In [1]:
```
!pip install pyprind
```

```
Requirement already satisfied: pyprind in /usr/local/lib/python3.6/dist-packages (2.11.2)
```

In [2]:
```
from PIL import Image
import itertools
import numpy as np
import pandas as pd
import warnings
import nltk
import math
import time
import re
import os
import pickle
import pyprind
#Deep-Learning Library
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dropout, Flatten, Dense
from keras import applications
from keras.preprocessing.image import load_img,img_to_array
from keras import backend as k
from keras.applications.vgg16 import preprocess_input
```

```
Using TensorFlow backend.
```

In [3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In [4]:
```
path="/content/drive/My Drive/Colab Notebooks/41k_images/"

data_cnn_vgg16=path+"41k_image_feature.npy"
data_filenames_vgg16=path+"41k_image_asins.npy"
```

In [5]:
```
# Start Clock for time tracking
start_time = time.clock()
# dimensions of our images.
img_width, img_height = 299, 299
epochs = 50
batch_size = 1
#print("Started")
```

In [6]:
```
# The Below Function Genrates and Return Numpy 2-D Array for Training Images
# dimensions of our images.
img_width, img_height = 224, 224

top_model_weights_path = 'bottleneck_fc_model.h5'
#train_data_dir =
nb_train_samples = 41022
epochs = 50
batch_size = 1
image_size = (224, 224)
print("Started")

def save_bottlebeck_features():

    #Function to compute VGG-16 CNN for image feature extraction.

    asins = []
    features=[]
    datagen = ImageDataGenerator(rescale=1. / 255)
    print("asins and datagen done")
    # build the VGG16 network
    model = applications.VGG16(include_top=False, weights='imagenet')
    print("Model settled")
    generator = datagen.flow_from_directory(
        path,
        target_size=(img_width, img_height),
        batch_size=batch_size,
        class_mode=None,
        shuffle=False)
print("Genrator work finish")
j=0
bar=pyprind.ProgBar(41022)
for i in generator.filenames:

    image_path=path+i
    try:
      img = load_img(image_path, target_size=image_size)
      asins.append(i[14:24])
      x = img_to_array(img)
      x = np.expand_dims(x, axis=0)
      x = preprocess_input(x)
      feature = model.predict(x)
      flat = feature.flatten()
      features.append(flat)
    except:
      j+=1
    bar.update()
print(j)
print(len(asins))
#print(asins)
features=np.array(features)
print(features.shape)

print("For loop ended")
print(generator)
#print("training start")
"""
bottleneck_features_train = model.predict_generator(generator, nb_train_samples // batch_size)
print("training finised")"""

#print("reshape the trained vector")
#bottleneck_features_train = bottleneck_features_train.reshape((41022,25088))
#print("Trainning done")
np.save(open(data_cnn_vgg16, 'wb'), features)
np.save(open(data_filenames_vgg16, 'wb'), np.array(asins))
print("npy file saved")


save_bottlebeck_features()

print(time.clock() - start_time, "seconds")
```

```
Started
asins and datagen done
Model settled
Found 41022 images belonging to 1 classes.
Genrator work finish
```

```
0% [##############################] 100% | ETA: 00:00:00
Total time elapsed: 05:49:48
```

```
410
40612
(40612, 25088)
For loop ended
<keras.preprocessing.image.DirectoryIterator object at 0x7f7de4042e48>
npy file saved
40454.357902 seconds
```

In [6]:

In [6]:

In [ ]:

In [6]:

In [6]:

In [6]: