Piramal Finance Hackathon Submission

Darshan Makwana

January 25, 2023

Abstract

To segregate and extract texts from large documents has been an important problem in many large b2c markets. To identify the documents and extract data from them, a significant amount of human processing is required. Here we present a machine learning based approach to classify and extract vital informations from the documents and to make the process more efficient

1 Our Approach

The problem statement consisted of two parts. The first part being able to classify and segregate the different types of documents that is being provided as the input. To achieve that we used the VGG16 model with some customizations. VGG-16 was one of the best performing architectures in the ILSVRC challenge 2014. It was the runner up in the classification task with a top-5 classification error of 7.32% (only behind GoogLeNet with a classification error of 6.66%). It was also the winner of localization task with 25.32% localization error.

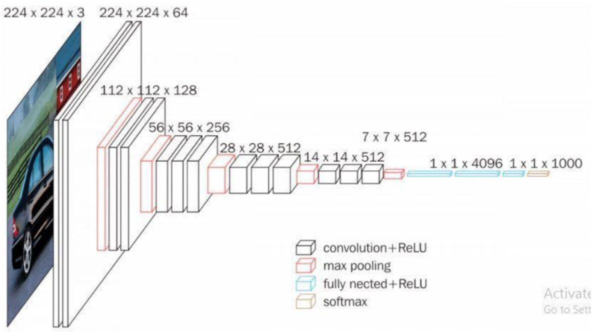

Figure 1: VGG16 Architecture

For the second task of extracting useful textual information from the documents we used an OCR model for localizing and extracting the texts. Specifically we used easyocr which is a python library for extracting the texts from the documents.

2 Results

2.1 Document Segregation

We created a dataset of 1000 images scrapped from the internet. The images consisted of various types of documents as mentioned in the problem statement booklet. We decided to combine the bank statement into salary slips and ITR form 16 into utility bills for increasing the accuracy of the model trained. After scrapping we perfomed data preprocessing on the documents and we were left with 634 images.

We experimented with various batch sizes and images sizes and decided to move forward with 500 pixels in size which requires average time as well as provides accurate local information inside the image so as to classify them

Image Size	Accuracy	Training Time
250	82.764	12.02 min
300	87.145	17.34 min
500	92.063	25.45 min

Table 1: Accuracy and time for various image sizes

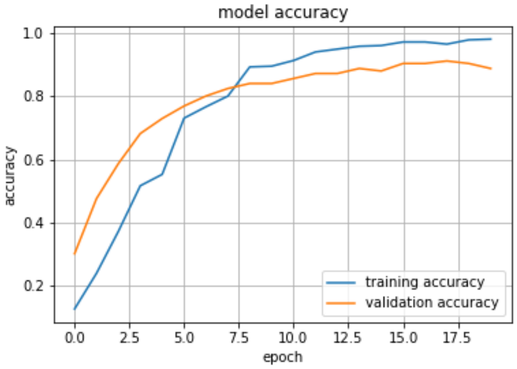

Figure 1: model accuracy for image size 300

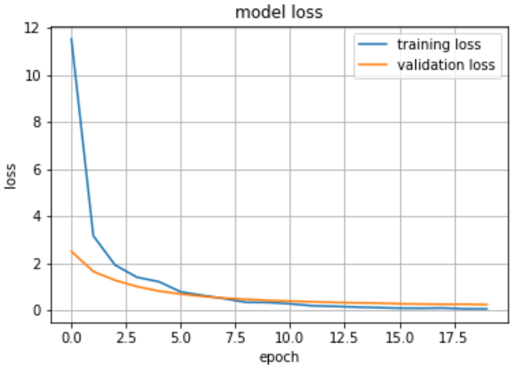

Figure 2: model loss for image size 300

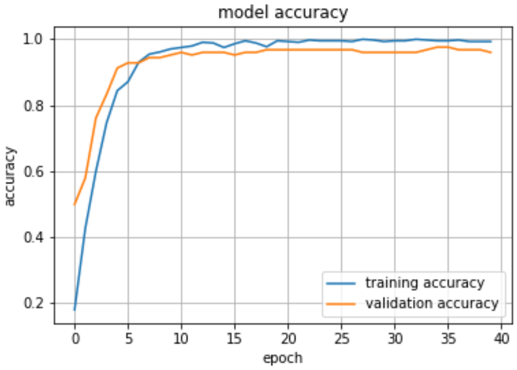

Figure 1: model accuracy for image size 500

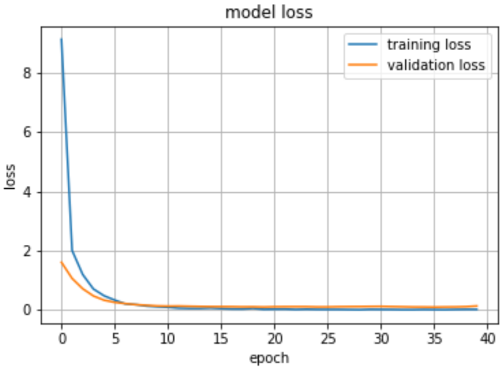

Figure 2: model loss for image size 500

Finally we created a class for documents and using that class we can load in the data to be classfied along the various models to be used and classify the documents. We achived an accuracy of 92.063 over the testing datase

2.2 Text extraction

For extracting the texts from the documents we using easyocr we created a script which can iterate iterate through all the documents which are classified and extract the texts from them and store them

3 Problems faced and our solutions

We were occasionally facing large gradient oscillations after 10+ epochs so to overcome that we customized the model by adding 2 dense layer with relu and softmax activation functions resulting in lesse oscillations

References

Link to the github repo: https://github.com/darshanmakwana412/document-classification-extraction
Link to google drive for code: https://drive.google.com/drive/folders/1VWoCIuD1mTMHoc8KAvDf2I2MShs_pOIq?usp=share_link
VGG16 Architecture: https://arxiv.org/pdf/1409.1556.pdf
EasyOCR: https://github.com/JaidedAI/EasyOCR