CS-682 Computer Vision Deepak Kanuri(G01070295)

My website: <http://mason.gmu.edu/~dkanuri>

Password: deepak

Part 1:

I created a dialog box using ‘tkinter’ package. It reads the name of the image with extension and produces the histogram of the inputted image. Then the image is shown over which you can move the mouse and can see the position of the pixel and the R,G,B values at that point. After closing this image, the image is opened again, where when the cursor is moved, a 11x11 window is shown, which is the area around the cursor and the mean, standard deviation and intensity value is printed. To close this image press the letter ‘c’.

4. When the cursor is over a white area the mean is 255 in all channels and standard deviation is 0 in all channels. This area is where I can see homogeneous distribution. So a homogeneous area is where we see an extreme value in mean or standard deviation. If the window does not have all white, then it is a inhomogeneous area and the mean and standard deviation varies.

Part 2:

I read all the 99 images, found the histogram of 512 bin for each image after converting the color value to an index. Two empty matrices are created of size 99x99. The chi square distance and intersection of each image with the rest of the images are found. To scale the output, each intersection value is multiplied by 255 to get a range between 0 and 255. For Chi square, the maximum value of the matrix is taken and is divided by 255 to get the scaling factor. This value is then used to divide every value in chi-square matrix. Thus every value in the matrix is between the range 0 and 255.

Part 1 code:

import numpy as np

import cv2

from matplotlib import pyplot as plt

from tkinter import \*

import mpldatacursor

img=None

def execute():

global img

img=e1.get()

master.destroy()

img=cv2.imread(img,1)

img=cv2.resize(img, (1000, 500))

cv2.imshow('nibondara',img)

color = ('b','g','r')

for i,col in enumerate(color):

histr = cv2.calcHist([img],[i],None,[256],[0,256])

plt.plot(histr,color = col)

plt.xlim([0,256])

plt.show()

img=cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)

fig, ax = plt.subplots()

ax.imshow(img, interpolation='none', extent=[0, 1.5\*np.pi, 0, np.pi])

mpldatacursor.datacursor(hover=True, bbox=dict(alpha=1, fc='w'),

formatter='i, j = {i}, {j}\nz = {z}'.format)

plt.show()

img=cv2.resize(img, (500, 500))

img=cv2.cvtColor(img,cv2.COLOR\_RGB2BGR)

cv2.namedWindow('image')

cv2.setMouseCallback('image',hover\_and\_crop)

while True:

cv2.imshow('image',img)

key = cv2.waitKey(1) & 0xFF

if key == ord("c"):

break

cv2.destroyAllWindows()

def hover\_and\_crop(event, x, y, flags, param):

global img

if event == cv2.EVENT\_MOUSEMOVE:

win=img[y-5:y+6,x-5:x+6]

emo= cv2.copyMakeBorder(win,5,5,5,5,cv2.BORDER\_CONSTANT,value=(255,255,255))

cv2.namedWindow('window',cv2.WINDOW\_NORMAL)

cv2.imshow('window',emo)

r,g,b=img[x,y]

print("the intensity value at that point is",(r+g+b)/3)

(means, stds) = cv2.meanStdDev(win)

print("the mean at this point for all channels are",means)

print("the standard deviation at this point for all channels are",stds)

master = Tk()

Label(master, text="Enter Image name with extension P.S.:Please close the windows to continue with operations").grid(row=0)

e1 = Entry(master)

e1.grid(row=0, column=1)

Button(master, text='Execute', command=execute).grid(row=3, column=1, sticky=W, pady=4)

mainloop( )

Part 2 code:

import numpy as np

import cv2

from matplotlib import pyplot as plt

import os

import glob

def convind(img):

img=cv2.imread(img,1)

img=cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)

(r,g,b)=(img[:,:,0], img[:,:,1], img[:,:,2])

conv=((r//32)\*64)+((g//32)\*8)+(b//32)

return conv

def intersect(h1,h2):

mini=0

maxi=0

for i in range(0,len(h1)):

mini+= min(h1[i],h2[i])

maxi+= max(h1[i],h2[i])

return float(mini/maxi)

def chi(h1,h2):

chi=0

for i in range(0,len(h1)):

if (h1[i]+h2[i])>5:

chi+=(((h1[i]- h2[i])\*\*2)/float(h1[i]+h2[i]))

return chi

hhist=[]

for img in glob.glob('ST2MainHall4/\*.jpg'):

histr,bins = np.histogram(convind(img),512,[0,512])

hhist.append(histr)

intersectmat = np.zeros(shape=(99,99))

chimat = np.zeros(shape=(99,99))

for i in range(0,99):

for j in range(0,99):

intersectmat[i][j]=(intersect(hhist[i],hhist[j]))\*255

chimat[i][j]=chi(hhist[i],hhist[j])

scaling=(max(chimat.flatten()))/255

for i in range(0,99):

for j in range(0,99):

chimat[i][j]=(chimat[i][j]/scaling)

plt.imshow(intersectmat)

plt.colorbar()

plt.savefig('Intersection\_Comparison.png')

plt.title('Intersection\_Comparison')

plt.show()

plt.imshow(chimat)

plt.colorbar()

plt.savefig('Chi\_Square\_Comparison.png')

plt.title('Chi\_Square\_Comparison')

plt.show()