

```
!pip install datasets
```

Saved successfully!



```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting datasets
  Downloading datasets-2.4.0-py3-none-any.whl (365 kB)
    |████████████████████████████████████████| 365 kB 5.1 MB/s
Requirement already satisfied: dill<0.3.6 in /usr/local/lib/python3.7/dist-packages
Collecting responses<0.19
  Downloading responses-0.18.0-py3-none-any.whl (38 kB)
Collecting multiprocessing
```

```
import os
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
import random
from datasets import Dataset
```

```
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.7/dist-packages
```

```
Downloading xxhash-3.0.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

```
!pip install transformers
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Requirement already satisfied: transformers in /usr/local/lib/python3.7/dist-packages (
```

Saved successfully!

```
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.13,>=0.11.1 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from transformers==4.11.0)
```

```
import tensorflow as tf
import tensorflow_hub as hub
from PIL import Image
import numpy as np
```

```
Uninstalling urllib3-1.24.3:
```

**WARNING: The following packages were previously imported in this runtime:**

## ▼ Took Efficientnet-lite0 for Feature Extraction

```
model_url = "https://tfhub.dev/tensorflow/efficientnet/lite0/feature-vector/2"
```

```
IMAGE_SHAPE = (224, 224) #img- size
```

```
layer = hub.KerasLayer(model_url)
model = tf.keras.Sequential([layer])
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
!cp '/content/drive/MyDrive/visualsimilarity.zip' '/content/'
```

```
!unzip '/content/visualsimilarity.zip'
```

```

inflating: bottoms_resized_png/13460090XMR.png
inflating: bottoms_resized_png/13460096XMR.png
inflating: bottoms_resized_png/13461429GQD.png
inflating: bottoms_resized_png/13461429GQR.png
inflating: bottoms_resized_png/13461429PCD.png
inflating: bottoms_resized_png/13461429PCR.png
inflating: bottoms_resized_png/13462253WCD.png
inflating: bottoms_resized_png/13462253WCR.png
inflating: bottoms_resized_png/13462369OCD.png
inflating: bottoms_resized_png/13462369OCR.png
inflating: bottoms_resized_png/13464005MWD.png
inflating: bottoms_resized_png/13464005MWR.png
inflating: bottoms_resized_png/13468156KAD.png
inflating: bottoms_resized_png/13468156KAR.png
inflating: bottoms_resized_png/134684520JD.png
inflating: bottoms_resized_png/134684520JR.png
inflating: bottoms_resized_png/13470449FFD.png
inflating: bottoms_resized_png/13470449FFR.png
inflating: bottoms_resized_png/13471485JED.png
inflating: bottoms_resized_png/13471485JER.png
inflating: bottoms_resized_png/13476253KTD.png
inflating: bottoms_resized_png/13476253KTR.png
inflating: bottoms_resized_png/13480173LGD.png
inflating: bottoms_resized_png/13480173LGR.png
inflating: bottoms_resized_png/13481506UHD.png
inflating: bottoms_resized_png/13481506UHR.png
inflating: bottoms_resized_png/13482030TID.png
inflating: bottoms_resized_png/13482030TIR.png
inflating: bottoms_resized_png/13485245LFD.png
inflating: bottoms_resized_png/13485245LFR.png
inflating: bottoms_resized_png/13492905VMD.png
inflating: bottoms_resized_png/13492905VMR.png

```

Saved successfully!

