GOA COLLEGE OF ENGINEERING

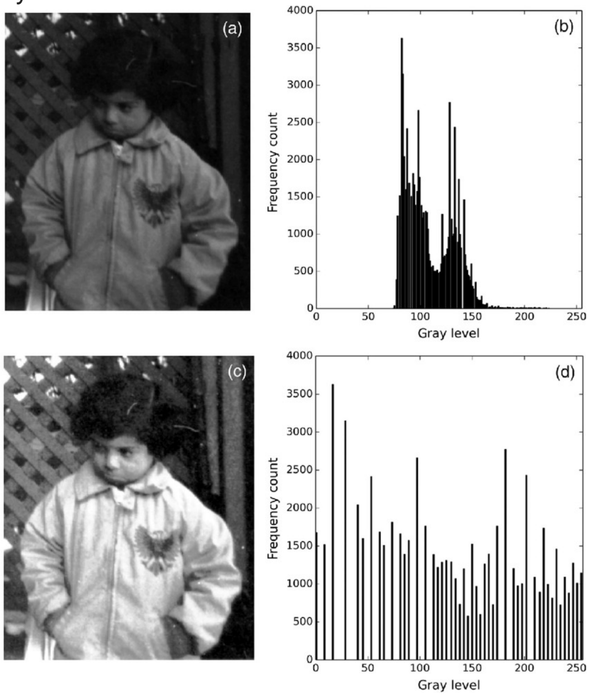
“Bhausaheb Bandodkar Technical Education Complex”

Experiment No: 7 **Date:**

Aim: Write a Java program to read an image from a file system and perform histogram equalization of the image and write it back with a different name.

Theory: Histogram is a graphical representation of the intensity distribution of an image. In simple terms, it represents the number of pixels for each intensity value considered.

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.



The above image shows 2 images and corresponds to 2 histograms. The above image is the original image with histogram indicating low contrast dark image. After equalization the intensity is distributed across the range resulting in increasing contrast and a relatively clearer image (shown in the lower image and histogram of the above figure)

Algorithm :

1. Read the image and store it in a container to perform operations on it.
2. Convert this image into gray scale image
3. Read all pixel values and store the frequency count for each pixel
4. Now find the cumulative probability for each pixel intensity by multiplying the pixel frequency with total number of pixels
5. Now find the sum of the cumulative probability by adding the frequency value to the previous frequency value. Eg. Pixel 0 (0.117) & Pixel 1 (0.123) . Now in the table the value of sum for Pixel 1 will be 0.117 + 0.123 = 0.240 and so on
6. Now multiply each sum value with 2^B - 1 where B is the no. of bits with which the pixel is represented
7. Now round this value to the nearest integer
8. Now map this rounded value to its corresponding original frequency
9. Calculate the new frequency value of each pixel according to map
10. Now iterate through the image and replace the pixel value with new value according to the mapped values found out earlier

11. Choose a directory to store the new image
12. Store the equalised image into the selected directory

Program:

Python Code:

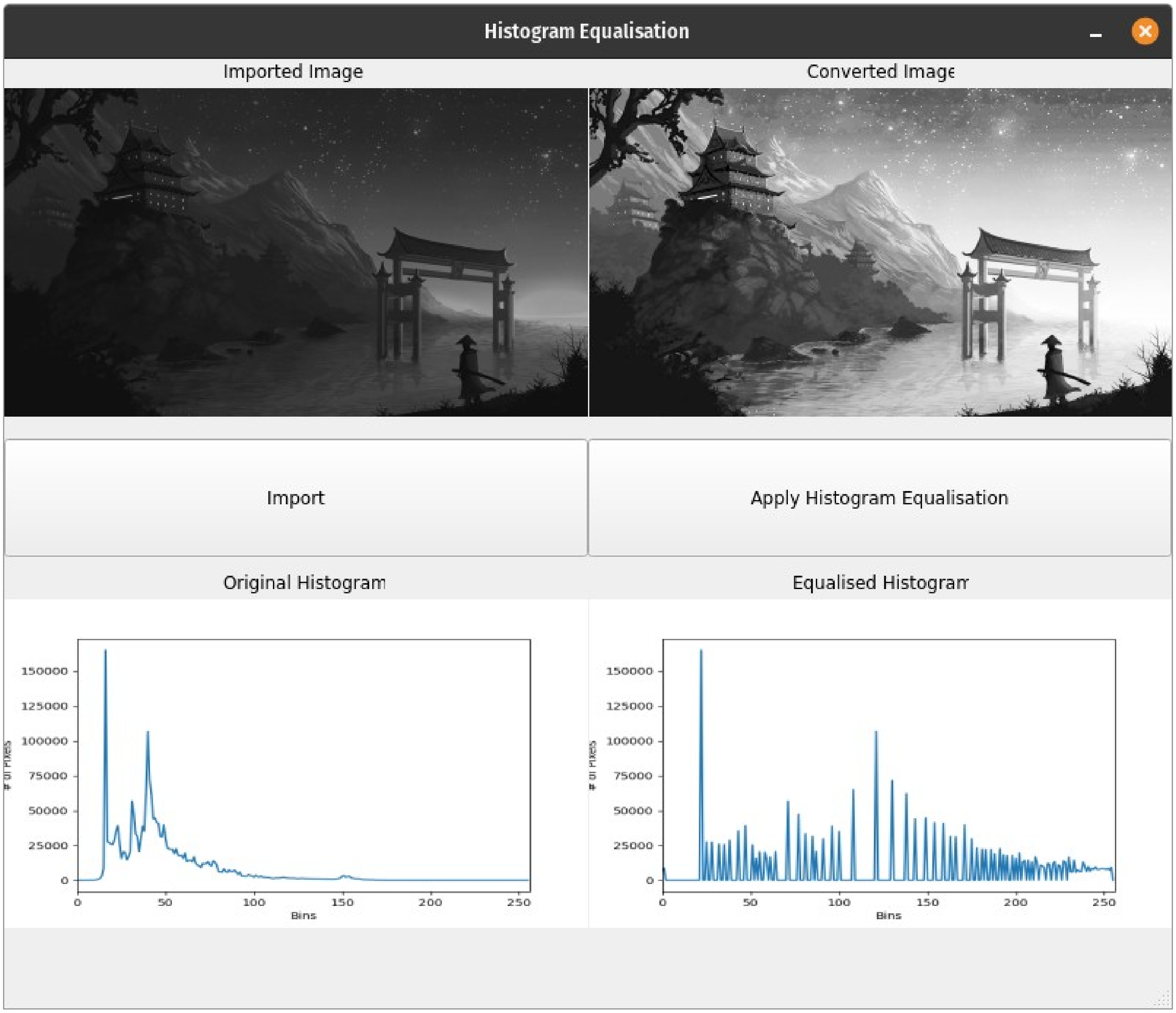
```
def show_image(self):
file_filter = 'Image File (*.jpg *.png)'
fname = QtWidgets.QFileDialog.getOpenFileName(parent=self.centralwidget,
caption='Select an Image',
directory="/home/deeprajb/Downloads/",
filter=file_filter)
self.img = cv2.imread(fname[0], cv2.IMREAD_GRAYSCALE)
self.img1 = QtGui.QImage(self.img.data, self.img.shape[1], self.img.shape[0], QtGui.QImage.Format_Grayscale8)
self.hist = cv2.calcHist([self.img], [0], None, [256], [0, 256])
self.imageinput.setPixmap(QtGui.QPixmap.fromImage(self.img1))
plt.figure()
plt.xlabel("Bins")
plt.ylabel("# of Pixels")
plt.plot(self.hist)
plt.xlim([0, 256])
plt.savefig('Original_Histogram.png')
# plt.show()
self.imageinput_2.setPixmap(QtGui.QPixmap("Original_Histogram.png"))
def getCounts(self,width,height,px):
counts=[0]*255
for i in range(width):
for j in range(height):
counts[px[i,j]]+=1
return counts

def histogram_equalisation(self):
(row, col) = self.img.shape[0:2]
# final = self.hist.copy()
# ejo = self.hist.copy()
preture = self.img.copy()
counts=self.getCounts(row,col,preture)
# ehi /= row*col
pS=0
for i in range(len(counts)):
pS+=255*(counts[i]/row*col)
counts[i]=pS
for i in range(row):
for j in range(col):
preture[i,j]=int(counts[preture[i,j]])
```

```
eq_hist = cv2.calcHist([preture], [0], None, [256], [0, 256])
cv2.imwrite('eh_output.jpg',preture)
self.imageoutput.setPixmap(QtGui.QPixmap("eh_output.jpg"))
plt.figure()
plt.xlabel("Bins")
plt.ylabel("# of Pixels")
plt.plot(eq_hist)
plt.xlim([0, 256])
plt.savefig('Equalised_Histogram.png')
self.imageoutput_2.setPixmap(QtGui.QPixmap("Equalised_Histogram.png"))
```

Output:

Python GUI Output:



Conclusion: Program to read an image and perform histogram equalization on it was written and executed successfully.