



Image Classification Model

Submitted by:

RAHUL SINGH

ACKNOWLEDGMENT

Image classification Project is a deep learning-based Image Classification model. The Images of multiple products like saree, trousers for men, jeans for men has been download from Amazon website to build the model.

The car data has been scrap using selenium web scrapping method.

The Image classification model is a deep learning-based Image Classification model.

INTRODUCTION

- **Business Problem Framing**

The Image classification model is basically predict the image of multiple products downloaded for amazon website.

- **Conceptual Background of the Domain Problem**

The deep learning model is related to predict the image after passing the image details for which model has been build.

- **Review of Literature**

This model is basically build by first do the image scrapping using the automated selenium tool then after that building the model using the scrapped image.

The image has been download from the amazon website. The total image scraped is approx. 150 for each product and using the image model has been build.

- **Motivation for the Problem Undertaken**

The main challenge to build this model is download the image from the amazon.

After downloading the image for multiple product building the deep learning model using the image.

Analytical Problem Framing

- Data Sources and their formats

The Dataset is in jpg format of multiple product that has been download using selenium web driver and then used to build the deep learning model.

```
time.sleep(3)
#fetching the product url
prd_url=[]
for q in driver.find_elements_by_xpath('//ul[@class="a-pagination"]/li/a'):
    prd_url.append(q.get_attribute('href'))

img_urls = []
img_data = []
for i in prd_url[:3]:
    driver.get(i)

    images = driver.find_elements_by_xpath('//img[@class="s-image"]')
    for image in images:
        source= image.get_attribute('src')
        if source is not None:
            if(source[0:4] == 'http'):
                img_urls.append(source)

for i in range(len(img_urls)):
    if i >= 160:
        break
    print("Downloading {0} of {1} images" .format(i, 160))
    response= requests.get(img_urls[i])
    file = open(r"D:\Amz Product\Saree"+str(i)+".jpg", "wb")
    file.write(response.content)
```

```
Downloading 0 of 160 images
Downloading 1 of 160 images
Downloading 2 of 160 images
Downloading 3 of 160 images
Downloading 4 of 160 images
Downloading 5 of 160 images
Downloading 6 of 160 images
Downloading 7 of 160 images
Downloading 8 of 160 images
Downloading 9 of 160 images
Downloading 10 of 160 images
Downloading 11 of 160 images
Downloading 12 of 160 images
```

- Data Pre-processing Done

The Data Pre-processing consist of below mentioned points:

1. The model has built on Google Colab.
2. Using the InceptionV3 to process the image.
3. The size of the image has been set [224, 224] for the image model building.

4. Softmax activation has been used to build the model as It has more than 2 type of images.
5. Building the model and the check the model loss and accuracy with the help of graph.
6. Predicting the model with by passing the image and then check It will predict correctly.

- **Hardware and Software Requirements and Tools Used**

The Tool used to build the model is anaconda jupyter, Google colab and selenium web driver.

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

Multiple problem faced during the model building. All are listed bellows.

1. Downloading the image from the websites for model building.
2. Checking the model accuracy by passing the random image and make model predict.

- **Testing of Identified Approaches (Algorithms)**

This is a deep learning-based Image Classification problem as the target column has three kind of image so deep learning method has been used to build model.

InceptionV3 has been used to build the image classification model.

- Run and Evaluate selected models

1. InceptionV3: Inception v3 is a convolutional neural network for assisting in image analysis and object detection.

```
# \provide the same target size as initialised for the image size
df_set = df_datagen.flow_from_directory('/content/drive/MyDrive/AmazonData',
                                       target_size = (224, 224),
                                       batch_size = 16,
                                       class_mode = 'categorical')
```

Found 463 images belonging to 3 classes.

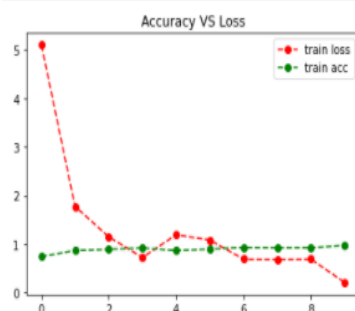
```
# fit the model
# Run the cell. It will take some time to execute
r = model.fit_generator(
    df_set,
    epochs=10
)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

```
Epoch 1/10
29/29 [=====] - 135s 4s/step - loss: 5.0987 - accuracy: 0.7430
Epoch 2/10
29/29 [=====] - 49s 2s/step - loss: 1.7749 - accuracy: 0.8683
Epoch 3/10
29/29 [=====] - 49s 2s/step - loss: 1.1481 - accuracy: 0.8920
Epoch 4/10
29/29 [=====] - 49s 2s/step - loss: 0.7246 - accuracy: 0.9179
Epoch 5/10
29/29 [=====] - 50s 2s/step - loss: 1.1902 - accuracy: 0.8683
Epoch 6/10
29/29 [=====] - 50s 2s/step - loss: 1.0820 - accuracy: 0.8963
Epoch 7/10
29/29 [=====] - 50s 2s/step - loss: 0.6900 - accuracy: 0.9309
Epoch 8/10
29/29 [=====] - 50s 2s/step - loss: 0.6858 - accuracy: 0.9222
Epoch 9/10
29/29 [=====] - 49s 2s/step - loss: 0.6906 - accuracy: 0.9266
Epoch 10/10
29/29 [=====] - 49s 2s/step - loss: 0.2096 - accuracy: 0.9719
```

We can clearly see that the for every epoch loss is decreasing and the accuracy is increasing till epoch equals to 10. After that using the matplotlib lib to check the accuracy vs the loss.

```
# plot the Loss
plt.plot(r.history['loss'], 'go--', c='red', label='train loss')
plt.plot(r.history['accuracy'], 'go--', label='train acc')
plt.legend()
plt.title('Accuracy VS Loss')
plt.show()
```



- **Visualizations**

Visualization is basically finding some outcomes after visualizing the data in form of some graph or some plots. Different visualizing methods has been use the to do the analysis on the data. In this model multiple plots has been used to do analysis on the provided data.

1. Matplotlib Plot – Used to plot the model accuracy and loss.

- **Interpretation of the Results**

The Image Classification model is used to do the prediction on the image, which has been passing to it.

CONCLUSION

- **Key Findings and Conclusions of the Study**

Downloading the image from the websites and using the NLP and deep learning for model building.

- **Learning Outcomes of the Study in respect of Data Science**

Using the deep learning and NLP to build the image classification model, Using Inception v3 which is convolutional neural network for assisting in image analysis and object detection.