Mujtaba Shahid Faizi

BSCS-5A

#131818

LAB 6 of DIP

**Code:**

**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
**import** cv2  
**from** PIL **import** Image  
  
**def** show\_hist(img): *#to show histograms* plt.hist(img.ravel(), 256, [0, 256])  
 plt.show()  
  
img = cv2.imread(**'Monkman\_MABKS\_D30112.jpg'**)  
gray\_image = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  
cv2.imshow(**'image'**,gray\_image)  
show\_hist(gray\_image)  
arr = np.asarray(gray\_image)  
  
rows = arr.shape[0] *#width & height of the image i.e. no. of pixels*cols = arr.shape[1]  
  
  
pixels=[0]\*256 *#number of pixels of each intensity*pdf =[0]\*256 *#pdf of all pixels*cdf=[0]\*256 *#cdf of all pixels*T=[0]\*256 *#transformation***for** x **in** range(0, rows):  
 **for** y **in** range(0, cols):  
 pix=arr[x,y]  
 pixels[pix]+=1  
  
totalpixels=0 *#total no. of pixels in image***for** i **in** pixels:  
 totalpixels = totalpixels + i  
  
**for** a **in** range(0,256):  
 pdf[a]=pixels[a]/totalpixels  
  
sum=0  
**for** a **in** range(0,256):  
 **for** b **in** range(0,a+1):  
 sum=sum+pdf[b]  
 cdf[a]=sum  
 sum=0  
 T[a]=round((cdf[a]\*255),0) *#255 being L-1=256-1=255***for** x **in** range(0, rows): *#reconstruct image on basis of transformation values* **for** y **in** range(0, cols):  
 arr[x,y]=T[arr[x,y]]  
  
im = np.array(arr \* 255, dtype = np.uint8)  
  
cv2.imshow(**'image'**,im)  
show\_hist(im)

**Screenshot:**

The equalized histogram has a uniform distribution acc. to the second figure.







