Yoshika Takezawa

5/11/19

CS 558, hw3

Code :

# Yoshika Takezawa

# 5/11/19

# HW 3 : image classifier based on color histograms

# I pledge my honor that I have abided by the Stevens Honor System

import numpy as np

import sys

import os

import cv2

import math

def initialize\_models(path, bins=8):

# return an array of coast, forest, inside city

# each category is set up as (img1 blue, green, red), (img2 blue, green, red), etc

coast = list()

forest = list()

insidecity = list()

for image in os.listdir(path):

if "train" not in image:

continue

img = cv2.imread(os.path.join(path, image), cv2.IMREAD\_COLOR)

imBlue = cv2.calcHist([img], [0], None, [bins], [0, 256])

imGreen = cv2.calcHist([img], [1], None, [bins], [0, 256])

imRed = cv2.calcHist([img], [2], None, [bins], [0, 256])

temp = (imBlue, imGreen, imRed)

if "coast" in image:

coast.append(temp)

elif "forest" in image:

forest.append(temp)

elif "insidecity" in image:

insidecity.append(temp)

else:

continue

# print(coast, forest, insidecity)

return coast, forest, insidecity

def testing(path, coast\_models, forest\_models, insidecity\_models, bins=8):

# uses 1 nearest neighbor

count = 0

total = 0

for image in os.listdir(path):

if "test" not in image:

continue

total += 1

img = cv2.imread(os.path.join(path, image), cv2.IMREAD\_COLOR)

imBlue = cv2.calcHist([img], [0], None, [bins], [0, 256])

imGreen = cv2.calcHist([img], [1], None, [bins], [0, 256])

imRed = cv2.calcHist([img], [2], None, [bins], [0, 256])

small\_dist = math.inf

label = "undef"

for cm in coast\_models:

b, g, r = cm

b\_diff = np.sum(np.power((b - imBlue), 2))

g\_diff = np.sum(np.power((g - imGreen), 2))

r\_diff = np.sum(np.power((r - imRed), 2))

temp = b\_diff + g\_diff + r\_diff

if temp < small\_dist:

small\_dist = temp

label = "coast"

for fm in forest\_models:

b, g, r = fm

b\_diff = np.sum(np.power((b - imBlue), 2))

g\_diff = np.sum(np.power((g - imGreen), 2))

r\_diff = np.sum(np.power((r - imRed), 2))

temp = b\_diff + g\_diff + r\_diff

if temp < small\_dist:

small\_dist = temp

label = "forest"

for icm in insidecity\_models:

b, g, r = icm

b\_diff = np.sum(np.power((b - imBlue), 2))

g\_diff = np.sum(np.power((g - imGreen), 2))

r\_diff = np.sum(np.power((r - imRed), 2))

temp = b\_diff + g\_diff + r\_diff

if temp < small\_dist:

small\_dist = temp

label = "insidecity"

if ("coast" in image) & (label == "coast"):

count += 1

elif ("forest" in image) & (label == "forest"):

count += 1

elif ("insidecity" in image) & (label == "insidecity"):

count += 1

print(image + " is classified as " + label)

print("accuracy = " + str(count/total))

return

def testing3nn(path, coast\_models, forest\_models, insidecity\_models, bins=8):

# uses 3 nearest neighbors

count = 0

total = 0

for image in os.listdir(path):

if "test" not in image:

continue

total += 1

img = cv2.imread(os.path.join(path, image), cv2.IMREAD\_COLOR)

imBlue = cv2.calcHist([img], [0], None, [bins], [0, 256])

imGreen = cv2.calcHist([img], [1], None, [bins], [0, 256])

imRed = cv2.calcHist([img], [2], None, [bins], [0, 256])

small\_dist = [math.inf, math.inf, math.inf]

label = ["undef", "undef", "undef"]

for cm in coast\_models:

b, g, r = cm

b\_diff = np.sum(np.power((b - imBlue), 2))

g\_diff = np.sum(np.power((g - imGreen), 2))

r\_diff = np.sum(np.power((r - imRed), 2))

temp = b\_diff + g\_diff + r\_diff

if temp < max(small\_dist):

ind = small\_dist.index(max(small\_dist))

small\_dist[ind] = temp

label[ind] = "coast"

for fm in forest\_models:

b, g, r = fm

b\_diff = np.sum(np.power((b - imBlue), 2))

g\_diff = np.sum(np.power((g - imGreen), 2))

r\_diff = np.sum(np.power((r - imRed), 2))

temp = b\_diff + g\_diff + r\_diff

if temp < max(small\_dist):

ind = small\_dist.index(max(small\_dist))

small\_dist[ind] = temp

label[ind] = "forest"

for icm in insidecity\_models:

b, g, r = icm

b\_diff = np.sum(np.power((b - imBlue), 2))

g\_diff = np.sum(np.power((g - imGreen), 2))

r\_diff = np.sum(np.power((r - imRed), 2))

temp = b\_diff + g\_diff + r\_diff

if temp < max(small\_dist):

ind = small\_dist.index(max(small\_dist))