```

inflating: bottoms_resized_png/13493522ITD.png
inflating: bottoms_resized_png/13493522ITR.png
inflating: bottoms_resized_png/13498054VRD.png
inflating: bottoms_resized_png/13498054VRR.png
inflating: bottoms_resized_png/13498125OND.png
inflating: bottoms_resized_png/13498125ONR.png
inflating: bottoms_resized_png/13498134WRD.png
inflating: bottoms_resized_png/13498134WRR.png
inflating: bottoms_resized_png/13512944AOD.png
inflating: bottoms_resized_png/13512944AOR.png
inflating: bottoms_resized_png/13513815TJD.png
inflating: bottoms_resized_png/13513815TJR.png
inflating: bottoms_resized_png/13517520KHD.png
inflating: bottoms_resized_png/13517520KHR.png
inflating: bottoms_resized_png/13524052ORD.png
inflating: bottoms_resized_png/13524052ORR.png
inflating: bottoms_resized_png/13528768GRD.png
inflating: bottoms_resized_png/13528768GRR.png
inflating: bottoms_resized_png/13529169NVD.png
inflating: bottoms_resized_png/13529169NVR.png
inflating: bottoms_resized_png/13529187UGD.png
inflating: bottoms_resized_png/13529187UGR.png
inflating: bottoms_resized_png/13529612CUD.png
inflating: bottoms_resized_png/13529612CUR.png
inflating: bottoms_resized_png/13530476UDD.png
inflating: bottoms_resized_png/13530476UDR.png
inflating: bottoms_resized_png/13531518MKD.png

```

Saved successfully!

## Extracting feature vector for each vecto and storing it as JSON object

```

def extract(file):
    file = Image.open(file).convert('L').resize(IMAGE_SHAPE)
    #display(file)

    file = np.stack((file,)*3, axis=-1)

    file = np.array(file)/255.0

    embedding = model.predict(file[np.newaxis, ...])
    #print(embedding)
    vgg16_feature_np = np.array(embedding)
    flattened_feature = vgg16_feature_np.flatten()

    #print(len(flattened_feature))
    #print(flattened_feature)
    #print('-----')
    return flattened_feature

vector_dict = {}
path_dict = {}

```

```
for ind, item in enumerate(os.listdir( '/content/bottoms_resized_png' )):  
    print(ind,item)  
    path = os.path.join('/content/bottoms_resized_png', item)  
    #img = image.load_img(path, target_size=(32, 32))  
  
    #x = image.img_to_array(img)  
    #x = np.expand_dims(x, axis=0)  
    #images = np.vstack([x])[0].tolist()  
    vector = extract(path)  
    path_dict[item[:-4]] = path  
    vector_dict[item[:-4]] = vector.tolist()
```

```
102 13224618QUR.png  
103 13586192QXD.png  
104 13586572WAD.png  
105 35468811TWD.png  
106 13587435BQD.png  
107 13585856KWD.png  
108 13110484CLR.png  
109 13468156KAR.png  
110 13587314XJR.png  
111 13587597BDD.png  
112 13586498PUD.png  
113 13588959GJR.png  
114 13585814LFR.png  
115 35469133OUD.png  
116 35468684LHD.png
```

Saved successfully!



```
117 13588111KXR.png  
120 35468935XLR.png  
121 13581217MCD.png  
122 35467798JQD.png  
123 13524052ORD.png  
124 13586399JIR.png  
125 13585560NFD.png  
126 13588022DJD.png  
127 13586098VWD.png  
128 13587355WLD.png  
129 13583370LCD.png  
130 13587499WKR.png  
131 13585109SXD.png  
132 13586939KAD.png  
133 13577361LLR.png  
134 135799880FR.png  
135 13587559SSD.png  
136 35468920XHR.png  
137 13589244AJR.png  
138 35468641USD.png  
139 35468998UGR.png  
140 35468462BJD.png  
141 13470449FFR.png  
142 35468470GCD.png  
143 13587189HMD.png  
144 13585804KIR.png  
145 13587741KUD.png
```

```

146 13585861XGR.png
147 35469325AMD.png
148 13584807VGR.png
149 13468156KAD.png
150 13586162UOR.png
151 13628197SNR.png
152 13586129QMD.png
153 13632427XJR.png
154 13589722RJD.png

155 13586416QDR.png
156 13529169NVR.png
157 13581097NPR.png
158 35469830CLD.png
159 13582867PNR.png
160 354697777777

```

```

from pprint import pprint
pprint(vector_dict['35468464IMD'])

```

```

[0.0,
 0.0,
 1.2059400081634521,
 0.02823033183813095,
 0.008607905358076096,
 0.0644526481628418,
 0.05020752176642418,
 0.6326779127120972,

```

Saved successfully!

