FLIP ROBO

IMAGE SCRAPING AND CLASSIFICATION PROJECT

Submitted by:

Bhawani Singh

Shekhawat

ACKNOWLEDGMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Flip Robo Technologies Bangalore for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I want to thank my SME Miss. **Sapna Verma** for providing the project and helping us to solve the problem and addressing out our Query in right time.

I would like to express my gratitude towards my parents & members of Flip Robo for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

INTRODUCTION

- Business Problem Framing

Classification is a systematic arrangement in groups and categories based on its features. Image classification came into existence for decreasing the gap between the computer vision and human vision by training the computer with the data. The image classification is achieved by differentiating the image into the prescribed category based on the content of the vision. We use Deep Learning to accomplish the task of image Classification.

In this project we have to classify whether is image is of a jeans , a trouser or a saree. We could we such model either for automatic segmentation of items using IOT and techniques for Industry 4.0

Conceptual Background of the Domain Problem

Deep learning (DL) is a sub field to the machine learning, capable of learning through its own method of computing. A deep learning model is introduced to persistently break down information with a homogeneous structure like how a human would make determinations. To accomplish this, deep learning utilizes a layered structure of several algorithms expressed as an artificial neural system (ANN). The architecture of an ANN is simulated with the help of the biological neural network of the human brain. This makes the deep learning most capable than the standard machine learning models.

In deep neural networks every node decides its basic inputs by itself and sends it to the next tier on behalf of the previous tier. We train the data in the networks by giving an input image and conveying the network about its output. Neural networks are expressed in terms of number of layers involved for producing the inputs and outputs and the depth of the neural network.

- 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 in order 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 in order to make the 'training' session to be more efficient and fast. 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

The problem was undertaken in order to classify images efficiently using best deep learning algorithms and data augmentation techniques.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  We have scraped images of all 3 classes that is men jeans, men trouser and saree from the internet 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 over 250 images per class.

- Data Preprocessing Done

  We have first labelled the image data.

  We have manually removed irrelevant images that were downloading via the script

  We have removed the duplicate images using a script.

  We have also performed multiple data augmentation techniques in order to train the model better or multiple angles and orientation of the same image.

- 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

- Hardware and Software Requirements and Tools Used • 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)

We will be first scraping the data from amazon and then cleaning the data to be further used for training the model.

Then splitting the data into train and test set to model evaluation.

We would use VGG16, A transfer learning SOTA model for training our data on.

We would perform multiple data augmentation techniques on the images for better generalization of our model.

We could check and identity an optimal batch size and number of epoch to Train the model using trial and error method also keeping the epoch loss for both training set and validation in mind.

• Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

• Run and Evaluate selected models

Model summary:

```
Model: "model_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
================================================================= input_5
(InputLayer)         [(None, 256, 256, 3)]      0
_________________________________________________________________
```

```
block1_conv1 (Conv2D)          (None, 256, 256, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)          (None, 256, 256, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)     (None, 128, 128, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)          (None, 128, 128, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)          (None, 128, 128, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)     (None, 64, 64, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)          (None, 64, 64, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)          (None, 64, 64, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)          (None, 64, 64, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)     (None, 32, 32, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)          (None, 32, 32, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)          (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)          (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)     (None, 16, 16, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)          (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)          (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)          (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)     (None, 8, 8, 512)         0
_________________________________________________________________
flatten_4 (Flatten)            (None, 32768)             0
_________________________________________________________________ dense_4
(Dense)                    (None, 3)                    98307
=================================================================
Total params: 14,812,995
Trainable params: 98,307
Non-trainable params: 14,714,688
_________________________________________________________________
```

```
# compile the model model.compile(optimizer='adam',
              loss='categorical_crossentropy',
metrics = ['accuracy'])
```

- Key Metrics for success in solving problem under consideration

  We have considered the model accuracy and loss for both the training and validation data.

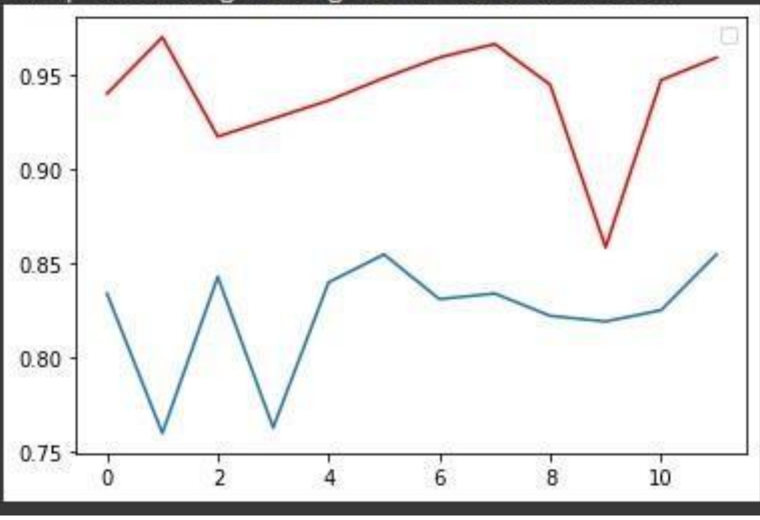
- Visualizations

  A Random image for a training set:

  model Accuracy:

```
plt.plot(losses['accuracy'],color='red')
plt.plot(losses['val_accuracy'])
plt.legend()
```

```
No handles with labels found to put in legend.
<matplotlib.legend.Legend at 0x7fc97c61b990>
```

