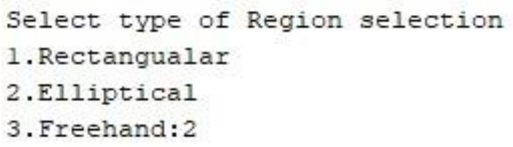
Merging Images - REPORT

A complete program has been coded in Matlab to perform image blending and the functioning of the code is given here in short(the subsections are elaborated later in the report):

1.) The two images are read using imread() function.

2.) A function had been designed to resize foreground image with a black background (position of start of image is taken as input which is chosen manually).

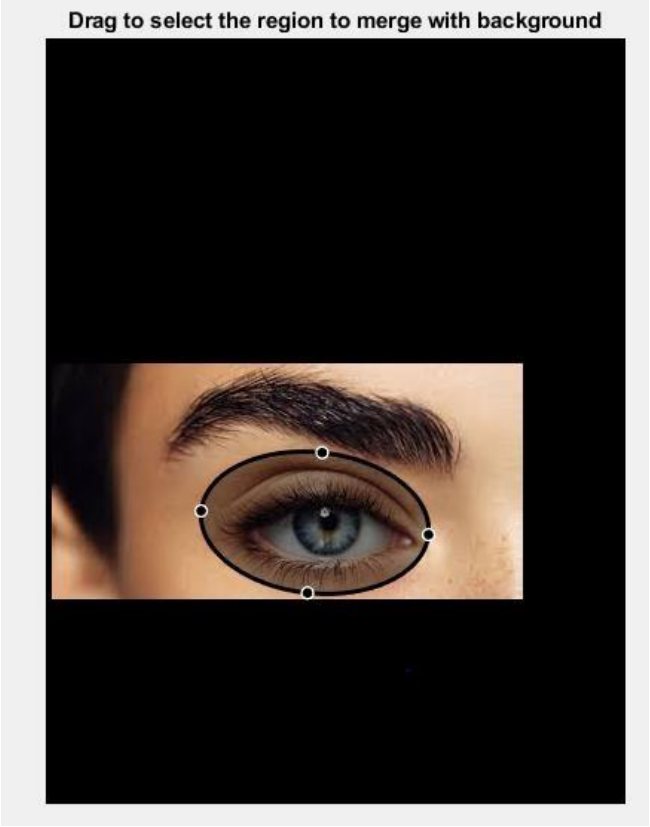
3.) Program takes an input to decide the type of region to select:



```
Select type of Region selection
1.Rectangualar
2.Elliptical
3.Freehand:2
```

(eg. If input is 2 elliptical region can be chosen)

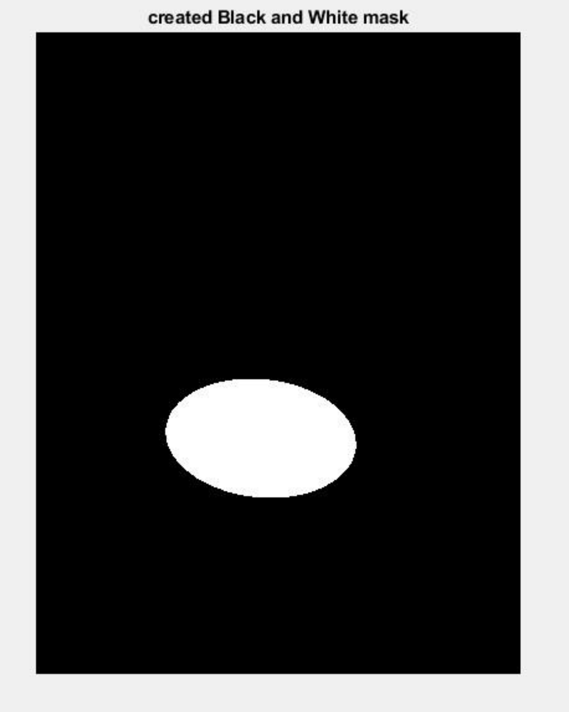
4.) Now GUI displays foreground image and allows to select region.



