

Assignment 1

Computer Vision

Image Cartoonifier

<i>Esraa Wael El-Hawash</i>	<i>ID:3959</i>
<i>Guehad Mohamed</i>	<i>ID:3861</i>

Image Cartoonifier

Applying Image Processing Filters For Image Cartoonifying

In this assignment we want to make the real world images look like they are genuinely from a cartoon. The basic idea is to fill the flat parts with some colour and then draw thick lines on the strong edges. In other words, the flat areas should become much more flat and the edges should become much more distinct. We will detect edges and smooth the flat areas, then draw enhanced edges back on top to produce a cartoon or comic book effect.

Step 1:

we need to generate a black & white sketch using edge detection bilateral filtering (laplacian to be precise) but also alongside noise reduction using a Median filter.

First thing we did was import the **cv2** and **numpy** libraries to use them

Next we take the image as an input using **cv2.imread()** function

We then downsample the image then apply the bilateral filter with the dimension parameter = 9.

we upsize the image again so it's not pixelated and we use **cv2.imshow()** to show the result of the filter.

Step 2:

Next to reduce the noise we use the **cvtColor()** function to convert to grayscale and then we use the **cv2.medianBlur()** with parameter 7 (7x7) as asked to do the smoothing.

Last thing in this step is the laplacian function with aperture size 5.

Step 3:

In this step we attempt the thresh-holding to create edge mask detect and enhance edges using **cv2.adaptiveThreshold()** function

Step 4:

Lastly we combine the colour image with the edge mask and convert it back to colour image to see the cartoon effect

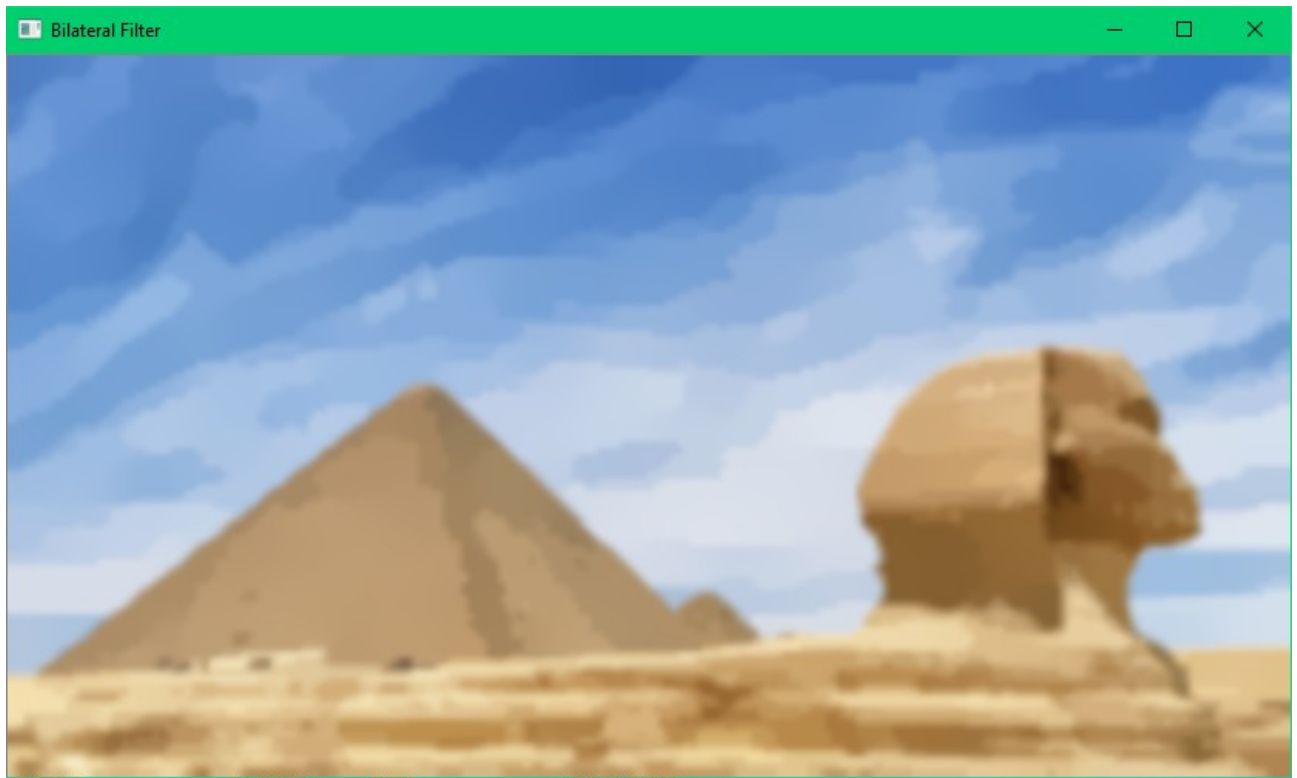
using `cvtColor()` on the output from `threshold()`
we extract whichever input image's size using `img.shape()` function we then resize the output to that size of the input image.

