

End-to-End Tabular Information Extraction In Datasheets With Deep Learning

by

Ertugrul Kara

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

© Ertugrul Kara, Ottawa, Canada, 2019.

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The advent of Industry 4.0 phenomenon has been transforming the information management regarding the specifications of electronic components. This change affects many organizations, including global supply chains that optimizes many product chains, such as raw materials or electronic components. Supply chains consist of thousands of manufacturers and connect them to other organizations and end user, and they include billions of distinct components. The digitization of critical information has to be carried out automatically since there are millions of documents. Although the documents vary greatly in shape and style, the essential information is usually presented in the tables in a condensed format. Extracting the structured information from tables are done by human operators, which costs human effort, time and corporate resources. Based on the motivation that AI-based solutions are automating many processes, this thesis proposes to use deep learning-based solutions for three main problems: (i) table detection, (ii) table internal structure detection and (iii) End-to-End (E2E) tabular structure detection. To this end, deep learning models are trained mostly with public datasets, and a private dataset (after labelling 2000+ documents) which was provided to us by our industry partner. To achieve accurate table detection, we propose a method based on the successful Mask-Region-Based Convolutional Neural Network (Mask-RCNN) instance segmentation model. With some modifications to the training set labels, we have achieved state-of-the-art detection rates with 99% AP and 100% recall. We use the PASCAL Visual Object Classes (VOC) 11-point Average Precision (AP) metric to compare the evaluated deep learning-based methods. Detecting tables is the initial step towards semantic modeling of e-components. Therefore, structure should also be detected in order to extract information. With this in mind, we introduce another method based on the Mask-RCNN model, which is able to detect structure at a with around 96% AP. Combining these two networks, or developing a new model is a necessity. To this end, inspired by the success of Mask-RCNN models, we introduce the following Mask-RCNN based models to realize E2E tabular structure detection: *Stitched E2E model* achieved by bridging the output of table detection model into the structure detection model, attained more than 77% AP on the difficult public UNLV dataset with various post-processing steps applied when bridging the two network. *Single-pass E2E detection networks* were able to attain a higher AP of 86% but with lower recall. This thesis concludes that deep learning-based object detection and instance segmentation networks can accomplish state-of-the-art performance.

Acknowledgements

I would like to start with expressing my honest gratitude to my supervisor Dr. Burak Kantarci, for his unlimited support, guidance and enthusiasm. His continuous assistance and passion are what made this progress possible in such a short time. Without his guidance, expertise and attention to detail, I would not be able to overcome the work we have done and publish our findings. I am glad and thankful that I had Dr. Burak Kantarci as my supervisor, and I could not imagine going through my studies without his supervision.

I would like to thank Mark Traquair, whom we have worked together throughout a part of the project and Dr. Murat Simsek for his endless enthusiasm, support and constructive criticism. Their knowledge and assistance were invaluable, and we had a productive collaboration. I would like to thank Lytica Inc. and Gnowit Inc. for supporting this project, providing a dataset and resources of which made this thesis possible.

Finally, I must express my very profound appreciation to my family. They always believed in me, and they have been my greatest supporters. Throughout my years of study, I have always felt their encouraging supports behind me, and this support is what kept me motivated. This thesis would not have been possible without them.

This thesis is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) ENGAGE Program and Ontario Centres of Excellence Voucher for Innovation and Productivity(VIP) I Program under Project Number 29931, Gnowit Inc. and Lytica Inc.

Table of Contents

List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Contributions	5
1.4 Thesis Outline	6
2 Related Work	8
2.1 Hand-crafted Heuristics and Machine Learning	9
2.1.1 Hand-crafted Rules	9
2.1.2 Machine Learning	10
2.2 Table Detection	12
2.2.1 Rule-Based Approaches	12
2.2.2 Machine Learning-Based Approaches	16
2.3 Table Structure Detection	22
2.3.1 Rule-Based Approaches	22
2.3.2 Machine Learning-Based Approaches	24

3	Methodologies	26
3.1	Deep Learning-Based Methods	27
3.1.1	Base Networks	27
3.1.2	R-CNN Family	34
3.1.3	Feature Pyramid Networks	48
3.1.4	Fully Convolutional Networks	51
3.1.5	RetinaNet	53
3.2	Rule-Based Mathematical Morphology Method	56
4	Table Detection	58
4.1	Datasets	59
4.2	Methods	66
4.3	Training Details	69
4.4	Results	73
4.5	Conclusion	75
5	Table Structure Detection	81
5.1	Datasets	83
5.2	Methods	84
5.3	Training Details	88
5.4	Results	92
5.5	Conclusion	94
6	End-to-End Tabular Structure Detection	101
6.1	Dataset	103
6.2	Methods	104
6.3	Training Details	116
6.4	Results	118
6.5	Conclusion	123

7 Conclusion	124
7.1 Future Directions	125
References	128

List of Tables

2.1	Overview of table detection and table structure detection research. TD: Table Detection, CD: Column Detection and RD: Row Detection	25
4.1	Datasets and distribution of the images.	59
4.2	Table detection performance on public ICDAR datasets.	74
4.3	Table detection performance on public UNLV dataset and our private dataset.	74
5.1	Comparison of considered methods.	91
6.1	Table detection performance on public ICDAR datasets.	119
6.2	Table detection on private dataset and on public UNLV dataset.	119
6.3	E2E performance of the proposed methods	119

List of Figures

1.1	Example document flow which exists throughout all industry.	4
3.1	VGG-16 architecture used in the thesis	30
3.2	ResNet-50 architecture used in the thesis.	32
3.3	ResNeXt-101 architecture used in the thesis.	35
3.4	R-CNN method architecture used in the thesis. Operations are shown for a single bounding box proposal.	37
3.5	Fast-RCNN method architecture used in the thesis.	39
3.6	Overview of the Region Proposal Network used in the thesis.	42
3.7	Overview of the Faster-RCNN method architecture used in the thesis. . . .	44
3.8	Mask-RCNN method architecture used in the thesis.	46
3.9	Feature pyramid networks used in the thesis. Coloured boxes represent the images and white boxes stands for feature maps.	49
3.10	A fully convolutional architecture based on VGG-16 to classify pixels. FCN-32s upsamples the predictions made with 32 stride back to pixels. FCN-16s combines the output of both the last layer and the 'pool4' layer which has stride of 16. FCN-8s is the combination of information from pool3, pool4 and conv7. It provides further precision and allow the network to keep the high-level semantics as well as providing finer details about the input. . . .	52
3.11	The RetinaNet architecture used in the thesis. Purple boxes represent the input image and white boxes stand for feature maps.	54
3.12	The morphological method examples. All operations are shown with a simple 5×5 square kernel.	56

4.1	Example images from the ICDAR 2013 dataset.	61
4.2	Example images from the ICDAR 2017 dataset.	62
4.3	Example images from the Marmot dataset.	64
4.4	Example images from the UNLV dataset.	65
4.5	Example images from our private dataset.	67
4.6	Training loss over time for the Mask-RCNN models.	72
4.7	Example detection results by our Faster-RCNN based model with VGG-16 backbone network. Detections are not accurate, however it shows promise on these hard examples and even though it has only 16 layers CNN, captures the idea of tables.	76
4.8	Example detection results by our RetinaNet based model with ResNet-50 backbone network.	77
4.9	Example detection results by our RetinaNet-based model with ResNeXt-101 backbone network. Detection is precise but not all tables are detected. . .	78
4.10	Example detection results by our Mask-RCNN based model with ResNet-50 backbone feature extractor.	79
5.1	Example images from the UNLV table structure dataset. As it can be seen, some tables have dense rows and columns (left) and a table of content example (right).	84
5.2	Example detection results by our Faster-RCNN based model with VGG-1 backbone network.	95
5.3	Example detection results by our Faster-RCNN based model with ResNet-50-C4 backbone network.	96
5.4	Example detection results by our RetinaNet based model with ResNet-50 backbone network.	97
5.5	Example detection results by our Mask-RCNN based model with ResNet-50 backbone feature extractor.	98
5.6	Example detection results by our heuristics-based method. Multi-line rows are merged into a single row which shows the effectiveness of the connected components.	99
6.1	System flow diagrams for proposed E2E systems.	105

6.2	Pure E2E Tabular Structure detection examples from Model D.	106
6.3	Agreement module - table agreement algorithm flow	108
6.4	Agreement module - line agreement algorithm flow	110
6.5	Example case for the Line Agreement module steps.	110
6.6	Example cases for the Judge mechanism . Different colors stand for segmen- tation of different instances of the same class.	112
6.7	The architectures of the proposed binary table classifiers.	113
6.8	System flow diagrams for proposed post-processing steps.	115
6.9	False-positive detection examples from the proposed table detection approach.	117
6.10	E2E detection examples for proposed models.	122

Nomenclature

β	Scaling factor in focal loss.
γ	Focusing parameter of the focal loss.
Ω	Minimum side length condition when stretching the dataset.
ϕ_5	denotes that the features extracted from the fifth convolutional layer.
b^l	Bias vector of the layer l .
$Conf$	Confidence of a detection.
i and j	Indices for matrices and vectors.
L_1	L1-norm or known as Least absolute deviations.
L_2	L2-norm or known as Least square errors.
L_{cls}	Classification loss.
L_{mask}	Segmentation loss.
L_{reg}	Regression loss.
$LReLU$	Leaky ReLu: An activation function.
p	Represents a probability.
$smooth_{L_1}$	Smooth- L_1 loss.
t_θ	Coordinates of the predicted bounding box where θ is one of $x, y, width, height$.
w^l	Weights of the layer l .

y^l Output of the layer l .

z^l Input to the layer l .

Glossary

AP Average Precision.

AR Average Recall.

BatchNorm Batch Normalization.

CNN Convolutional Neural Network.

CRF Conditional Random Field.

DOM Tree Document Object Model Tree.

FCN Fully Convolutional Network.

FPN Feature Pyramid Network.

GroupNorm Group Normalization.

ICDAR International Conference on Document Analysis and Recognition.

ILSVRC ImageNet Large Scale Visual Recognition Competition.

IoU Intersection over Union.

mAP Mean Average Precision.

MS COCO Microsoft Common Objects in Context.

OCR Optical Character Recognition.

PASCAL VOC PASCAL Visual Object Classes.

POD Page Object Detection.

ResNet Residual Networks.

RoI Region of Interest.

RPN Region Proposal Network.

SCM Supply Chain Management.

SVM Support Vector Machine.

Publications of the Candidate During MCS Studies

Direct outcomes of the thesis:

- **E. Kara**, M. Traquair, B. Kantarci, and S. Khan, Deep learning for recognizing the anatomy of tables on datasheets, in *IEEE Symposium on Computers and Communications (ISCC)*, (Barcelona, Spain), June 2019
(Accepted)
- M. Traquair, **E. Kara**, B. Kantarci, and S. Khan, Deep learning for the detection of tabular information from electronic component datasheets, in *IEEE Symposium on Computers and Communications (ISCC)*, (Barcelona, Spain), June 2019
(Accepted)
- **E. Kara**, M. Traquair, M. Simsek, B. Kantarci, and S. Khan, Holistic Design for Deep Learning-Based Discovery of Tabular Structures in Datasheet Images, *Engineering Applications of Artificial Intelligence*, 2019
(Under Review)

Chapter 1

Introduction

1.1 Motivation

The world bank estimated that there were more than 3.4 billion internet users throughout the world in 2016 [1]. The wide acceptance of internet opened new possibilities; Cloud computing matured, Internet of Devices (IoT) systems have been accepted and started to be used, new Industrial revolutions are started to happen. Industry 3.0 began with the use of computers, automation and information technologies (IT) in general. It was aimed at automating manufacturing processes with simple computers such as Programmable Logic Controllers (PLC) and required human input. Shortly after, Industry 4.0 is surfaced. The new industrial revolution includes smart devices in which they communicate with each other, send information, and the end goal is to automatize all the processes with no human intervention what so ever. The new revolution and IoT devices are now becoming pervasive and applied everywhere which enables large-scale sensing to improve various processes and operations. The information generated by these inventions is mostly digitally structured since no human intervention is wanted. However, even though some operations are digitized, digitization of every process is still far ahead. Until the digitization of the world is complete, the industry will continue to benefit from traditional formats such as printed papers or visual documents which are designed to be read, inspected and reviewed by humans. These data formats are mostly in PDF, DOC(x) or scanned documents format and can be said to be designed for human use and computers cannot comprehend them easily. Therefore, even though the data in an industrial process is automated and benefits from the ubiquity of the Internet, cloud computing, AI and IoT; inter-organizational processes still rely on conventional methods and is needed to be converted.

The data generated by the industry is available online and since the internet came into our houses and became the heart of many organizations, organizing manufacturing processes and connecting retailers with manufacturers became harder to manage. To solve this issue, a concept called *supply chains* emerged. Supply chains are defined as a group of organizations communicating with each other to make the flow of services and information between different entities easier [2]. Management of the supply chains, called Supply Chains Management (SCM), is poised to benefit from the ubiquity of the IoT, Internet and Artificial Intelligence (AI). The main goal of an SCM is to create value, leverage logistics and synchronize the demand with supply. An SCM has to be innovative to support and grow the commerce market that they support and to achieve this goal they have to establish intelligent methods and cater innovative insights about the processes to provide the necessary assistance for rapid growth [3]. Since the analysis of the data flowing through the supply chains is essential and SCMs are information-intensive, the data should be available and structured to be analyzed. A typical SCM includes a flow of products and data from raw materials until to the product which is sent to the end user. In the middle parts of this chain, manufacturers, retailers and other organizations have to share data and find suitable suppliers for themselves. Considering Industry 5.0, which planned to bring full automation in the process, is not here yet, human operations is still a significant player in the chain and therefore, the formatted data is usually prepared with humans in mind. Human operators are still significant members of an organizational process but optimizing the supply chain and extracting the maximum value can only be achieved with data-driven solutions.

