Report Document Classify

Duy Nguyen Ngoc

July 2021

1 Introduction

Document classification is an example of Machine Learning (ML) in Convolutional Neural Network (CNN) or the form of Natural Language Processing (NLP). By classifying text, we are aiming to assign one or more classes or categories to a document, making it easier to manage and sort. This is especially useful for publishers, news sites, blogs or anyone who deals with a lot of content. Broadly speaking, there are two classes of ML techniques: supervised and unsupervised. In supervised methods, a model is created based on a training set. Categories are predefined and documents within the training dataset are manually tagged with one or more category labels. A classifier is then trained on the dataset which means it can predict a new document's category from then on. Depending on the classification algorithm or strategy used, the classifier might also provide a confidence measure to indicate how confident it is that the classification label is correct.

We are a company specializing in Document Processing Outsourcing and we need to apply ML and DL in data classification before proceeding to extract the necessary information from the document. Because for each type of document we have different requirements or methods for processing so document classification is necessary. So we made this document to test and compare some of the methods in ML that best suit we needs.

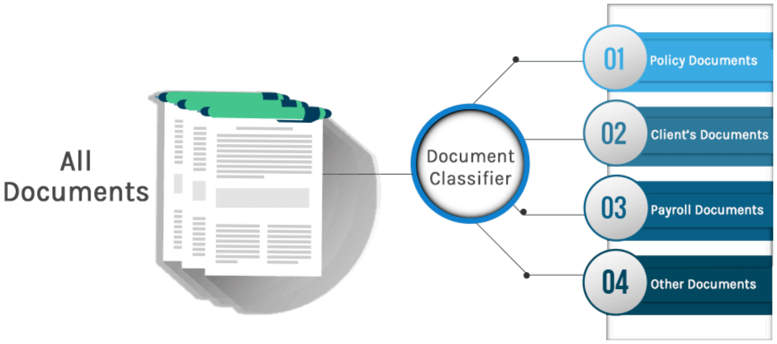

Figure 1: A simple Illustration of Document Classification

There are two methods of using ML to perform document classification: using CNN and using NLP. For CNN, the document image is fed through a pre-configured CNN network to extract features of the image based on pixels. As for the method using NLP, the input image is then extracted the content of the image through an OCR model and used the content as a feature map as a basis for document classification.

1.1 Target

The goals we need to achieve for our system include factors of accuracy, desired number of samples, number of classes, flexibility in adding new classes. Towards a system with each corresponding document layer, it is only necessary to use a small amount of sample (10-100 for each layer) but still ensure the accuracy (> 85%) of the model. Besides, another aspect to consider is how to easily add new classes without having to rerun the training model process.

Target/Goals	**Requirements**	**Notes**
Accuracy	> 85%	Hope to > 90%
Small of sample	(10-100)/class	
Flexibility	Obligatory	Add new class and managing database

Table 1: Requirements for system

1.2 Solution

In this document, we use two methods CNN and NLP to compare the results obtained from the two methods. Our model includes many components including document Pre-Processing (Pre Data) system that ensures processing of bad input images such as denoising rotation, cropping, filter noise ... We will update our Pre-data system in another document. In (Figure 2) only a few illustrations of the document after going through the Pre-data system. For CNN method we have one CNN(VGG-19) system to converts the image from Pre-data into a vector carrying the corresponding label. For the NLP method we have OCR (Tesseract) extract the content of the image and NLP (PhoBert) convert the extracted content into a vetor with labels, Finally, the Elasticsearch system for storage, management, and search engine features(Bert similar search). If you are a researcher, create your own search engine, a small suggestion is to use the method to calculate the euclidean distance between vectors.

At the same time, we also perform an effective comparison with the document classification method based on CNN (VGG-16) with the use of a large number of samples on the class (10000 samples/class) of traditional methods to get an overview.

