# REPORT ASSIGNMENT 1

## NEGATIVE:


original image


neagtive image

## RESIZE:

## NEARESTNEIGHBOUR

**original**

In case of nearest neighbour, RMSE was coming to be 0 telling that the implementation is similar to the inbuilt function.

**resized using nearest neighbour(2,2)**

RMSE ERROR=0 (inbuilt function used: resize)

In bilinear, little error was observed telling that there was a small difference in my and the inbuilt implementation. (inbuilt function used: resize)

**bilinear(0.5,0.8)**
RMSE=0.645217

## LOG TRANSFORMATION

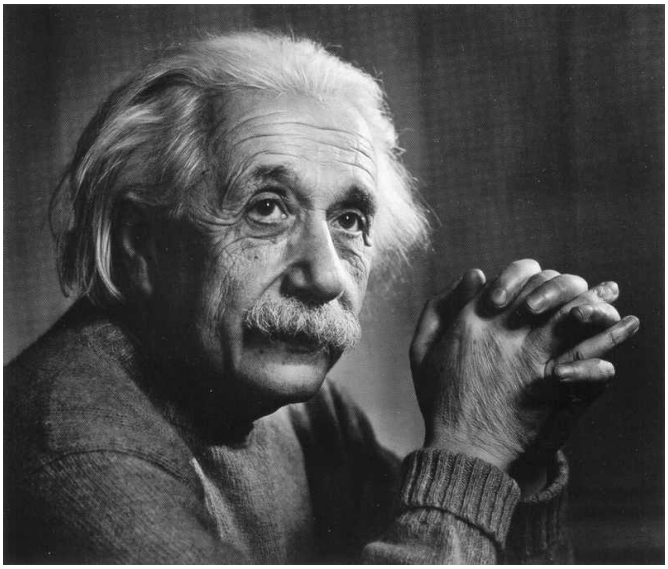
log transformed image (c=30)

## GAMMA TRANSFORMATION

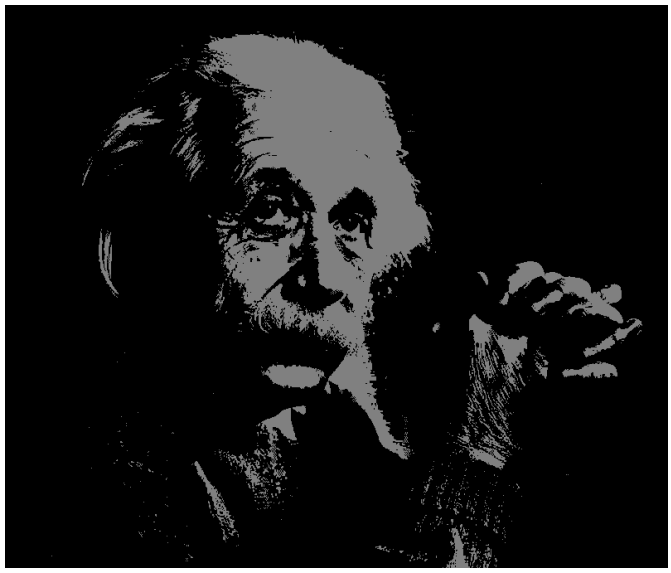
gamma transformed (gamma=1.2 , c=0.2)

