from keras.models import load\_model

import pickle, os

import cv2

import matplotlib.pyplot as plt\

from sklearn.cluster import KMeans

from sklearn.neighbors import KNeighborsClassifier

import numpy as np

import preprocessing

def visualize\_test(img, model):

pred = model.predict(np.expand\_dims(img / 255.0, axis=0))[0]

pred = (pred \* 255.0).astype(np.uint8)

fig, ax = plt.subplots(2, 1)

ax[0].imshow(img)

ax[1].imshow(pred)

plt.show()

class BoW:

def \_\_init\_\_(self, code\_size=None, n\_clusters=None, centroids\_path=None):

self.code\_size = code\_size

self.kmeans = KMeans(n\_clusters=n\_clusters)

if centroids\_path is not None:

self.kmeans.cluster\_centers\_ = np.load(centroids\_path)

def fit(self, data, save\_path=None):

self.kmeans.fit(data)

if save\_path is not None:

np.save(save\_path, self.kmeans.cluster\_centers\_)

def transform(self, codes):

labels = self.kmeans.predict(codes)

bow = np.zeros(self.kmeans.n\_clusters)

for i in labels:

bow[i] += 1.0

return bow