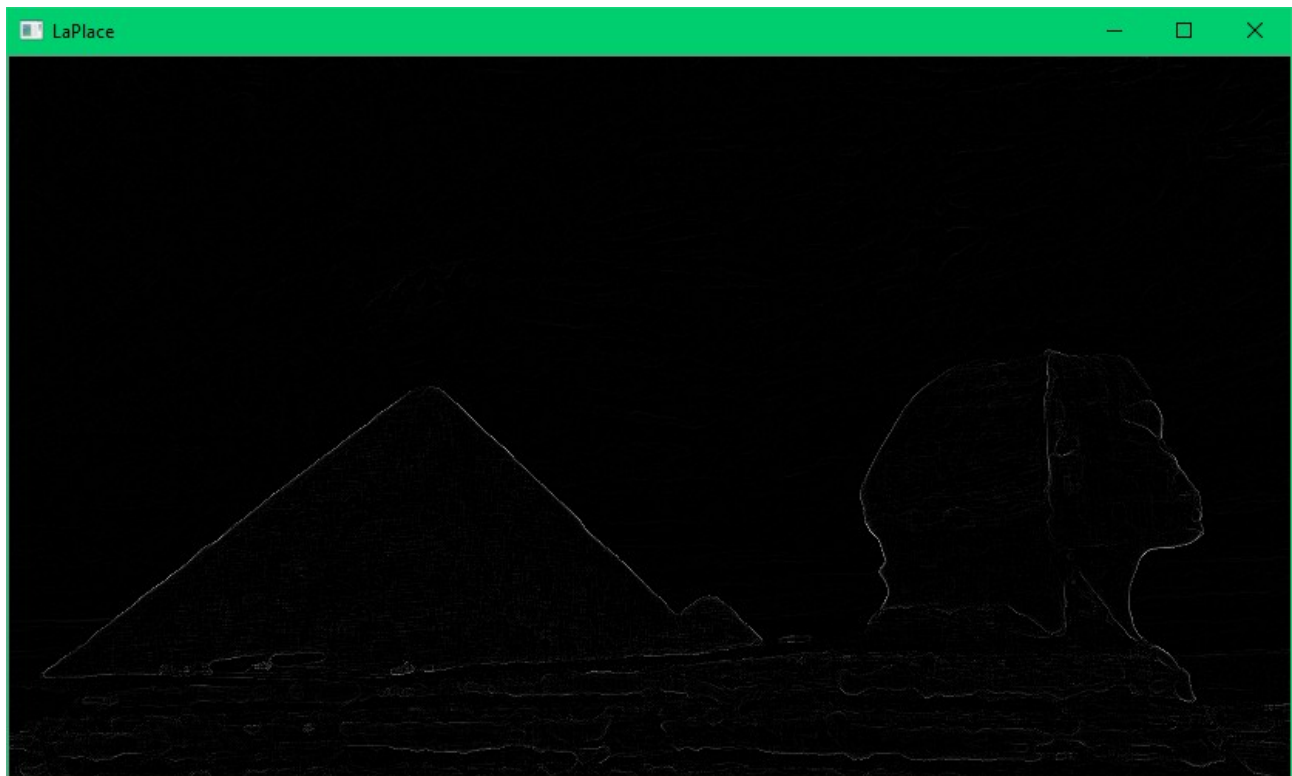
Finally we use `cv2.bitwise_and(colouredImage, EdgedImage)` to combine the two images. And we show the final result the cartoon.