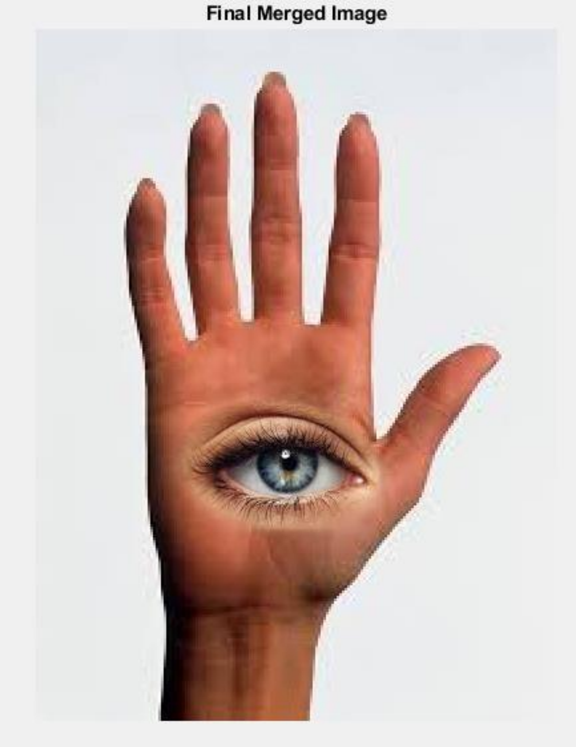
Drag to select the region to merge with background

We can select the required region in the image

5.) The mask corresponding to the chosen region is displayed



6.) The program performs blending and displays the final image.



The various functions used in the program are described below:

PART A) Gaussian and Laplacian Pyramid:

1.) The function defined is `[gPyr,lPyr]=ComputePyr(input_image,num_layers)` The function takes in image and number of layers and calculates the Gaussian and laplacian pyramid stored in a cell array.

2.) The number of default layers is calculated based on Log2 of the dimensions of the image. If the input argument number of level is greater than the default, it is invalid and only the default level is considered.

3.) The Gaussian Pyramid is computed by Gaussian filtering the image, downsampling to half size.

4.) For filtering operation the kernel convolve function from project 2 is used(myconvolve_2D).

5.) This function takes in the input image, kernel generated, type of padding and performs filtering operation.(1 represents 0 padding)

6.) The Gaussian filter is designed using MATLAB function `fspecial` and sigma of gaussian is taken as 4/6.

7.) A sample of Gaussian pyramid generated by code is given below considering 4 layers is given in the following page

8.) Laplacian pyramid is calculated in the same function using the Gaussian pyramid layers.

