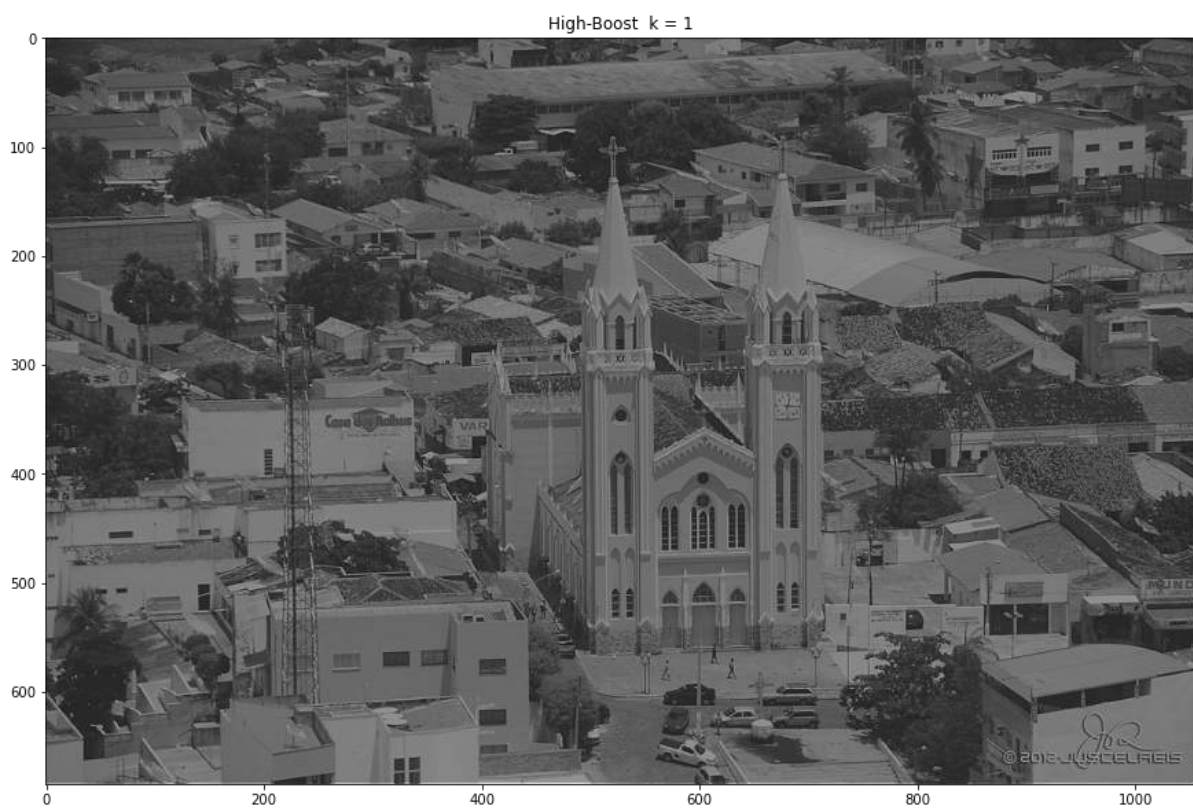


Figura 20: Imagem aplicada a filtragem high-boost com  $k = 1$

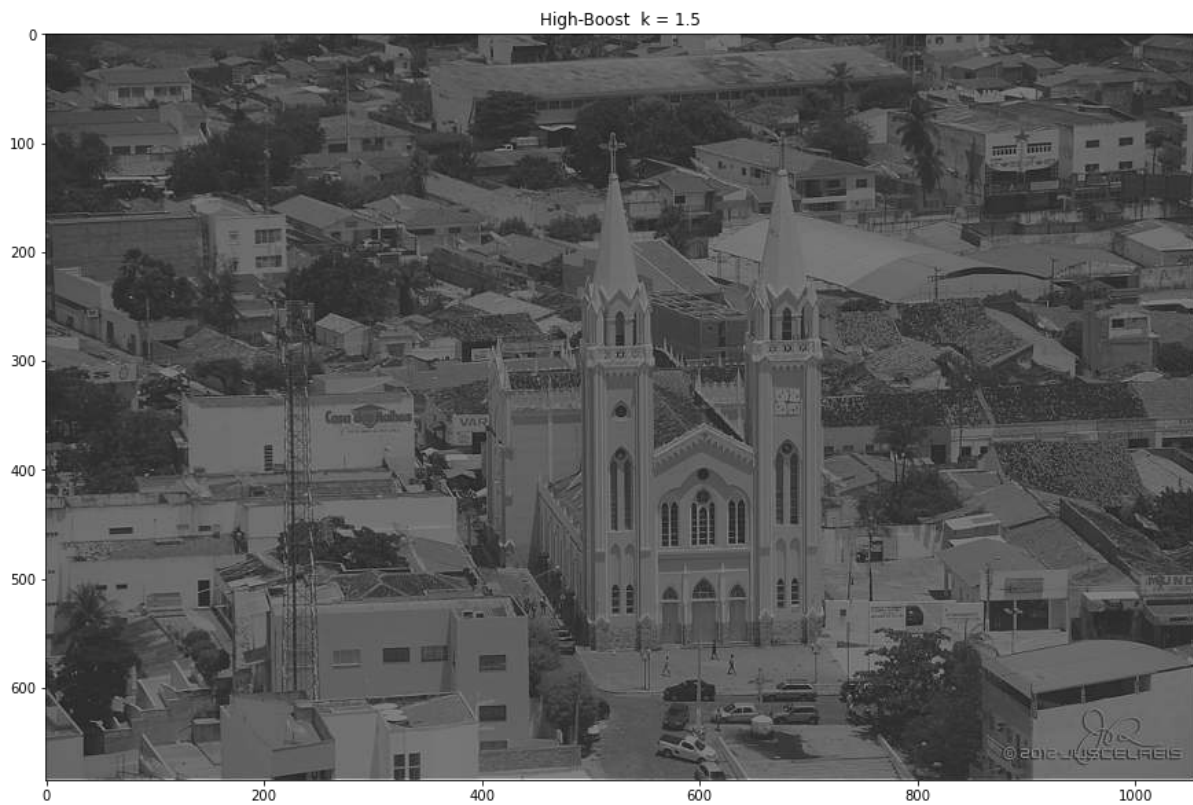


Figura 21: Imagem aplicada a filtragem high-boost com  $k = 1.5$

### Questão 5:

A adição de ruído foi realizada com a função `random_noise` do `skimage`. Utilizou-se o ruído ‘pepper’ (pimenta) que substitui pixels aleatórios por da imagem por zero 0, conforme Figura 22.

Para ambos os filtros, o algoritmo desenvolvido substitui a intensidade de cada pixel da imagem ruidosa pela média ou mediana das intensidades na sua vizinhança baseada em uma janela pré-definida. Além disso, para ambos os filtros foram aplicados uma janela 3x3 para cálculo da média ou mediana, em seguida o centro do recorte da imagem ruidosa teve seu valor substituído pela média ou mediana da janela calculada.

**Ao aplicar o filtro da média**, a imagem resultante apresentou um aumento do ruído na imagem (não resolvendo o problema), conforme Figura 23.

**Ao aplicar o filtro da mediana**, a imagem resultante apresentou uma redução drástica nos níveis de ruído em comparação a imagem ruidosa, conforme Figura 24.