# BITPLANE SLICING

**original image**



**bitplane slicing (bit plane=8)**



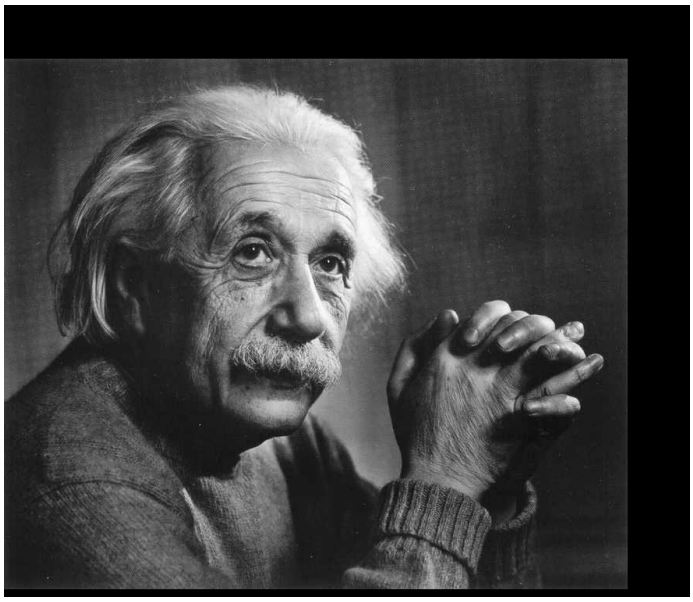
More significant features are observed when bitplane is higher as there is more difference in the pixel values highlighting important features.

**bitplane=5 and 6**

2 bitplanes are added but the image is still less clear than the 8 bitplane image.

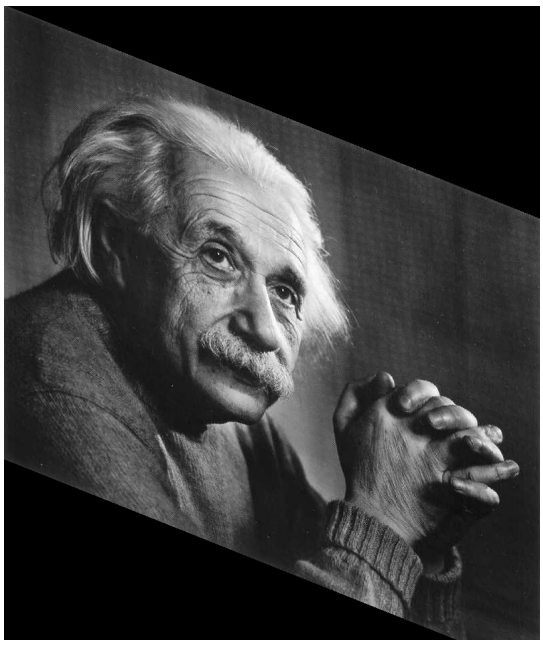
## TRANSLATION

translation
(30,-40)

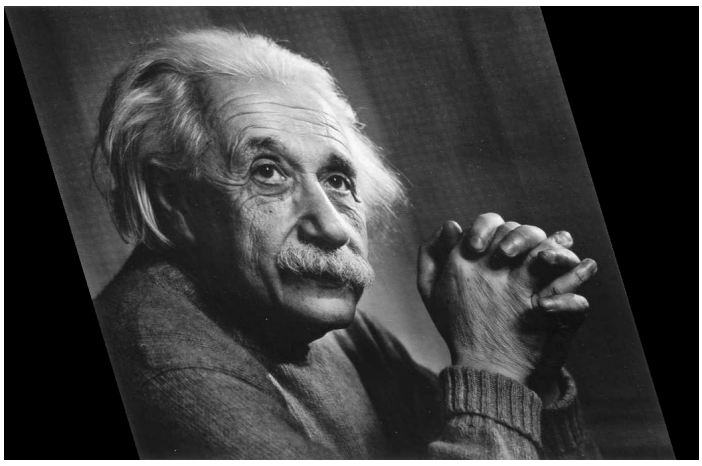


## SHEAR

vertical shear (0.4)
(using nearest neighbour)



horizontal shear (0.3)
(using bilinear)