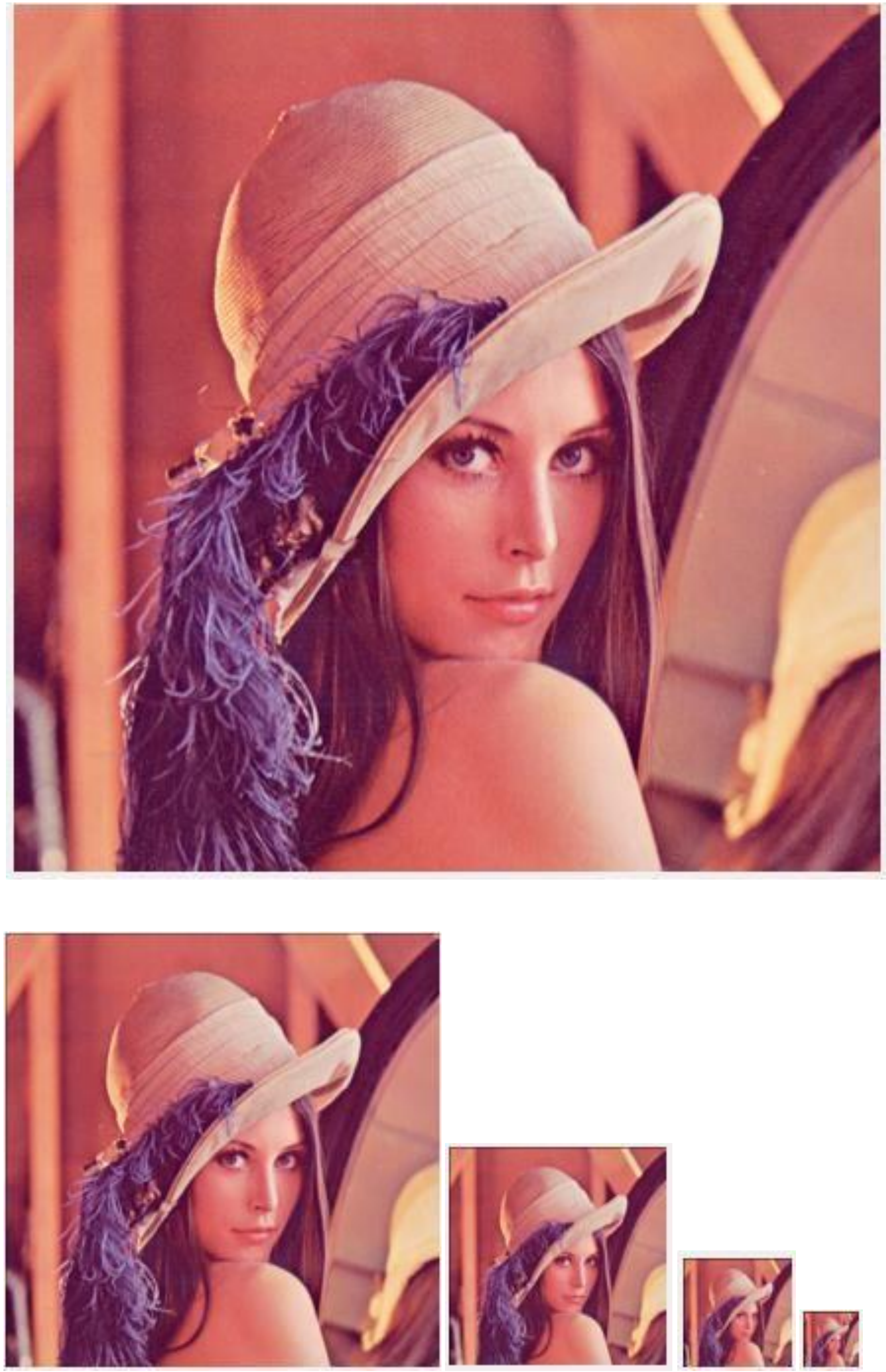
9.) It is calculated by interpolating a layer of Gaussian and subtracting it from the previous layer.

10.) A sample of laplacian pyramid is also given in the subsequent page as generated by the program for 5 layers.

