[PROJECT CS-41 EXTRACT TABLES FROM FINANCIAL DOCUMENTS]

Project Proposal

Information Technology Capstone Project

COMP5703

Group Members

1. Mengting Wu (480551009)
2. Wanying Zhou (480256333)
3. Yiwen Gong (480481728)
4. Matthew Leung (480498601)
5. Jieming Ying (480538538)

ABSTRACT

This proposal is to propose a project for table extraction in financial literature text-based pdf documents. It consists of a brief introduction to the whole development of this project, and an application industrial literature review. Then the technical gap between current table extraction methods/tools and our project product is analysed. After specifying the project objectives and deliverable scope, the chosen methodologies for our application are illustrated in detail. Since our project development heavily depends on external resources such as python libraries/APIs, annotated table dataset, and pre-trained CNN models, a section will be included in this proposal to elaborate on them. In the end, our project's expected outcomes are summarised, and project milestones and schedules are presented accordingly for time management.

TABLE OF CONTENTS

1. INTRODUCTION

Given our project is targeting financial document information detection and extraction, the data analysis for given training and testing data will be concluded. Our techniques to detect and extract tables from pdf documents require several resources to support, such as a computer equipped with GPU, multiple OS/software environment dependencies, and pre-trained neural network models. All these resources are imperative for our expected outcomes and we will elaborate them in detail. We design our project outcomes into two categories, the first one is basic model deliverables, which contain the ones correctly detect tables from randomly given financial documents, recognize table structures and extract information clearly without noise or discarding information. The second one is an extended application based on former models, we would try to implement a platform to realise uploading PDF documents and loading customised forms to achieve a standard output for our product. At the end of this proposal, a project schedule that contains milestones and detailed timelines will be presented.

2. RELATED LITERATURE

2.1 Literature Review

Table detection is a crucial and a necessary field to discover and experiment on. However, it still consists of a lot of problems due to varying layouts and encodings of the tables. The major reason for causing failures of these methods is researchers rely on hand-engineered features which are not robust to different layout configurations.

2.2 PDFMiner

PDFMiner was built by Yusuke Shinyama in 2004 out of boredom and the first version was released in 2007. It focuses entirely on getting and analysing text data. PDFMiner allows one to obtain the exact location of text in a page (Khemiri, 2019). Strengths of PDFMiner include conversion functions that allow many format conversions such as html, xml and text retrieval. However, not every part is needed for most PDF processing tasks (Shinyama, 2013). Weaknesses of PDFMiner includes it is hard to keep the table layout as for extracting texts only. Locating the table from pure text is extremely difficult and infeasible.

2.3 PyPDF2

PyPDF2 was launched in 2005, the purpose of building this tool are document manipulation: by-page splitting, concatenation, and merging, document introspection, page cropping and document encryption and decryption. PyPDF2 provides a useful tool to extract information from a PDF file and it is easy to implement as well (Driscoll, 2018).The strength of PyPDF2 is that all contents would be extracted so the completeness of the extracted file is high-quality. One of the weaknesses is locating the table is the same issue of PyPDF2, this method cannot provide where the table is from the document. Extracting documents into HTML format may provide a way to check the <td> and <tr> tag.

2.4 Tabula

Tabula' s first version was released in 2016 by Aki Ariga. Copying tabular data and information from PDF documents is not an easy task. Tabula allows you to extract that data into a CSV or Microsoft Excel spreadsheet (Tabula: Extract Tables from PDFs, 2020). Strength of Tabula is the conversion function that is convenient to change the documents and extract information into different formats. However, a common weakness that Tabula also has is that it does not recognize ASCII letters. Thus, a single row will be split into several rows in this scenario.

2.5 Camelot (Excalibar)

Camelot is the major source that allows Excalibur to function. Camelot has two flavours, which represents the mode that you want to set for Camelot to detect the tables (Mehta, 2020). The Stream is for tables formed with whitespace, usually tables with no frames and borders will be applicable and have a more satisfying result. The Lattice flavor is for tables that formed with lines (Mehta, 2018).

