

```

import pickle
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ResNet50,
preprocess_input
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.models import Sequential
from numpy.linalg import norm
from sklearn.neighbors import NearestNeighbors
import cv2

features_list = pickle.load(open("image_features_embedding.pkl", "rb"))
img_files_list = pickle.load(open("img_files.pkl", "rb"))

print(np.array(features_list).shape)

model = ResNet50(weights="imagenet", include_top=False, input_shape=(224,
224, 3))
model.trainable = False

model = Sequential([model, GlobalMaxPooling2D()])

img = image.load_img('sample/shoes.jpg',target_size=(224,224))
img_array = image.img_to_array(img)
expand_img = np.expand_dims(img_array,axis=0)
preprocessed_img = preprocess_input(expand_img)
result_to_resnet = model.predict(preprocessed_img)
flatten_result = result_to_resnet.flatten()
# normalizing
result_normlized = flatten_result / norm(flatten_result)

neighbors = NearestNeighbors(n_neighbors = 6, algorithm='brute',
metric='euclidean')
neighbors.fit(features_list)

distance, indices = neighbors.kneighbors([result_normlized])

print(indices)

for file in indices[0][1:6]:
    print(img_files_list[file])
    tmp_img = cv2.imread(img_files_list[file])
    tmp_img = cv2.resize(tmp_img, (200,200))
    cv2.imshow("output", tmp_img)
    cv2.waitKey(0)

```