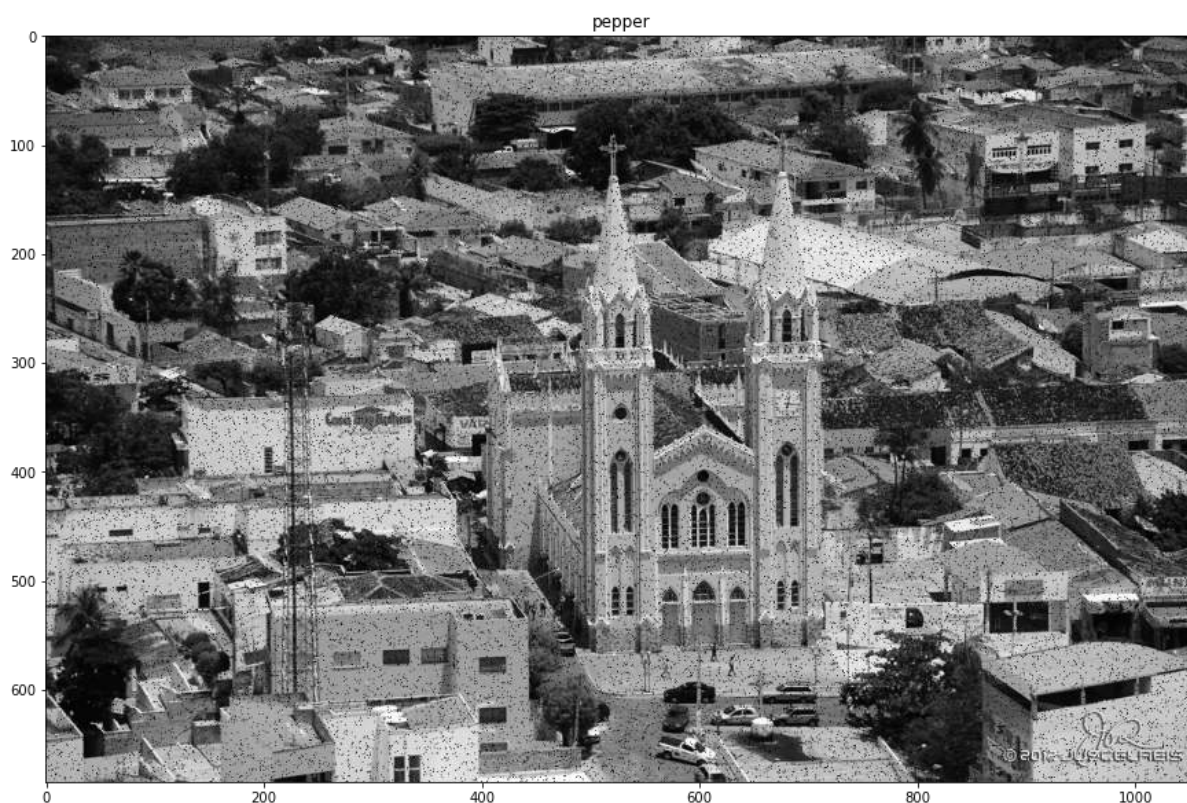


Figura 22: Imagem com ruído pimenta (‘pepper’)

