Lab 1

Image Cartoonifier

Introduction :

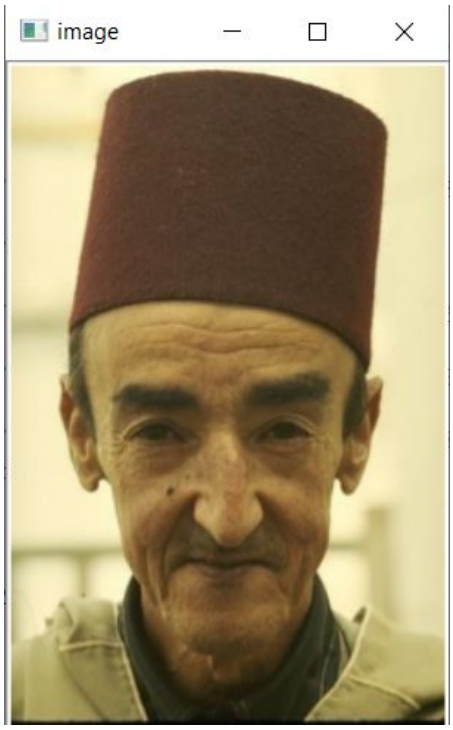
Our project is about images into its cartoonized version. To be able to do this process to we need two main steps: detecting edges in these photos and then combine them with a smoothed version of the original image.

Steps:

1. First we used OpenCV library to read images cv2.imread() and also used for displaying them cv2.imshow().
2. We used cv2.cvtColor() to convert the image from RGB to grayscale.
3. We applied a median filter of size (7*7) to reduce noise using function cv2.medianBlur()
4. Then we applied laplacian filter to detect edges of the grayscale image by function cv2.Laplacian() and we tried different kernel sizes:
    a. For example k =5 tends to detect more details and more edges were obvious , k=3 detected stronger borders and k=1 barely detected any edges.
5. We applied threshold to the output of laplacian filter to make edges either white or black with a threshold value 125 using function cv2.threshold()
6. We applied bilateral filter to the original image to smooth flat regions while keeping edges sharp. The image tends to be smoother as we increased the sigmacolor and sigmaspace parameters. We used the function cv2.bilateralFilter() and applied it multiple times to get more cartoonized effect.
7. Last, we needed to combine the output of the bilateral filter and the pervious detected edges in one image to get the cartoonized image. We used cv2.bitwise_and() to combine them

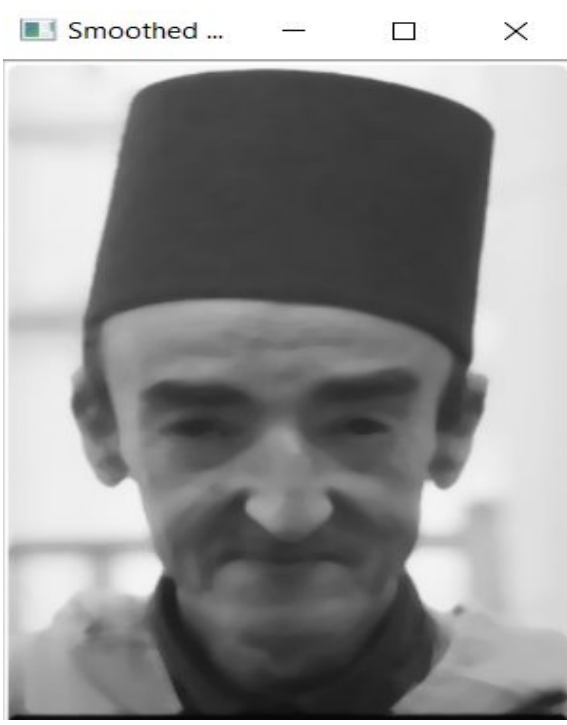
Samples:

Original image



Grayscale image

Smoothed Grayscale



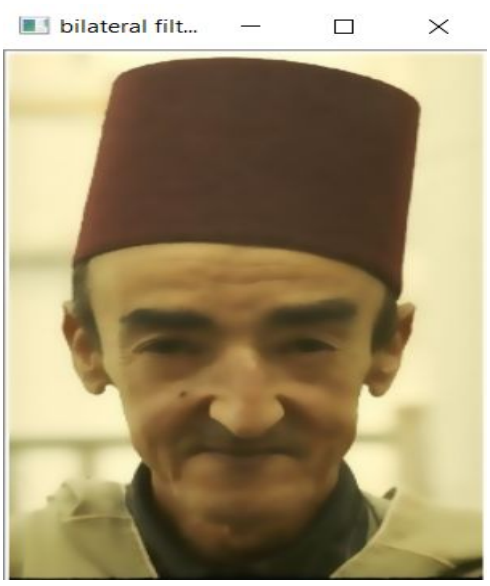
Laplacian filter k=5

After Thresholding with 125

Detected Edges

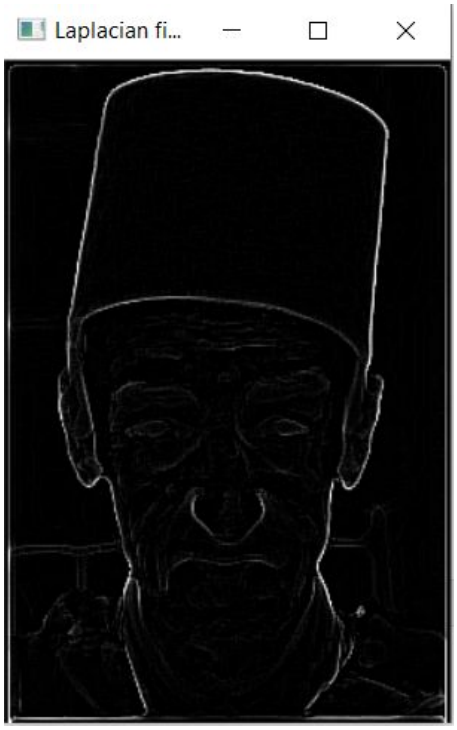
Bilateral Filter

Bilateral filter with other parameters

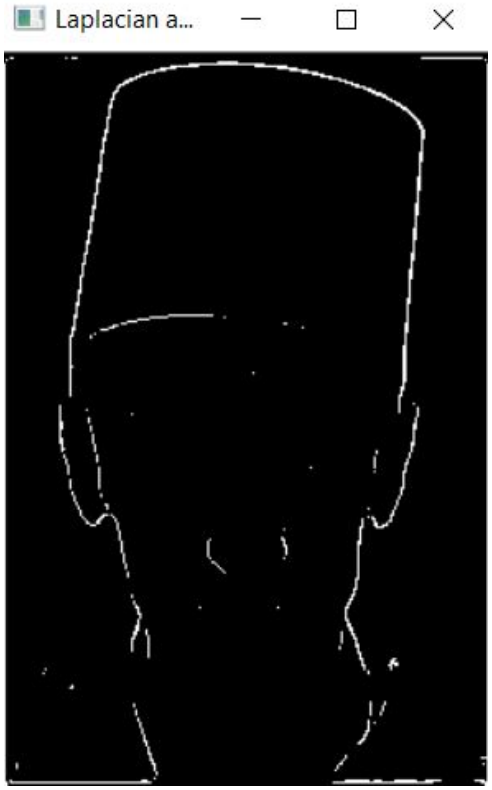


Cartoonized image

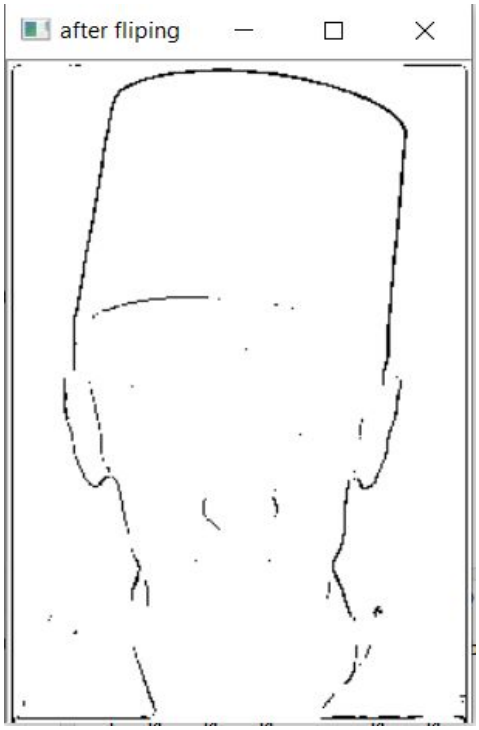
Laplacian with k=3



Thresholding

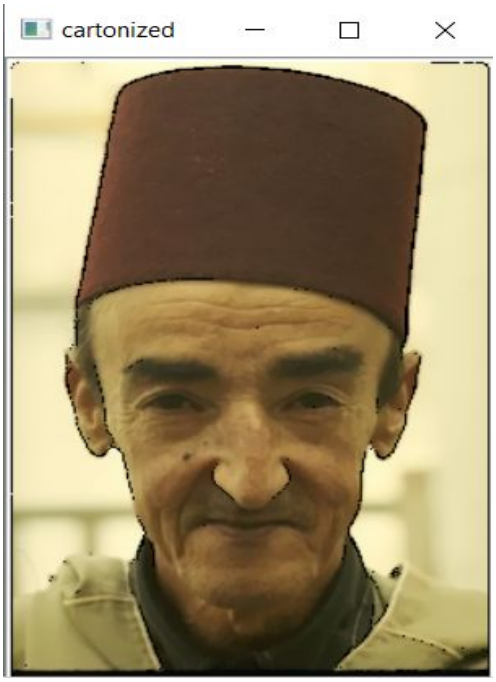


Detected edges

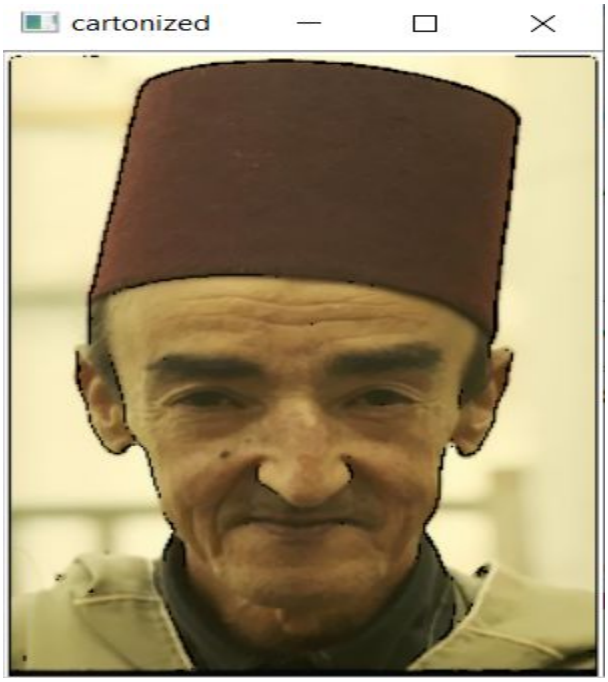


K=1

Cartoonized for k=1



Cartoonized for k=3



Other Samples :