CLOTHING IMAGE CLASSIFICATION

Submitted by:

AVINASH PATEL

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my teacher as well as my intern company FLIPROBO who gave me this golden opportunity to do this wonderful project on the topic CLOTHING IMAGE CLASSIFICATION, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them. Secondly I want to mention some sites which helped me when I got stuck somewhere while completing my projects.

Those sites are :

http://scikit-learn.org

https://www.w3schools.com

https://www.youtube.com

https://www.kaggle.com

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. We are trying to give you an exposure of how an end to end project is developed in this field.
  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.

- Conceptual Background of the Domain Problem

  I learned about convolution neural network (CNN) and other concepts regarding it suck as how multi layered perceptron works and what is the basic principle behind that. Then I learned about weight initialisers and back propagation I also learned about losses and the I learned about different convolution al models and then I chose the mobile-net convolution al model to be my model to develop this model.

- Review of Literature

  I need to scrape images from e-commerce portal, Amazon.com. The clothing categories used for scraping will be:

  - Sarees (women)
  - Trousers (men)
  - Jeans (men)

  You need to scrape images of these 3 categories and build your data from it. That data will be provided as an input to your deep

learning problem. You need to scrape minimum 200 images of each categories. There is no maximum limit to the data collection. You are free to apply image augmentation techniques to increase the size of your data but make sure the quality of data is not compromised. Customers through a strategy of disruptive innovation that focuses on the subscriber.

- Motivation for the Problem Undertaken

As I am an intern I want to work on as many project as possible for me. So ,I am highly motivated to do this project and to learn new thing and also learn from my mistakes.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  First, I organised all the images into train, test and validation folders and the test and validation folder contains 20 and 16 images respectively and the rest of the images goes to the train folder.

  After segregating all the data into these three folders I started with the initial pre-processing of the images because the mobile net model only takes inputs in a certain format.

  Now I see the model for its layers and after some experimentation I concluded that only the last layer is required to remove and the I replaced the last layer with the 3-output layer which is required to predict out three classes saree, trouser and jeans.

  After that I freeze all the layers up to 23 layers as we cannot spend so much resources to train lacks of datapoints so we are using a technique called transfer learning to reduce our work load as the previous layers have already been trained.

  The finally I trained our data on our training set and passed validation data for validation score and then as we reached our 30 epoch the validation score is very close to our accuracy score that means the model is performing well and finally, I tested the model on our testing data and saved model as .h5 format which hierarchal format data.

- Data Sources and their formats

  As no Data is provided by our client, I scraped almost 450 images for each of the classes i.e. saree, trousers and jeans .All of the data is in .jpg format with name as number from zero to the last number of image there is.

- Data Pre-processing Done

  After segregating all the data into these three folders I started with the initial pre-processing of the images because the mobile net model only takes inputs in a certain format.

  Now I see the model for its layers and after some experimentation I concluded that only the last layer is required to remove and the I replaced the last layer with the 3-output layer which is required to predict out three classes saree, trouser and jeans.

  After that I freeze all the layers up to 23 layers as we cannot spend so much resources to train lacks of datapoints so we are using a technique called transfer learning to reduce our work load as the previous layers have already been trained.

  The finally I trained our data on our training set and passed validation data for validation score and then as we reached our 30 epoch the validation score is very close to our accuracy score that means the model is performing well and finally, I tested the model on our testing data and saved model as .h5 format which hierarchal format data.

- Data Inputs- Logic- Output Relationships

  The Remaining Data after all the pre-processing acts as an input data for the Model(in our case XGBoost Regressor) the already trained model gives us an output of an estimated price.

- Hardware and Software Requirements and Tools Used

  Ram:16GB
  ROM:200MB
  Processor : Ryzen 7 5700u(8 cores 16 threads)
  Tool : Jupiter Notebook
  Language Used: Python

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  First I imported all the basic libraries of python used in the problem solving. Then I read carefully the data and the description given to me. After that I imported the scraped images file into the Jupiter notebook.

  Now I selected mobile-net architecture of my convolutional neural network as its size is small and it also gives a pretty good result.