We focus on the optimization of the aforementioned SCM operations. Much of the information used in an SCM is shared with mostly *digital-born* documents where the document is created and read in computers. However, PDF files are not created to be used by computers, but they are formatted with visual considerations. In addition, the same problem can be extended beyond the supply chains. For example, government offices (i.e. hospitals) still use conventional papers where the document is printed, filled by a person and then usually a human operator registers the information. Scanning the printed documents, extracting structured information and analysis of the historical documents is a popular research topic [4, 5, 6] but not the main focus of this research. Instead, we primarily focus on information extraction from digital-born documents, but since we use image formats as input to our system, scanned documents work just as well. Since information transfer is at the heart of SCM and a time-consuming task for almost every business and organization, in this thesis and in our previous works [7, 8, 9] we focus and solve the information extraction from documents. We show that the most essential and important information is given in tables in a document and this is based on the observations we

made. Some examples of technical supply chain document can be seen in Figure 4.5. In the electronic component datasheets tables contain information about the voltage values, capacitors and more. Therefore, we believe that tables are the most critical elements in a document in terms of ease of understandability and compactness.

In addition to tables, the main elements of documents are text blocks, formulas and figures. These, although less than tables, provide critical information as well. Since these documents are created to appeal to humans, they adopt distinct and unique styles differing in between documents from various sources. Commercial tools that address this task only operates on pre-set templates and cannot accurately detect tables with unique styles. To solve all of the points mentioned, we are motivated to solve the problem of table discovery and information extraction in almost all documents without previous knowledge on the document while staying fast and accurate. Operating on non-image files (i.e. PDFs) would allow us to use metadata, but then, we would be forced to operate on one kind of documents only. Therefore, we chose document images as input and train deep object detection networks as prior works have shown that they give viable solutions on table detection task [7, 9, 10]. After the table detection, we detect smaller structures that are semantically relevant to each other (rows and columns) from table crops. This is also a challenging task. In table detection, the system should differentiate figures and tables, but in structure detection, every structure possibly can have figures and tables inside them; hence creating a nested table structure. In our previous works [8, 9] we have used Mask Region-Based Convolutional Neural Network (Mask-RCNN) [11] and presented that a deep learning based object detection or instance segmentation model is sufficient to provide state-of-the-art performance with little to no pre and post-processing.

1.2 Objectives

When a simple SCM or a government is considered, we can see that possibly millions of documents that contain valuable information is present in these ecosystems and great benefits await upon analysis of the data. For instance, in an electronic component ecosystem, there are many retailers, manufacturers of components and many more entities in the chain. Assuming every new component is released with accompanying technical documents, there will be millions of documents that contains vital information for many other manufacturers, retailers and the end user. To optimize a supply chain and provide innovative insights to the entities inside, **converting these documents to viable formats which are suitable for further analysis** is crucial. The analysis results will give matching products for businesses and possibly with the lowest cost and improve the organizations in a chain

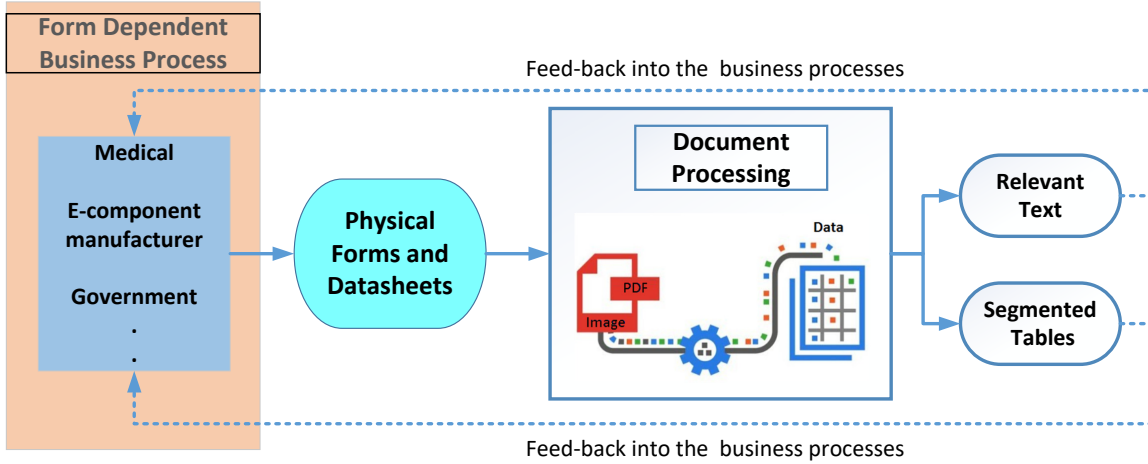


Figure 1.1: Example document flow which exists throughout all industry.

as a whole. An overview of a use case of information extraction from documents can be seen in Figure 1.1. Our objective is extracting information contained in tables in document images. We formulate our solutions such that we accept any image as input. Therefore, if the document is scanned and in pristine condition or if it is born-digital, we can utilize our End-to-End (E2E) system. This thesis divides the information extraction into two stages; (i) table detection and (ii) table structure detection. Tables are structured components of a document and composed of rows and columns which determines the reading order of the table and creates key-value pairs. Detecting these structures directly in a single-stage is also possible. However, as we show in Chapter 5, rows are only a few pixels in height, which makes them visually challenging to distinguish. Moreover, we employ deep learning based solutions (specifically deep object detection networks), and since they rely on visual saliency, the task of detecting structure directly from the document pages is challenging. Therefore, we exercise a two-stage system wherein the first stage we detect and crop a table region, and this crop is used in the second stage to detect rows and columns that belong to the table to finalize the tabular structure detection. These detected structures are then ready to undergo semantic meaning extraction phase, which is not a focus of this thesis.

1.3 Contributions

After conducting a thorough literature review, we introduce improvements upon previous works, stated the limitations, found possible future directions with high possibility to improve the information extraction from documents further. We propose a two-stage deep object detection network based system to detect tables from arbitrary documents and extract structures from the tables to achieve our objectives described in Section 1.2.

Our main contributions can be summarized as follows;

- The first contribution is that we propose to use object detection networks to detect tables in document images and to prove that it works, we compare state-of-the-art object detectors; Faster-RCNN [12], RetinaNet [13] and Mask-RCNN [11]. To train these networks, we use public and private datasets and adapt pre-trained feature extractors such as ResNet [14] to our domain with transfer learning. The most important contribution of this thesis is that we prove that an instance segmentation network Mask-RCNN achieves the new state-of-the-art performance on table detection task and outperforms all other methods that we studied. We conclude that this is largely due to the underlying architectures such as Feature Pyramid Networks (FPNs) [15], Fully Convolutional Networks (FCNs) [16] and Group Normalization (GroupNorm) [17]. We also observe that accepting larger input images allows the model to learn better representations which leads to higher accuracy. However, since GPU rams are limited, we cannot use large mini-batch sizes and Batch Normalization (BatchNorm) [18] renders useless. In this case, Mask-RCNN with GroupNorm performs better and allows the network to learn better.
- The second contribution of this thesis is that we propose a deep learning based structure detection method powered with state-of-the-art object detectors and feature extractors. Unlike the previous work, we do not detect the structures from the document page. Instead, we crop the table regions first and apply structure detection on top. To achieve accurate table detections, we test the same deep learning models as well as a mathematical morphology based method with heuristics. We conclude that Mask-RCNN based model performs best, in alignment with the table detection stage.
- We propose various novel E2E detection system designs to detect tables as well as rows and columns in the documents. The first proposal is called the Stitched E2E system from now on and is based on Mask-RCNN instance segmentation method as we show that it outperforms other object detection models. This system split the

problem into two; (i) Table detection from document image and cropping the detected region (ii) Using the crop from the first stage, detect rows and columns. While bridging the two models, we also proposed different post-processing steps aimed at eliminating false-positives and feeding better table crops to structure detection stage. We show that they have a positive effect on precision. The second proposed E2E system is the Pure E2E table structure system designed to detect rows and columns only. We also propose the Pure E2E tabular structure system where the output includes tables as well. We conclude that Pure E2E tabular structure model attains the highest precision with lower recall than the Stitched E2E system and explain the reasons.

1.4 Thesis Outline

The thesis is comprised of seven chapters. In Chapter 2 we present a thorough literature study and explain works that studied the same problem. In Section 2.1, we present a summary of the methodology used in the previous work. We describe the hand-crafted rules that are used in either this task or similar tasks (Subsection 2.1.1) and then describe machine learning and deep learning based solutions used in previous work (Subsection 2.1.2). Then in Section 2.2, a thorough background on table detection task is presented both in hand-crafted rule-based approaches (Subsection 2.2.1) and Machine/Deep learning based solutions (Subsection 2.2.2). Apart from the table detection, we also present previous work on table structure detection task in Section 2.3 although there are not many. Table structure detection models are again divided into two; hand-crafted heuristics based solutions are summarized in Subsection 2.3.1 and machine learning based approaches in Subsection 2.3.2.

In Chapter 3 we propose the methodology and explain the methods we are using throughout the paper. The chapter starts by studying the deep learning based solutions we benefit from in Section 3.1. Our deep learning models use other deep learning image classifiers as feature extractors and Subsection 3.1.1 explains both of the networks we use. Subsubsection 3.1.1 introduces the VGG [19] networks and Subsubsection 3.1.1 summarizes the ResNet [14] networks. Subsection 3.1.2 introduces the background of our Mask-RCNN object detector and introduces R-CNN [20] (Subsubsection 3.1.2), Fast-RCNN [21] (Subsubsection 3.1.2), Faster-RCNN [12] (Subsubsection 3.1.2) and Mask-RCNN [11] (Subsubsection 3.1.2) incrementally. Then, we introduce two architectures that we use in Mask-RCNN model with ResNet feature extractor. They are FPN, explained in Subsection 3.1.3 and FCN which is introduced in Subsection 3.1.4. Then we introduce our latest deep

learning based model RetinaNet [13] in Subsection 3.1.5. Finally, this chapter concludes with presenting the only heuristics-based model we use; Mathematical Morphology [22] in Section 3.2.

In Chapter 4, we present our solution to the table detection problem from documents. First, our datasets are defined in Section 4.1 followed by the detailing the methods we use for this purpose in Section 4.2. Then, the training process is summarized in Section 4.3 and then we give our results (Section 4.4) and conclude the table detection model (Section 4.5), which forms the first stage of our Stitched E2E system design.

In Chapter 5, similar flow to the previous chapter is observed where we describe our approach to the table structure detection problem. Section 5.1 gives information about the structure dataset and Section 5.2 introduces the models we compare. Then, the training processes of the deep learning based methods are described in Section 5.3 and then Section 5.4 displays our results and findings, and the approach is finalized in Section 5.5. This chapter forms our second stage in the Stitched E2E system.

Chapter 6 is where we form the Stitched E2E system and propose Pure E2E systems. Section 6.1 talks about the datasets we use and Section 6.2 introduces the methodology concerning the proposed E2E systems and discusses the design choices. The training details of the models are discussed in Section 6.3 and we present our findings in Section 6.4. Section 6.5 comes to a conclusion about the comparison of the proposed E2E systems.

Ultimately, Chapter 7 winds up the approaches discussed in the thesis, a final outcome is conferred, and we review possible additions to be made and give future directions in Section 7.1 and finalize the thesis.

Chapter 2

Related Work

Nowadays many of the physical documents never see the light of the analog world. They are created, filled, stored and processed in computers. Of course, we cannot ignore the physical data, and they have to be converted to digital formats, and there is research done focused explicitly on this problem [23, 24, 7, 8, 9]. However, the same problem persists even when historical documents are scanned and converted to digital documents. These documents hold essential information such as product specifications, availability or hazardous warnings in the supply chains perspective and safety-critical information regarding citizens or other organizations for the government perspective.

As our industry partner experiences, the most valuable information is mostly presented in tables, figures and formulas. Tables contain key-value pairs where there is usually one key paired with many values, and the information in the rows and columns are important to understand the overall content. Therefore, discovery and the extraction of the information from the tables is a crucial element to understand the document.

The methods developed for table detection range from hand-crafted rule-based methods to deep learning-based methods which stem from object detection problem. Treating the tables as objects in a document image or making use of alignments, tabs and white-spaces or lines are the two most popular choice for this problem. Structure detection has the same characteristics as tables, and the same ideas apply. In addition to other methods, prior research also proposed a pixel-level classification for structure detection.

2.1 Hand-crafted Heuristics and Machine Learning

Electronic documents can be separated into two main categories; documents as images and documents in PDF. Electronic documents do not necessarily mean PDF files. Some examples are; scanned documents (images), online documents (HTML) or documents in other formats, such as docx or odt. Since these other types are not as popular as PDFs when sharing the documents, the research focuses on these two types. Especially since we can convert any of these document types to document images, concentrating on solving the tabular data extraction on document images encapsulates all of these sub-problems.

Extracting unstructured information from the printed documents is a need and being studied for a long time [7]. Prior work is mostly based on hand-crafted rules that are extracted with a visual inspection by the human operators [25, 26, 27]. More recent works, however, benefit from the image classification networks success in a wide range of problems and often based on machine learning and deep learning.