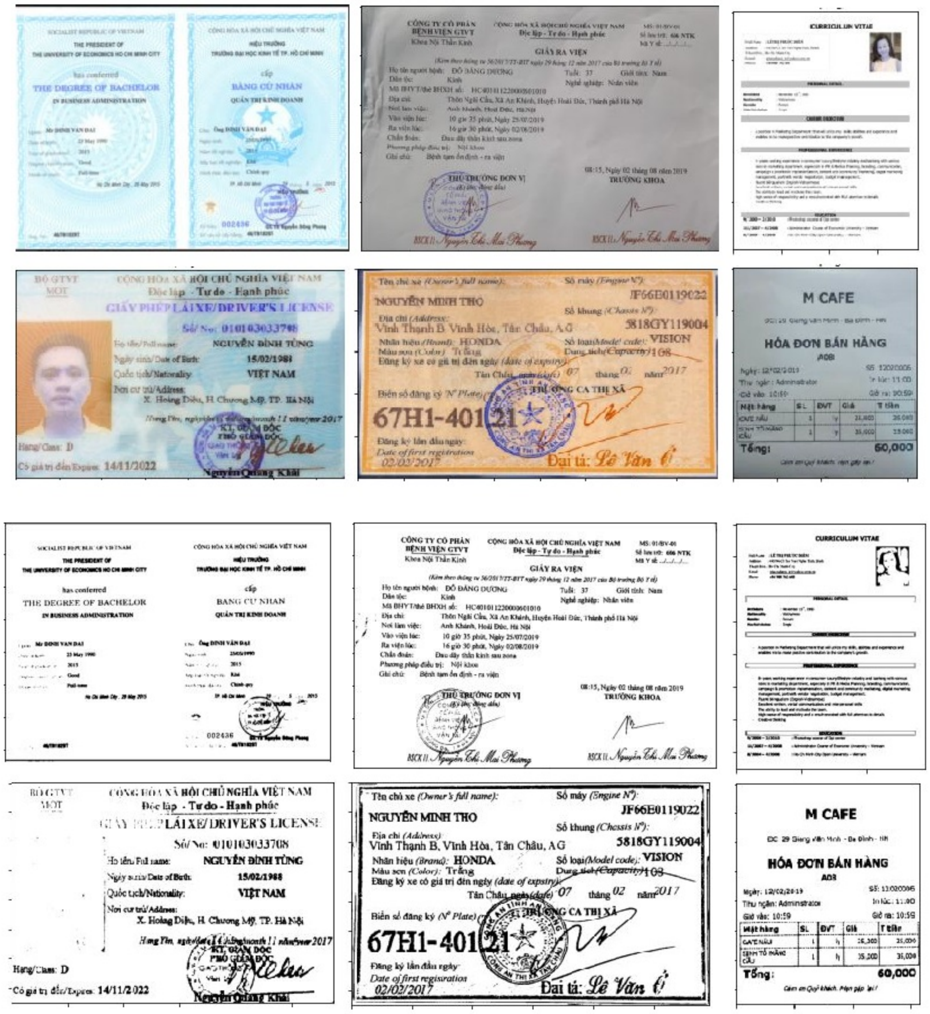

Figure 2: Document image after and before pre-processing

2 Benchmark

For the NLP model we performed on an extremely small train dataset with only 10 (samples/class) and applied to 6 types of documents including: degree of bachelor, discharge record, driver license, invoice, resume and vehicle registration certificate . The accuracy of the model is quite high with 87.3%, especially for documents with good noise removal processing such as discharge record, degree of bachelor, the accuracy is very high up to 96.6%. From the results, the main factor that determines the accuracy of the model is the preprocessing process. Besides, another aspect that is also measured is that the processing speed of the system on a document reaches 3.5s/document (excluding pre-processing system time), this is quite dependent on the configuration of the system. all system. Information on the system that conducts the parameter test is mentioned in (Table 2)

Component	**ML System**	**Elasticsearch**
CPU	4 CPU Cores	16 CPU Cores
GPU	$\geq$ GeForce GTX 1080 Ti	No
Memory	4 GB RAM	32 GB RAM
Storage	50GB HDD	1 TB SSD 3k dedicated IOPS

Table 2: System Requirement

Component	**NLP**	**CNN Inception resnet v2, VGG16**
Accuracy	87.3%	89%
Train sample	10/class	>10000/class
Flexibility add class	Yes	Need re-train
Processing speed/doc	<3.5s	<2s
Training time/class	<60s	>3000s

Table 3: Comparing NLP and CNN InceptionResNetv2 + VGG16

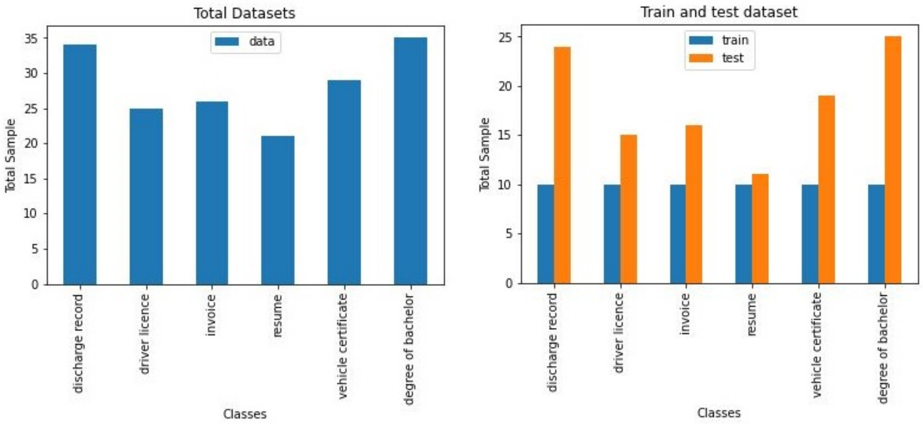

Figure 3: Dataset NLP

We also tried replacing NLP with CNN method with the above data and the result is not very likely with <60 % accuracy. Besides, a comparison was also made between NLP model and CNN combine models (Inception resnet and VGG16) on a dataset larger than 300000 document images on 14 classes are also updated in (Table 3). including a number of comparative parameters such as

