IMAGE SCRAPING AND CLASSIFICATION PROJECT

Submitted by:

Hinal Seth

ACKNOWLEDGMENT

I wish to express my sincere gratitude to DataTrained Academy and FlipRobo Technologies who gave me the opportunity to do the IMAGE SCRAPING AND CLASSIFICATION PROJECT. It helped me to do a lot of research and I have grasped many new things.

I have put good effort in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I would also thankful to the blogs/articles through online platforms which gave me a lot of information in finishing this project within the limited time

INTRODUCTION

- ## Business Problem Framing

  Data scrapping used for collecting data of particular site for insufficient data and for image classification CNN method is used to teach a machine about that particular image. More than 25% of the whole revenue in E-Commerce is attributed to apparel & accessories. a serious problem they face is categorizing these apparels from just the pictures especially when the categories provided by the brands are inconsistent. This poses a stimulating computer vision problem that has caught the eyes of several deep learning researchers.

- ## Conceptual Background of the Domain Problem

  Data science plays an important role to solve business problems by which companies increase their profits and improve business strategies. Our objective is to create a Classification model that classifies the image of each clothing category (which is scraped from an E-Commerce website) focusing on changing trends.

- ## Review of Literature

  Recently, image classification is growing and becoming a trend among technology developers especially with the growth of data in different parts of industry such as e-commerce, automotive, healthcare, and gaming. The most obvious example of this technology is applied to Facebook. Facebook now can detect up to 98% accuracy to identify your face with only a few tagged images and classified it into your Facebook's album. The technology itself almost beats the ability of human in image classification or recognition.

One of the dominant approaches for this technology is deep learning. Deep learning falls under the category of Artificial Intelligence where it can act or think like a human. Normally, the system itself will be set with hundreds or maybe thousands of input data to make the 'training' session to be more efficient and faster. It starts by giving some sort of 'training' with all the input data (Faux & Luthon, 2012). Image classification has become a major challenge in machine vision and has a long history with it.

The challenge includes a broad intra-class range of images caused by color, size, environmental conditions, and shape. It is required big data of labelled training images and to prepare this big data, it consumes a lot of time and cost as for the training purpose only in this project we will be using a transfer learning state of the art model for getting the best results that is VGG 16. Which was a winner in the ImageNet competition.

- Motivation for the Problem Undertaken

  Every problem begins with ideas that are further developed and inspired to address a variety of situations and circumstances. Learning the theoretical background for data science or machine learning are often a frightening experience, because it involves multiple fields of mathematics and an extended list of online resources. By proper practical research and practice I can become better in this field. These suggestions are derived from my mentors/SME's and my own experience in the beginner projects.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  We have scraped images of all 3 classes that is men jeans, men trousers and saree from the amazon website and we have built our model by training it on this data.

  We have used transfer learning to get state of the art results for our model.

- Data Sources and their formats

  Data has been scraped from amazon.com using a python script which with selenium. All the data is in the .jpg image format. We have 200+ images per category.

- Data Pre-processing Done

  The obtained data is split into train and test (80:20) and as input and output respectively using split folders.

```
#splitting the data
import splitfolders

input = "/content/drive/MyDrive/Category"
Output = "/content/drive/MyDrive" #where you want the split datasets saved. one will be created if none is set

splitfolders.ratio(input, output="Output", seed=42, ratio=(0.8,0.2))
```

- Data Inputs- Logic- Output Relationships

  We have given raw yet cleaned images to the machine learning model and the output produced is a label of the image on which the model is predicted.

- State the set of assumptions (if any) related to the problem under consideration

  No relevant assumptions taken by me.

- Hardware and Software Requirements and Tools Used
- Hardware Required:
  - A computer with a processor i3 or above.
  - More than 4 GiB of Ram.
  - GPU preferred.
  - Around 100 Mib of Storage Space.
- Software Required:
  - Python 3.6 or above
  - Jupyter Notebook.
  - Google Collab. Tools/Libraries Used:

  1. Computing Tools:

  - Numpy
  - Pandas
  - Tensorflow

  2. Visualizing Tools:

  - Matplotlib

  3. Saving Tools:

  - keras.savemodel

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Scrapped images are classified into three categories and are tested using the model.

- Testing of Identified Approaches (Algorithms)

  Inception model is used and Imagenet as weights, optimizer RMSprop, convolution 2D filters, loss as categorical_crossentropy and accuracy as metrics.