The Strength of Camelot is that the table detection is relatively accurate but the heading may be split into several rows because Camelot does not recognize ASCII letters. Therefore, it will split the header into several cells.

2.6 Neural Network

The appearance of using deep learning for object detection and table detection has created a rapid advancement in the research area. Fast and Faster R-CNN allows a purely data-orientated approach to detect tables without the need for predefined rules. TableBank has created a table-orientated dataset to perform data extraction.

TableBank is built with novel weak supervision from Word and Latex documents on the Internet. It is built by the researchers from Beijing University from China and also with the help of the team of Microsoft in China (Beckmann, 2020).

A sufficient amount of training dataset is needed, a domain-orientated dataset is also another key characteristic for our training dataset. Current research for image-based table extraction methods and tools usually use out-of-domain data with few thousands of human-labelled examples to train on the dataset, which causes difficulty to generalize on real-world scenarios. With TableBank that contains 417K high-quality labelled tables, it is built on several strong baselines using state-of-the-art models with deep neural networks (Li, et al., 2019).

There are three major errors when the model is used. Partial detection is the first one, un-detection is the second error type and the third error type is misdetection.

3. RESEARCH/PROJECT PROBLEMS

In media-retrieval, especially textual media retrieval, the main text mining techniques are provided for extracting information from text. However, in financial literature, essential information is commonly located in tables, which are neglected in many data retrieval approaches due to its difficulties and complexities (Nikola Milosevic, 2019). Therefore, our research is aiming to examine and design methods that can better locate financial tables and extract its numerical (Class, Certificate balance/price) and simple textual (the rating) information in cells from financial pdf documents.

3.1 Research/Project Aims & Objectives

This project is commissioned to deliver a standardized table format that contains information that is of particular interest accurately extracted from the pdf documents.

3.2 Research/Project Questions

As the most popular documentation format, even a text-based PDF file cannot keep the original layout and structure of tables. Tables' structure and edges are commonly lost during the file conversion using different pdf generators. The question of our clients is actually to get an application that can reverse the tables conversion process in pdf files.

Problem clarification from our team in a technical way: First, the client needs an application that can take a financial pdf document as the input, then the output will be the critical tables automatically detected from the input. Second, it will be optimal if this application can also acquire the semantic meaning of each cell in different tables and optimize the outputs in a standardized way.

3.3 Research/Project Scope

3.3.1 Project deliverables

Our project deliverables are categorized into two genres, management-related deliverables, and product-related deliverables. The detail will be elaborated in section 6 Expected Outcomes.

3.3.2 Scope Statement

- In-Scope: A fully-tested application with its source code. ReadMe guidance will be provided to help users to install and run our application. Detection Model. Extraction Model. Graphical User Interface (optional). Maintenance and Update of Applications
- Out-of-Scope: Hardware to run our application will not be included. The client needs to install/run our application on his or her own devices. Software used in our project. The installation and maintenance of prerequisite dependencies for our application. Only concise dependency installation guidance will be provided, the client needs to follow the instruction and set up the work environment for our application by self. The labelled-dataset we used for our application model will not be provided, the client needs to apply for it from our team if it is necessary. We cannot guarantee the dependencies' updates will not compromise our application's performance in the future.

4. METHODOLOGIES

In terms of the methodologies, we tend to adapt the convolutional neural networks model as the detection and image-to-text model to implement the extraction. LabelImg is the main method to annotate the dataset and ROI curve is used to show the testing result and the performance of models.

4.1 Methods

4.1.1 LabelImg

LabelImg is an open-source, free of charge graphical image annotation labeling tool that is written in Python and uses Qt for its graphical interface. LabelImg supports bounding boxes for one-class tagging. ImageNet is an image database organized according to a lexical database of English hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images. ("tzutalin/labelImg", 2020).