Here are some test images and their results with each stage shown:

Original image 1:





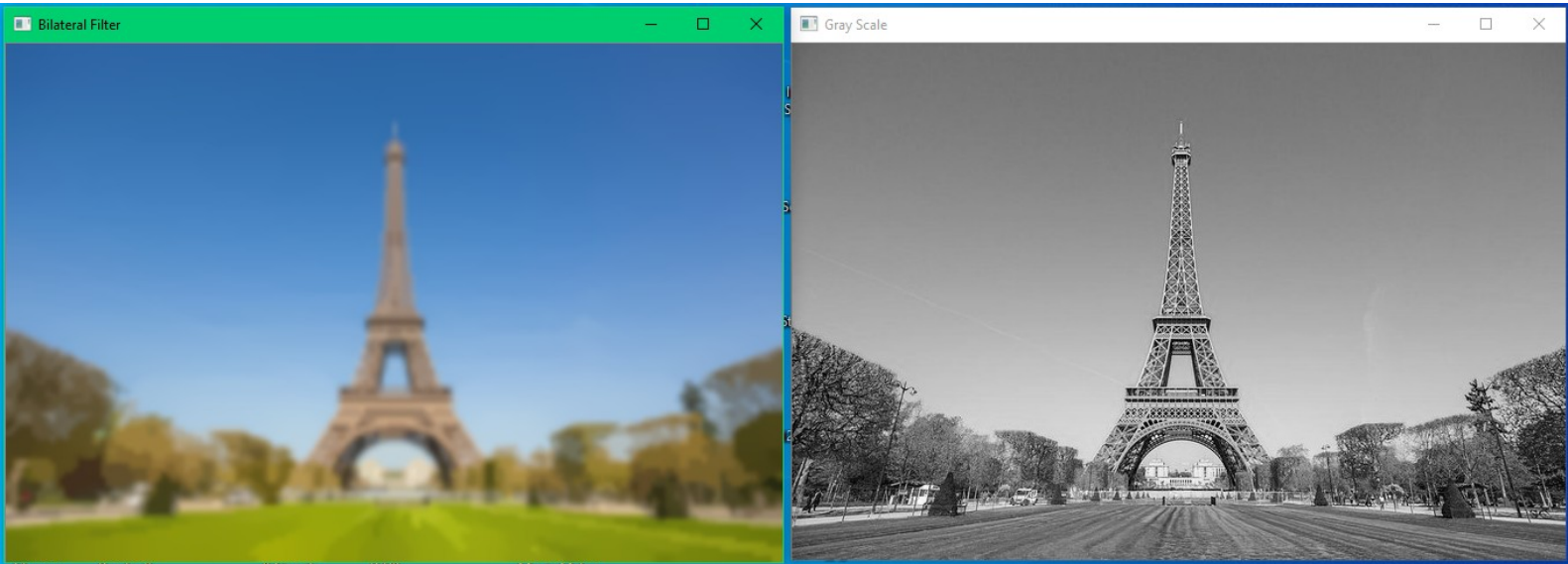


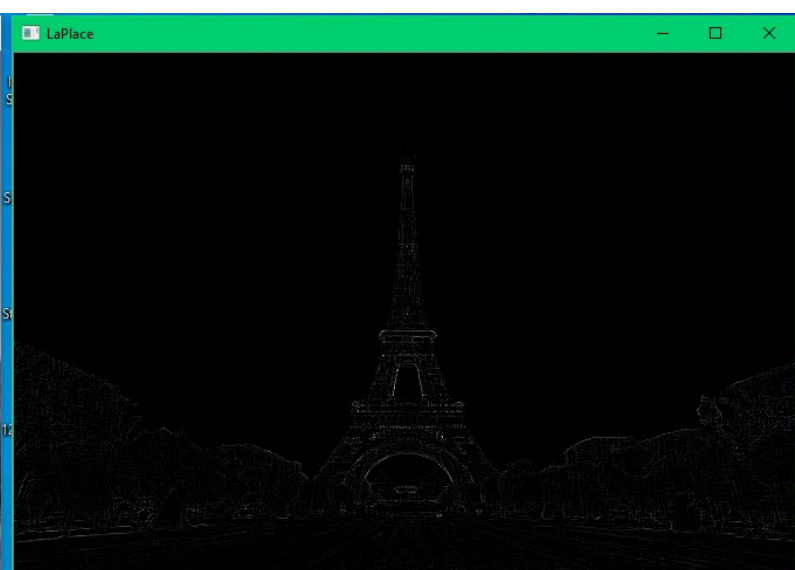