```

0.0,
0.0,
0.0,
0.008355788886547089,
0.0,
0.011677905917167664,
0.0,
0.0,
0.0,
0.0,
0.008628073148429394,
0.0,
0.0,
0.12080425769090652,
0.0,
0.0,
0.3310277462005615,
0.12133216857910156,
0.44846251606941223,
0.8173195719718933,
0.6641164422035217,
0.0,
0.11397874355316162,
0.0,
0.9434801936149597,
0.23542749881744385,
0.0,

```



```
from scipy.spatial import distance
```

```
metric = 'cosine'
```

```
!pip install cv2
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/pub>  
ERROR: Could not find a version that satisfies the requirement cv2 (from versions: none)  
ERROR: No matching distribution found for cv2

```
import cv2
from google.colab.patches import cv2_imshow
img1= cv2.imread('/content/bottoms_resized_png/13079565VVR.png')

cv2_imshow(img1)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Saved successfully!



```
import cv2
```



```
from google.colab.patches import cv2_imshow
img1= cv2.imread('/content/bottoms_resized_png/13079565VVD.png')

cv2_imshow(img1)

cv2.waitKey(0)

cv2.destroyAllWindows()
```



Saved successfully!



```
import cv2
from google.colab.patches import cv2_imshow
img1= cv2.imread('/content/bottoms_resized_png/13322221VND.png')

cv2_imshow(img1)

cv2.waitKey(0)

cv2.destroyAllWindows()
```



```
dc = distance.cdist([jean1], [jean1], metric)[0]
print(dc)
print("the distance between jean1 and the jean1 is {}".format(dc))
```

Saved successfully!



the jean1 is [0.]

```
dc = distance.cdist([jean1], [jean2], metric)[0]
print(dc)
print("the distance between jean1 and the jean2 is {}".format(dc))
```

```
[0.29833915]
the distance between jean1 and the jean2 is [0.29833915]
```

```
dc = distance.cdist([jean1], [skirt], metric)[0]
print(dc)
print("the distance between jean1 and the skirt is {}".format(dc))
```

```
[0.45481624]
the distance between jean1 and the skirt is [0.45481624]
```

## ▼ Performing Query Based search for particular image

```
from scipy.spatial import distance
metric = 'cosine'
```

```
def find_similar_images(img_path, vector_dict):
```

```
def find_similar_images(img_path, vector_dict):
```

```
    feature_vector = extract(img_path)
    similarity_dict = {}
    for key,value in vector_dict.items():

        sim = distance.cdist([feature_vector], [value], metric)[0]
        similarity_dict[key] = sim
    sorted_similarity = sorted(similarity_dict.items(), key=lambda x: x[1], reverse=False)
    suggested_ls = []
    for i in range(0,10):
        suggested_ls.append(sorted_similarity[i][0])
        print(sorted_similarity[i][0],sorted_similarity[i][1])
    return suggested_ls
```

```
suggested_ls = find_similar_images('/content/bottoms_resized_png/13586062VER.png', vector_dict)
```

```
13586062VER [0.]
13588319VTD [0.14460529]
13643117AAR [0.17039089]
13586472KMD [0.17595911]
13428218QFD [0.17746354]
13573117RRD [0.17778571]
13584405AHD [0.17922314]
13586572WAR [0.179826]
13585945CJD [0.18153457]
13585406MCD [0.18182743]
```

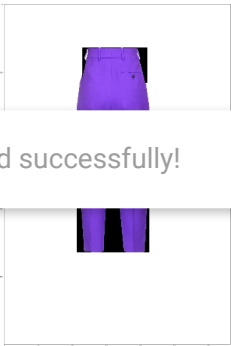
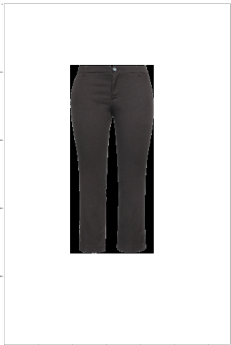
Saved successfully!