In our project, we will use LabelImg to annotate each table from the pdf documents, no matter what format or layout the table is, and all these annotations will be done manually.

4.1.2 ResNeXt Model

ResNext is based on ResNet as well as the pattern of split-transform-merge inception. The advantage of this structure is that the precision can be improved by a wider or deeper network with a guaranteed number of flops and parameters, and each path in each block is called a cardinality in the article (Xie, Girshick, Ross, Piotr, & Kaiming, 2017). The essence of ResNeXt is group convolution, which controls the number of groups through cardinality. It is precisely because of cardinality, so that ResNeXt has better accuracy and effect, as Figure 2 shows below.

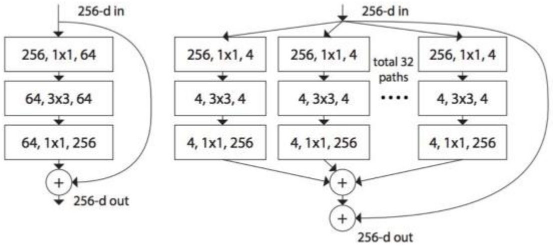
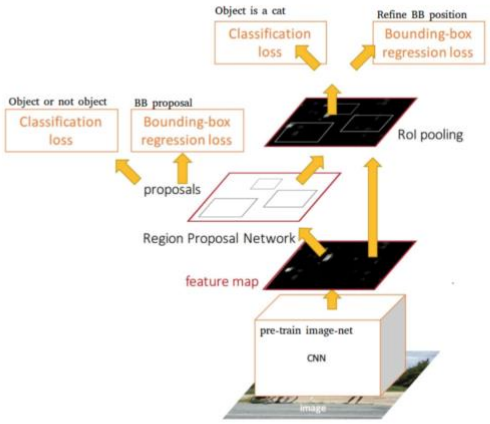


Figure 2. ResNet versus ResNeXt

As a result, we use the ResNeXt structure-based pre-trained neural network for data training and testing. In addition to the simple structure of the model, which is easier to understand and apply, it can also better prevent training overfitting when the given data set is relatively limited.

4.1.3 Faster R-CNN Model

The basic structure of Faster R-CNN is composed of the following four parts: feature extraction, Region proposal network, proposal layer and ROI pooling, as the following Figure 3 (Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, 2016):



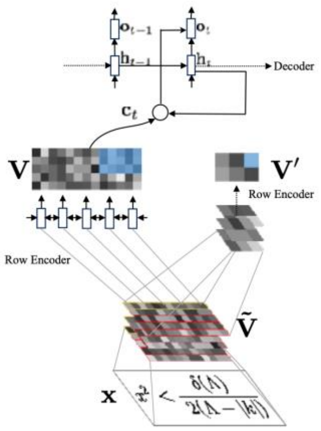


Figure 3. Structure of Faster-RCNN Figure 4. Network Structure (Deng et al., 2016)

As an end-to-end CNN object detection model, Faster-RCNN replaces the slow selective algorithm with RPN to generate region proposals and shares the underlying computation with the previous Fast structure. Therefore, the processing speed is greatly improved based on regional feature extraction. At the same time, the accuracy keeps like the original Fast-RCNN while it is very convenient and fast in the detection stage. Based on the ResNeXt structure in the feature extraction part, we selected the Faster-RCNN model to achieve higher detection accuracy and faster training speed.

4.1.4 Image-to-markup Model

Following the detection model above, all PDF pages have been transferred into JPEG images where tables are framed by the bounding box. The rectangles for tables are used as the input image of the structure analysis that is the main procedure to extract all contents which is a challenging task. Our base model is image-to-markup model (Deng et al., 2016) which is widely used in image captioning. It contains one row encoder, one full grid encoder for image input and one decoder for the text output, which are based on recurrent neural networks (RNN). The architecture for the whole model is shown in Figure 4.