2.1.1 Hand-crafted Rules

Before AlexNet [28] showed the power of the CNNs, object detection was usually performed with more traditional methods where we extracted features and apply the k-Means algorithm to the structures such as bag-of-features. However, since those methods were not entirely compelling, creating heuristics to detect structured information resulted in a better performance.

The most commonly used rules are based on detecting lines and white spaces to detect tabular regions as in [29, 30, 31, 32]. These structures helped achieve state-of-the-art results since many tables have bounding lines and spaces to separate rows and columns which makes the table detection task easier. One other common method is to segment words into blocks and then use alignment and neighbouring relations with other words or space around them [33, 29]. Finding bounding boxes for words is straightforward for PDF documents, but when the document is in image format, another layer of processing to find words from the pixels is needed.

Mathematical morphology [22] has two methods that can be used in combination to extract text blocks as connected components. Morphological dilation of the pixels followed by morphological erosion fills in the small gaps between characters in an image and ideally, all the characters in a word will now be connected. Following the morphological closing, a connected component finding algorithm (Union-Find) is applied. Resulting connected components are then classified as words or lines, according to their size. The outcome is

used as word bounding boxes which form the basis of many research on tabular structure detection [34, 31, 8].

2.1.2 Machine Learning

Although hand-crafted heuristics are widely adopted, more recent research explored the machine learning-based approaches [7, 9]. Support Vector Machines (SVMs) is a supervised learning method which finds a linear (or non-linear) separator to classify data [35]. In machine learning approaches, line or word bounding box detection is still necessary, and after the extraction phase, extracted features are inputted to an SVM classifier to detect tables. [36, 37, 38] demonstrate that this process yields good performance.

In addition to SVMs, others have also tested out machine learning approaches like Decision Trees [39], Hidden Markov Models (HMMs) [40] and Conditional Random Fields (CRFs) [41]. Decision Trees are a machine learning algorithm which produces a tree-like structure from the training set to create a tree which classifies the input. Since decision trees require a structured input, they are mostly used to detect tables in the context of HTML documents. Hidden Markov Models are probabilistic Markov models with unobserved states. With information like the appearance of images and certain words or hyperlinks, derived from the document metadata, HMMs can be used to locate the tables. CRFs are generally used for pattern detection and in the tabular structure detection context, they are used to model the dependencies within the data.

Deep Learning

Deep learning name is attributed to a small subset of the machine learning algorithms. Where learning with multi-layer perceptrons or more specialized architectures such as LSTMs, or CNNs are involved. This thesis solely focuses on the widely accepted deep learning framework for computer vision problems, CNNs. Since we focus on tabular structure detection on document images, CNNs are best suited for this task.

CNNs are specialized deep neural networks based on the multilayer perceptrons and are designed to match the visual processing of animals. First, Kunihiko Fukushima proposed the neocognitron (1980) [42]. It was inspired by the prior research conducted on monkeys that some neurons in visual cortexes of these animals only respond to specific regions of the visual field [42]. During these times, backpropagation (the de facto optimization method to use in neural networks) was not as developed as it is today. Hence these early networks were trained with different optimization methods and were not successful. Yann LeCun

(1989) [43] proposed a neural network where learning was fully automatic and trained with backpropagation and gradient-descent. [43] can be said to be the first of today's CNNs and afterwards, CNNs are used and developed for different problems, but the long-awaited breakthrough did not happen until AlexNet [28].

Although the performance of SVMs was increasingly powerful with the kernel functions [44], AlexNet [28] won the ImageNet competition [45] in 2012 and outdistanced the previous winner with a considerably high-accuracy. Krizhevsky et al. (2012) [28] proved that CNNs surpasses all other methods in image classification and merit further inspection. Their CNN AlexNet's success was possible due to the recent increase in computational power, consequently, with the possibility of having many layers and parameters. Although AlexNet seems to have reached a new peak, following CNN-based methods performed increasingly better on similar tasks [46, 14, 47]. Simonyan and Zisserman proposed the VGG method (2014) [19]. Authors observed the modifications made since the AlexNet and addressed the depth parameter. They have tested different depths and concluded that 16 layer deep CNN was the best performing model. In 2014, Szegedy et al. [46] introduced the *Inception Module* which is several convolutional neural networks piled next to each other and then the outputs are concatenated. They then stacked several inception modules and created the *GoogLeNet*. Which, at the time, was the one of the deepest convolutional neural networks with 22 layers.

VGG and GoogLeNet tried training CNNs with more layers, but the accuracy was degrading. The main reason behind this phenomena is called *vanishing gradient problem*. Since layers are equipped with convolution and max-pooling layers and activation functions like sigmoid, which shrinks the values to 0-1 range, after subsequent multiplications, the gradient is lost or not the same as in the last layer where it is calculated. He et al. (2015) [14] proposed a novel method called *Residual Learning* and *Skip Connections* (ResNet) to overcome this issue. They incorporated forward connections between convolutional layers to backpropagate the gradient with less information loss on the way. This way, they were able to train deeper CNNs and achieved 1202 layers [14].

These recent augmentations, allowed CNNs to be used in many computer vision tasks involving visually salient objects/environments with more than adequate performance such as pedestrian detection [48] and event detection [49]. They have even been tried on time-series tasks and achieved good results and is used instead of LSTMs in some cases (LSTMs are a particular type of neural networks designed for time-series problems) [50].

Although object detection is one of the anchor tasks of computer vision, sometimes segmentation is preferred. There are two types of image segmentation; (i) semantic segmentation and (ii) instance (instance-aware) segmentation. Semantic segmentation is when

an image is segmented in a class-agnostic way where distinct instances of a class are not important and not separated. When we only want to know the content of an image and objects are not of the main goal, semantic segmentation methods are used. Contrarily, if we care about objects and instances of classes as well, then instance segmentation methods are used. Semantic segmentation is a crucial problem and a well-studied one. Research has been done on semantic segmentation ranging from image retrieval to autonomous driving [51, 52, 53]. Instance-aware segmentation can be achieved by giving classes to segmented pixels and hence, only a small overhead. Li et al. [54] proposed a CNN based E2E instance segmentation method which performs pixel level segmentation (mask) and classifies the masks simultaneously. Mask-RCNN [11] uses the same idea in Faster-RCNN method.

In addition to CNNs’ general success, machine learning methods and specifically CNNs have outstanding performance on tabular structure detection task, as it can be seen in the Subsection 2.2.2 and Subsection 2.3.2. Hence, we make use of deep learning in the feature extraction phase and benefit from the capabilities of the methods mentioned in the tabular structure detection task.

2.2 Table Detection

Many businesses and government operations need data to be able to continue processing as usual. The data is in the shape of inventory records, tax filings or product datasheets. Collected data is collected for humans to process but, we tend to skip-read when reading a long text; hence, we can observe that the most critical information is mostly given in the tables. Therefore, it is clear to see why table detection is a significant step in the document analysis problem, and many people work on this problem to find a better solution. This section provides a brief overview of the previous work done on table detection problem. Two main approaches to table detection are; (i) Rule-based approaches, (ii) Machine Learning-based approaches.

2.2.1 Rule-Based Approaches

Document analysis has always been at the centre of the research since the task is boringly easy for humans and people think that structures in the documents have seemingly trivial delimiters that separate them from the document. In very early work in 1968 [23], researches developed hardware explicitly designed for detection of text and other structures as needed. Their hardware can operate with a human operator which gives input in the

runtime for more sophisticated documents such as journal papers. It can also operate under unsupervised mode. This work only, with exclusive hardware design, shows that information extraction from documents is at utmost importance for many organizations.

Since the table detection task is critical to information extraction facilities, it has also been worked on for a long time. Early works usually benefit from the heuristics extracted by visual inspection and basic feature extraction. Wahl et al. (1982) [25] focuses on digitized documents hence works on the document image domain. They design a binary operation to run over the image to fill small gaps with black pixels. This operation is done horizontally and vertically, and then the results are smoothed. As a result, the authors divide the image into 'meaningful blocks'. To separate the discovered blocks, they employ a rule which starts at the left-hand of the image and iterates to the right. Each adjacent black pixel is appointed to the same label. Then with the features such as the height of the block segment and ratio of the block pixels to the bounding line these blocks are classified as one of many page objects.

Pyreddy and Croft (1997) [55] proposed a system called *TINTIN*, a heuristics based table detection method that works on electronic documents. Table extraction phase is developed by looking at the white spaces in the text. Whitespaces are checked against alignment as well as the number of whitespaces between the text. The system first detects columns with whitespaces. For example, to separate the columns, there should be at least three whitespaces between the text. There are a couple of parameters that are set by visual inspection. Following the extraction of table blocks, the second phase tags the different components. *TINTIN* showed that such trivial heuristics are capable of detecting tables and open to further development.

Jain and Yu (1998) [24] target the problem of converting papers to their electronic version and is one of the earlier works. This work, like many other earlier works, focuses on one specific type of documents and that is technical journal papers. The extraction process is based on the geometric layout analysis and the extraction of the connected components on binary images of the documents. Since all the papers have similar layouts, the authors were able to incorporate domain knowledge to the solution. Connected components are found with block adjacency algorithm and table, text and image regions are detected with the help of ruling lines. Apart from other similar works, they also estimate and correct the orientation of the document.

Kieninger and Dengel (1999) [56] introduced the T-Recs table recognition system which is a widely accepted and successful method [29, 30]. The work focuses on document segmentation on arbitrary documents and is based on four hierarchical parts; words, lines, blocks and the document. T-Recs is not based on line detection but rather a bottom-up

approach where blocks are detected from the words. The recognition of the tables starts with a 'word', which is a bounding box. This box then linked to the neighbouring boxes if they are close to each other to form 'blocks'. If the resulting blocks are horizontally neighbours, they are assumed to belong to the same table. As a next step, authors created rules to detect internal structures (rows and columns) from these assumed table regions. In a follow-up work, Kieninger and Strieder (1999) [33] improved the T-Recs detection model soon after publishing the first version. In the original T-Recs method, tables are detected by linking the blocks together. However, in [33], columns are detected by the vertical alignment between the blocks and then the columns are aggregated to form tables. T-Recs is an important step in achieving a robust table detection method that does not rely on non-tabular structures like lines. That is, lines are not essential for a tabular region.

Tabfinder, proposed by Cesarini et al. (2003) [57], is another method to exploit the lines surrounding the tables. Tabfinder separates from the earlier work mentioned by utilizing a modified version of XY-trees on the page images. Every node in this tree represents a part of the image that might contain a table and nodes are created when a new line is detected in the sub-image. This tree is recursively created for probable table locations. Again, as in others, tables are verified using either lines or appropriate spacing between words which might form rows or columns. Tabfinder shows us that employing additional data structures is possible to improve hand-crafted heuristics based solutions further.

Gatos et al. (2005) [58] proposed a table detector which is not dependent on any heuristics and can work on any document images. The method starts with character size estimation and line estimation based on black runs processing. Continuous black pixels are connected (black runs) and labelled to create connected components. To finalize the detection, the authors first detect the intersection of the detected lines and remove all the lines. Then, intersection points are checked against the alignment to conclude the table detection. This method heavily depends on lines in the tables (both horizontal and vertical) as well as scan quality.

Lin and Xiong (2006) [59] take a different path to detect tabular regions. Unlike other conventional works such as applying machine learning techniques or line detection with heuristics, the authors study the text in the documents that may refer to the table regions. However, they specifically focus on the table of contents region in this work. The process is simple, choose pages that may be the table of contents page and then assign a score to it. If it is higher than some threshold, this is a table of contents and entries in this table linked to the corresponding pages. Scoring of the pages and the linking of the entries are dependent on the OCR and text mining. [59] can be taken as a proof-of-concept that text mining from the documents may help improve the table detection results and was one of the first papers that did not use rules to locate the tables. The methods we propose in this

thesis can detect the table of contents page as a table and requires no further training to do so.

Detection of tables on the web (HTML) has two possible solutions; (i) look for *table* tag, (ii) just like other table detection methods, check for visual cues. Although table detection is essential for many domains and a well-studied research area, researches also focused on later stages of the process. For example, Son et al. (2008) [60] focuses on discriminating the web tables concerning their value. However, this work focuses on the detection problem. Therefore, we do not go into more detail for these kinds of research. Gatterbauer et al. (2007) [61] discuss the downsides of the detection with *table* tags and concludes HTML tags are not stable and that it is not a trivial task with the total amount of tables present are growing. In [61], the authors chose to form twenty rules to detect a table. These rules are applied to frames of web pages. The interpretation phase follows locating the tables. They again use visual rules to extract information from the tables.

TARTAR [26] is another method proposed to extract tables from arbitrary documents including PDFs and HTML, but the only testing was done on HTML tables. Unlike [61], TARTAR focuses on table detection based on HTML code represented in a Document Object Model (DOM) tree. After a table is detected within the HTML, the table is processed to surface underlying structural and functional organizations. They clean the HTML DOM tree which outputs a table where multi-column spanning columns are separated into multiple columns, and the column header is repeated. TARTAR separates the table to logical sub-units which then checked with common string patterns and neighbouring relations. Although the authors claim that TARTAR works on any document, they heavily rely on DOM tree and functional relations that are extracted.

The research is not just focused on table detection from images or HTML code; extraction from PDF files is another path that previous work chose. Liu et al. (2009) [62] makes an initial assumption that every table starts with a keyword such as; *table* or *form*. The detection process starts by detecting sparse text lines. The authors say that sparse lines are a candidate for a table row. After the classification of sparse lines using the relations between text sequences, table caption and these lines are combined and tested with a hand-crafted heuristic to locate the tables. Although the results in [62] are promising, they only focused on detection from PDF files with two-column documents and table captions which narrows down the scope.