- Run and Evaluate selected models

  Hyper parameter tuning is done to know the best fit parameters such as filters and layers by setting some number of trials and found the metrics of the model.

```
#model cost and optimization
model.compile(
    loss='categorical_crossentropy',
    optimizer='rmsprop',
    metrics=['accuracy']
)
```

```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
# Make sure you provide the same target size as initialied for the image size
training_set = train_datagen.flow_from_directory('/content/Output/train',
                                                 target_size = (80,80),
                                                 batch_size = 16,
                                                 class_mode = 'categorical')
```

```
Found 574 images belonging to 3 classes.
```

```
test_set = test_datagen.flow_from_directory('/content/Output/val',
                                            target_size = (80,80),
                                            batch_size = 16,
                                            class_mode = 'categorical')
```

```
Found 144 images belonging to 3 classes.
```

```
training_set.class_indices
```

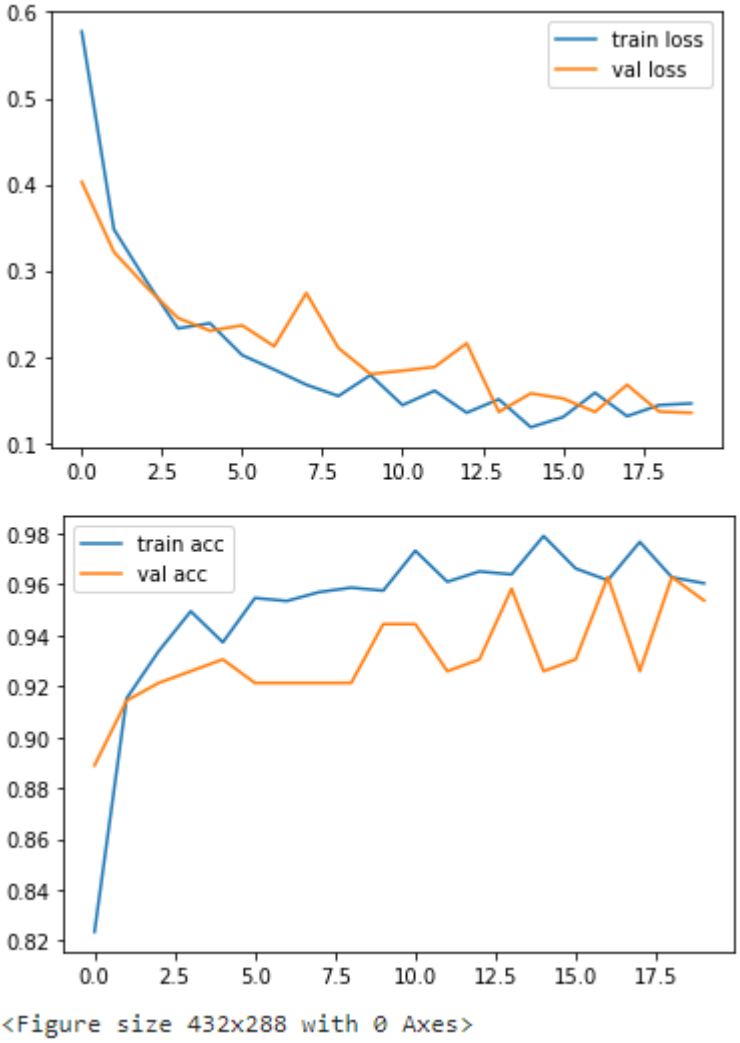
```
{'Jeans': 0, 'Saree': 1, 'Trouser': 2}
```

```
# We got good accuracy but let's do hyper parameter tuning
def build_model(hyperparameter):
  model = keras.Sequential([
    keras.layers.Conv2D(
        filters=hyperparameter.Int('conv_1_filter', min_value=32, max_value=128, step=16),
        kernel_size=hyperparameter.Choice('conv_1_kernel', values = [3,5]),
        activation='relu',
        input_shape=(80,80,3)
    ),
    keras.layers.Conv2D(
        filters=hyperparameter.Int('conv_2_filter', min_value=32, max_value=128, step=16),
        kernel_size=hyperparameter.Choice('conv_2_kernel', values = [3,5]),
        activation='relu'
    ),
    keras.layers.Flatten(),
    keras.layers.Dense(
        units=hyperparameter.Int('dense_1_units', min_value=32, max_value=128, step=16),
        activation='relu'
    ),
    keras.layers.Dense(3, activation='softmax')
  ])

  model.compile(optimizer=keras.optimizers.RMSprop(hyperparameter.Choice('learning_rate', values=[1e-2, 1e-3])),
              loss='categorical_crossentropy',
              metrics=['accuracy']
  )

  return model
```

- Key Metrics for success in solving problem under consideration

  The metrics used for the model are training and validation accuracy.

- Visualizations



<Figure size 432x288 with 0 Axes>

From the above plots, it is clear that Accuracy of both training and validation accuracy increasing and both the loss are decreasing respectively.

- Interpretation of the Results
  1. Given imagenet as weights for the inception model in layers.
  2. Hyper parameter tuning is done for best fit parameters and good performance metrics.

CONCLUSION

- Key Findings and Conclusions of the Study

  Predicting the image uploaded to Google colab and predicted with the model and got the expected result.

- Learning Outcomes of the Study in respect of Data Science

  1. By using selenium scrapping images from amazon made easy.

  2. By using optimizers, layers and filters observed that there is difference in scoring.

  3. RMSprop optimizer gave best score.

- Limitations of this work and Scope for Future Work

  One could always provide more data for training the model to get better results. We could use to model for apparel segmentation at Supermarkets and shopping stores.