4.2 Data Collection

In this project, our data are entirely from University of Sydney Business School, during the early phase of this project, the business school selects 21 sample pdf files from its database of over 4,000 pdf files. These sample PDF files are used as training dataset and testing dataset.

4.2.1 Table Source Analysis & Table Format Analysis

In the early phase of the project, our team is provided with 21 sample pdf files from USYD Business School, and these pdf files are offering the certificates. In this project, three areas are of particular interest to us as the following:

1. Classes of certificate: each class of certificate will receive monthly distributions of interest, principal or both, commencing on a specific date. The pattern of the class is defined as “Class A-1” for instance.
2. Original class certificate balance: the aggregate principal amount of a certain class upon initial issuance. The pattern of the certificate balance is defined as “$” + “100,000” for instance.
3. Pass-through rating: the “pass-through rate” for each class of certificate will be as set forth according to class classification. The pattern of the rating is defined as “AA+” for instance.

Varied kinds of table formats appear in these 21 sample pdf files. Each pdf sample file has 30 tables on average, and these tables are highly diversified into dozens of formats, such as tables with unparalleled headers and multiple headers, which present great challenges as we detect and extract information from the tables.

4.3 Deployment

Troubleshooting documents will be delivered as one of the product-related deliverables to our client. Environments of different operating systems, modules that needed to be downloaded to run PyTorch, Caffe2 and Detectron etc are all aspects that might cause bugs and errors.

The document will provide a detailed explanation of how you can resolve the bug with commands and how users can change their system settings to use our product.

4.4 Testing

Confusion matrix would be the major tool that we used to test our project. Precision, recall, accuracy and F score will all be used frequently in our testing phase.

In precision, we want to check how many tables are all identified correctly over the actual results, which actual result is the sum of true positive and false negative cases. On the other hand, recall refers to the percentage of total relevant results correctly classified by your algorithm. It looks for how many true positive cases being identified over the predicted results, which is the sum of true positive and false negative cases.

Accuracy represents the percentage of correctly identifying results over the total number of cases. The F score is used to measure a test's accuracy, and it balances the use of precision and recall to do it. The F score can provide a more realistic measure of a test's performance by using both precision and recall.

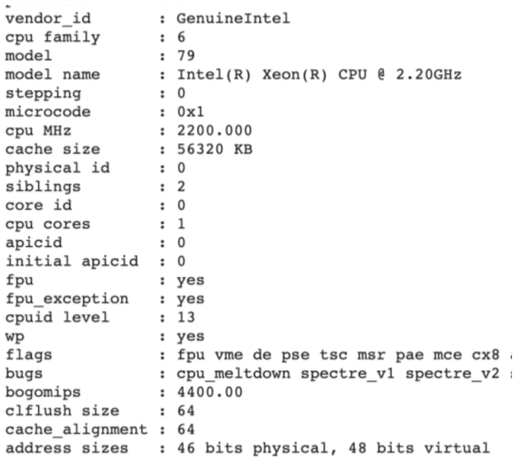
5. RESOURCES

5.1 Hardware & Software

The main software tools that will be used in the project are some open-source frameworks, Python modules and libraries which are:

- Detectron is high quality and high performance in object detection and supports the state-of-the-art object detection algorithms.
- OpenNMT is used for neural machine translation and neural sequence learning with two deep learning frameworks, OpenNMT-py and OpenNMT-tf.
- Python is the principal programming language and the whole project is built based on modules, PyYAML (version 3.12), Poppler and Pdf2image.
- LabelImg is the open source tool providing the support for the annotation for the dataset.