In [27], Shafait and Smith (2010) perform the detection by first detecting the column layout of the document. The table detection is based on detecting column partitions, and these partitions are detected with the connected components and tab-stops between them. The algorithm relies on hand-crafted rules to identify these regions and would have a hard

time to detect tables where there are unusual cases. Fang et al. (2011) [32] proposed a table detection algorithm from PDF documents. Detection again depends on white spaces and ruling lines from the document. Preliminary white space processing discriminates columns of the page and the tables. Afterwards, the method finishes the process with the table detection that heavily relies on hand-crafted rules crafted by visual inspection and insights.

In another work, Dey et al. (2016) [63] use a consensus-based algorithm to extract tables in document images. As in [27, 58, 64, 24], the main building block is the connected components and is reliant to the hand-crafted heuristics. Then, from these components, features that represent colour and stroke are extracted. Consensus-based clustering then applied on every pair of features to statistically generate similarity and predict connections between the connected components. As the last step, extracted graphs are labelled as tables. Tran et al. (2016) [31] is another method stems from the connected component extraction. The authors propose a morphology-based method [22] to extract connected components which are then merged to form Region of Interests (RoIs) by using hand-crafted rules and thresholds. After the RoI is created, the white gap between the connected components is considered to decide if they are separate or not. Then, the extracted text blocks are checked against vertical alignment to form columns. These columns form the tables. Every step of the process is heavily engineered and dependent on the heuristics which means when the document context is changed, the model may spawn unreliable tables.

2.2.2 Machine Learning-Based Approaches

Wang et al. (2001) [65] proposes, not a detection method but zone classification solution by using customized decision trees to classify zones into graphical regions such as text, image and table. Classification uses vertical and horizontal white spaces, as well as inherent rows and columns which are given as input. This work is not as relevant as others, but this is one of the earlier applications of machine learning algorithms to the document analysis problem. Wang and Hu (2004) [37] in a later work focused on the classification of web tables. Tables are classified as genuine or non-genuine with decision trees and SVMs. Features to train these methods include; statistical information about rows, columns and cells. One other feature, which we found interesting is the average content type consistency over rows and columns. [37] reveals promising results that attest that machine learning techniques can master how a table is formed accurately.

Shetty et al. (2007) [66] try the CRF algorithm trained to maximize pseudo-likelihood of the documents. First, the document images are divided into small patches which cover

connected components and are determined by the foreground and background pixels. Then, characteristics of these patches and the neighbouring relations are used to extract features. CRF is used as the last stage to label these pages as hand-writing, machine or noise. [66] is not proposing a machine learning approach but we think this work is a good showcase where CRFs potential on document analysis has been shown.

Around the same time, Ng et al. (2007) [67] tried C4.5 decision tree algorithm and a simple feed-forward neural network with backpropagation as classifiers. As in previously introduced machine learning-based works, Ng et al. benefits from these methods at the last step and domain expertise is still needed. Features are extracted from horizontal lines since they might form the boundaries of the tables, rows and columns. The models are trained to detect boundaries of a table and rows and columns. Therefore, it is a three-step algorithm where first the boundaries are detected, and the vertical and horizontal lines inside this boundary are classified for possible rows and columns. When the order of processing is considered, this work has a similar idea as ours where table structure detection is preceded by table detection.

In 2009, Silva proposed a table detection method [68] based on HMM and lines in the document. The main idea is to group lines to detect table regions with data-driven methods. They have also tested Decision Trees and different structures such as charts or titles. HMMs are used to classify lines to a *table line* or *non-table line*. They propose multiple models wherein some of them, charts, titles or texts are also classified using HMM. Since HMMs model the neighbourhood relationships when classifying a node, the addition of other classes was made to increase the accuracy of the HMM. The findings supported that the additions increased the completeness and the purity of the table detections.

Kasar et al. (2013) [38] proposed another method with hand-crafted feature extraction to be used to infer the tables with a machine learning algorithm. The proposed method is not dependent on the text in tables or documents but is dependent on the ruling lines. First, horizontal and vertical lines are detected, and their intersections are located which then used to obtain twenty-six features. Features include the area of the encapsulated convex region, the ratio of the length of a line with max-length line detected and so on. [38] is one of the first occurrences of SVM in table detection domain, and it confirmed that machine learning methods promise a future on this problem and require more attention from the researchers.

Fan and Kim (2015) [69] uses a method called *Distant Supervision* to form their dataset. *Distant Supervision* is a method where a simpler heuristics-based unsupervised classifier is used to label the unlabelled line data. Since the classifier used to label the data would not be 100% accurate, there will be errors, but since machine learning methods are robust to

noisy data [70], the ensemble learning used by the authors would learn to detect tables. The ensemble method includes SVM, Logistic Regression and Naive Bayes algorithms. Along with weakly labelled features, textual features are extracted to support the prediction further. The authors employ prior knowledge in the feature extraction phase as well. For example, they extract a feature on the basis of there will be more nouns in the table than adjectives and adverbs. Another machine learning-based information extraction method from the PDF documents is *CERMINE*, proposed by Tkaczyk et al. (2015) [71]. CERMINE analyses the PDF file to label the raw PDF to lines, headers, zones, reading order and similar hierarchical structures. CERMINE is a multi-step algorithm where the first step is to examine the layout to determine zones. Layout examination phase benefits from the traits like line-spacing, text orientation and neighbouring relations. Following the discovery of the zones, they are classified to be one of the classes: metadata, content, references or others. The classifier is a regular SVM trained with geometrical, lexical, sequential and heuristical features. CERMINE proves that machine learning algorithms are able to operate well on a wide range of tasks on document domain.

After the deep learning breakthrough in the computer vision field [28], the researches that work on document analysis from the document images began to adopt deep learning to their solutions. Hao et al. (2016) [72] proposes a table detection method that utilizes heuristics-based proposals and CNNs. First, certain areas are detected and marked as possible table areas. Locating these areas are done with heuristics-based search with the horizontal and vertical lines detected from the document image. [72] uses CNN to approve the located table with the heuristics. Consequently, deep learning is just used as a judge at the end of a traditional table detection method. An intriguing property of this system is that some information extracted from the original PDF is embedded either to the input or the output of the CNN. The extracted information crops include coordinates of lines or words. Authors show that, when the information is added to the CNN input, there can be a significant increase in the precision on ICDAR 2013 [73] dataset.

Schreiber et al. (2017) [10] proposes an E2E table detection and table structure detection method. Previously these methods included hand-crafted rules in at least one phase of the process. However, in DeepDeSRT [10], the detection process is E2E that means, given the document image, the decision is made without any human intervention. To train a deep learning model E2E requires enormous amounts of training data which in tabular structure detection domain, is hard to come by. Therefore, the authors choose to apply transfer learning. Transfer learning is to use a previously trained deep neural network on a large database and use a smaller dataset to fine-tune the existent network and transfer the learnt knowledge to the new domain. The proposed table detection method uses Faster-RCNN [12] and outperforms many others. The test results on ICDAR 2013 [73] show that

advanced CNN architectures are promising in this field and need further work and data to train on.

Yang et al. (2017) [74] propose an E2E network design for extracting the semantic structure from document images. The proposed network consists of an encoder and two decoders. One of the decoders try to re-build the input with deconvolutions to make sure the network does not *forget* information from the input image. The other decoder is to segment the pixels to corresponding classes. Predicted classes include; lists, paragraphs, tables and figures. Along with skip-connections, text embeddings from the document is also fed to the network in the later stages of the flow. A significant contribution and the lesson to take from [74] is, when the text from the document is extracted with OCR engines, the segmentation becomes slightly better. However, if the real text extracted from the PDF is given to the network instead of the OCR, one can see a significant increase in detection performance when compared with the network with no text connection is present.

Pursuing the great success of the deep learning methods, many people continued examining new ways with these methods. Gilani et al. (2017) [75] proposed another E2E table detection method based on document images with Faster-RCNN [12] architecture. Unlike DeepDeSRT[10], they transform the input image to make it look more like natural images (i.e. real-life, RGB images). Instead of RGB, they have applied three distance metrics for each dimension of the image. The distance metrics represents the distance between text regions and white spaces. This transformation means only the architectural remains will be present in the image but not the figures, lines or text. [75] shows that transforming the image may improve recall and precision.

Li et al. (2018) [76] propose a system to recognize multiple page objects including tables and formulas. The input to the system is the binarized document image. After connected components are found with the contour tracing method, the page image is segmented into columns and then lines. Machine learning algorithm CRF is applied at this level to classify each line region into one of the four classes. Apart from the classification, the same model is also used to determine the link between the lines. As a post processing, heuristics are applied to fix possible miss-classifications. In the end, a verification model is also employed. The authors claim that since the CRF takes a fixed-size input, some line regions may be misclassified due to the information loss caused by the resizing. Therefore, a final deep learning method, CNN, is used to verify the previous classification. [76] achieves a promising F1 value of 96% on ICDAR 2017 POD Dataset [77].

Oliviera et al. (2018) [78] propose a method called *dhSegment* with the historical documents are a priority. *dhSegment* is a pixel-wise detector coupled with task dependent enhancements. The first step is to infer probable segmentation fragments that might

belong to some classes via the dhSegment method. dhSegment is a fully-convolutional neural network which is a modified version of pre-trained ResNet-50 [14] and includes skip connections as well. Created segments are not complete rectangles. Therefore the authors proceed with the post-processing methods. Post-processing in [78] focuses on completing the segmentation and eliminate the fragments. Methods to accomplish that goal are; thresholding between the class predictions, applying morphological closing to fill in the small gaps, filter out small remainders after the morphological closing operation, and finally, the region is translated to bounding box coordinates. [78] proves that a simple fully convolutional neural network that is pre-trained on a natural image dataset can be employed to segment document images into meaningful parts successfully.

After deep learning methods attracted enormous attention by the table detection community, Kerwat et al. (2018) [36] decided to compare several modern deep learning-based CNN architectures using public ICDAR 2013 [73] test set. Compared methods are; Faster-RCNN, Single Shot Detector (SSD), and You Only Look Once (YOLO) [12, 79, 80]. SSD is a single-stage object detector with the idea of anchors. Anchors are rectangles with pre-determined ratios placed at each grid location to predict bounding boxes. SSD [79] predicts bounding boxes while performing bottom-up processing. That leads to predictions performed at multiple scales; hence, many predictions are produced on a single image in one pass. Whereas YOLO [80] concentrates more on the speed and performs only one detection in a single pass. The comparison of these methods shows that Faster-RCNN has a significant lead in the correctness of the table and figure predictions [36].

Siddiqui et al. (2018) [81] proposes a Faster-RCNN [12] based model with deformable ResNet-101 [14] as the backbone feature extractor. In the conventional CNNs, features are extracted from a rectangular area and towards the upper layers, these rectangles gets bigger. However in deformable CNNs, feature extraction is not straightforward but, the receptive field of the neurons in the network may be adjusted because it is dynamic and learnable. The proposed method [81] shows the near-perfect result on ICDAR 2013 dataset and exceptional performance on ICDAR 2017 dataset [73, 77]. Although the excellent performance on ICDAR datasets, the model struggles on the UNLV dataset which includes scanned documents and presents more difficulty. The experiments show that adding deformable layers to ResNet-101 increases the F1 score by 4% in ICDAR 2017 but has no effect on ICDAR 2013. This might mean that deformable layers overfitted to the table layouts and therefore when the style/purpose of the documents are changed, performance degrades.

Kavasidis et al. (2018) [82] proposes a deep CNN to detect tables and charts in digital document images. The detection network is based on VGG-16 [19] but to further tackle the problem of the small/thin objects (lines, white spaces, rows and columns), filter sizes

are modified. Following the convolutional layers, dilational convolution block takes the resulting feature map (75x75) and outputs a new feature map with the same sizes. Dilational convolutions extend the receptive field of the filters/kernels. It is reasonably similar to the deformable CNN architecture with a difference; dilation size is not learnable. Binary classifiers are connected to the output of the convolutional layers to impose class related losses and benefit from it in the back-propagation. As a final post-processing method to improve the noisy segmentation, the authors utilize a CRF model. It can be seen from [82] that, addition of the binary classifiers to the CNN improved the precision by a lot. When combined with CRFs too, it further improved the test set results on ICDAR 2013 [73].

Arif and Shafait (2018) [83] studies the previous deep learning-based table detection methods and proposes improvements on top of them. The improvements stem from the belief that tables include more numerical data than other parts of the document. The deep learning method under study is Faster-RCNN [12] and it is tested on the public UNLV dataset¹. Pre-processing (improvements) steps can be divided into two: (i) colorization where the text is coloured to green and numbers converted to red, (ii) image transformation based on the distance on the blue pixels, similar to [75, 83]. The trained network is Faster-RCNN [12] with ResNet-101 [14] backbone feature extractor. They, similar to us, labelled their own training set and tested on the UNLV dataset. The results show around 90% F1 score. Hence, we can conclude that either Faster-RCNN has limits and cannot go higher, or the transformations are not as effective as one would hope.

In recent years, with the advancement and popularity of deep learning, as we have seen in [82, 7, 78] general focus of the research was on applying the state-of-the-art object detectors to the table detection domain. However, there are still work under progress for more focused information extraction. Milosevic et al. (2019) [84] focused on a specific task of information extraction from biomedical literature. However, in [84], authors focus on only one dataset where the tables are already annotated.