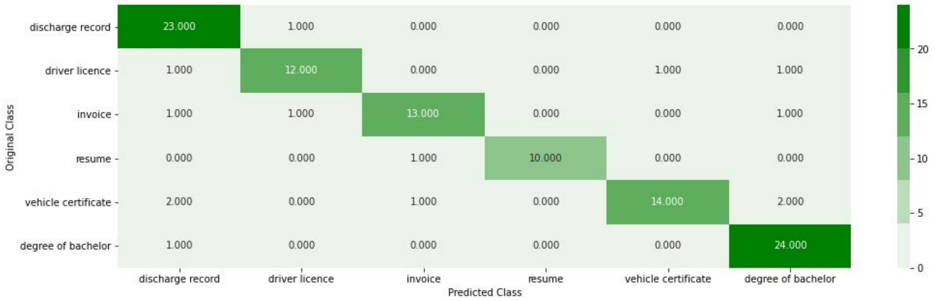

Figure 4: Confusion matrix NLP

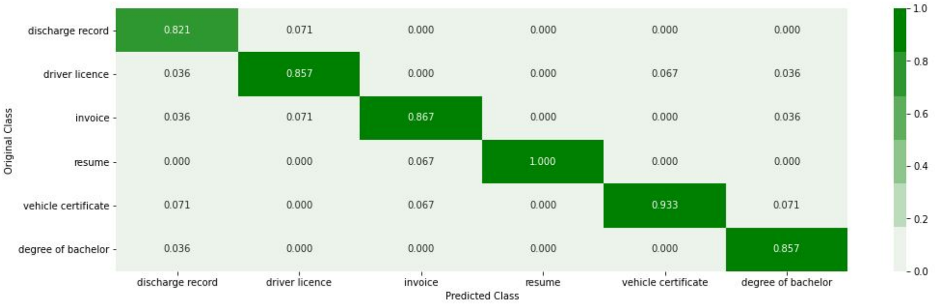

Figure 5: Precision matrix NLP

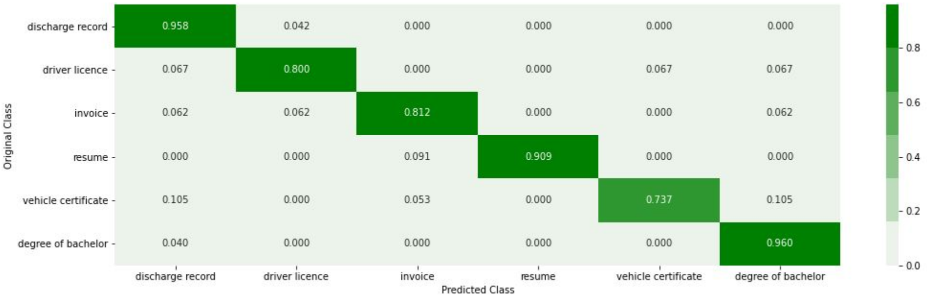

Figure 6: Recall matrix NLP

accuracy, train time, processing speed, with the goal of comparing the criteria of the report that we have set to determine if an alternative NLP can be applied.

3 Summary

For the NLP model, it is the best fit for our system in terms of accuracy, processing speed and flexibility. From the test results, the following issues need to be done to improve the accuracy, including: First, enhance the input image processing because this is a decisive factor in the results of extracting content from the document image. Whether. The second also related to information extraction from images is that the current OCR model using OCR Tesseract (Vietnamese, English) needs to be enhanced or replaced. followed by a model (phobert) embedding content into tensors (not recommended) and finally a redesign of the analog search system to replace the currently used ElasticSearch (not recommended).

References

[1] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis: The RVL-CDIP Dataset,
`https://www.cs.cmu.edu/ aharley/rvl-cdip`

[2] Arpan Das, Han Sheng, and Konstantinos G. Derpanis: Scanned-Document-Classification,
`https://github.com/arpan65/Scanned-document-classification-using-deep-learning`

[3] Xingjiao Wu, Ziling Hu, Xiangcheng Du, Jing Yang and Liang He: Document Layout Analysis via Dynamic Residual Feature Fusion,
`https://www.arxiv-vanity.com/papers/2104.02874/`

[4] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, Jimmy Lin: DocBERT: BERT for Document Classification,
`https://arxiv.org/abs/1904.08398`

[5] Elastic Stack and Product Documentation,
`https://www.elastic.co/guide/index.html`

[6] Jay Alammar: A Visual Guide to Using BERT for the First Time,
`https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time`

[7] Arthur Flôr: Text Segmentation,
`https://arthurflor23.medium.com/text-segmentation-b32503ef2613`

[8] Dat Quoc Nguyen, Anh Tuan Nguyen: PhoBERT: Pre-trained language models for Vietnamese,
`https://arxiv.org/abs/2003.00744`