```
['13079565VVR', '13589082JLR', '13400677WER', '13586212XER', '13586197WBD', '13586572WA
```



```
import numpy as np
import matplotlib.pyplot as plt
import cv2

fig = plt.figure(figsize=(200, 200))
columns = 2
rows = 5
for i in range(1, columns*rows +1):
    path = path_dict[suggested_ls[i-1]]
    img = cv2.imread(path)
    fig.add_subplot(rows, columns, i)
    plt.imshow(img)
plt.show()
```



Saved successfully! ✕



```
# code for displaying multiple images in one figure

#import libraries
import cv2
from matplotlib import pyplot as plt

# create figure
fig = plt.figure(figsize=(100,100))

# setting values to rows and column variables
rows = 5
columns = 2

# reading

Image1 = cv2.imread('/content/bottoms_resized_png/13589082JLR.png')
Image2 = cv2.imread('/content/bottoms_resized_png/13400677WER.png')
Image3 = cv2.imread('/content/bottoms_resized_png/13400677WER.png')
Image4 = cv2.imread('/content/bottoms_resized_png/13586197WBD.png')

fig.add_subplot(rows, columns, 1)

# showing image
plt.imshow(Image1)
plt.axis('off')
plt.title("First")

# Adds a subplot at the 2nd position
fig.add_subplot(rows, columns, 2)

# showing image
plt.imshow(Image2)
plt.axis('off')
plt.title("Second")

# Adds a subplot at the 3rd position
fig.add_subplot(rows, columns, 3)

# showing image
plt.imshow(Image3)
plt.axis('off')
plt.title("Third")

# Adds a subplot at the 4th position
fig.add_subplot(rows, columns, 4)

# showing image
```

Saved successfully!



```
plt.imshow(Image4)
plt.axis('off')
plt.title("Fourth")
```

```
Text(0.5, 1.0, 'Fourth')
```



Third

Fourth

Saved successfully!



## Method2: Using VGG-16 feature extractor and using L2

```

from PIL import Image
from feature_extractor import FeatureExtractor
from pathlib import Path
import numpy as np

if __name__ == '__main__':
    fe = FeatureExtractor()

    for img_path in sorted(Path("/content/bottoms_resized_png").glob("*.png")):
        print(img_path) # e.g., ./static/img/xxx.jpg
        feature = fe.extract(img=Image.open(img_path))
        feature_path = Path("/content/feature") / (img_path.stem + ".np") # e.g., ./static/feature/xxx.npy
        np.save(feature_path, feature)

```

```

/content/bottoms_resized_png/13585972BLR.png
/content/bottoms_resized_png/13585972CAD.png
/content/bottoms_resized_png/13585972CAR.png
/content/bottoms_resized_png/13585975IND.png
/content/bottoms_resized_png/13585975INR.png
/content/bottoms_resized_png/13585983RQD.png
/content/bottoms_resized_png/13585983RQR.png
/content/bottoms_resized_png/13585998HWD.png
/content/bottoms_resized_png/13585998HWR.png
/content/bottoms_resized_png/13586005VED.png

```

Saved successfully!

```

3586005VER.png
3586013SED.png
3586013SER.png
/content/bottoms_resized_png/13586023VQD.png
/content/bottoms_resized_png/13586023VQR.png
/content/bottoms_resized_png/13586043JCD.png
/content/bottoms_resized_png/13586043JCR.png
/content/bottoms_resized_png/13586043TOD.png
/content/bottoms_resized_png/13586043TOR.png
/content/bottoms_resized_png/13586044RQD.png
/content/bottoms_resized_png/13586044RQR.png
/content/bottoms_resized_png/13586048CLD.png
/content/bottoms_resized_png/13586048CLR.png
/content/bottoms_resized_png/13586055AWD.png
/content/bottoms_resized_png/13586055AWR.png
/content/bottoms_resized_png/13586056MDD.png
/content/bottoms_resized_png/13586056MDR.png
/content/bottoms_resized_png/13586062VED.png
/content/bottoms_resized_png/13586062VER.png
/content/bottoms_resized_png/13586067UGD.png
/content/bottoms_resized_png/13586067UGR.png
/content/bottoms_resized_png/13586069LPD.png
/content/bottoms_resized_png/13586069LPR.png
/content/bottoms_resized_png/13586086SID.png
/content/bottoms_resized_png/13586086SIR.png
/content/bottoms_resized_png/13586089CBD.png
/content/bottoms_resized_png/13586089CBR.png
/content/bottoms_resized_png/13586091AMD.png
/content/bottoms_resized_png/13586091AMR.png
/content/bottoms_resized_png/13586091NKN.png

```