NVIDIA GPU is required to support the detectron and Google Colab is the platform with free GPU with the hardware illustrated in the figure below:



```
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU @ 2.20GHz
stepping        : 0
microcode       : 0x1
cpu MHz         : 2200.000
cache size      : 56320 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8
bugs            : cpu_meltdown spectre_v1 spectre_v2
bogomips        : 4400.00
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
```

Figure 5. Hardware of Google Colab

5.2 Materials

Google search engine provides websites introducing related knowledge and tools, such as Tabula, Camelot, PDFminer, PyPDF2 and so forth. GitHub has a big collection of open-source software for a pre-trained model which provides some reference and inspiration to promote the project. Google Scholar contains millions of metadata of scholarly literature as the reference and supporting material for the project.

5.3 Roles & Responsibilities

There are 5 main roles in our team:

Names	**Roles**	**Responsibilities**
Menting Wu	Team Leader	- Lead and manage the project team - Track and coordinate activities - Communicate with supervisors
Yiwen Gong	Leading developer	- Decide coding standards - Solution design and implementation - API configuration and methods selection.
Jieming Ying	Administer	- Take notes for weekly meetings - Prepares and facilitates meeting agendas - Conduct management documents
Jane Zhou	Data analyst	- Define the problems in the dataset - Figure out the relationship between the metadata - Translate issues to developers
Matthew Leung	Tester	- Test design - Internal testing - External testing

Table 1 Roles of Team Members

6. EXPECTED OUTCOMES

The expected product deliverables can process PDF documents and output all the table information. Although tables may exist in different forms, the deliverable will convert the table files in a general way into readable, complete tables in a uniform form.

6.1 Project Deliverables

6.1.1 Project Management Deliverables

- Project Proposal

The document that outlines all necessary elements to initiate the project, including introduction, related literature, research and project problems, methodologies, resources, expected outcomes and milestones.

- Project Charter

In particular, the project description and general description of deliverables are specified explicitly.

- Project Scope Statement

Further, the project scope, project justification & needs, milestones and deliverables are elaborated.

- Project WBS

The 2-level work breakdown structure is implemented.

- Project Time Schedule

The schedule is defined mainly in the form of Gantt charts.

- Documentation

Other relevant documents, such as software development documentation, test documentation, maintenance documentation and communication backlogs.

- Final Presentation

The relevant materials presented finally, mainly composed of models and application presentations and slideshows.

6.1.2 Product Deliverables

- Basic Model

The basic models are mainly composed of table detection, table structure recognition and information extraction.

- Extended Application

Further developed graphical user interface until tested application can realise uploading PDF documents and loading unified forms.

- User Manual and source code

The source code of the application along with all dependency's installation guidance, in the form of an instruction book.

6.2 Implications

Tables are an important source of information in documents, and their core is to provide users with information that can help them make decisions. Especially for financial documents, the table information including product rating and the unit price is of high reference and analysis value.

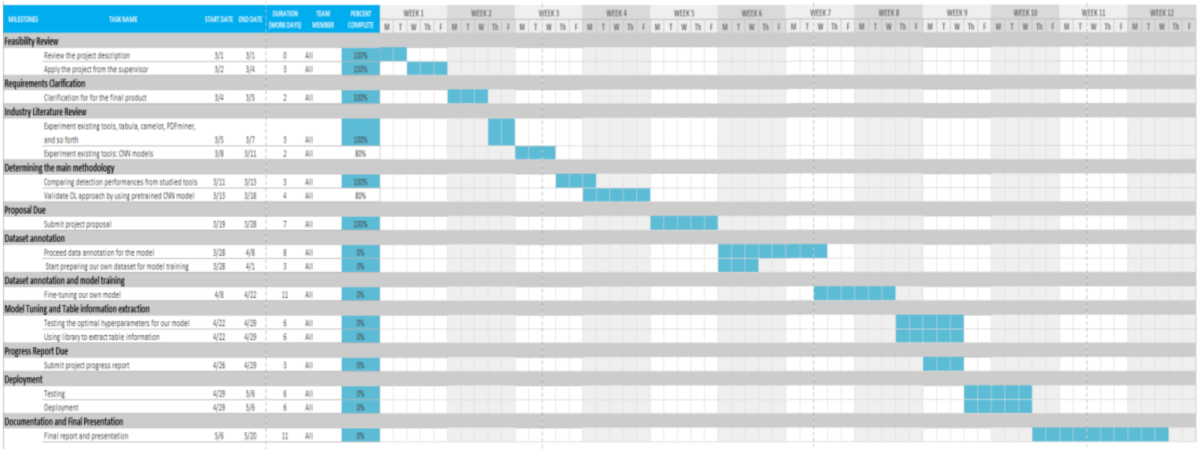
However, manual extraction of all forms is a heavy task, the high intensity of the work makes the member easy to fatigue and make mistakes in the state of repetitive work. In the actual operation, this situation leads to the low efficiency of basic data collection as well as the lag of comprehensive statistical data, which leads to the performance of other business information management systems (such as ERP and CRM) of the company is greatly reduced, thus affecting the correct decision-making of the enterprise.

