```
from keras.preprocessing.image import img_to_array
from keras.models import load_model
import numpy as np
import argparse
import imutils
import pickle
import cv2
import os
from keras.applications.vgg19 import VGG19
from keras.models import Model
from keras.layers import Dense, GlobalAveragePooling2D, Dropout
from keras.optimizers import SGD
import matplotlib.pyplot as plt
```

```
Using TensorFlow backend.
```

```
# Run this cell to mount your Google Drive.
from google.colab import drive
drive.mount('/content/drive')
```

```
!unzip drive/My\ Drive/final_data-fynd.zip
```

```
labels = ['backstrap_BV',
 'backstrap_NBV',
 'buckle_BV',
 'buckle_NBV',
 'hook&look_BV',
 'hook&look_NBV',
 'lace_up_BV',
 'lace_up_NBV',
 'slip_on_BV',
 'slip_on_NBV',
 'zipper_BV',
 'zipper_NBV']
```

```
vgg_model = VGG19(weights='imagenet', include_top=False)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0630 08:11:12.310020 139902118483840 deprecation_wrapper.py:119] From /usr/lo

W0630 08:11:12.360449 139902118483840 deprecation_wrapper.py:119] From /usr/lo

W0630 08:11:12.368828 139902118483840 deprecation_wrapper.py:119] From /usr/lo

W0630 08:11:12.434321 139902118483840 deprecation_wrapper.py:119] From /usr/lo

Downloading data from https://github.com/fchollet/deep-learning-models/release
80142336/80134624 [==============================] - 1s 0us/step
W0630 08:11:14.500201 139902118483840 deprecation_wrapper.py:119] From /usr/lo

W0630 08:11:14.501507 139902118483840 deprecation_wrapper.py:119] From /usr/lo
```

```
x = vgg_model.output
x = GlobalAveragePooling2D()(x)

# add fully-connected layer
```

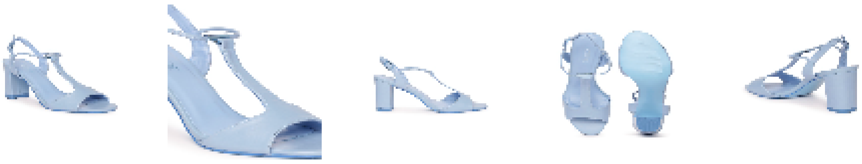
```
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)

# add output layer
predictions = Dense(12, activation='softmax')(x)

model = Model(inputs=vgg_model.input, outputs=predictions)
model.load_weights('drive/My Drive/fine_tune_shoes_multiclass.best.hdf5')
```

```
W0630 08:11:19.682404 139902118483840 deprecation.py:506] From /usr/local/lib/
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - ke
```

```
def identify_type(images , labels):

    w=300
    h=300
    fig=plt.figure(figsize=(8, 8))
    columns = 5
    rows = 1
    for i in range(1, len(images)+1):
        img = cv2.imread(images[i-1])
        img = cv2.resize(img, (w,h))
        fig.add_subplot(rows, columns, i)
        plt.axis('off')
        plt.imshow(img)
    plt.show()

    best_image_index = 0
    type_score = dict()
    for i,j in enumerate(images) :
        image = cv2.imread(j)
        image = cv2.resize(image, (128,128))
        image = img_to_array(image)
        image = np.expand_dims(image, axis=0)
        proba = model.predict(image)[0]
        idxs = np.argsort(proba)[::-1][:1]
        type_score[i] = {proba[idxs[0]] : labels[idxs[0]]}

    print (type_score)
    best_image = dict()
    for key,i in type_score.items() :
        type_image  = list(i.values())[0]
        score = list(i.keys())[0]
        print(type_image.split('_')[1])
        if type_image.split('_')[1] == 'BV':
            best_image[score] = type_image.split('_')[0]
    temp_dict = dict()
    for key,value in type_score.items():
        temp_dict[list(value.keys())[0]]=list(value.values())[0]

    if len(best_image) == 1:
        print ( 'Type of footwear = ',(list(best_image.values())[0]).split('_')[0])
        best_image_index = list(temp_dict.keys()).index(list(best_image.keys())[0])+1
        print ('Best Image Index =' , best_image_index )

    if len(best_image) > 1 :
        max_score = max(list(best_image.keys()))
        best_image_index = list(temp_dict.keys()).index(max_score)+1
        print ('Type of footwear = ',best_image[max_score].split('_')[0])
        print ('Best Image Index =' , best_image_index )

    if len(best_image) == 0 :
        max_score = max(list(temp_dict.keys()))
        best_image_index = list(temp_dict.keys()).index(max_score)+1
        print ('Type of footwear = ',temp_dict[max_score].split('_')[0])
        print ('Best Image Index = ', best_image_index)

    print ("Best Image ")
    img = cv2.imread(images[best_image_index-1])
    img = cv2.resize(img, (w,h))
```

```
        plt.axis('off')
        plt.imshow(img)
```

```
    images = ['data/BV/backstrap/view_1-01e90b3bf103420892f98767e10bffd4.jpg',
              'data/NBV/backstrap/view_2-01e90b3bf103420892f98767e10bffd4.jpg',
              'data/NBV/backstrap/view_3-01e90b3bf103420892f98767e10bffd4.jpg',
              'data/NBV/backstrap/view_4-01e90b3bf103420892f98767e10bffd4.jpg',
              'data/NBV/backstrap/view_5-01e90b3bf103420892f98767e10bffd4.jpg']
    identify_type(images , labels)
```



```
{0: {1.0: 'backstrap_BV'}, 1: {1.0: 'zipper_NBV'}, 2: {0.9978325: 'backstrap_N
BV
NBV
NBV
NBV
NBV
{1.0: 'backstrap_NBV', 0.9978325: 'backstrap_NBV'}
Type of footwear =  backstrap
Best Image Index = 1
Best Image
```