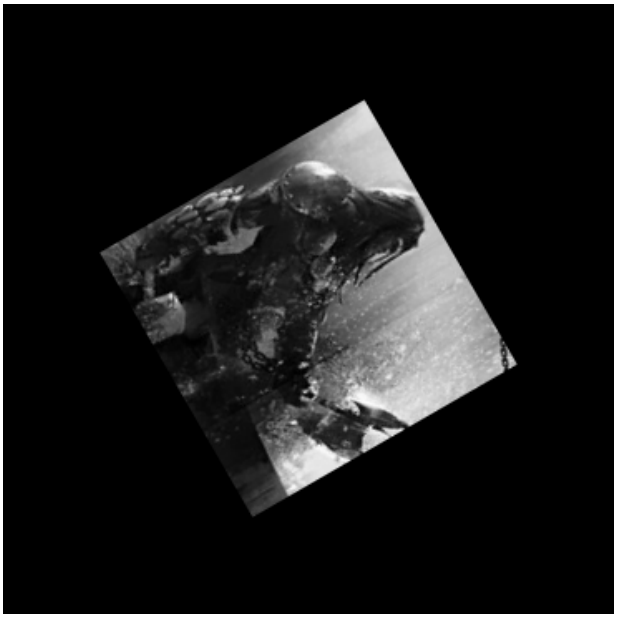


**ROTATION**

original image

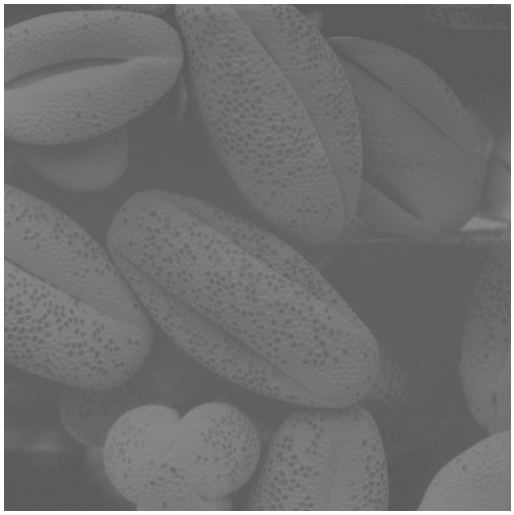
rotated image 45 degrees (nearest neighbour)

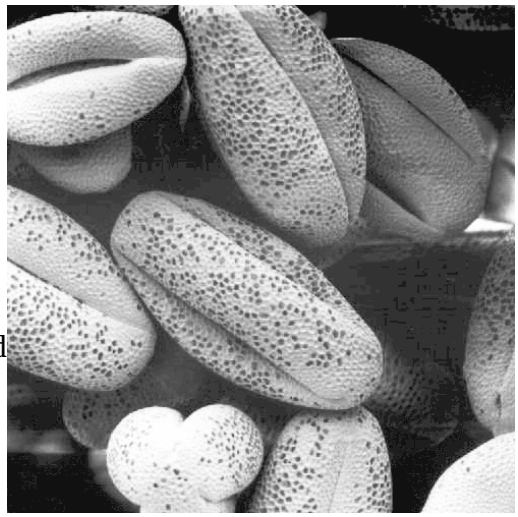
rotated image -60 degrees (bilinear)



## HISTOGRAM EQUALIZATION

**original image**

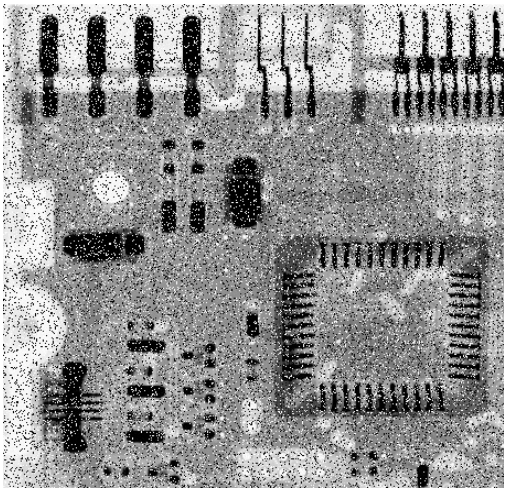


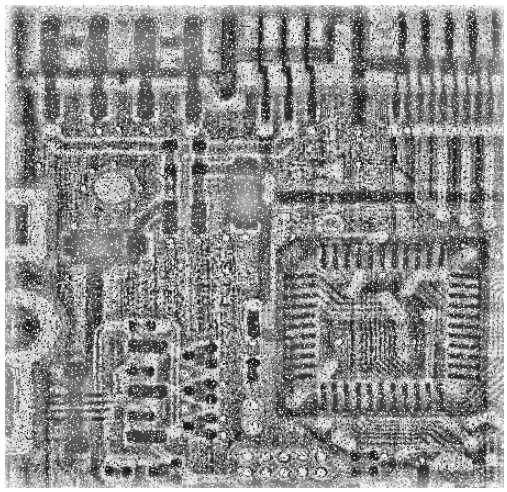


**histogram equalized image**

As evident from output,histogram equalization significantly improves the image by contrast stretching and making important details visible. Initial image had all the pixel values in a limited range which were stretched to get a beautiful output image.

