IMAGE CLASSIFICATION- A Deep Learning Project

Submitted by-
Anurag Kumar

ACKNOWLEDGMENT

This project needed data provided by Flip Robo Technology. Necessary technical help. Some websites such as stackoverflow.com, Datatrained.com, guru99.com helped a lot for doubt clearing. Mentor at Flip Robo guided me through all the speed brakers I faced during this project completion. Keras & TensorFlow websites provided great insights during image classification model building.

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

Conceptual Background of the Domain Problem-

- There are many domains which required image classification even fashion and apparel sector. So, here I tried to visualize and classify different apparel into classes with accuracy,

Motivation for the problem undertaken-

- 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. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users.

Data Collection Phase:

In this section, we needed to scrape images from e-commerce portal, Amazon.com. The clothing categories used for scraping will be:

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

We scraped images of these 3 categories and build your data from it. That data will be provided as an input to your deep learning problem. We needed to scrape minimum 200 images of each category. There is no maximum limit to the data collection. Then we applied image augmentation techniques to increase the size of your data but make sure the quality of data is not compromised.

Remember, in case of deep learning models, the data needs to be big for building a good performing model. More the data, better the results.

Model Building Phase:

After the data collection and preparation is done, we needed to build an image classification model that will classify between these 3 categories mentioned above. After that we played around with optimizers and learning rates for improving your model's performance.

Data Collection Phase:

- Scraped images from e-commerce portal, Amazon.com. The clothing categories used for scraping will be:

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

- Image scrapped was done using Selenium.
- More than 250 images were scrapped from different images url.
- Scrapped image further stored on google drive in train and test folder dataset, which further used for model building.

Model Building Phase:

- After the data collection and preparation is done, I build an image classification model that will classify between these 3 categories mentioned above.
- I played around optimizers and learning rates for improving your model's performance using-

I. BATCH_SIZE = 32

II. IMAGE_SIZE = 256

III. CHANNELS=3

IV. EPOCHS=50

Image Classification Approach:

- Transfer Learning- (VGG16)
- A CNN architecture that is 16 layers deep.
- It loads a pertained version of the network trained on more no. of images.

The pertained network can classify image into 1000 object categories

Data Sources and their formats-

Image data was scrapped in JPG format from amazon websites which contains image of three different dress types i.e. a) Jeans b) Trousers c) Sarees. This image stored in dataset folder which contains three subfolders of three different names. Which further uploaded on google drive for working, there drive is mounted in google colab before uploading image.

Pre-processing of Image data-

First hurdle regarding image data set is its image size and number of images which is being uploaded on colab. To counter these problems, we first send theses images on batches and resize these images.

BATCH_SIZE = 32
IMAGE_SIZE = 256
CHANNELS=3
EPOCHS=50

Image Inputs and Training-

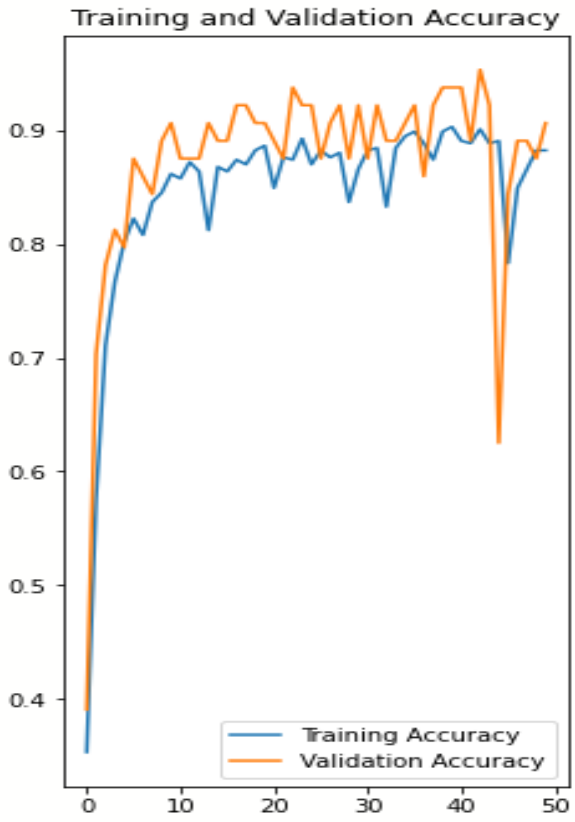
Before training image data, its needed to split the given image as following-

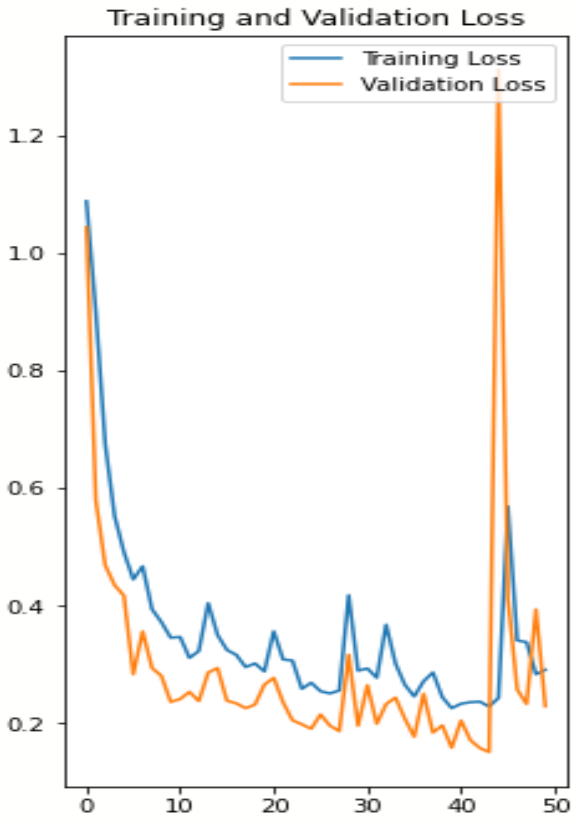
train_split=0.8
Val split=0.1
test_split=0.1
shuffle=True
shuffle size=10000

Model Compilation & KEYS-

- optimizer='adam',
- loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
- metrics=['accuracy’]
- KEYS= ['loss', 'accuracy', 'val_loss', 'val_accuracy’]

MODEL EVALUATION-





MODEL SUMMARY-

- Model Score- loss: 0.2625 - accuracy: 0.9167
- Parameters- 'epochs': 50, 'steps': 16, 'verbose': 1
- Activation function- relu & softmax