In our previous work [7], we have compared two state-of-the-art object detectors. Namely, Faster-RCNN [12] and RetinaNet [13]. In this work, we have trained the Faster-RCNN model with the popular and fast VGG-16 [19] backbone network and the RetinaNet with ResNet-50 [14] and ResNeXt-101 [85] backbone networks. ResNeXt is an augmented version of the regular ResNet networks where the blocks are repeated for better information storage. More detail about the methodology and the networks will be given in Chapter 3. In another previous work, we have implemented Mask-RCNN [11] model with the ResNet-50 [14] feature extractor and we have achieved the state-of-the-art performance for the table detection task on multiple public datasets (ICDAR 2013, ICDAR 2017, UNLV) [9].

¹<http://www.isri.unlv.edu/ISRI/OCRtk> (currently not available)

In [9], we tried several augmentations which we found to be very useful. The first proposal we did was to implement a judge-like algorithm and by benefiting from both the table and the structure detection models, eliminate false-positives. Then, we proposed the line-agreement model where we have added a line and intersection detection algorithm followed by fitting a tight bounding box to these intersecting lines. That bounding box then used to further improve table detection. Even though the line-agreement model does not affect recall and precision, due to the nature of our two-step approach, we observed an increase in end-to-end precision [9] when judge and agreement models are used together.

2.3 Table Structure Detection

We mentioned the importance of table detection and extraction; however, employing only the table detection cannot eliminate human operators which inscribes the information in the tables to the computer. Therefore, to have a fully autonomous information extraction from the documents, individual cells should be extracted from the tables, but these cells can inherently be inferred from the intersections of the columns and rows, thus require no further processing. As we will explain in the upcoming sections, the conventional way of detecting table structures is doing it E2E. That is, the columns and rows are detected directly from the input file. Whereas, we propose a different two-step approach where structures are detected from the tables that are previously detected by another method. The details of our E2E system will be explained in Chapter 6

2.3.1 Rule-Based Approaches

The rule-based table detections as we have seen in the Section 2.2 usually follow the bottom-up approach where the first step is to detect smaller structures such as words or columns or rows [56, 38, 27]. Locating the columns and rows can be accomplished with alignment or white space or ruling line heuristics whereas tables are more complex structures and there are no alignment restrictions for the content in the tables, in general. In this section, we remind those research that uses bottom-up processing or a different approach and what heuristics they have used to detect table structures.

Pyreddy and Croft (1997) [55] proposed a system called TINTIN to extract structured information from the documents. TINTIN’s first step is to extract the tables which fulfilled with white space heuristics. After the table is extracted, the component extraction phase starts. Heuristics are based on syntactic features of every table line and not semantic. That

is, components are detected by how things look not what they possibly mean. Heuristics range from the alignment of the words to the gap between them. Results have shown that the visual of the table itself is enough to recognize the structure of the table.

Kieninger and Dengel proposed the T-Recs system (1999) [56] to detect tables. T-Recs looks at the word bounding boxes and their neighbouring word bounding boxes to cluster them with the white spaces in mind. As a result, columns are supposed to be structured together and multi-column areas are said to be possible table areas. In [64], the author is presenting a structure segmentation model that operates on table images domain. In this work, tables are given to the system as input, and from the table images, a table grid structure is extracted (rows and columns). A table grid is a layout that the table has been built upon and assumed to be like a spreadsheet. The column detection is based on separator lines or the inter-column gaps followed with the k-means algorithm to further justify the choice. The row detection on the other hand, as the authors stated, an extra detailed task [64]. To detect rows, [64] assumes several possible row divisions and then heuristics and hand-crafted rules identify the rows. If there are no lines detected to separate columns and rows, connected components checked against the alignment with hand-crafted rules embedded in every phase of the process.

TARTAR [26] is a table extraction method based on DOM trees extracted from the HTML code. TARTAR builds the physical structure of the documents by capturing the relative positions of the cells. These cells are then examined by their functionality which is the purpose of the cells and their semantic meaning which is the meaning of the text inside the cells and their relationship between their orders. In short, TARTAR extracts structures by their physical, functional and semantic characteristics unlike [55]. Shafait and Smith (2010) [27] suggests a method that works on multi-column pages. How they proposed it is they detect columns by predetermined heuristics and connected components. After the columns are aggressively detected by tab-stops, white spaces and vertical alignment, they are combined to form table regions. Although the rows are not considered, [27] shows that such simple heuristics can be formed to discover structures on a minimal level. Tran et al. (2016) [31] also presented a solution that performs bottom-up table prediction with connected components. The mathematical morphology-based [22] closing operation is used to find connected components to be used to form RoIs with hand-crafted heuristics. RoIs are possible table locations and created with checking the existence of possible columns formed by connected components. White space and alignment of the connected components are checked to form the columns.

2.3.2 Machine Learning-Based Approaches

Ng et al. (2007) [67] divides the tabular structure detection problem into three steps that follow each other. First, a decision tree is trained to detect the tables by the classification of the horizontal lines. Then, every vertical line inside the table region is classified into one of the five classes to finalize column regions. The last step is to look at horizontal lines inside the table region and classify the lines to one of the two classes. These classes determine if the line is the start of a cell (row or column), the end or is the line inside the cell. Either a decision tree or a feed-forward neural network is used to come up with a classification based on the features are extracted from the lines.

Fang et al. (2012) [86] proposes a method based on SVM and Random Forest classifiers that identify table headers from PDF documents. From the table, rows and columns are detected with features such as; the number of cells, percentage of italic characters, percentage of numeric characters and average overlap portion between cells. Preliminary results show that Random Forest classifier outperforms other methods and columns are detected better than the rows [86]. After deep learning dominated computer vision field, table detection methods started to switch to use deep learning instead of the old-fashioned heuristics, as we have seen in Subsection 2.2.2. One successful example is the method proposed by Schreiber et al. (2017) [10]. In [10], both the table detection and the table structure detection is solved in E2E fashion. Authors used Faster-RCNN [12] for table detection whereas, for the table structure detection, an FCN was utilized to perform pixel-level segmentation for rows and columns.

In our previous work [8], we have focused on the second stage of our two-stage method. The second stage is when we have the table crop extracted, and we need to detect columns and rows from the table. We have trained several deep learning methods where every one of them had different advantages and disadvantages. We have concluded that FCN and FPN architectures help massively and single-stage detectors have a harder time detecting the structure mainly since they are tiny [8]. We have also tested connected components based structure detection method and concluded that even though it includes no learning, the column detections were more accurate than the row detections due to the nature of these classes' shape [8]. We have also observed that increasing the performance of the models is possible by stretching the images vertically to 200% of their original size. In another previous work, we have proposed two E2E structure detection systems where the input was the whole document image [9]. We have compared two E2E structure extractors with one model trained on detection rows and columns and where another model is trained to detect tables, rows and columns. We concluded that including the table into the detections, improves the overall performance of the deep learning model [9]. In [9] we have also

proposed a two-stage tabular structure detector, and the details will be given in Chapter 6.

Table 2.1: Overview of table detection and table structure detection research. TD: Table Detection, CD: Column Detection and RD: Row Detection

Image						Born-Digital					
ML and DL			Rule-Based			ML and DL			Rule-Based		
TD	CD	RD	TD	CD	RD	TD	CD	RD	TD	CD	RD
[65, 66]	[8, 9]		[25, 57]	[31, 27, 64]	[64]	[37, 67]	[67, 86]		[55, 24]	[55]	[55]
[38, 10]	[10]		[58, 63]			[68, 69]			[56, 33]	[56, 32]	
[74, 75]			[31, 64, 27]			[71, 72, 86]			[59, 60]		
[76, 78]									[61, 26, 32]		
[36, 81]											
[82, 83]											
[84][7, 9]											

Chapter 3

Methodologies

Tables can be found in many document types like technological reports, electronic component datasheet, medical documents and more. Since many of these organizations are driven by data, it became crucial to extract the structured data from the documents. As we show in Chapter 2, starting from the 1980s and notably as years passed, many researchers proposed methods to solve this issue. The most recent methods are based on deep learning, and here in this chapter, we define and detail the methods that we study to propose our table detection, table structure detection and finally, E2E detection system.

We propose a two-stage E2E tabular structure detection method based on deep learning-based object detection networks. The first stage is to detect and crop the table regions. In the first stage, we compare three deep learning networks; (i) Faster-RCNN [12], (ii) Mask-RCNN [11] and (iii) RetinaNet [13]. We explain the architectures of these methods and the reasoning behind these methods as well as discussing their pros and cons. The second stage in our proposal is to detect rows and columns from these table crops. In the second stage, our methodology does not only include deep learning methods, but we use the mathematical morphology [22] method to test the idea of structure detection based on connected component extraction and the heuristic-based checks as many of the related works did. We also explain and detail the deep learning methods used in structure detection which are; (i) Faster-RCNN [12], (ii) Mask-RCNN [11], (iii) RetinaNet [13] and (iv) morphology based method. We start by defining the deep learning based methodologies followed by the morphology method that we used.

3.1 Deep Learning-Based Methods

This section starts by detailing the backbone feature extractors to be used in the deep learning object detectors, followed by in depth analysis of the deep learning methods under study and discusses their differences and advantages and disadvantages.

3.1.1 Base Networks

Computer vision enjoyed the benevolent success of the CNNs and deep learning in recent years. At first, regular CNNs with Dropout [87] and BatchNorm [18] were enough to make a breakthrough on computer vision competition datasets such as ImageNet Large Scale Visual Recognition Competition (ILSVRC) [45] and so on. Therefore, many research has devoted to these areas, and the results are fruitful. First successful deep learning-based CNN methods that had an impact are; AlexNet [28], GoogLeNet/Inception Module [46], VGG [19] and ResNet [14] of which previously detailed in Subsubsection 2.1.2. All these different networks showed us the advantages of seeking different architectures. For example, GoogLeNet gave insight into the advantages of stacking convolutions side-by-side and then concatenating their outputs back and repeating it for layers after layers. VGG showed us that even the networks with fewer layers and parameters are capable of achieving excellent results although it only has 16 or 19 layers, and it is still used extensively because of its size and speed. ResNet, on the other hand, introduced *Skip Connections* and went deeper, up to more than 1200 layers. Their findings show that going deeper is not always helpful, and we have to look for better architectures to improve computer vision algorithms further.

Hidden layers of these networks are usually equipped with Rectified Linear units (ReLU) [88] activation function. ReLU is a non-linearity function used commonly in deep learning models [19, 14, 28]. The function is straightforward; returns 0 if the input to the function is negative and for positive values, it returns the value without change. Hence, the formula is;

$$ReLU(x) = \max(0, x) \quad (3.1)$$

Activation functions are introduced to deep learning models because the operations are linear, and the resulting model would be similar to the linear regression model. Therefore, to add non-linearity to the networks and to be able to represent complex non-linear arbitrary mappings between input and output, non-linearity functions are added. Another essential property of the activation functions is that they should be differentiable to be able to optimize the model with the backpropagation. When one looks at the Eq. (3.1), we can

see that the ReLu function is not differentiable at $x = 0$. However, it is still used because gradient descent cannot arrive at a local minimum, easily. Also, if it arrives, implementations of the functions choose a derivative from one of the sides of the function. Another point that should be cleared is that activation functions should be non-linear, whereas ReLu is two linear units connected. Since models have many layers that are connected, each layer changes the slope of the linear function at different spots and in the end, we have a function with many different slopes. These back-to-back connections with ReLu can be used to approximate smooth (non-linear) functions [89].

Historical activation functions such as *tanh* and *sigmoid* have S-shaped curves that help them to retain the small changes in the input and change the output of the function. However, when these activation functions are stacked, except for the minimal area (the middle part of the S-shape), the derivative is close to zero. These flat derivatives make it harder to update the weights of the layers and create the *vanishing gradient problem*. When ReLu is used with a reasonably sized batch, some nodes will have non-zero activations, which would increase the average derivative. Although it is common to use ReLu, activations may turn out to be zero layers after layers which would result in the problem of *dead neurons*. A dead neuron is when a neuron is never activated (activation output > 0) and to solve this issue a modification to the ReLu is proposed; Leaky-ReLu [90]. Leaky-ReLu can be described as;

$$LReLU(x) = \max(\alpha x, x) \quad (3.2)$$

Where the α value is less than 1.0. When the input is negative, a slight activation is preserved in order to keep the weight updates going and neurons alive.

Although pure CNN networks such as Resnet and VGG outperforms all previous methods on the image recognition/classification task, they have stuck in a plateau and could not improve further. To escape from this plateau, researches have proposed advanced architectures. In a convolutional network, every layer encodes distinct features that represent the input in some abstraction level, and as shown by [91], stacked convolutional layers can encode hierarchical structures that represent the input. For example, first layers can encode edges, lines and corners, and as we go up the layers, encoded information converts to upper-level abstractions such as doors or wheels of a car. Ignoring the successes of the previous methods is not a sound idea, and therefore, they are still used, as feature extractors and usually called *backbone networks*. Although these CNN models are originally proposed to classify images, current object detectors only re-use them to extract features from the internal layers. New generation object detectors are divided into two classes; (i) single-stage detectors and (ii) two-stage detectors. Single-stage detectors use the features extracted by the backbone network and place *anchors* to denote possible object locations.

Anchors are pre-determined shapes (usually by k-means) that represent the most common shapes of instances in the dataset. Then, the image is divided into grids and anchors are placed onto these grids' centers. These anchors are then classified to detect objects. Two-stage detectors, on the other hand, have an additional neural network (or a search algorithm in older methods) to propose possible object locations. Since two-stage detectors have an additional overhead, they are usually slower but more accurate.