small\_dist[ind] = temp

label[ind] = "insidecity"

mode\_label = max(set(label), key=label.count)

if ("coast" in image) & (mode\_label == "coast"):

count += 1

elif ("forest" in image) & (mode\_label == "forest"):

count += 1

elif ("insidecity" in image) & (mode\_label == "insidecity"):

count += 1

print(image + " is classified as " + mode\_label)

print("accuracy = " + str(count/total))

return

if \_\_name\_\_ == '\_\_main\_\_':

dirName = os.path.dirname(\_\_file\_\_)

path = os.path.join(dirName, "ImClass")

c8, f8, i8 = initialize\_models(path, 8)

print("results for 8 bins and 1 nearest neighbor: ")

testing(path, c8, f8, i8, 8)

print("\n -------------------------------------\n")

c4, f4, i4 = initialize\_models(path, 4)

print("results for 4 bins and 1 nearest neighbor: ")

testing(path, c4, f4, i4, 4)

print("\n -------------------------------------\n")

c16, f16, i16 = initialize\_models(path, 16)

print("results for 16 bins and 1 nearest neighbor: ")

testing(path, c16, f16, i16, 16)

print("\n -------------------------------------\n")

c32, f32, i32 = initialize\_models(path, 32)

print("results for 32 bins and 1 nearest neighbor: ")

testing(path, c32, f32, i32, 32)

print("\n -------------------------------------\n")

print("results for 8 bins and 3 nearest neighbors: ")

testing3nn(path, c8, f8, i8, 8)

**Results:**

D:\cs 558\hw3>python ./hw3.py

results for 8 bins and 1 nearest neighbor:

coast\_test1.jpg is classified as coast

coast\_test2.jpg is classified as coast

coast\_test3.jpg is classified as coast

coast\_test4.jpg is classified as coast

forest\_test1.jpg is classified as forest

forest\_test2.jpg is classified as forest

forest\_test3.jpg is classified as forest

forest\_test4.jpg is classified as forest

insidecity\_test1.jpg is classified as forest

insidecity\_test2.jpg is classified as forest

insidecity\_test3.jpg is classified as insidecity

insidecity\_test4.jpg is classified as insidecity

accuracy = 0.8333333333333334

-------------------------------------

results for 4 bins and 1 nearest neighbor:

coast\_test1.jpg is classified as coast

coast\_test2.jpg is classified as coast

coast\_test3.jpg is classified as coast

coast\_test4.jpg is classified as coast

forest\_test1.jpg is classified as forest

forest\_test2.jpg is classified as forest

forest\_test3.jpg is classified as forest

forest\_test4.jpg is classified as forest

insidecity\_test1.jpg is classified as forest

insidecity\_test2.jpg is classified as insidecity

insidecity\_test3.jpg is classified as insidecity

insidecity\_test4.jpg is classified as coast

accuracy = 0.8333333333333334

-------------------------------------

results for 16 bins and 1 nearest neighbor:

coast\_test1.jpg is classified as coast

coast\_test2.jpg is classified as coast

coast\_test3.jpg is classified as coast

coast\_test4.jpg is classified as coast

forest\_test1.jpg is classified as forest

forest\_test2.jpg is classified as forest

forest\_test3.jpg is classified as forest

forest\_test4.jpg is classified as forest

insidecity\_test1.jpg is classified as forest

insidecity\_test2.jpg is classified as forest

insidecity\_test3.jpg is classified as forest

insidecity\_test4.jpg is classified as insidecity

accuracy = 0.75

-------------------------------------

results for 32 bins and 1 nearest neighbor:

coast\_test1.jpg is classified as coast

coast\_test2.jpg is classified as coast

coast\_test3.jpg is classified as coast

coast\_test4.jpg is classified as coast

forest\_test1.jpg is classified as forest

forest\_test2.jpg is classified as forest

forest\_test3.jpg is classified as forest

forest\_test4.jpg is classified as forest

insidecity\_test1.jpg is classified as forest

insidecity\_test2.jpg is classified as forest

insidecity\_test3.jpg is classified as forest

insidecity\_test4.jpg is classified as insidecity

accuracy = 0.75

-------------------------------------

results for 8 bins and 3 nearest neighbors:

coast\_test1.jpg is classified as coast

coast\_test2.jpg is classified as coast

coast\_test3.jpg is classified as coast

coast\_test4.jpg is classified as coast

forest\_test1.jpg is classified as forest

forest\_test2.jpg is classified as coast

forest\_test3.jpg is classified as forest

forest\_test4.jpg is classified as insidecity

insidecity\_test1.jpg is classified as forest

insidecity\_test2.jpg is classified as insidecity

insidecity\_test3.jpg is classified as forest

insidecity\_test4.jpg is classified as coast

accuracy = 0.5833333333333334

The image classifier based on image histograms and Euclidean distance results in an accuracy ranging from 0.533 and 0.833. As the training images did not have a large enough variety, the accuracy was not as stable as it could be. Despite this, it produces decent results. The algorithm bases heavily on brute force, calculating solely on the color distance, and a low level of machine learning.