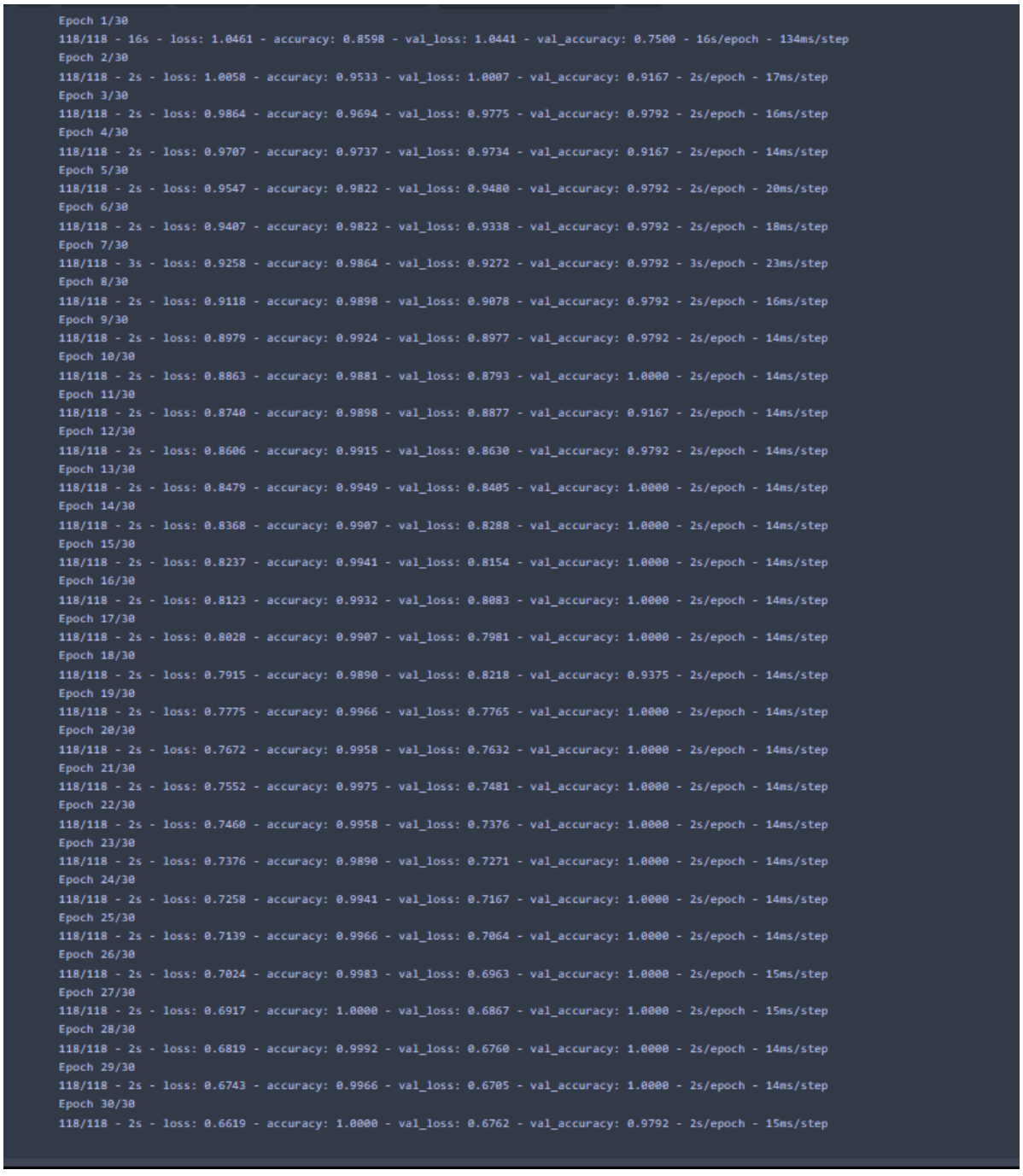
  Then I pre-process the images for the inputs of the mobile net model and then I use transfer learning to freeze sone layers and modified the model to satisfied my needs.

- Testing of Identified Approaches (Algorithms)

  - Mobile-net CNN model.

