FLIP ROBO

Image Scrapping and Classification Project

Submitted by:

C S Manjunath Reddy

ACKNOWLEDGMENT

I would like to thank FlipRobo for giving me this opportunity. The DataTrained institute classes helped me to solve this problem.

The language used for the project Python, Selenium and BeautifulSoup for Image scrapping, Tensorflow and Keras used to build the image classification model.

INTRODUCTION

- Business Problem Framing

  Images are one of the major sources of data in the field of data science and AI. This field is making appropriate use of information that can be gathered through images by examining its features and details.

- Conceptual Background of the Domain Problem

  Image classification for scrapped images from the e commerce portal

- Motivation for the Problem Undertaken

The idea behind this project is to build a deep learning-based Image Classification model on images that will be scraped from e-commerce portal. This is done to make the model more and more robust.

Analytical Problem Framing

- Data Sources and their formats

  Data is scrapped from ecommerce website Amazon.in

- Data Preprocessing Done

  The images will have different sizes. So we with the help of ImageDataGenerator imported from tensorflow.keras will help us to rescale all the images for the same size. Here we took 224 and 224 values for Height and weight.

- Hardware and Software Requirements and Tools Used

Selenium and BeautifulSoup is used for Image scrapping(Data Collection).

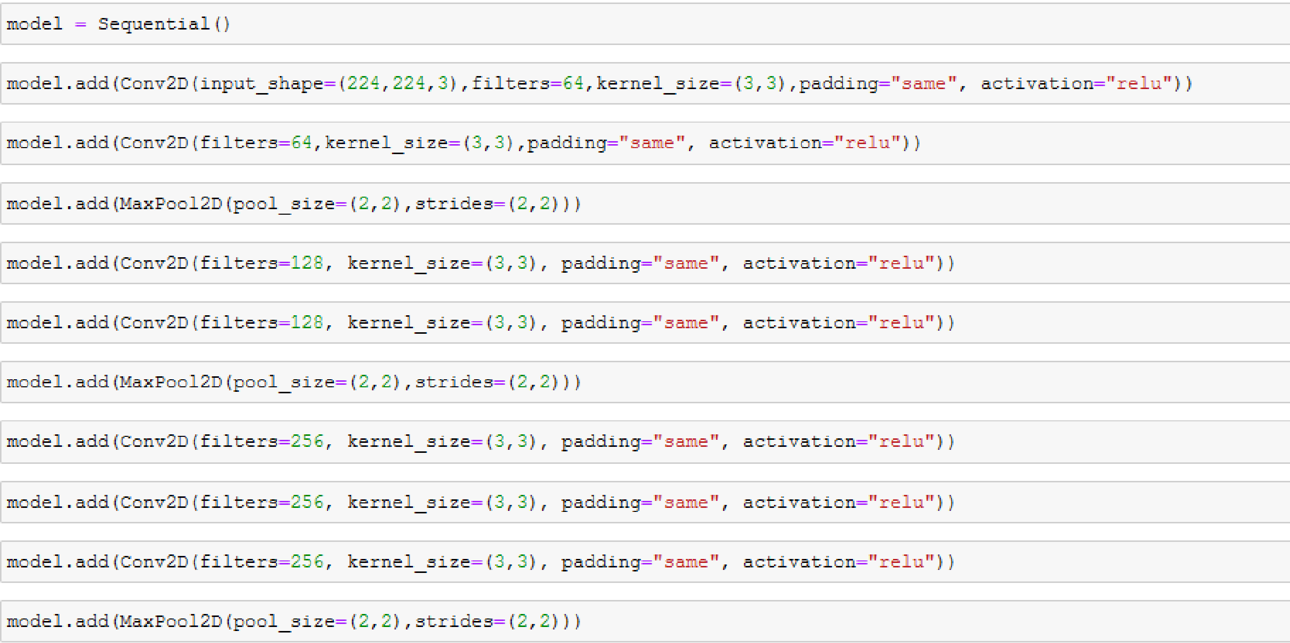
Tensorflow and Keras is used to build the vgg16 CNN model.

Model/s Development and Evaluation

- Testing of Identified Approaches (Algorithms)

Used Vgg16 convolutional Neural Network model is used to train the data for image classification. This model is built with 16 CNN layers with only 3x3 kernels.

The network comprises of 134,272,835 parameters.



```
model = Sequential()
```

```
model.add(Conv2D(input_shape=(224,224,3),filters=64,kernel_size=(3,3),padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
```

```
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
```

