**Project Title:** Multiclassification of fashion apparel images from Fashion MNIST

**Author:** Rajesh Nachaloor

**Executive summary:**

Create and train a model to classify fashion apparel images. Create a model from scratch and compare with another model (VGG16) using transfer learning.

**Rationale:**

The idea is to learn from this project and then extend it to a real-life scenario in my company (GM) to classify different types of street parking in autonomous driving situations.

**What are you trying to answer?**

Multiclass Images classification can be done very well with transfer learning from existing models such as VGG16, thereby we can reduce a lot of time and energy and obtain great results.

**What data will you use to answer your question?**

I would be using the fashion MNIST images. There is a total of 70000 monochrome small images (28x28) that is available for download from Keras.

**Methodology**

1. Create a model from scratch using Tensor Flow and Keras and build a neural network to classify images. Split the data in train and test images and then look at the accuracy score.
2. Create another model from VGG16. Do transfer learning, just remove the top and replace with the softmax layer (10 way classification). Obtain the accuracy score.
3. Compare the accuracy scores between the models. Calculate Precision, Recall and F1 scores.

**Results**

1. In my case base model written from scratch performed slightly better than the VGG16 transfer learning model.
2. The reason for the discrepancy is the image set difference. Although I used the same image set, my base model used 60000 images to train and 10000 images to test. However, VGG16 model required a color image instead of the monochrome images. So the image size was increased 3 times, which meant that I couldn’t have all these images in memory to perform the operations. Google colab often crashed due to lack of RAM and hence I trained with 5000 images and tested on 1000 images.
3. Due to above base model performed better as it had 12 times more images for training and 10 times more images for testing. When I made the image set of the base model for both training and testing, equal to the VGG16 model, VGG16 performed slightly better. I am confident however that if the image set were increased to the 60k + 10 k ((train + test), VGG16 will outperform the base set, as more the images, the better the enhanced model would perform.

**Outline of the project**

Created 2 notebooks, one for the base model and another for the VGG16 extended model. Trained and tested both the models. Base model was trained with 60k images and tested on 10k images. VGG16 extended model’s input images were changed from monochrome to color (just repeated the same monochrome pixels for RGB) and trained/tested on 5k/1k images.

Finally Test loss, Test Accuracy, Precision Score, Recall Score and F1 scores were calculated for both the models.

**Base model results**

Test Loss: 0.33752474188804626

Test Accuracy: 0.8834999799728394

Precision Score: 0.8835

Recall Score: 0.8843346210518979

F1 Score: 0.8834642962652971

Confusion Matrix

[[858 4 15 34 0 0 127 0 8 0]

[ 1 969 0 5 0 0 0 0 0 0]

[ 13 3 851 8 113 0 125 0 0 0]

[ 11 17 12 876 41 1 21 0 4 0]

[ 3 4 61 22 775 0 65 0 3 0]

[ 1 0 0 0 0 971 0 19 1 6]

[108 2 61 52 68 0 656 0 9 1]

[ 0 0 0 0 0 13 0 939 4 24]

[ 5 1 0 3 3 0 6 0 971 0]

[ 0 0 0 0 0 15 0 42 0 969]]

**VGG16 enhanced model results**

Test Loss: 6.774092674255371

Test Accuracy: 0.8510000109672546

Precision Score: 0.8524820681143188

Recall Score: 0.8633787342709537

F1 Score: 0.8549271593130932

Confusion Matrix

[[ 92 2 4 11 1 0 13 0 2 0]

[ 0 100 0 0 0 0 0 0 0 0]

[ 0 0 79 1 7 0 6 0 0 0]

[ 1 3 0 67 2 0 2 0 0 0]

[ 2 0 11 4 98 0 13 0 0 0]

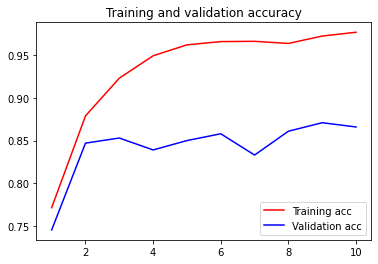
[ 0 0 0 1 0 76 0 0 1 1]

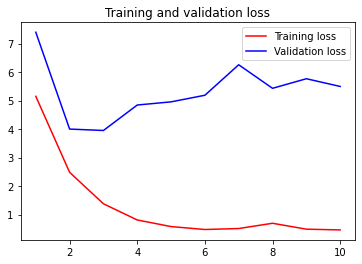
[ 12 0 17 9 7 0 63 0 0 0]

[ 0 0 0 0 0 8 0 94 1 3]

[ 0 0 0 0 0 1 0 0 91 0]

[ 0 0 0 0 0 2 0 1 0 91]]

****

****

**Base model code link:** https://github.com/rnachloo/RajeshAIML/blob/main/Image\_Classification\_with\_base\_model.ipynb

**VGG16 enhanced code link:** <https://github.com/rnachloo/RajeshAIML/blob/main/vgg16model.ipynb>

**Contact Details**

Rajesh Nachaloor, nachaloor@gmail.com