## ADAPTIVE HISTOGRAM EQUALIZATION



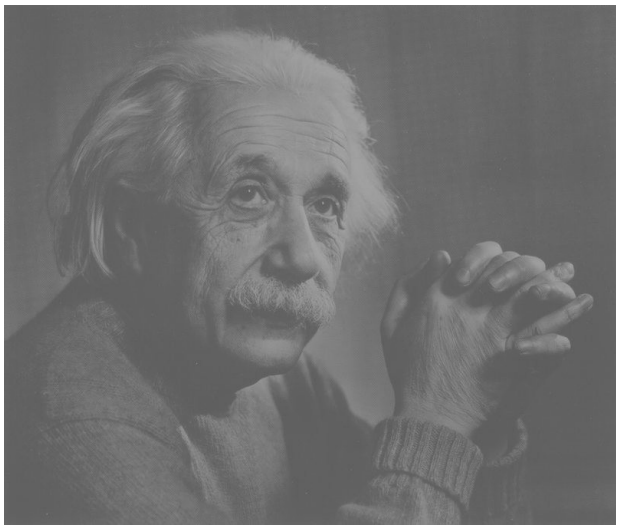


original image

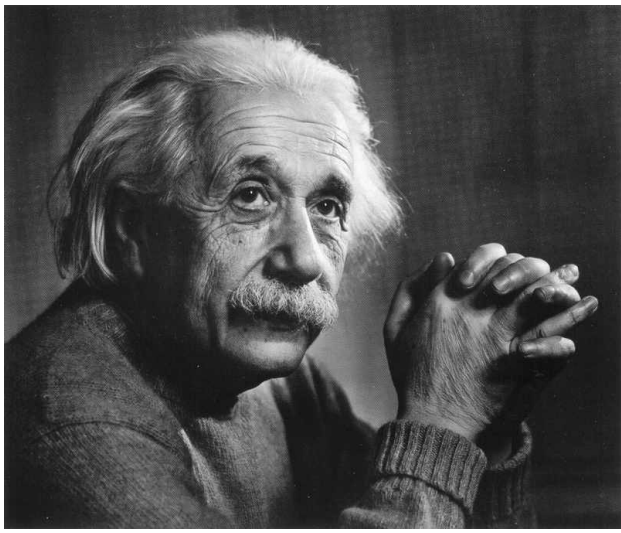
adaptive histogram equalized image

In adaptive histogram little details are also highlighted. It can be used to focus on little things in an image.