The application uses this technology to automatically identify and grab the table, through batch scanning and recognition of the way to collect the form information, greatly improve the efficiency of data collection, the heavy repetitive work to the computer to deal with, and give full play to the advantages of computer processing technology.

7. MILESTONES / SCHEDULE

The milestones and tasks build up a three-layers work breakdown structure. And the schedule of the whole project is presented in Gantt Chart for future project time management.

1 Milestone	Tasks	Reporting	Date
Week-1	1. Review the project description. 2. Apply the project.	Project outline review and summarization	1-03-2020
Week-2	1. Clarifying requirements for the project.	Meeting to review the requirements	04-03-2020
Week-3	1. Experiment existing tools.	Slides for testing results	11-03-2020
Week-4	1. Compare detection performances using various tools. 2. Testing a CNN pretrained model.	Client meeting to present current progress.	18-03-2020
Week-5	Proposal Report Due	Proposal Report	28-03-2020
Week-6	1. Proceed data annotation for the model. 2. Start preparing our own dataset for model training.	The dataset annotation and model training progress.	01-04-2020
Week-7	1. Dataset labeling. 2. Build our own model and keep fine-tuning our own model.	Trained model with different hyperparameters and Training results.	08-04-2020
Week-8	1. Testing the optimal hyperparameters for our models. 2. Develop the way to extract information.	Client meeting to present our application’s performance.	22-04-2020
Week-9	Progress Report Due	Progress Report	29-04-2020
Week-10	1. Deployment 2. Testing	Client meeting to deploy the system	06-05-2020
Week-11	1. Collate documents	Related documents	13-05-2020
Week-12	Final Presentation	Slides for Presentation	20-05-2020
Week-13	Final Report (thesis)	Final Report	27-05-2020



MILESTONES	TASK NAME	START DATE	END DATE	DURATION (WORK DAYS)	TEAM MEMBER	PERCENT COMPLETE
Feasibility Review						
	Review the project description	3/1	3/1	0	All	100%
	Apply the project from the supervisor	3/2	3/4	3	All	100%
Requirements Clarification						
	Clarification for for the final product	3/4	3/5	2	All	100%
Industry Literature Review						
	Experiment existing tools, tabula, camelot, PDFminer, and so forth	3/5	3/7	3	All	100%
	Experiment existing tools: CNN models	3/8	3/11	2	All	80%
Determining the main methodology						
	Comparing detection performances from studied tools	3/11	3/13	3	All	100%
	Validate DL approach by using pretrained CNN model	3/13	3/18	4	All	80%
Proposal Due						
	Submit project proposal	3/19	3/28	7	All	100%
Dataset annotation						
	Proceed data annotation for the model	3/28	4/8	8	All	0%
	Start preparing our own dataset for model training	3/28	4/1	3	All	0%
Dataset annotation and model training						
	Fine-tuning our own model	4/8	4/22	11	All	0%
Model Tuning and Table information extraction						
	Testing the optimal hyperparameters for our model	4/22	4/29	6	All	0%
	Using library to extract table information	4/22	4/29	6	All	0%
Progress Report Due						
	Submit project progress report	4/26	4/29	3	All	0%
Deployment						
	Testing	4/29	5/6	6	All	0%
	Deployment	4/29	5/6	6	All	0%
Documentation and Final Presentation						
	Final report and presentation	5/6	5/20	11	All	0%