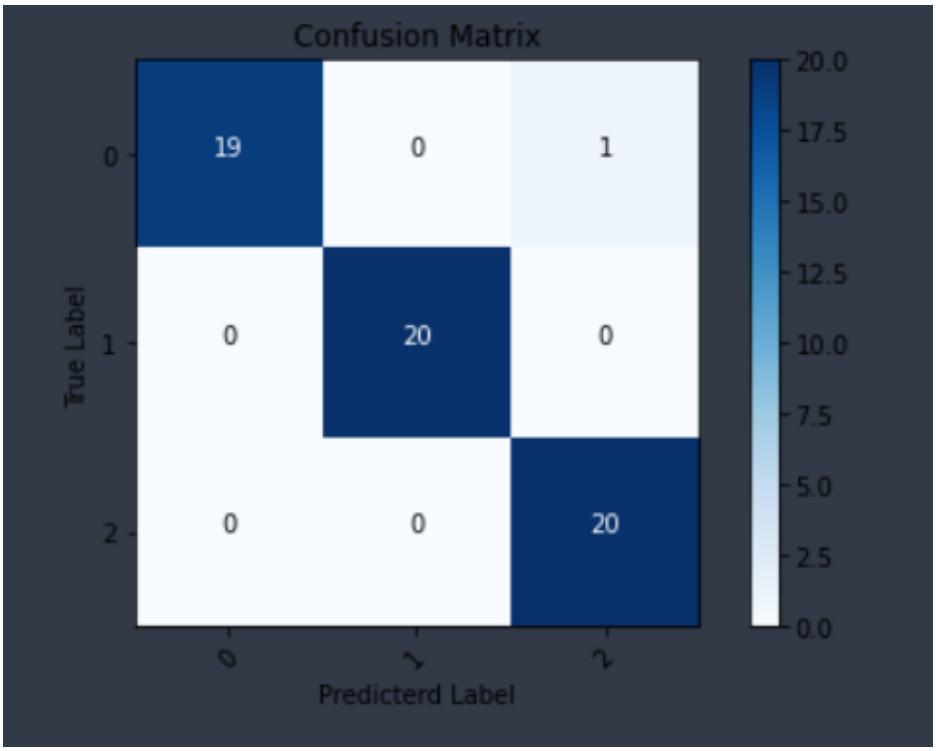
- Run and Evaluate selected models

After running for 30 epoch the accuracy come very close to 100%.



```
Epoch 1/30
118/118 - 16s - loss: 1.0461 - accuracy: 0.8598 - val_loss: 1.0441 - val_accuracy: 0.7500 - 16s/epoch - 134ms/step
Epoch 2/30
118/118 - 2s - loss: 1.0058 - accuracy: 0.9533 - val_loss: 1.0007 - val_accuracy: 0.9167 - 2s/epoch - 17ms/step
Epoch 3/30
118/118 - 2s - loss: 0.9864 - accuracy: 0.9694 - val_loss: 0.9775 - val_accuracy: 0.9792 - 2s/epoch - 16ms/step
Epoch 4/30
118/118 - 2s - loss: 0.9707 - accuracy: 0.9737 - val_loss: 0.9734 - val_accuracy: 0.9167 - 2s/epoch - 14ms/step
Epoch 5/30
118/118 - 2s - loss: 0.9547 - accuracy: 0.9822 - val_loss: 0.9480 - val_accuracy: 0.9792 - 2s/epoch - 20ms/step
Epoch 6/30
118/118 - 2s - loss: 0.9407 - accuracy: 0.9822 - val_loss: 0.9338 - val_accuracy: 0.9792 - 2s/epoch - 18ms/step
Epoch 7/30
118/118 - 3s - loss: 0.9258 - accuracy: 0.9864 - val_loss: 0.9272 - val_accuracy: 0.9792 - 3s/epoch - 23ms/step
Epoch 8/30
118/118 - 2s - loss: 0.9118 - accuracy: 0.9898 - val_loss: 0.9078 - val_accuracy: 0.9792 - 2s/epoch - 16ms/step
Epoch 9/30
118/118 - 2s - loss: 0.8979 - accuracy: 0.9924 - val_loss: 0.8977 - val_accuracy: 0.9792 - 2s/epoch - 14ms/step
Epoch 10/30
118/118 - 2s - loss: 0.8863 - accuracy: 0.9881 - val_loss: 0.8793 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 11/30
118/118 - 2s - loss: 0.8740 - accuracy: 0.9898 - val_loss: 0.8877 - val_accuracy: 0.9167 - 2s/epoch - 14ms/step
Epoch 12/30
118/118 - 2s - loss: 0.8606 - accuracy: 0.9915 - val_loss: 0.8630 - val_accuracy: 0.9792 - 2s/epoch - 14ms/step
Epoch 13/30
118/118 - 2s - loss: 0.8479 - accuracy: 0.9949 - val_loss: 0.8405 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 14/30
118/118 - 2s - loss: 0.8368 - accuracy: 0.9907 - val_loss: 0.8288 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 15/30
118/118 - 2s - loss: 0.8237 - accuracy: 0.9941 - val_loss: 0.8154 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 16/30
118/118 - 2s - loss: 0.8123 - accuracy: 0.9932 - val_loss: 0.8083 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 17/30
118/118 - 2s - loss: 0.8028 - accuracy: 0.9907 - val_loss: 0.7981 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 18/30
118/118 - 2s - loss: 0.7915 - accuracy: 0.9890 - val_loss: 0.8218 - val_accuracy: 0.9375 - 2s/epoch - 14ms/step
Epoch 19/30
118/118 - 2s - loss: 0.7775 - accuracy: 0.9966 - val_loss: 0.7765 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 20/30
118/118 - 2s - loss: 0.7672 - accuracy: 0.9958 - val_loss: 0.7632 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 21/30