In this subsection, we give more detail on the feature extractors that we benefit from. Namely, VGG-16, ResNet-50 and ResNeXt-101. These feature extractor backbone networks are used in our deep learning based methods to extract features.

VGG-16

Simonyan and Zisserman (2015) [19] observed the rise of the convolutional neural networks in the computer vision fields and developed their image recognition model. Their models, named VGG, has a fixed-size RGB image (224x224x3) input. The model includes a pre-processing, and that is calculating the mean over the entire training set and subtracting it from the images. The convolution filters have sizes 3x3 and stride of 1, which makes it suitable for our task because it will not skip over large chunks of information. Although filters are small, max-pooling is applied with 2x2 size and a stride of 2 which prone to information loss. In the original VGG architecture, three fully connected layers stacked on top of the convolutional layers. These layers are used to classify images. The first two have 4096 channels whereas the last one has 1000 channels because it was initially developed for 1000-way ILSVRC classification.

In VGG models, hidden layers are equipped with ReLu non-linearity, and the fully connected layers have dropout regularization [87]. Dropout is a regularization method that used to avoid overfitting in large networks, and it approximates training architecturally different neural networks in parallel. This is accomplished by randomly dropping a number of nodes in a layer. Therefore, it is similar to training a different neural network in every update. This modification to the layers makes the model robust to changes in the input by giving every node responsibility that eliminates the possibility of a node having high importance for the output [87]. Dropout randomly subsamples a layer which in turn reduces the capacity of the network and in the case of the network is being more complex than the problem required, the network size is thinned, which prevents overfitting. To understand the dropout better, consider a neural network with L hidden layers and let W^l and b^l denote the weights and the biases of a hidden layer $l \in \{1...L\}$. z^l indicates the input to the layer l and y^l denotes the output of the hidden layer l . After this formulation,

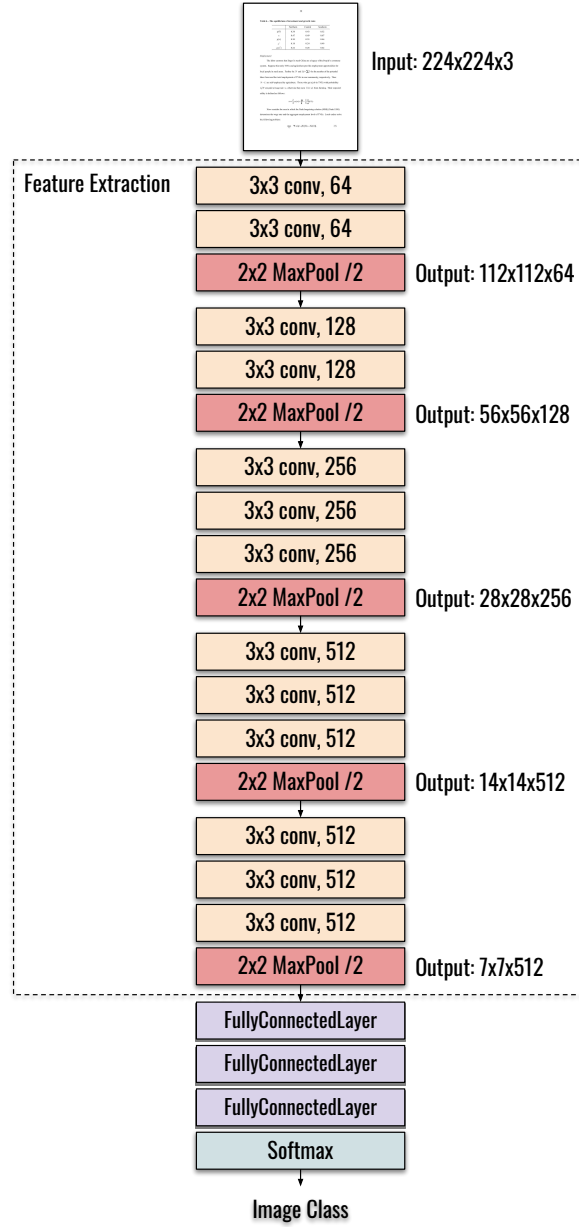


Figure 3.1: VGG-16 architecture used in the thesis

a feed-forward calculations of a neural network can be described as (i stands for nodes in hidden layer l);

$$z_i^{l+1} = w_i^{l+1}y^l + b_i^{l+1}, \quad (3.3)$$

$$y_i^{l+1} = f(z_i^{l+1}), \quad (3.4)$$

before the dropout where f represents an activation function. After the addition of the dropout regularisation, same operation becomes;

$$r_j^l = \text{Bernoulli}(p), \quad (3.5)$$

$$\hat{y}^l = r^l * y^l, \quad (3.6)$$

$$z_i^{l+1} = w_i^{l+1}\hat{y}^l + b_i^{l+1}, \quad (3.7)$$

$$y_i^{l+1} = f(z_i^{l+1}), \quad (3.8)$$

Where r^l is a vector of Bernoulli random variables with every entry having the probability p of having 1, which is taken as a parameter to the training. $*$ denotes the element-wise product between the vectors. The output of r^l decides on which nodes to 'close' by multiplying the vector r^l with the output of that layer and this operation is applied at each layer. This is equivalent to training a different sub-network in a larger network space. At test time, every node in the network is enabled and is used without dropout.

After giving all the details, the VGG network we are using in our models can be found in Figure 3.1. We are using VGG-16, which originally had 13 convolution layers and 3 fully connected layers. However, for the feature extraction purposes, we do not need the fully connected layers, and therefore, they are chopped off. The feature extraction block is denoted as 'Feature Extraction' in the Figure 3.1. After removing the fully connected layers (which takes $\sim 120\text{M}$ parameters of the total of $\sim 138\text{M}$), the network size is small and faster than its larger counterparts.

ResNet

Upon the increase in models and various architectures published by the researchers, many problems that have never been seen arose. One of them was vanishing gradient problem or the degradation in the accuracy when the depth of the network increased. He et al. (2015) [14] focuses on the degradation problem and proposed the residual learning method. They achieve this by shortcut connections. Shortcut connections are the connections that allow the feed-forward and back paths to skip over a couple of layers, where the outputs of the previous layers are added to the outputs of the later layers. To better

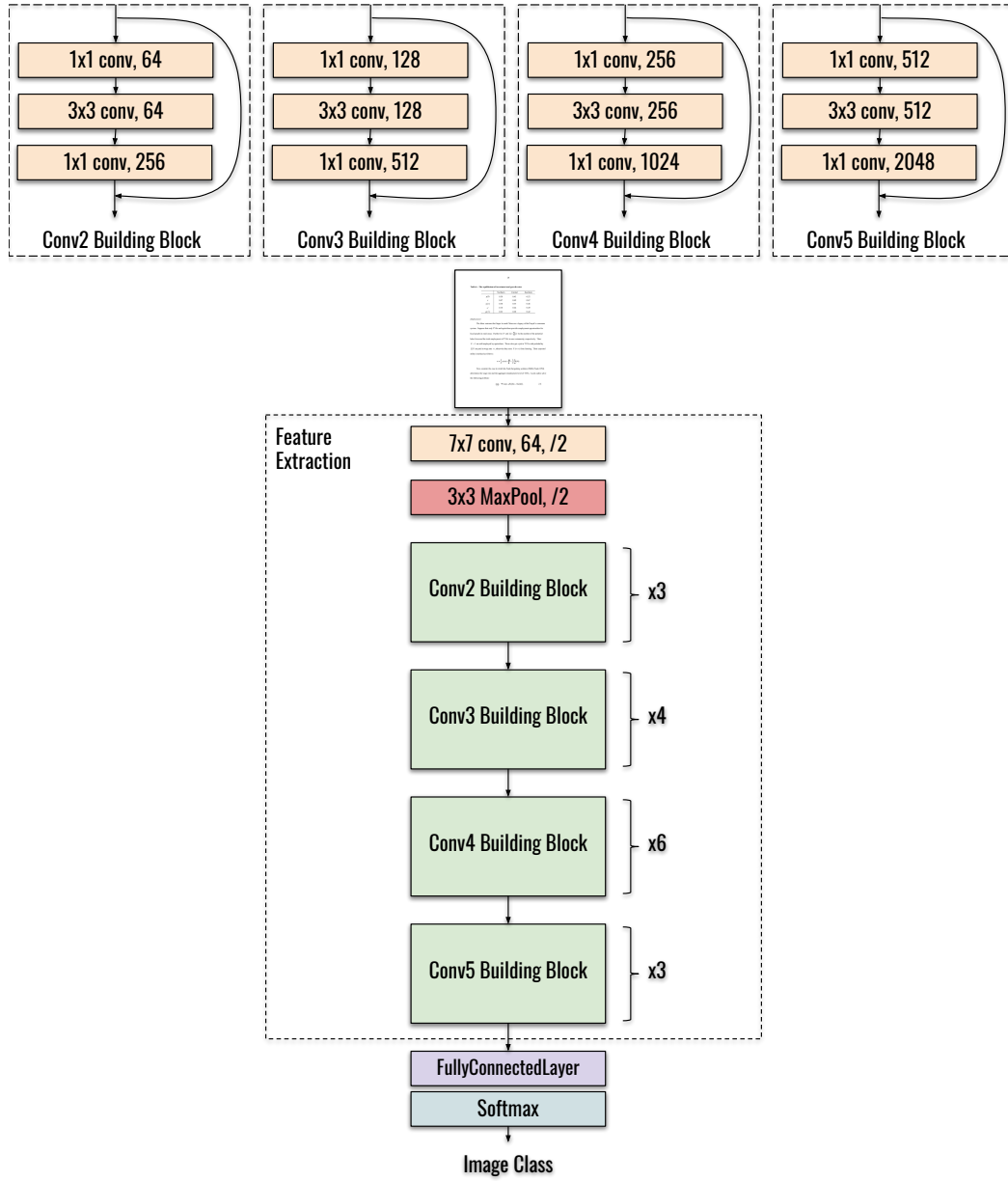


Figure 3.2: ResNet-50 architecture used in the thesis.

understand the problem ResNet tries to solve, look at this example. Assume we have a shallow network which outputs $f(x)$ when the input is x . When we add extra layers to this neural network, we hope to have better accuracy than $f(x)$. However, as observed in [14], this is not always the case, and as it turns out, deeper networks might have lower accuracy. Therefore, forcing the network to explicitly learn an identity mapping by learning the residual of input and output is the way ResNet chose. Assume the input to the network (or a subnetwork) is x and the **true** output is $H(x)$. The residual of this subnetwork is then defined as;

$$f(x) = H(x) - x \quad (3.9)$$

Since we are interested in finding the true output, the equation is reformed as;

$$H(x) = f(x) + x \quad (3.10)$$

This difference makes the distinction between residual learning and traditional neural networks. Former networks tried to learn $H(x)$ directly, whereas residual learning is interested in learning the residual. Consequently, the network can learn to make $f(x) = 0$ in the Eq. (3.10) and has the ability to 'skip' some subnetworks. This property can also be seen in the backpropagation phase where the network can choose to ignore the gradients in some subnetworks and send it back without modifying. The nature of CNNs with hierarchical feature encoding promised improvements with deeper networks. However, this was not possible before. He et al. [14] showed that the deeper networks are possible with residual learning and shortcut connections and confirmed that performance increases are possible with deeper networks. They also observed a relative improvement of 28% over VGG16 network with their ResNet-101 feature extractor when used in the Faster-RCNN [12] object detector.

In our experiments, we use the 50-layer version of the ResNet architecture, namely ResNet-50. The overall architecture can be seen in Figure 3.2. $f(x)$ mentioned in the Eq. (3.10) represents the outputs of the subnetworks i.e. building blocks. Each convolutional blocks are connected with strides. Which results in the following strides to the input image for each of the convolution blocks; 4,8,16,32 for conv2, conv3, conv4 and conv5 respectively. As in VGG-16 network, we only use the convolution part as feature extraction network, as it is denoted in Figure 3.2. We use a pre-trained ResNet-50 that is trained on ImageNet [92]. Then chop-off the last fully-connected layers (this model is denoted as ResNet-50) or we chop-off fully-connected layers as well as conv5 block, and we name that model as ResNet-50-C4. Removing conv5 block reduces the network size, which makes it easier to train when there is little data to train on. Also, when the stride of 32 with respect to the input image is considered, conv5 loses low-level semantics of the input image. To overcome

these issues, architectures like FPN are proposed as we explain the details in depth in the upcoming subsections.

ResNeXt

The ResNeXt model is a modification to the original ResNet model. Xie et al. (2017) [85] proposed a small change in the main building blocks of the ResNet and introduced a new hyperparameter called *cardinality*. The change in the architecture reminds the Inception module [46] with small differences. For example, in ResNeXt, the outputs from different branches are merged by adding them together wherein [46] they are concatenated depth-wise, and in Inception module, different branches of a block have different filter sizes, but in ResNeXt they have the same topology. The new parameter cardinality is to decide how many parallel blocks run in a single block. The authors conducted experiments that showed us increasing the cardinality improves model performance.

In our experiments, we used ResNeXt-101 pre-trained on ImageNet [92] where the overall architecture can be seen in Figure 3.3. To turn this network into a feature extractor, we cut-off the fully connected layers. We choose the cardinality value 32 as it is the default value and use ResNeXt-101 as the most complicated feature extractor in our table detection experiments.