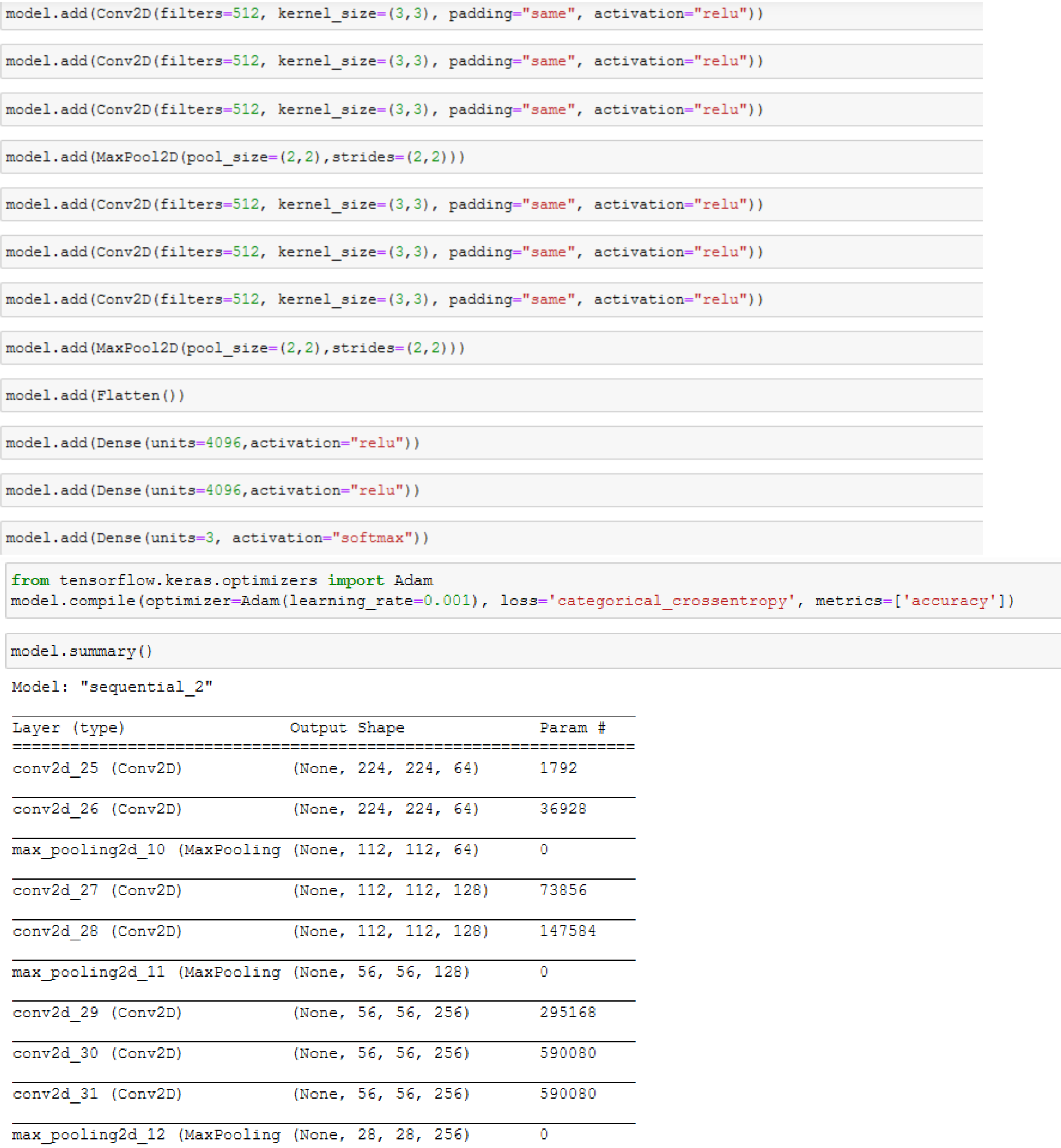
```
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```



```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
```

```
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
```

```
model.add(Flatten())
```

```
model.add(Dense(units=4096,activation="relu"))
```

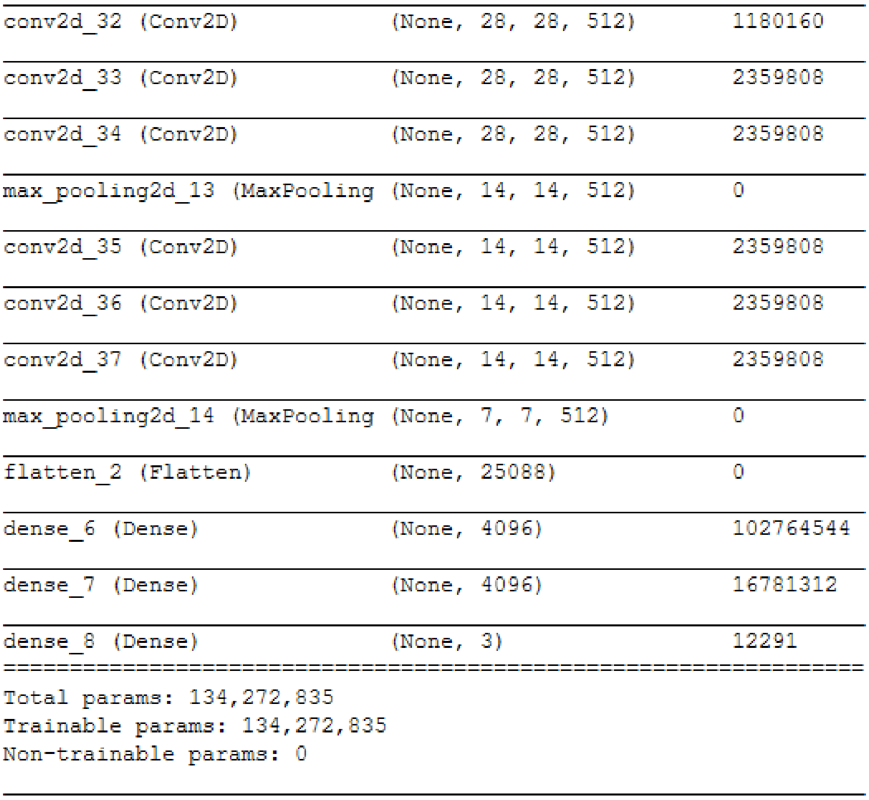
```
model.add(Dense(units=4096,activation="relu"))
```