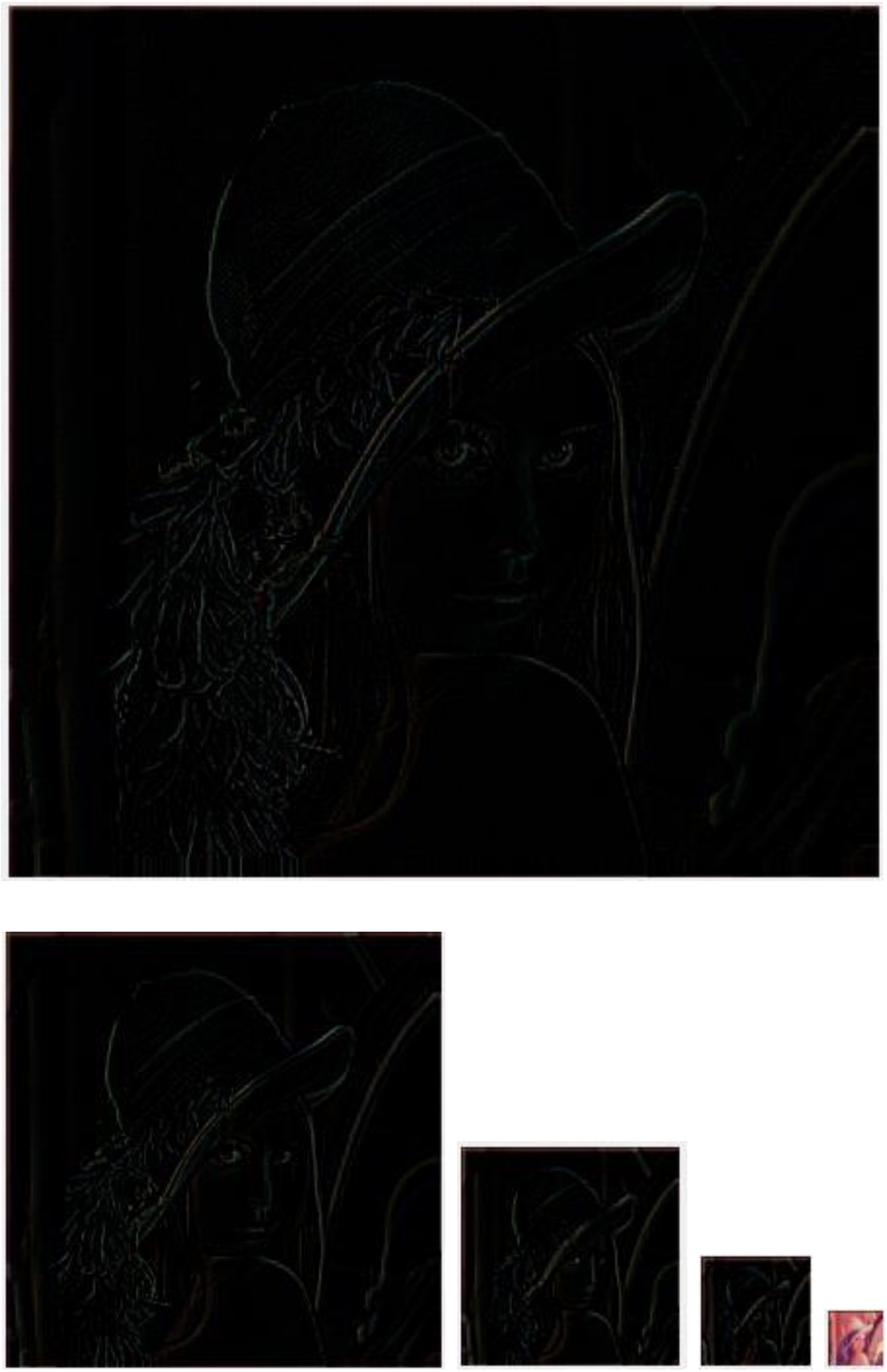
11.) The interpolation performed during the calculation is performed using the my_interpolate() function which performs nearest neighbour interpolation.

12.) The my_interpolate function takes in the image and returns the interpolated image

Example Gaussian Pyramid generated by the program:



Example Laplacian Pyramid generated by the program:

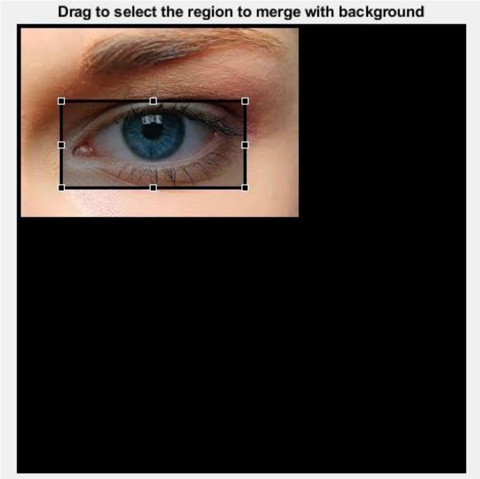


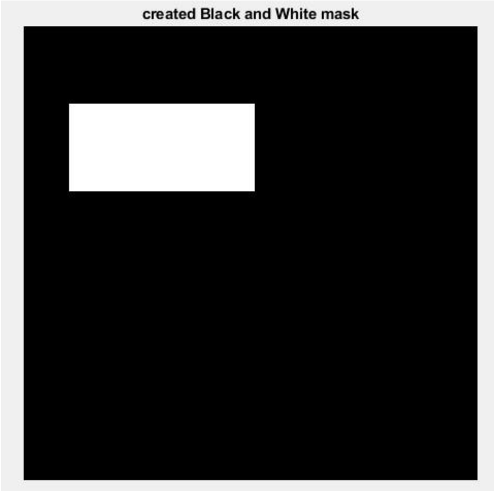
Note: generally last layer of Laplacian is same as last layer of Gaussian

PartB) GUI for region selection:

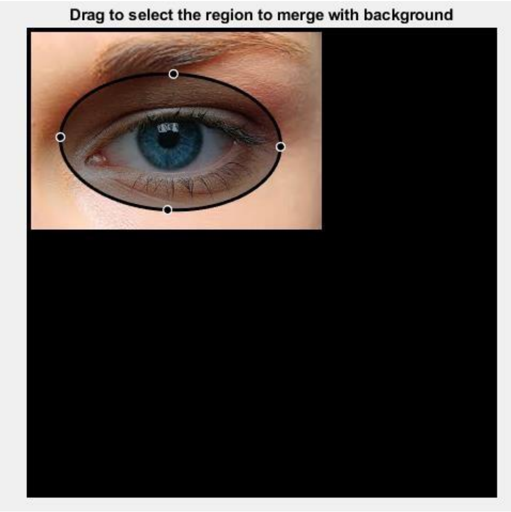
1.) Three choices are given to the user to select a region in the foreground image
2.) They are: rectangular, elliptical and freehand region.
3.) This is decided by the input 1,2 ,3 correspondingly.
4.) The region is chosen by dragging on the foreground image.
5.) Once the region is chosen the function generates a mask with chosen region in white and remaining black.
6.) Following are examples of the 3 types and corresponding masks generated.

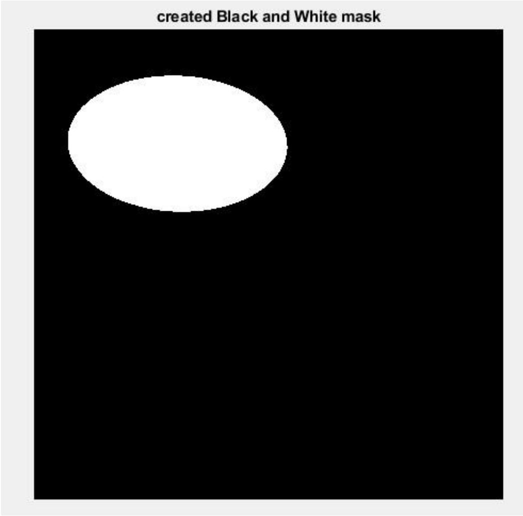
Rectangular:



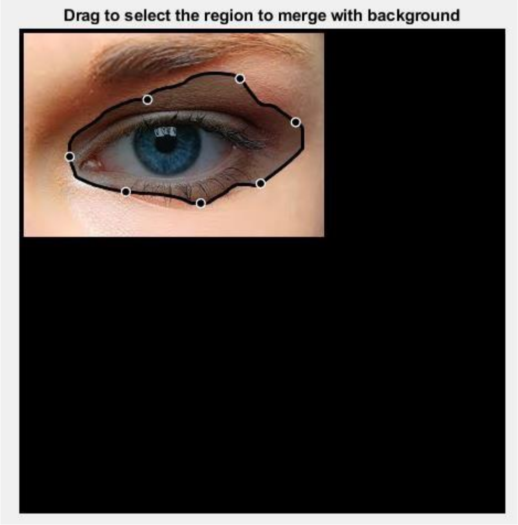


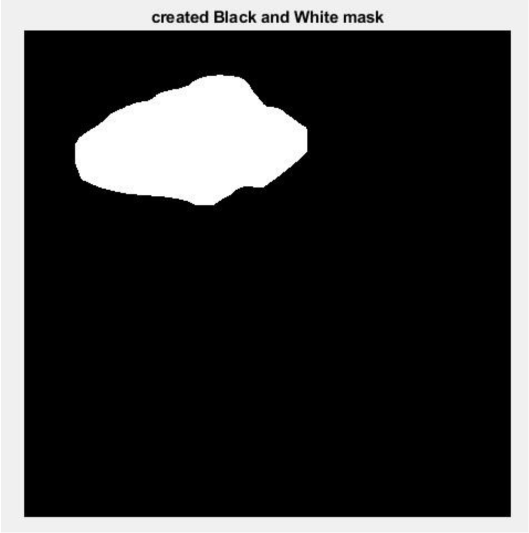
Elliptical:





Freehand:





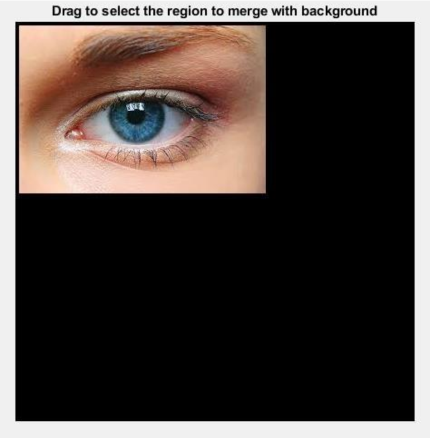
This mask generated is used for the merging process

PartC) Blending of images

1.) For blending operation we require the foreground and background image to be of equal size.
2.) Hence, as required in the question we define a function to impose the foreground image in a black background which is of the same size as the background(assuming foreground size<=background size)
3.) For this functionality the below function is defined in program
```
B=pic_resize(A,B_size,startx,starty)
```
4.) The function takes in the foreground image, background image size, and starting position pixel values of the foreground image in the black background.
5.) Based on the starting position pixel values,the function places the foreground image.
6.) The startx and starty values can be tuned manually and fed as input to the function based on which the positioning is varied.
7.) This is essential for merging in the required position.
8.) Below are some examples of foreground in the background based on the values of startx and starty.

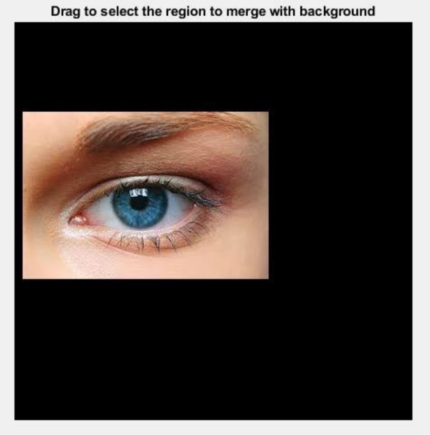
Examples for positioning foreground by controlling input to function my_resize manually:

startx=5 starty=5



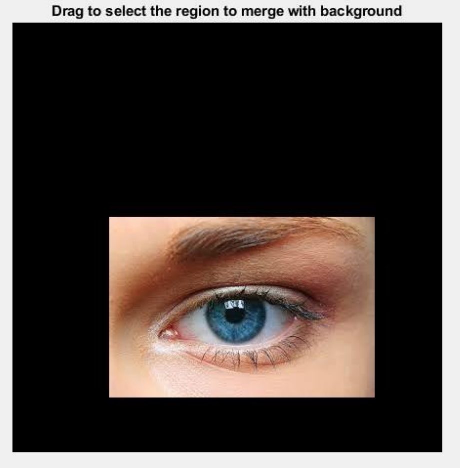
Drag to select the region to merge with background

startx=100 starty=10



Drag to select the region to merge with background

startx=200 starty=100



Drag to select the region to merge with background

Blending of images:

1.) The Laplacian Pyaramid is calculated for the foreground and background images.

2.) The Gaussian pyramid is calculated for the Black and white mask obtained from region selection.