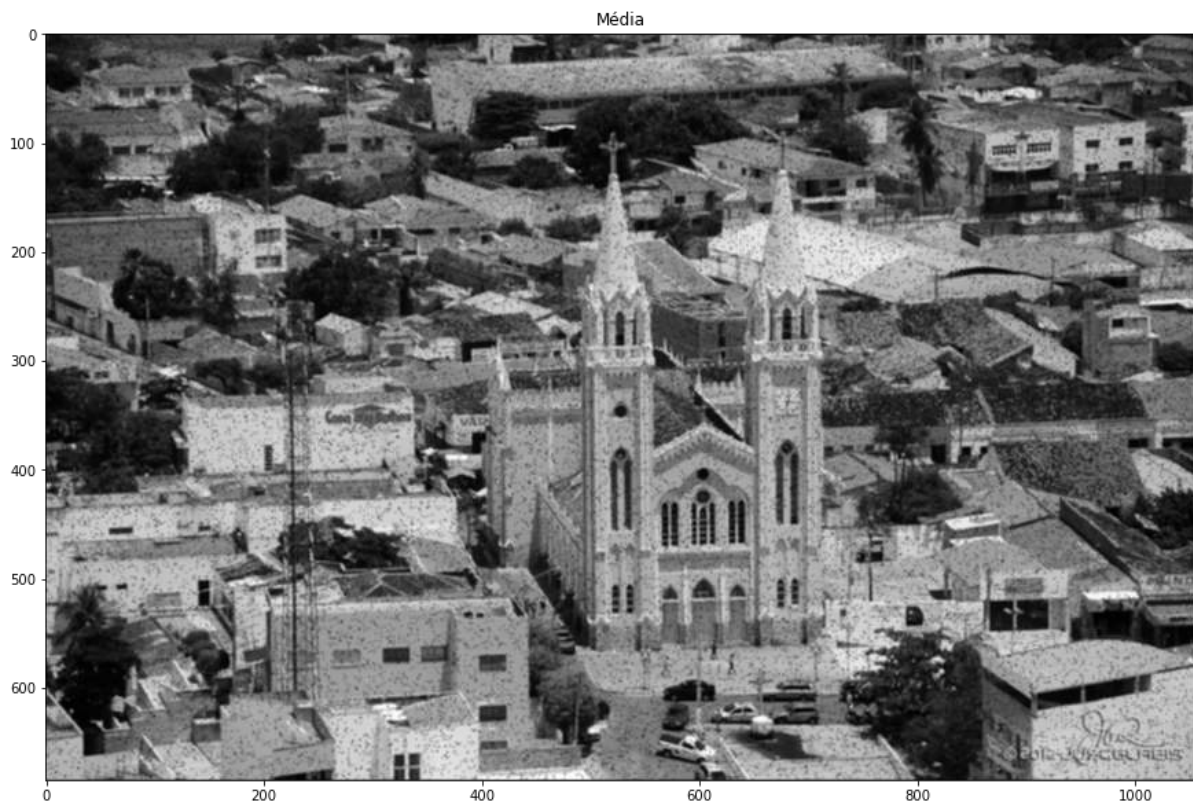


Figura 23: Imagem com aplicação do filtro da média

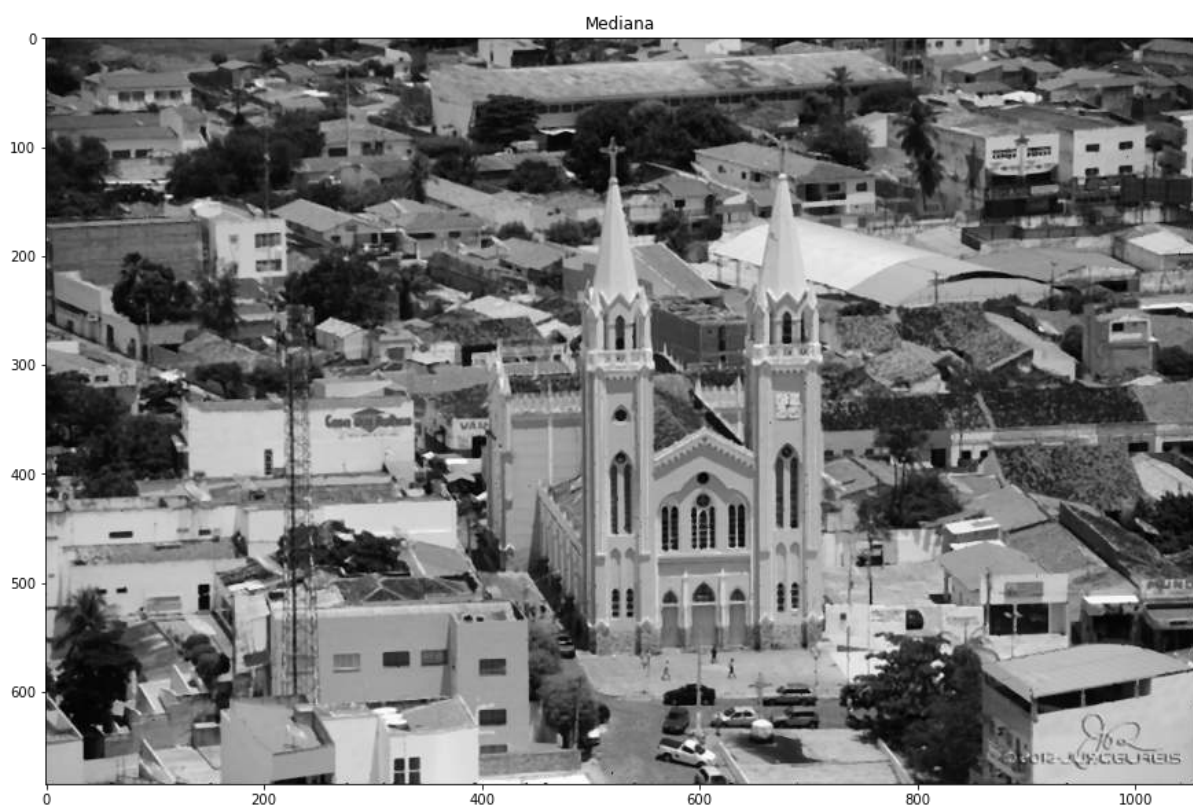


Figura 24: Imagem realçada com aplicação do filtro da mediana