```
model.add(Dense(units=3, activation="softmax"))
```

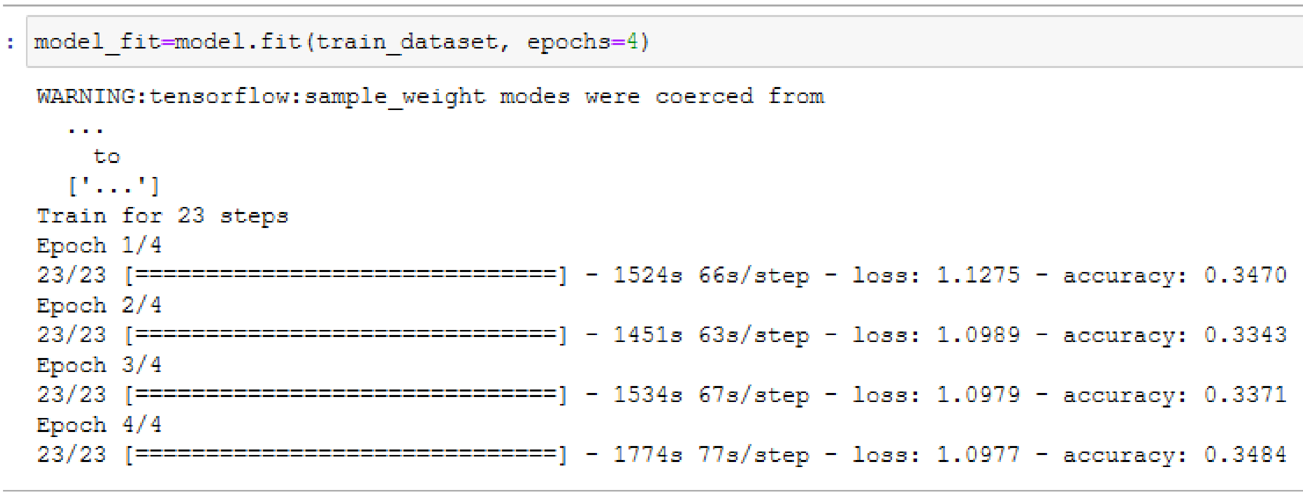
```
from tensorflow.keras.optimizers import Adam
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_25 (Conv2D)           (None, 224, 224, 64)      1792
_________________________________________________________________
conv2d_26 (Conv2D)           (None, 224, 224, 64)      36928
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 112, 112, 128)     73856
_________________________________________________________________
conv2d_28 (Conv2D)           (None, 112, 112, 128)     147584
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 56, 56, 128)       0
_________________________________________________________________
conv2d_29 (Conv2D)           (None, 56, 56, 256)       295168
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 56, 56, 256)       590080
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 56, 56, 256)       590080
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 28, 28, 256)       0
```



```
conv2d_32 (Conv2D)           (None, 28, 28, 512)       1180160
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 28, 28, 512)       2359808
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 28, 28, 512)       2359808
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_37 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 7, 7, 512)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_6 (Dense)              (None, 4096)              102764544
_________________________________________________________________
dense_7 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dense_8 (Dense)              (None, 3)                 12291
=================================================================
Total params: 134,272,835
Trainable params: 134,272,835
Non-trainable params: 0
_________________________________________________________________
```



```
: model_fit=model.fit(train_dataset, epochs=4)

  WARNING:tensorflow:sample_weight modes were coerced from
    ...
      to
    ['...']
  Train for 23 steps
  Epoch 1/4
  23/23 [==============================] - 1524s 66s/step - loss: 1.1275 - accuracy: 0.3470
  Epoch 2/4
  23/23 [==============================] - 1451s 63s/step - loss: 1.0989 - accuracy: 0.3343
  Epoch 3/4
  23/23 [==============================] - 1534s 67s/step - loss: 1.0979 - accuracy: 0.3371
  Epoch 4/4
  23/23 [==============================] - 1774s 77s/step - loss: 1.0977 - accuracy: 0.3484
```

- Key Metrics for success in solving problem under consideration

Key Metrics are loss rate and accuracy, the low the loss rate and high the accuracy considered as best model.

CONCLUSION

- Key Findings and Conclusions of the Study

  Trained the data with built vgg16 cnn model and checked the accuracy and loss rate.

- Learning Outcomes of the Study in respect of Data Science

  With this project I learnt how to build CNN model for image classification.