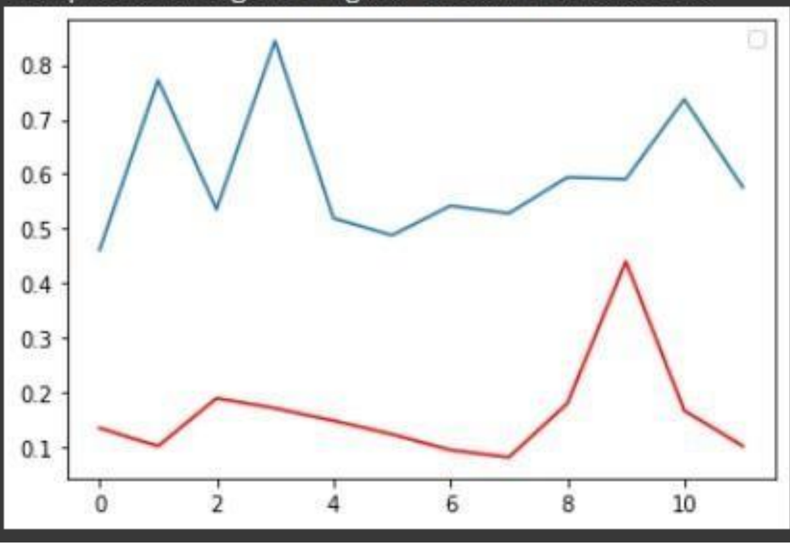


Model Loss:

```
# plotting the lss of the model

plt.plot(losses['loss'],color='red')
plt.plot(losses['val_loss'])
plt.legend()
```

```
No handles with labels found to put in legend.
<matplotlib.legend.Legend at 0x7fc97c605e10>
```



Evalutions metrics per Epoch:

```
losses = pd.DataFrame(model.history.history)
losses
```

	loss	accuracy	val_loss	val_accuracy
0	0.134504	0.939976	0.460784	0.833828
1	0.101309	0.969988	0.772406	0.759644
2	0.189138	0.917167	0.535361	0.842730
3	0.170861	0.926771	0.844059	0.762611
4	0.147873	0.936375	0.519072	0.839763
5	0.123221	0.948379	0.487635	0.854599
6	0.094144	0.959184	0.541837	0.830861
7	0.080961	0.966387	0.527974	0.833828
8	0.179659	0.944778	0.594119	0.821958
9	0.439795	0.858343	0.590677	0.818991
10	0.166460	0.947179	0.736998	0.824926
11	0.101558	0.959184	0.575776	0.854599

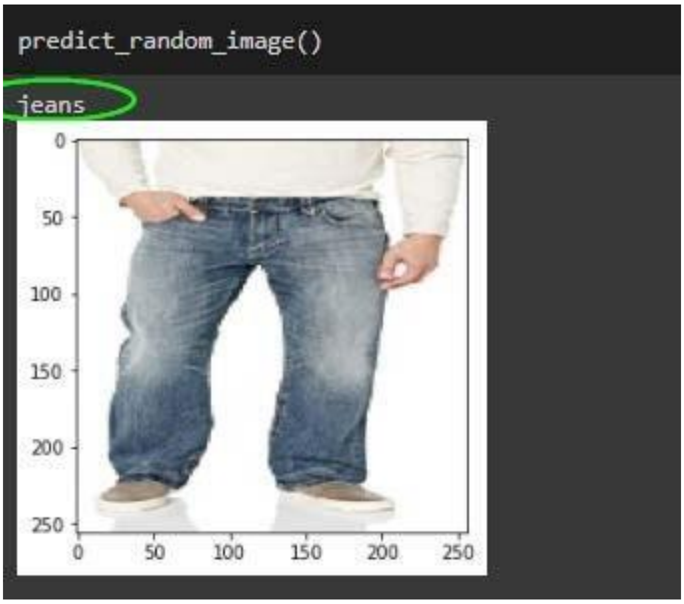
- Interpretation of the Results

```
#predicting any random image
import numpy as np
from google.colab import files
from keras.preprocessing import image


def predict_random_image():
  img = image.load_img(np.random.choice(test_files), target_size=IMAGE_SIZE)
  x = image.img_to_array(img)
  plt.imshow(img)
  x = np.expand_dims(x, axis=0) /255.0
  result_df = pd.DataFrame(list(model.predict(x)[0]),index=name_labels_df.names.values,columns=['result'])
  print(result_df.result.idxmax())
```

```
predict_random_image()
```

```
trousers
```



CONCLUSION

- Key Findings and Conclusions of the Study

  Our model was successfully to classify the images with an accuracy of 97 Percent on a validation set of around 70 images

- Learning Outcomes of the Study in respect of Data Science

  The larger the data the better the model could predict. Multi class prediction are somewhat relatively harder to train in comparison to a Binary class prediction.
  Data Augmentation is necessary where we have small datasets of images.

- Limitations of this work and Scope for Future Work

  One could always provide more data for training the model in order to get better results.

We could use to model for apparel segmentation at Supermarkets and shopping stores.