Figure 6. Gantt Chart

REFERENCES

Beckmann, C. (2019). TableBank: Benchmark for Image-based Table Detection and Recognition. Retrieved 27 March 2020, from https://syncedreview.com/2019/04/11/tablebank-benchmark-for-image-based-table-detection-and-recognition/

Brozovic, A. (2018). Opinion letter regarding the article Arch Toxicol https://doi.org/10.1007/s00204-018-2240-x. Archives Of Toxicology, 92(10), 3241-3241. doi: 10.1007/s00204-018-2272-2

Deng, Y., Kanervisto, A., Ling, J., & Rush, A. M. (2017, August). Image-to-markup generation with coarse-to-fine attention. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 980-989). JMLR. org.

Driscoll, M. (2018). Extracting PDF Metadata and Text with Python - The Mouse Vs. The Python. Retrieved 27 March 2020, from http://www.blog.pythonlibrary.org/2018/04/10/EXTRACTING-PDF-METADATA-AND-TEXT-WITH-PYTHON/

Fenniak, M., 2016. PyPDF2 Documentation — PYPDF2 1.26.0 Documentation. [ONLINE] pythonhosted.org. Available at: < https://pythonhosted.org/PyPDF2/> [ACCESSED 25 MARCH 2020].

Khemiri, A., 2019. PDF Processing with Python. [ONLINE] ,Towardsdatasceince. Available at: < https://towardsdatascience.com/pdf-preprocessing-with-python-19829752af9f> [ACCESSED 25 MARCH 2020].

Li,, M., Cui, L., Huang, S., Wei, F., Zhou, M. and Li, Z., 2019. TableBank: Table Benchmark for Image-based Table Detection and Recognition. [ONLINE] Available at: < https://www.researchgate.net/publication/331544220_TableBank_Table_Benchmark_for_Image-based_Table_Detection_and_Recognition> [ACCESSED 25 MARCH 2020].

Mehta, V., 2018. An Open-Source Tool to Extract Tables from PDFs into CSVs. Available at: < https://hackernoon.com/an-open-source-science-tool-to-extract-tables-from-pdfs-into-excels-3ed3cc7f22e1 > [ACCESSED 25 MARCH 2020].

Mehta, V., 2020. Camelot Documentation Release 0.7.3 [ONLINE] readthedocs.org. Available at: < https://readthedocs.org/projects/camelot-py/downloads/pdf/master/> [ACCESSED 25 MARCH 2020].

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. (2016, Jan 6). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Retrieved from https://arxiv.org/pdf/1506.01497.pdf

Shinyama, Y., 2013. PDFMiner Documentation. [ONLINE] pdfminer-docs.io. AVAILABLE AT: <https://pdfminer-docs.readthedocs.io/pdfminer_index.html> [ACCESSED 25 MARCH 2020].

Simonov, K. (2016). yaml/pyyaml. Retrieved 26 March 2020, from https://github.com/yaml/pyyaml/blob/master/README

Tabula.Technology. 2020. Tabula: Extract Tables From PDFs. [ONLINE] AVAILABLE AT: < https://tabula.technology/ [ACCESSED 25 MARCH 2020].

tzutalin/labelImg. (2020). Retrieved 27 March 2020, from https://github.com/tzutalin/labelImg

Xie, Girshick, Ross, Piotr, & Kaiming. (2017, April 11). Aggregated Residual Transformations for Deep Neural Networks. Retrieved from https://arxiv.org/abs/1611.05431