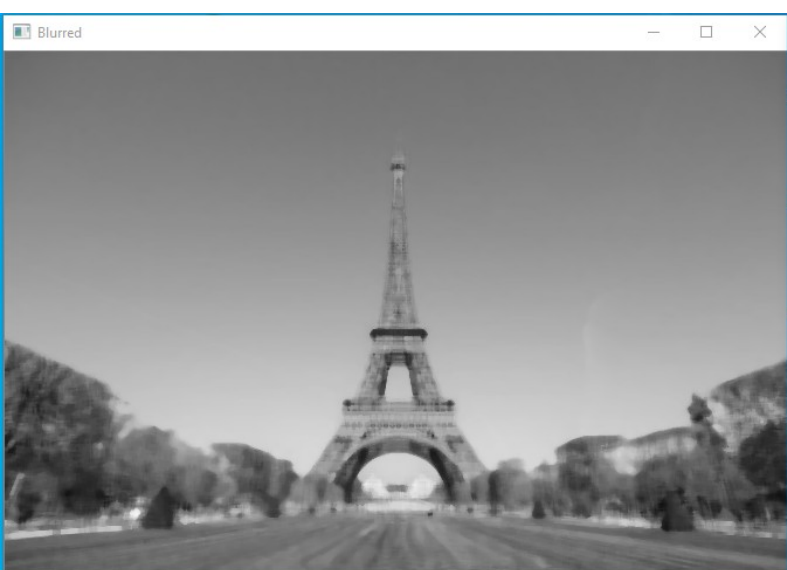


Final Cartoon 1

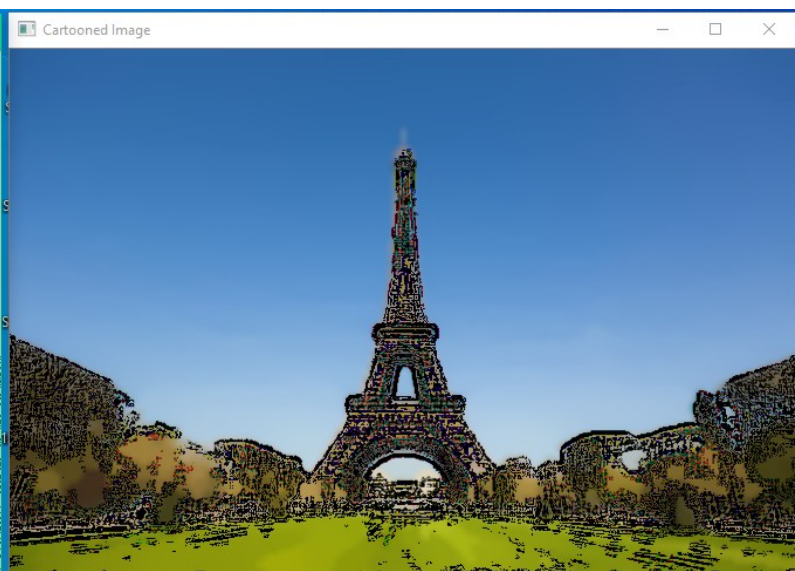
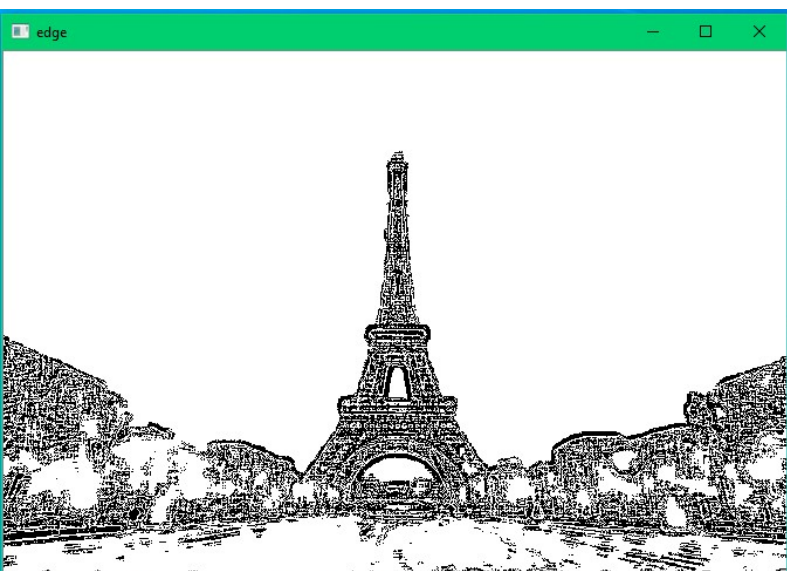


