In the Folder there are few files and two subfolder, files are the code and the two subfolder are input and output which takes the input during execution and saves the new image generated after execution respectively.

We need to have MATLAB(2016 or latest) in the system for the execution of the code.

I have used least amount of inbuilt function provided by the matlab to perform the task and the method i used i ‘ll explain that.

* Brightness- For varying brightness there is brightness.m which will vary the input image’s brightness by the provided value.

syntax:brightness(‘princeton\_small.jpg’,value) #you can put value according to your choice

**Explanation**-For brightness first we have to convert image to matrix form to manipulate it and then multiply it with the provided value .

* Contrast- For varying contrast there is contrast\_image.m which will vary the input image’s contrast by the provided value.

Syntax: contrast\_image(‘c.jpg’,value) #you can put value according to your choice and event the negative value

**Explanation**-For contrast first we have to convert image to matrix form to manipulate it and then multiply each layer of the matrix to the provided value and for the negative value we taken the compliment of the image.

* Blur- For blurring there is blur.m which will add blurr effect to the input image.

Syntax: blur(‘c.jpg’)

**Explanation**-for blurring the picture i used blur function provided by matlab whose concept is to vary the value of the image matrix little but randomly.

* sharp- For sharping image there is sharp.m which will vary the input image.

Syntax: sharp(‘princeton\_smalljpg’)

**Explanation**-for sharping i have used imsharpen function provided in the matlab.

* Edge detect- For detecting edge there is detect\_edge.m which will detect the detect the edges in the input image.

Syntax: edge\_detect(‘princeton\_small.jpg’)

**Explanation**-there are many method for detecting edges as no method was mention i have used canny method .

* Composite- there is file name compose.m to perform this function ai have included input inside the code so we don’t have to give any input.

Syntax: compose()

**Explanation**-it was little complicated but the given mask made the task simple and if the mask wouldn’t be given then we have to detect face and then we will generate mask out of it.

First i have superimposed mask on both of the image which cuts the face from both the images and then i added both of the images now the image generated has too high value (like negative) then two generate proper output i subtracted matrix by 255 and the result image is generated.

Output: the output generated are self compatible with .html file hence we don’t have to give any output name. Output will be automatically saved in output folder and just refreshing the html page will auto generate the page.

In output folder you can see the output generated by me.