3.1.2 R-CNN Family

Object detection is a lot more complex problem than image classification. This complexity arises from two problems; (i) object’s location has to be found, and (ii) the found region should be classified. Image classification was the more straightforward method to develop the methods on. A single image is classified into one of the given classes. However, in real life, when observing our environment, we also see objects and infer the context from the objects. Visual detection tasks were carried out with the use of computer vision algorithms such as Histogram of Oriented Gradients (HOG) [93] and Scale-Invariant Feature Transform (SIFT) [94]. SIFT and HOG are algorithms based on histograms of features and is similar to that of V1 layer (where the input from the human retina is connected in the brain) of the human visual recognition system, the first sensory areas in the visual path of the primates [20, 95]. However, when we observe the human visual recognition system, the visual system is divided into distinct visual areas where each one of these areas are connected in a feed-forward manner and represents a structure in a hierarchical way [95]. PASCAL VOC [96] is a popular and challenging object detection dataset, and the

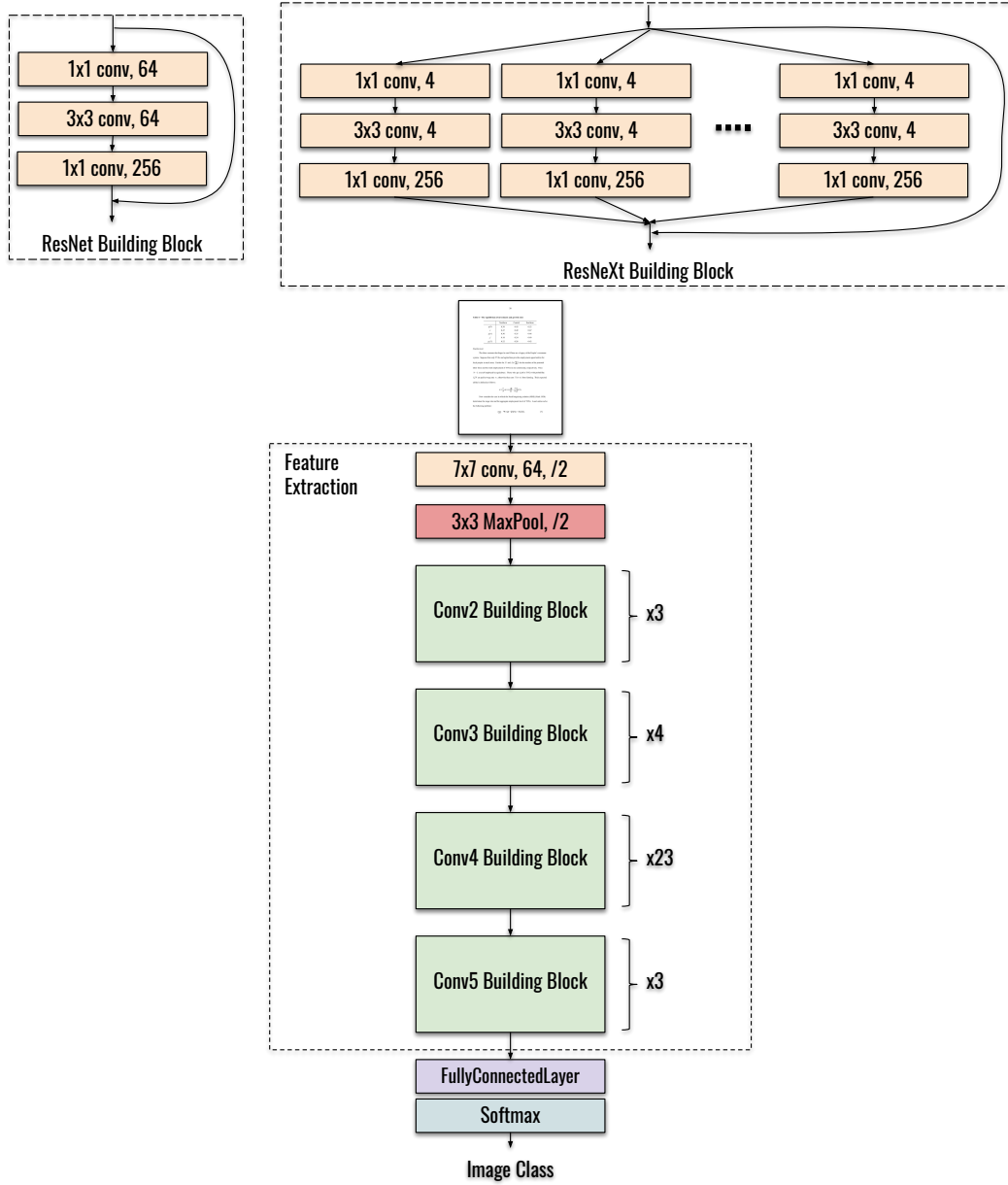


Figure 3.3: ResNeXt-101 architecture used in the thesis.

methods are usually compared on this medium. Although SIFT and HOG were dominating the PASCAL VOC competition, when one looks at the growth of the precision over time, in recent years (around 2012), a plateau was observed. To leave this plateau, hierarchical and deep learning based methods were also developed. As we discuss in the Subsubsection 2.1.2, Fukushima proposed 'neocognitron', inspired by human biology. This model was inspired by hierarchical and shift-invariant processing but lacked a suitable training method. After some time, stochastic gradient descent with backpropagation is applied to a similar network by LeCun et al. [43] to train CNNs, which extends the operations of the neocognitron. In the following years, starting with AlexNet [28], very successful entries to the ILSVRC [45] was made. However, these methods (and the ILSVRC) is oriented towards the image recognition problem, and this knowledge should be transferred to the object detection task.

In this subsection, we start by explaining the R-CNN network [20] in Subsection 3.1.2, which forms the base to many object detection networks nowadays. After the introduction of the R-CNN, we present modifications and updates and how they solved the prominent problems in object detection in the upcoming subsections.

R-CNN

Girshick et al. (2014) [20] developed a CNN-based method for object detection while uniting the knowledge from the image recognition methods' progress. Girshick et al. show that CNN-based object detection methods outperform other systems that use HOG with nearly 30% mean Average Precision (mAP) increase on VOC 2007. The CNN-way of detecting objects was the sliding-window method. This method applies convolution to a particular area of an image and then slides the image until all parts of the image are searched and classified. However, having large CNNs perform this operation makes it harder to decide on a precise location because of the long strides and filter sizes. Also, since the regions are exhaustively checked for objects, this method was considerably slow. R-CNN is a different approach than the sliding-window method where instead of exhaustively searching, the method generates object proposals from the input image that are class agnostic and classify those proposal areas with SVMs.

From the proposed areas, a fixed-size input is prepared to be used in a CNN hence the name, Regions with CNN features. Another contribution of [20] is that they pre-trained CNNs on ImageNet [92] and then fine-tuned on the smaller PASCAL dataset and showed that transfer learning is an effective way of training large CNNs. The flow of the algorithm is as follows. Firstly, the selective search [97] is used to detect possible regions. Then, the AlexNet network is used to extract fixed-size features (4096 features) from each of those

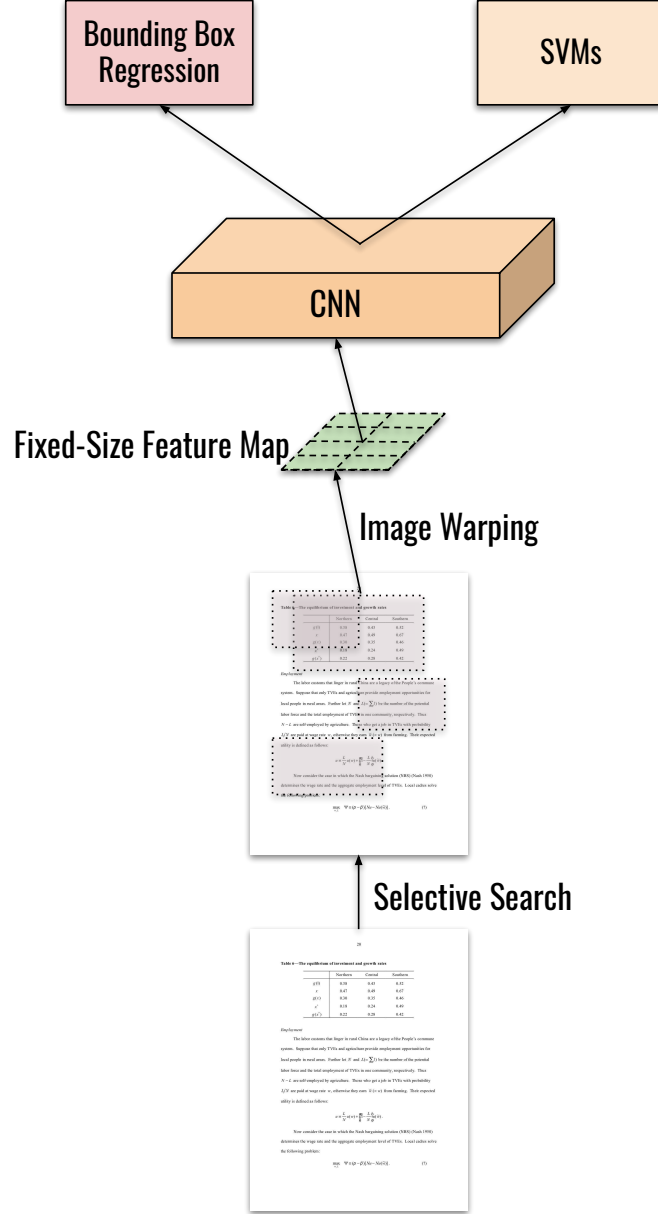


Figure 3.4: R-CNN method architecture used in the thesis. Operations are shown for a single bounding box proposal.

regions. Regions that do not meet the input criteria of the CNN network are warped to the correct size (4096 in this case). Lastly, fixed-size features are sent to the SVMs to conclude the object detection. The studies on the selective search algorithm show that around 2000 regions are selected for an image and selected regions have 91% recall on ILSVRC and 98% on PASCAL which clearly shows that there is plenty of room for improvement since the maximum recall attainable by the whole network cannot be higher than that of region proposal stage's. Class-specific SVMs give their scores to a region and afterwards, bounding-box regression is applied to those regions to predict a new bounding box. Let x, y, w, h be coordinates of a proposal bounding box and the width and height of it. Then, let x^*, y^*, w^*, h^* be coordinates and width and height for the ground-truth bounding box. Superscript i is used whenever needed, to denote the i th proposal. The learning process can be summed to learning w_θ in:

$$w_\theta = \underset{\hat{w}_\theta}{\operatorname{argmax}} \sum_i^N (t_\theta^i - \hat{w}_\theta^T \phi_5(P^i))^2 + \lambda ||\hat{w}_\theta||^2 \quad (3.11)$$

Where θ is one of x,y,w,h and $\phi_5(P)$ defined as the features extracted from the fifth (and final) convolution layer for the proposal P . To learn w_θ , we need the ground truth targets for the regression (t_θ) which are defined as (for a single proposal and related ground truth):

$$t_x = (x^* - x)/w \quad (3.12)$$

$$t_y = (y^* - y)/h \quad (3.13)$$

$$t_w = \log(w^*/w) \quad (3.14)$$

$$t_h = \log(h^*/h) \quad (3.15)$$

Finally, the network is trained as an optimization problem with the least square loss function and can be solved in closed form efficiently. R-CNN [20] method achieves 66% mAP on PASCAL VOC 2007 test set [98] and 62.4% mAP on the PASCAL VOC 2012 test set [99]. The overall architecture of an R-CNN is illustrated in Figure 3.4.