3.) Laplacian of the blended image is obtained as:

```
X{i}=(Lap1{i}.*Gau3{i})+(Lap2{i}.*(1-Gau3{i}))
```

X=laplcaian of blend, Lap1=Laplacian of foreground, Lap2=Laplacian of background, Gau3=Gaussian of mask.

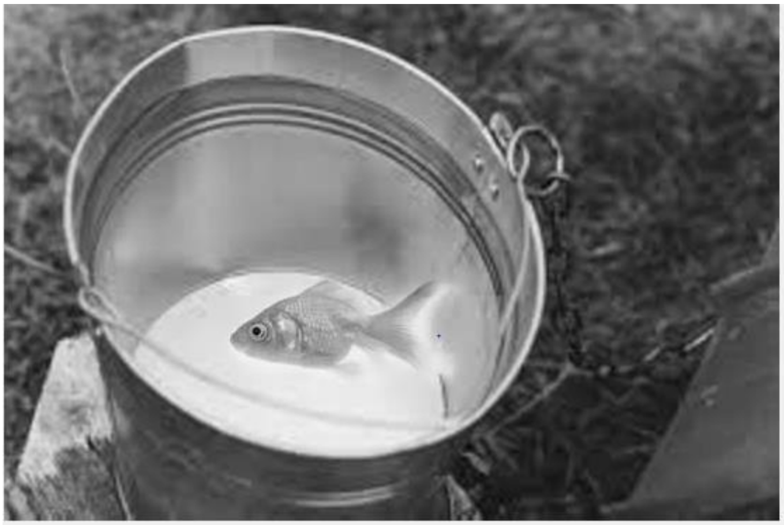
4.) After this pyramid is obtained, it is collapsed to obtain the merged image.

5.) Collapsing is performed by taking a layer of laplacian, upsampling it, filtering and adding it to the next big layer.

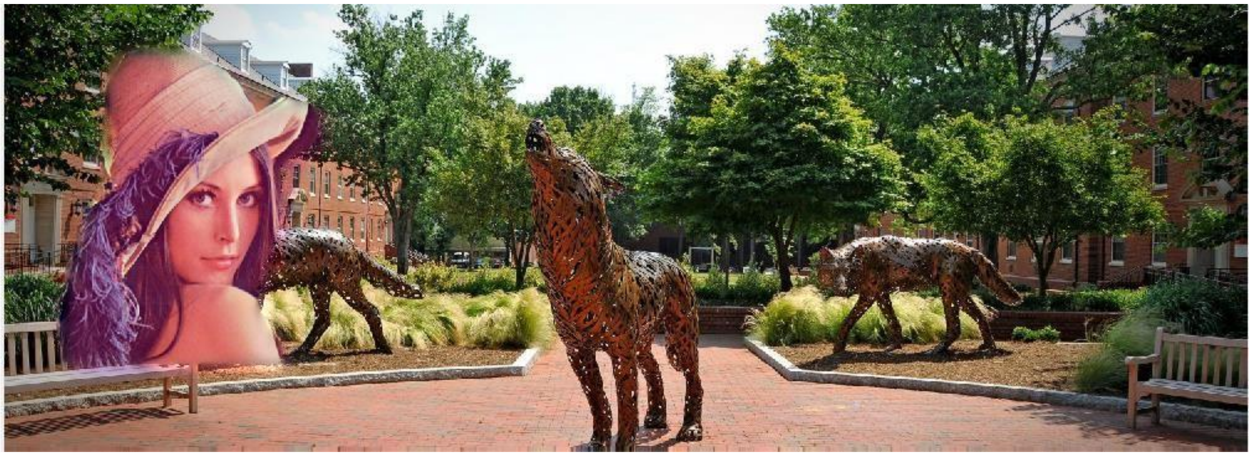
6.) This step is repeated from the smallest image to the largest image in the Laplacian pyramid.

7.) The function used for interpolation/upsampling is defined as my_interpolate and function for filtering is the one from project2.

8.) Following are various examples obtained for different positions and regions of blending.



Note:the code works for both RGB and colour as given above



Note: the pictures were resized before being input to program for adaptability to the code.