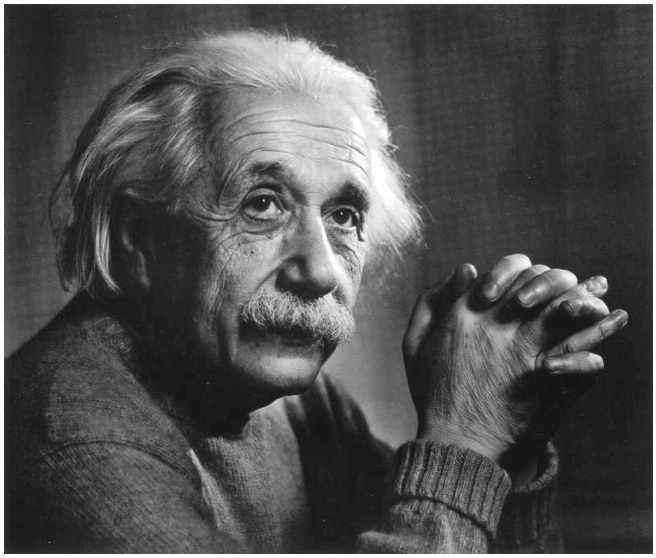
# HISTOGRAM MATCHING



**original**



**example image to be matched with**

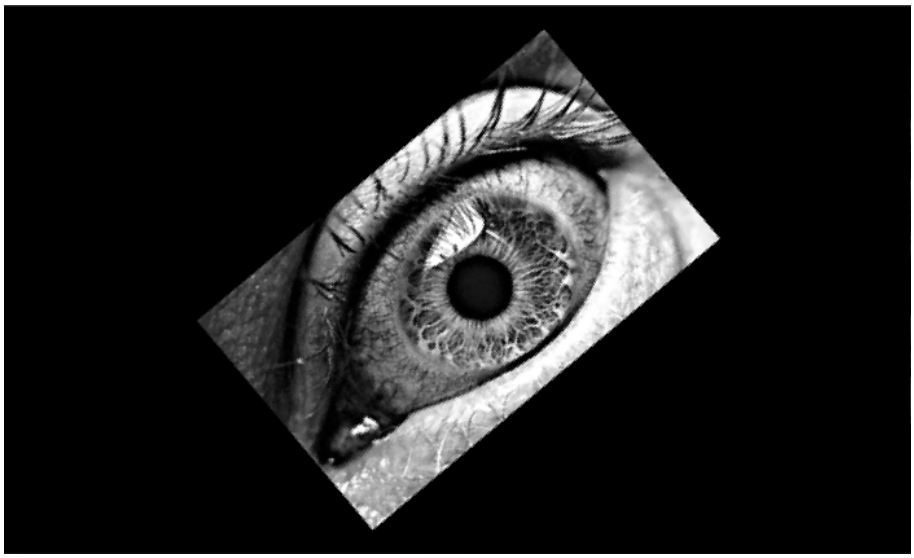


**histogram matched image**
RMSE from matched image= 2.02

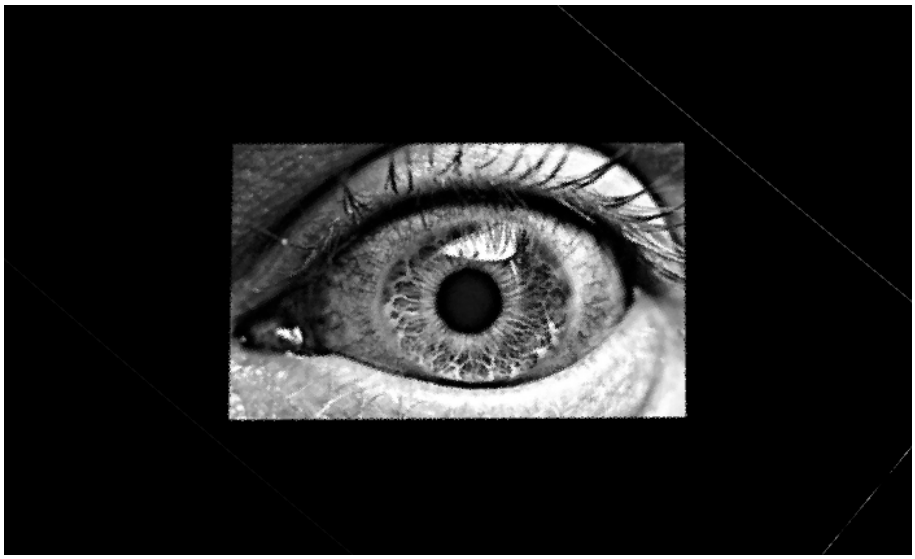
I matched the original image with the second one and the output image was almost same as the second image with RMSE of 2.02. The program, therefore, matches the histogram correctly.

**IMAGE RECONSTRUCTION**
**GIVEN 4 TIE POINTS**

original image



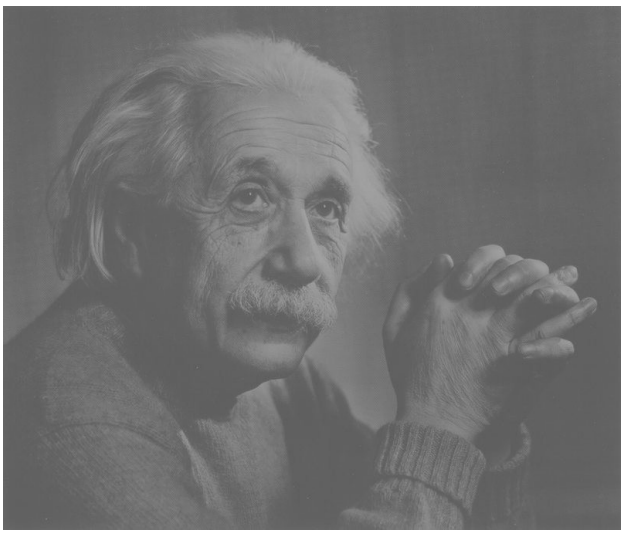
reconstructed image
input:
161 490 83 426
133 320 171 277
295 433 222 468
215 203 308 240



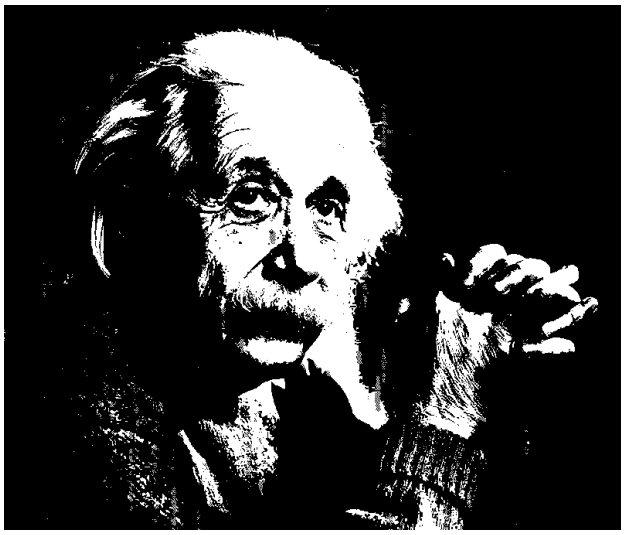
The 4 tie points were used to get the matrix
to get the coordinates of the original image.
In this,I have kept the window size to be the same
as that of the input.
(I took this image from shailendra to verify the result
and check if my program was correct).

# PIECEWISE LINEAR TRANSFORMATION

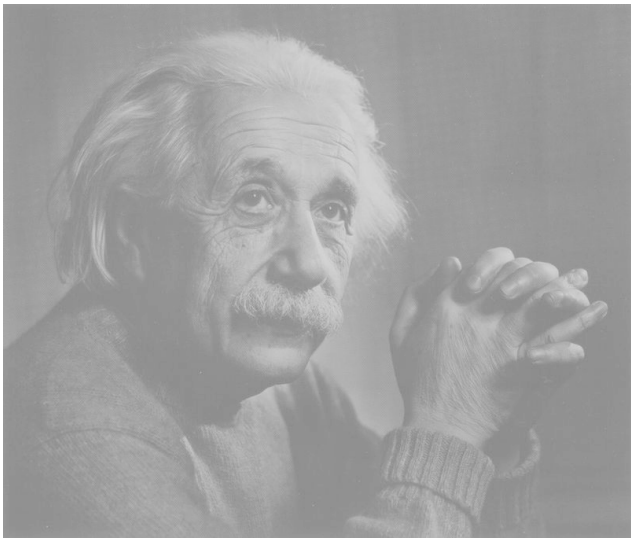
**original image**



**piece wise linear transformed (120,0) and (120,255)**



**piece wise linear transformed (100,150) and (200,250)**



In my program, I have considered 3 slopes one from (0,0) to first coordinate,second from first coordinate to second coordinate and thirs from second coordinate to (255,255). The 2 points can be adjusted to get the desired result.