```

/content/bottoms_resized_png/13586094NKR.png
/content/bottoms_resized_png/13586094NKR.png
/content/bottoms_resized_png/13586095TND.png
/content/bottoms_resized_png/13586095TNR.png
/content/bottoms_resized_png/13586098VWD.png
/content/bottoms_resized_png/13586098VWR.png
/content/bottoms_resized_png/13586101XHD.png
/content/bottoms_resized_png/13586101XHR.png

/content/bottoms_resized_png/13586111LCD.png
/content/bottoms_resized_png/13586111LCR.png
/content/bottoms_resized_png/13586119CUD.png
/content/bottoms_resized_png/13586119CUR.png
/content/bottoms_resized_png/13586119EDD.png
/content/bottoms_resized_png/13586119EDR.png
/content/bottoms_resized_png/13586128HWD.png
/content/bottoms_resized_png/13586128HWR.png
/content/bottoms_resized_png/13586129QMD.png
/content/bottoms_resized_png/13586129QMR.png
/content/bottoms_resized_png/13586135RCD.png

```

```
!cp -r '/content/feature' '/content/drive/MyDrive/'
```

```

features = []
img_paths = []
for feature_path in Path("/content/feature").glob("*.npy"):
    features.append(np.load(feature_path))
    img_paths.append(Path("/content/bottoms_resized_png") / (feature_path.stem + ".png"))

```

Saved successfully!

```

fe = FeatureExtractor()
query = fe.extract(img=Image.open('/content/bottoms_resized_png/13589082JLR.png'))
dists = np.linalg.norm(features-query, axis=1) # L2 distances to features
ids = np.argsort(dists)[:10] # Top 30 results
scores = [(dists[id], img_paths[id]) for id in ids]

```

```
print(str(scores[1][1]))
```

```
/content/bottoms_resized_png/13224618QUR.png
```

```

import numpy as np
import matplotlib.pyplot as plt
import cv2

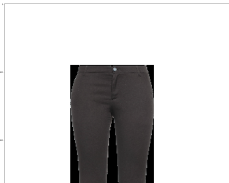
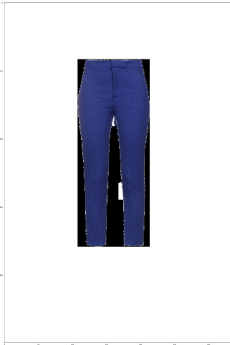
fig = plt.figure(figsize=(200, 200))
columns = 2
rows = 5
for i in range(1, columns*rows +1):
    path = str(scores[i-1][1])
    img = cv2.imread(path)
    fig.add_subplot(rows, columns, i)
    plt.imshow(img)
plt.show()

```



Saved successfully!

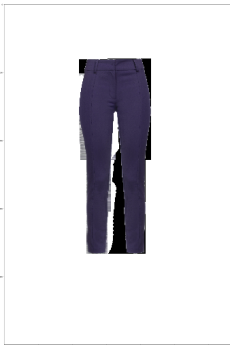
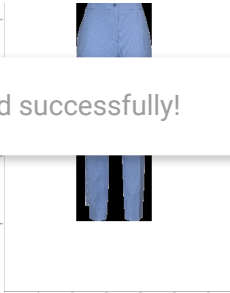




```
home = '/content/bottoms_resized_png/35469965SND.png'  
print(home[-15:-4])
```

35469965SND

Saved successfully! ×



✓ 17s completed at 23:17

● ✕