original image 2:

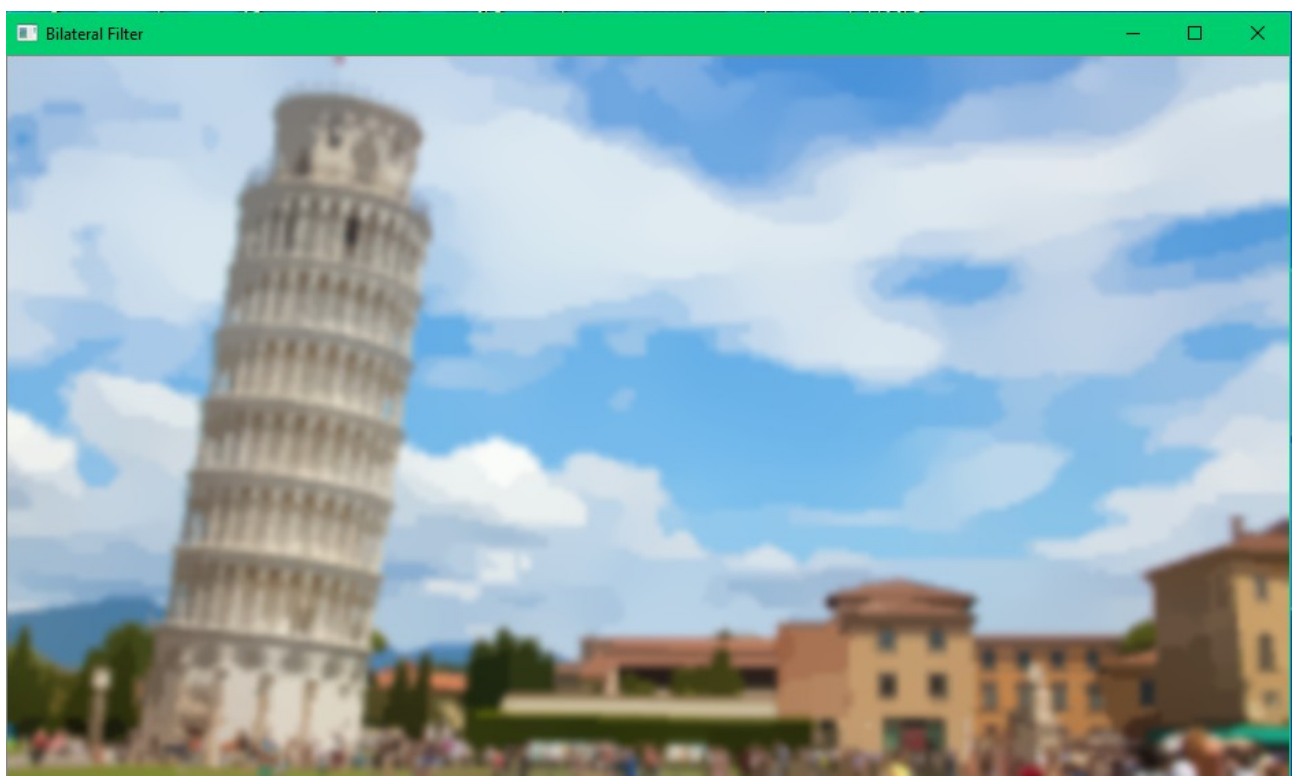




