**Name : Ali Hussein Ali.**

**Number : 39.**

**Problem Set 2**

**Image Cartooni\_er**

**Applying Image Processing Filters For Image Cartoonifying**

Generating a black-and-white sketch:

At first read the image with grayScale.

GrayImg = cv.LoadImageM(*"cat.jpg"*, cv.CV\_LOAD\_IMAGE\_GRAYSCALE)

this function load the image in grayScale.



Noise Reduction Using Median Filter

|  |  |
| --- | --- |
|  | * **src** – The source image * **dst** – The destination image * **smoothtype** –   Type of the smoothing:   * + **CV\_BLUR\_NO\_SCALE** linear convolution with \texttt{size1}\times\texttt{size2} box kernel (all 1’s). If you want to smooth different pixels with different-size box kernels, you can use the integral image that is computed using [**integral()**](http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html#void integral(InputArray src, OutputArray sum, int sdepth)) |

**CV\_MEDIAN** median filter with a \texttt{size1}\times\texttt{size1} square aperture

cv.Smooth(GrayImg, SmoothGrayImg ,cv.CV\_MEDIAN, param1=5, param2=0, param3=0, param4=0)

make smooth for the image by this function and make the image smoothed gray scale.



Edge Detection Using Laplacian Filter

cv.Laplace(SmoothGrayImg , EdgeDetection\_Img )

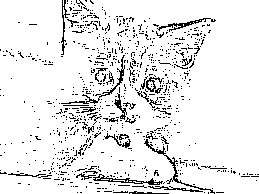
make edge detection by this function.



Edges Thresholding

cv.Threshold(EdgeDetection\_Img , Thresholding,20, 400, cv.CV\_THRESH\_BINARY\_INV)

put threshold for the edge detection by this function



Applying Bilateral Filter

cv.Smooth(im, BilateralImg, cv.CV\_BILATERAL, 100, 100, 100,100);

make smooth for the original image by this function



**CV\_BILATERAL** bilateral filter with a \texttt{size1}\times\texttt{size1} square aperture, color sigma= sigma1 and spatial sigma= sigma2. If size1=0 , the aperture square side is set to cvRound(sigma2\*1.5)\*2+1

Creating Cartoon Effect

By mix the two image edge detection and the image that make to it bilateral filter

cv.And(Sketch, Paint,finalImg)

