

```

import numpy as np
import pickle as pkl
import tensorflow as tf
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPool2D

from sklearn.neighbors import NearestNeighbors
import os
from numpy.linalg import norm
import streamlit as st

st.header('Fashion Recommendation System')

Image_features = pkl.load(open('Images_features.pkl','rb'))
filenames = pkl.load(open('filenames.pkl','rb'))

def extract_features_from_images(image_path, model):
    img = image.load_img(image_path, target_size=(224,224))
    img_array = image.img_to_array(img)
    img_expand_dim = np.expand_dims(img_array, axis=0)
    img_preprocess = preprocess_input(img_expand_dim)
    result = model.predict(img_preprocess).flatten()
    norm_result = result/norm(result)
    return norm_result
model = ResNet50(weights='imagenet', include_top=False, input_shape=(224,224,3))
model.trainable = False

model = tf.keras.models.Sequential([model,
                                     GlobalMaxPool2D()
                                     ])

neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute', metric='euclidean')
neighbors.fit(Image_features)
upload_file = st.file_uploader("Upload Image")
if upload_file is not None:
    with open(os.path.join('upload', upload_file.name), 'wb') as f:
        f.write(upload_file.getbuffer())
    st.subheader('Uploaded Image')
    st.image(upload_file)
    input_img_features = extract_features_from_images(upload_file, model)
    distance, indices = neighbors.kneighbors([input_img_features])
    st.subheader('Recommended Images')
    col1,col2,col3,col4,col5 = st.columns(5)
    with col1:
        st.image(filenames[indices[0][1]])
    with col2:
        st.image(filenames[indices[0][2]])
    with col3:
        st.image(filenames[indices[0][3]])
    with col4:
        st.image(filenames[indices[0][4]])
    with col5:
        st.image(filenames[indices[0][5]])

```