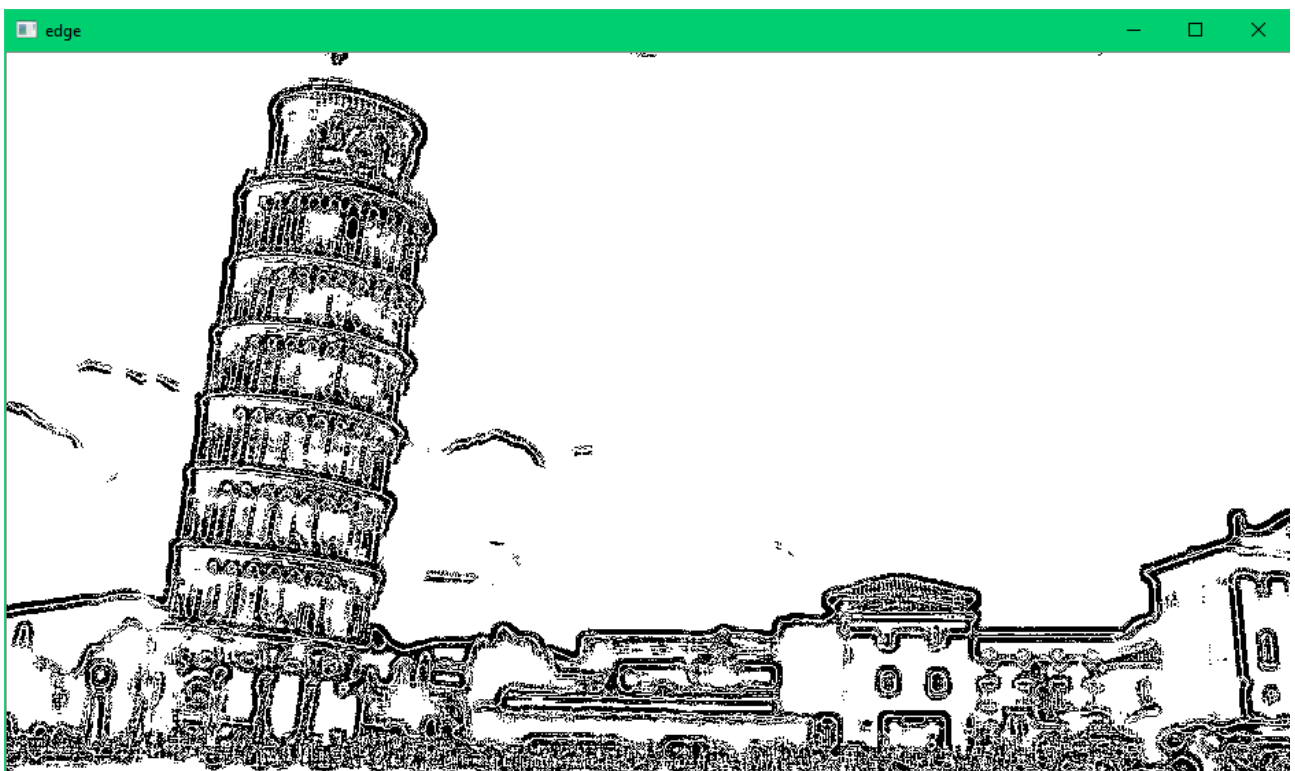
Final Cartoon 2:



Original image 3:







Final Cartoon 3:

