Mujtaba Shahid Faizi

BSCS-5A

#131818

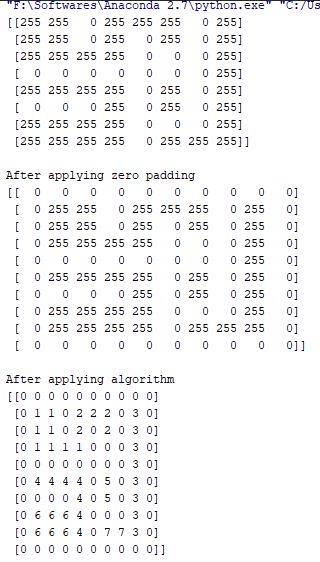
Lab 4 of DIP

**Task 1 (Using Connected Component Labelling Algorithm on the numpy array):**

**Code:**

**import** numpy **as** np  
  
a = np.array([[255, 255, 0, 255, 255, 255, 0, 255], [255, 255, 0, 255, 0, 255, 0, 255], [255,255,255,255,0,0,0,255],[0,0,0,0,0,0,0,255],  
 [255,255,255,255,0,255,0,255],[0,0,0,255,0,255,0,255],[255,255,255,255,0,0,0,255],[255,255,255,255,0,255,255,255]])  
**print** a  
  
count=0  
a=np.pad(a, ((1,1),(1,1)), **'constant'**) *#zero padding to avoid out of bound error when checking for neighbours*rows = a.shape[0] *#width & height of the image i.e. no. of pixels*cols = a.shape[1]  
  
**print "\nAfter applying zero padding"  
print** a  
  
**for** x **in** range(0, rows):  
 **for** y **in** range(0, cols):  
 **if** (a[x,y]==0): *#neglecting the black boundaries* **continue  
 else**:  
 conn\_labels = []  
 **if** (a[x - 1, y]!=255) **and** (a[x - 1, y]!=0): *#neglecting the neighbours who dont have any labels (yet)* conn\_labels.append(a[x - 1, y])  
 **if** (a[x + 1, y]!=255) **and** (a[x + 1, y]!=0):  
 conn\_labels.append(a[x + 1, y])  
 **if** (a[x , y-1]!=255) **and** (a[x , y-1]!=0):  
 conn\_labels.append(a[x, y - 1])  
 **if** (a[x , y+1]!=255) **and** (a[x , y+1]!=0):  
 conn\_labels.append(a[x, y + 1])  
  
 **if** len(conn\_labels)==0:  
 count=count+1 *#incrementing label when no neighbour labels found* a[x,y]=count *#assigning the label* **else**:  
 min\_label = min(conn\_labels) *#assigning the minimum label* a[x, y] = min\_label  
  
**print "\nAfter applying algorithm"  
print** a

**Screenshot:**



**Task 2 (Using Connected Component Labelling Algorithm on the image):**

**Code:**

**from** PIL **import** Image  
**import** numpy **as** np  
  
**def** binarize\_image(path,threshold):  
 *"""Binarize an image."""* img = Image.open(path)  
 img = img.convert(**'L'**) *# convert image to greyscale* img = img.point(**lambda** i: 0 **if** i < threshold **else** 255) *# if pixel intensity is less than threshold, change it to else 255* **return** img  
  
grayscale=binarize\_image(**"Lab4-image.png"**,145)  
arr = np.asarray(grayscale)  
  
count=0  
a=np.pad(arr, ((1,1),(1,1)), **'constant'**) *#zero padding to avoid out of bound error when checking for neighbours*rows = a.shape[0] *#width & height of the image i.e. no. of pixels*cols = a.shape[1]  
  
**for** x **in** range(0, rows):  
 **for** y **in** range(0, cols):  
 **if** (a[x,y]==0):  
 **continue  
 else**:  
 conn\_labels = []  
 **if** (a[x - 1, y]!=255) **and** (a[x - 1, y]!=0): *#neglecting the neighbours who dont have any labels (yet)* conn\_labels.append(a[x - 1, y])  
 **if** (a[x + 1, y]!=255) **and** (a[x + 1, y]!=0):  
 conn\_labels.append(a[x + 1, y])  
 **if** (a[x , y-1]!=255) **and** (a[x , y-1]!=0):  
 conn\_labels.append(a[x, y - 1])  
 **if** (a[x , y+1]!=255) **and** (a[x , y+1]!=0):  
 conn\_labels.append(a[x, y + 1])  
  
 **if** len(conn\_labels)==0:  
 count=count+1 *#incrementing label when no neighbour labels found* a[x,y]=count *#assigning the label* **else**:  
 min\_label = min(conn\_labels) *#assigning the minimum label* a[x, y] = min\_label  
  
im=Image.fromarray(a) *#convert array to image*im.show()

**Screenshot:**