118/118 - 2s - loss: 0.7552 - accuracy: 0.9975 - val_loss: 0.7481 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 22/30
118/118 - 2s - loss: 0.7460 - accuracy: 0.9958 - val_loss: 0.7376 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 23/30
118/118 - 2s - loss: 0.7376 - accuracy: 0.9890 - val_loss: 0.7271 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 24/30
118/118 - 2s - loss: 0.7258 - accuracy: 0.9941 - val_loss: 0.7167 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 25/30
118/118 - 2s - loss: 0.7139 - accuracy: 0.9966 - val_loss: 0.7064 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 26/30
118/118 - 2s - loss: 0.7024 - accuracy: 0.9983 - val_loss: 0.6963 - val_accuracy: 1.0000 - 2s/epoch - 15ms/step
Epoch 27/30
118/118 - 2s - loss: 0.6917 - accuracy: 1.0000 - val_loss: 0.6867 - val_accuracy: 1.0000 - 2s/epoch - 15ms/step
Epoch 28/30
118/118 - 2s - loss: 0.6819 - accuracy: 0.9992 - val_loss: 0.6760 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 29/30
118/118 - 2s - loss: 0.6743 - accuracy: 0.9966 - val_loss: 0.6705 - val_accuracy: 1.0000 - 2s/epoch - 14ms/step
Epoch 30/30
118/118 - 2s - loss: 0.6619 - accuracy: 1.0000 - val_loss: 0.6762 - val_accuracy: 0.9792 - 2s/epoch - 15ms/step
```

- Visualizations
  Confusion Matrix:



- Interpretation of the Results

  After all the pre-processing and the model building done I can say that the model is very good and can be implemented commercially as the results have an accuracy of 99% and the model also proves its legitimacy in predicting the test dataset with only one wrong prediction.

CONCLUSION

- Key Findings and Conclusions of the Study

  I find out that all the layers is important to build a good model and only the last output layer is needed to be deleted in order to modify the model to predict only three classes which is our target class.

  We have to freeze up to only 23 layers to get the best result in record time.

- Learning Outcomes of the Study in respect of Data Science

  As I have already told that I find out that all the layers are important to build a good model and only the last output layer is needed to be deleted in order to modify the model to predict only three classes which is our target class.

  I have to freeze up to only 23 layers to get the best result in record time.

  I have also plotted Confusion Matrix to see all of the correct and incorrect predictions.

- Limitations of this work and Scope for Future Work

  Everything has worked out as planned so I faced no such limitations to talk about but if I could get more data it will be better but as for the data I scraped I got a pretty good result of 99% and I don’t think there is much room to grow.