Assignment2

CHEN Xinyuan

April 27, 2022

1 Purpose

This assignment provide a satellite image, there is 5 categories in this image. Step1, We can randomly select 10 points per categories. Step2, calculate mean vector and covariance for each class. Step3, get train data for each pixel. Step4, calculate likelihood p(x|k) for all classes. Step4, classify the pixel (i,j) to class c, if p(x|c) = max p(x|k). Step6, repeat step3-5 for all pixel. Finally, we can click ramdom area on this image, and program will tell us the category of this area.

2 Method and Algorithm

For the algorithm, We use Maximum Likelihood Algorithm to estimate the class of selected area. Maximum Likelihood Estimation is simply known as MLE, is a traditional probabilistic approach.

Likelihood can be defined as the possibility that the parameters under consideration may generate the data. A likelihood function is simply the joint probability function of the data distribution. The main idea of Maximum Likelihood is to predict the class label y that maximizes the likelihood of our observed data x.

3 Results

First, we can run the python program click_class.py to select point for each class.

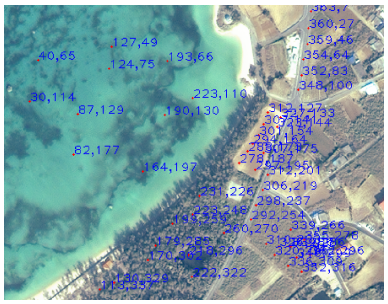

Figure 1: Image after selecting

Then click_class.py will generate the coordinates of selected points with a csv file. It will be used in likelihood.py in the future. After selecting the points, run likelihood.py and click the image, we can get the class info of which area we clicked.

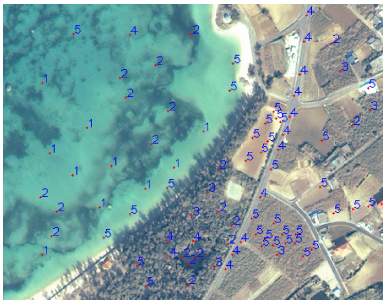

Figure 2: Classification result

4 Program

click_class.py

```
import cv2
import pandas as pd

img = cv2.imread('irabu_zhang1.bmp')

a = []
b = []
R = []
G = []
B = []

def onEVENT_LBUTTONDOWN(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        xy = "%d,%d" % (x, y)
        a.append(x)
        b.append(y)
        R.append(img[y,x,0])
        G.append(img[y,x,1])
        B.append(img[y,x,2])
        cv2.circle(img, (x, y), 1, (0, 0, 255), thickness=-1)
        cv2.putText(img, xy, (x, y), cv2.FONT_HERSHEY_PLAIN,
                    1.0, (255, 0, 0), thickness=1)
        cv2.imshow("image",img)



cv2.namedWindow("image",cv2.WINDOW_NORMAL)
cv2.setMouseCallback("image", onEVENT_LBUTTONDOWN)
cv2.imshow("image", img)
cv2.waitKey(0)

data = [R,G,B]
data = pd.DataFrame(data, index=['R', 'G', 'B'])
data = data.T
data.to_csv('pix.csv')
```

likelihood.py

```
import math
import numpy as np
import pandas as pd
import cv2

def coVariance(X):
    ro, cl = X.shape
    row_mean = np.mean(X, axis=0)
    X_Mean = np.zeros_like(X)
    X_Mean[:] = row_mean
    X_Minus = X - X_Mean
    covarMatrix = np.zeros((cl,cl))
    for i in range(cl):
        for j in range(cl):
            covarMatrix[i,j] = (X_Minus[:,i].dot(X_Minus[:,j].T)) / (ro-1)
    return covarMatrix

def getLikelihood(x,mean,std,x_inv):
    std = math.pow(std,1/2)
    exponent = np.exp(-(x-mean)@(2*x_inv)@(x-mean).T)
    print(exponent)
    return (1/(np.sqrt(2*math.pi)*std))*exponent

def onEVENT_LBUTTONDOWN(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        X = [img[y,x,0],img[y,x,1],img[y,x,2]]

        prob = np.zeros(5)
        for i in range(5):
            prob[i] = getLikelihood(X,m[i],x_norm[i],x_inv[i])
            label = str(np.argmax(prob)+1)
        cv2.circle(img, (x, y), 1, (0, 0, 255), thickness=-1)
        cv2.putText(img, label, (x, y), cv2.FONT_HERSHEY_PLAIN,
                    1.0, (255, 0, 0), thickness=1)
        cv2.imshow("image",img)

datafra = pd.read_csv("./pix_tmp.csv")

field1 = np.array(datafra[["R","G","B"]][:10])
field2 = np.array(datafra[["R","G","B"]][:20])
field3 = np.array(datafra[["R","G","B"]][:30])
field4 = np.array(datafra[["R","G","B"]][:40])
field5 = np.array(datafra[["R","G","B"]][:50])

m = np.zeros([5,3])
m[0] = [np.mean(field1[:,0]),np.mean(field1[:,1]),np.mean(field1[:,2])]
m[1] = [np.mean(field2[:,0]),np.mean(field2[:,1]),np.mean(field2[:,2])]
```

```
m[2] = [np.mean(field3[:,0]),np.mean(field3[:,1]),np.mean(field3[:,2])]
m[3] = [np.mean(field4[:,0]),np.mean(field4[:,1]),np.mean(field4[:,2])]
m[4] = [np.mean(field5[:,0]),np.mean(field5[:,1]),np.mean(field5[:,2])]

covarMatrix1 = coVariance(field1)
covarMatrix2 = coVariance(field2)
covarMatrix3 = coVariance(field3)
covarMatrix4 = coVariance(field4)
covarMatrix5 = coVariance(field5)

x_norm = np.zeros(5)
x_norm[0]=np.linalg.norm(covarMatrix1, ord=None, axis=None, keepdims=False)
x_norm[1]=np.linalg.norm(covarMatrix2, ord=None, axis=None, keepdims=False)
x_norm[2]=np.linalg.norm(covarMatrix3, ord=None, axis=None, keepdims=False)
x_norm[3]=np.linalg.norm(covarMatrix4, ord=None, axis=None, keepdims=False)
x_norm[4]=np.linalg.norm(covarMatrix5, ord=None, axis=None, keepdims=False)

x_inv = np.zeros([5,3,3])
x_inv[0] = np.linalg.inv(covarMatrix1)
x_inv[1] = np.linalg.inv(covarMatrix2)
x_inv[2] = np.linalg.inv(covarMatrix3)
x_inv[3] = np.linalg.inv(covarMatrix4)
x_inv[4] = np.linalg.inv(covarMatrix5)

img = cv2.imread('./irabu_zhang1.bmp')
cv2.namedWindow("image",cv2.WINDOW_NORMAL)
cv2.setMouseCallback("image", on_EVENT_LBUTTONDOWN)
cv2.imshow("image", img)
cv2.waitKey(0)
```