Fast-RCNN

Ross Girshick (2015) [21] proposed a series of innovations that build on top of their previous work, R-CNN. These innovations increased the detection accuracy while improving the speed as well. In this paper, the author proposed a deep learning method with a single-stage training process, i.e. E2E training of the whole system. The mentioned speed

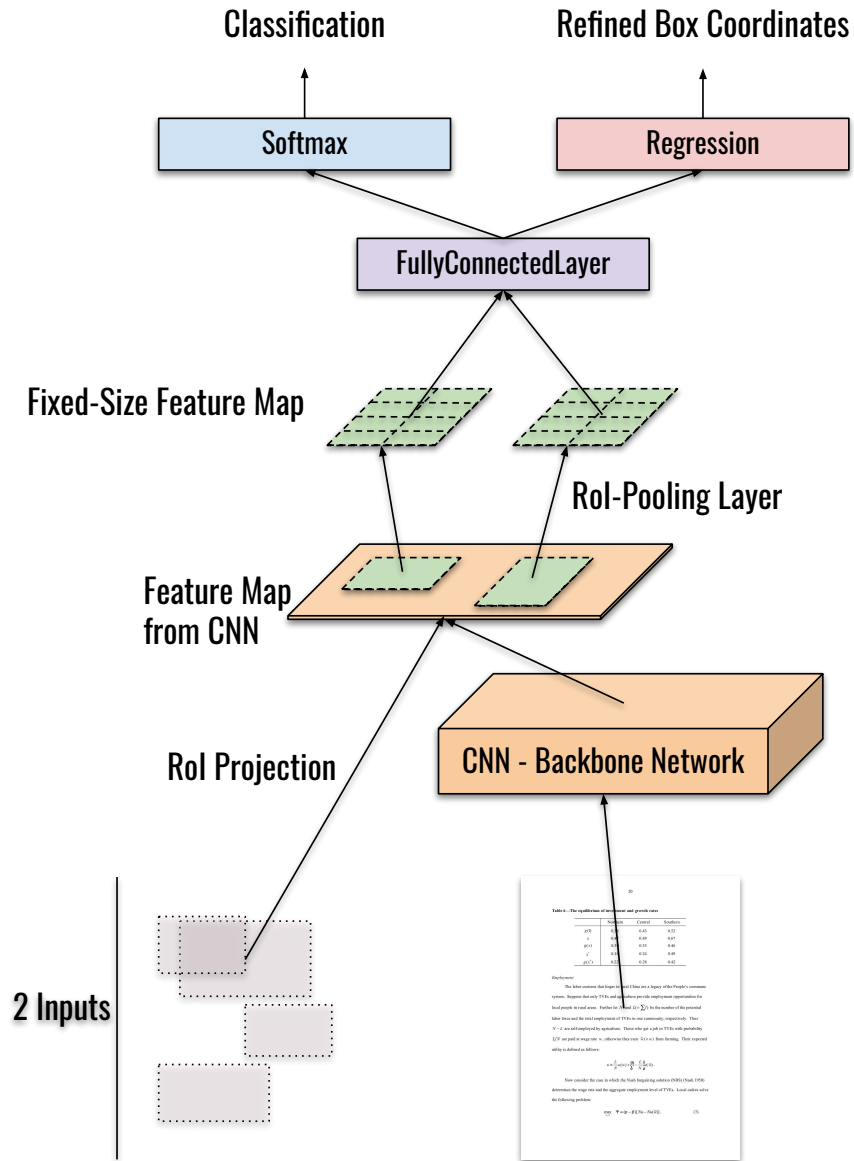


Figure 3.5: Fast-RCNN method architecture used in the thesis.

improvements are significant because R-CNN model with VGG16 takes about 47 seconds to process an image whereas Fast-RCNN takes around 0.3 seconds *without region proposal time taken into account* [21]. R-CNN, as we have mentioned, have several drawbacks. The training process is not trivial; where first, a CNN is trained and fine-tuned, then separate SVMs are trained for each class. Finally, bounding box regressors are trained. Moreover, the slowness of R-CNN comes from the fact that the underlying CNN model is run for every proposal, which there are about 2000 per image.

Fast-RCNN takes two inputs; image and proposals. Hence, the improvements are not at the bounding box proposal stage but at the later stages where convolution and SVMs are. First, the image is processed with the CNN feature extractor, which results in a feature map. Then, from each proposal, a fixed-length feature vector is extracted from the feature map via the RoI pooling layer. The change here is instead of running through CNN for each proposal; the image is processed only once. The fixed-size feature vector then fed into a fully-connected network which branches into two parts; a softmax layer for classification and another layer to refine the proposal boxes. The mentioned RoI-pooling layer uses max pooling to convert the features that are inside any regions of interest into fixed-size feature vectors. This fixed-size is given to the network as hyper-parameters, and assume them to be Y and Y' . Then the fixed-size small feature map would have the size $Y \times Y'$. Each RoI is defined with four coordinates (r, c, w, h) that denotes the top-left corner (r, c) and its width and height (w, h) . RoI max pooling layer divides this RoI into equal length spatial sub-windows of the approximate size of h/Y and w/Y' . Then, the regular max pooling operation is applied. To make use of successful CNNs, Fast-RCNN also uses pre-trained ImageNet [92] convolutional networks. The aforementioned RoI-pooling layer replaces the last pooling layers of the feature extractor CNN's.

Fast-RCNN takes the advantage that comes with architectural decisions. They share the features extracted for each RoI while training to prevent recalculating the fixed-size feature maps, which gave about 64 times faster training [21]. Also, R-CNN separately trained SVMs, but Fast-RCNN combines the box regression and classification branches with the CNN and RoI-pooling itself and trains the system as a whole. Therefore, Fast-RCNN network has two output layers. One, softmax layer, outputs a probability distribution per RoI, $p = (p_0, \dots, p_K)$ over $K + 1$ classes where K is the number of object classes in the dataset and 1 comes from the *background* class which represents the negative class. The second layer outputs offset for bounding boxes, $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$ for $u \in K$. A ground truth class k and a ground truth bounding box coordinate target v is associated with each training RoI. A multi-task loss L is then jointly trained for bounding box regression and RoI classification:

$$L(p, k, t^k, v) = L_{cls}(p, k) + \lambda[k \geq 1]L_{reg}(t^k, v), \quad (3.16)$$

in which L_{cls} and L_{reg} are defined as the following:

$$L_{cls}(p, k) = -\log(p_k) \quad (3.17)$$

$$L_{reg}(t^k, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^k - v_i), \quad (3.18)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (3.19)$$

Girshick [21] chosen smooth L_1 loss because it is less sensitive to changes and outliers than the L_2 loss used in R-CNN. In addition to this advantage, the classic L_1 loss is not differentiable around 0. Smooth L_1 loss converts into L_2 loss around zero to overcome this issue. Since the targets for the regression are unbounded, hyperparameters like learning rate have to be carefully tuned to prevent exploding gradients problem.

Fast-RCNN [21] method achieves 70% mAP on PASCAL VOC 2007 test set [98] and 68.4% mAP on the PASCAL VOC 2012 test set [99]. The overall architecture of an example Fast-RCNN system can be found in Figure 3.5.

Faster-RCNN

Faster-RCNN is another improvement made on the R-CNN object detection networks family by Ren et al. (2015) [12] further improving Fast-RCNN. This new method addresses some of the critical issues that are present in the deep learning based object detection methods. Selection search [97] was one of the many region proposal methods used by the research community. However, it was slow, two images per second, and since it is not learnable, we were missing out on learning domain-dependent features. Instead of the selective search algorithm that runs on CPU, Faster-RCNN proposes a deep convolutional neural network that shares convolutional layers with shared layers. The introduced network is called *Region Proposal Network (RPN)*. RPN consist of a few convolutional layers and thus can be trained E2E. [12] also proposed the term 'anchor' in object detection, which allows it to detect objects at different scales and ratios.

The RPN is designed to take a feature map as input and output region proposal with objectness scores assigned to each region. The RPN can simultaneously regress coordinates of the possible regions and give an objectness score to the regions. Objectness score denotes if the region belongs to the background class or one of the user classes. To speed up the process, RPN shares layers with the Fast-RCNN's convolutional network (backbone

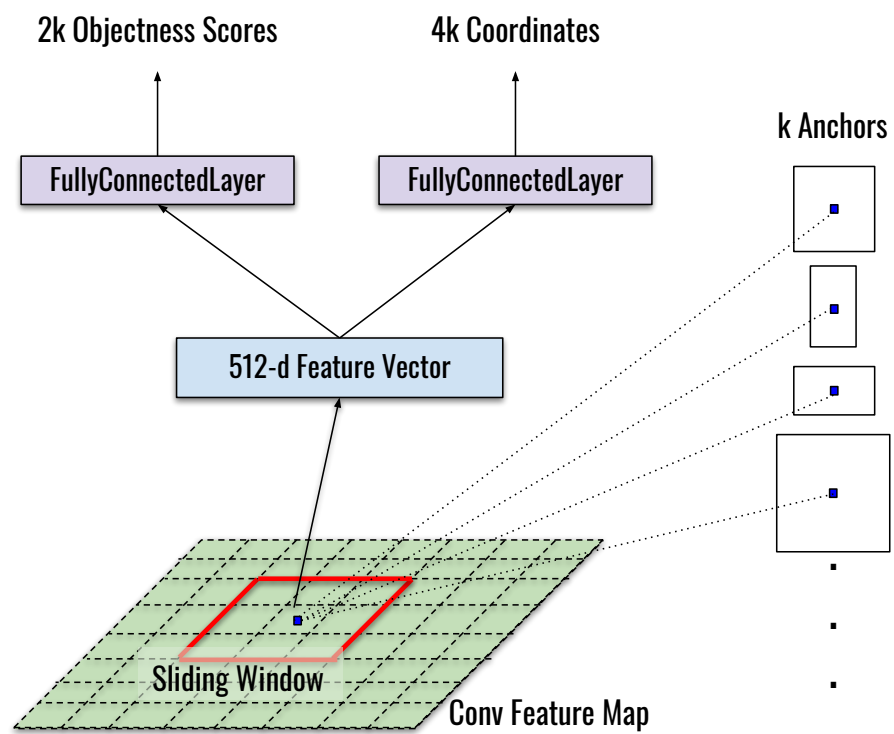


Figure 3.6: Overview of the Region Proposal Network used in the thesis.

network). For example, if VGG-16 [19] backbone is used, a feature map from the last shared convolutional layer (13th layer) is extracted, and then a small network is slid over this feature map. This small network’s input is $n \times n$. The output of RPN is then mapped to 512 dimension vector. Ren et al. propose to use $n = 3$, and we use it as it is. The extracted feature vector (512-d) is processed by the objectness layer and regression layer of which are implemented as fully connected layers. The overall RPN architecture can be found in Figure 3.6. The anchors, as mentioned earlier, allow the RPN to produce a proposal on different scales and ratios. An input to the system is k , the maximum number of proposals for each location. Therefore, regression layer has $4k$ outputs with the coordinates and the classification layer outputs $2k$ objectness score. These k proposal are made with anchors. Anchors have different scales (3 by default) and ratios (by default 2:1, 1:2 and 1:1) giving $k = 9$ anchors to be placed at each location. In every sliding window slice, all these anchors are placed, and objectness scores are given. Hence, If the size of the feature map is $W \times H$, then there are $W \times H \times k$ anchors in total. The scaling is added to benefit from the feature pyramids structure, similar to the FPNs [15]. Resizing the anchors adds smaller computational head when compared to scaling the image itself and repeating the detection process. However, significant gains can be observed with this trick where using anchors with different scales gives about 2% mAP on the PASCAL VOC 2007.

In training the network, anchors are assigned a score of 1 if they have IoU of 0.7 or higher with any ground truth box. 0 otherwise. Then, the loss function is just an adaptation from the Fast-RCNN. Let i be an index for an anchor, in a mini-batch, with a probability p_i of being an object. Also, the ground truth label for the anchors is denoted as p_i^* . Lastly, t_i represents the four coordinates that define a predicted bounding box and t_i^* is the coordinates for the corresponding ground truth bounding box. Then the multi-task loss L is defined as:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*). \quad (3.20)$$

The loss L_{cls} is same as Eq. (3.17) as defined in Fast-RCNN chapter. The regression loss is smooth L_1 loss (Eq. (3.19)), adapted from the Fast-RCNN [21] and defined as:

$$L_{reg}(t_i, t_i^*) = smooth_{L_1}(t_i - t_i^*) \quad (3.21)$$

In Faster-RCNN’s multi-task loss function, classification loss is normalized by the batch size (N_{cls}) and the regression loss is normalized by the number of anchor locations (N_{reg}).

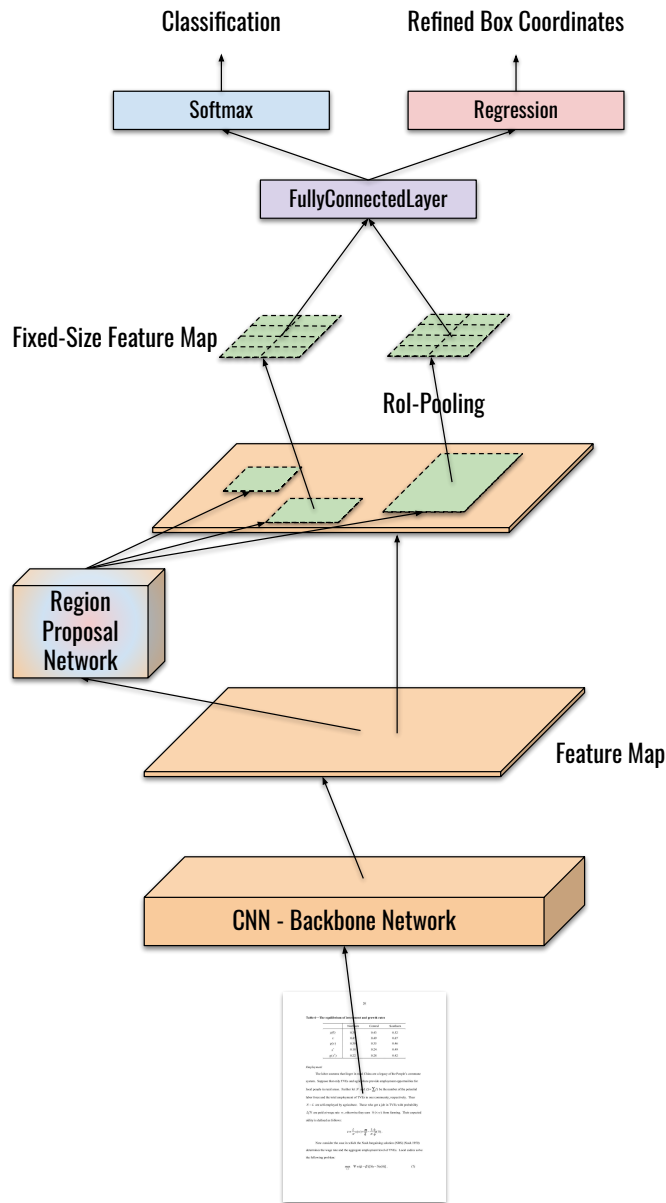


Figure 3.7: Overview of the Faster-RCNN method architecture used in the thesis.

The regression loss ($L_{reg}(t_i, t_i^*)$) is borrowed from the parameterizations in the R-CNN paper [20] and given as follows:

$$\begin{aligned} t_x &= (x - x_a)/w_a, & t_y &= (y - y_a)/h_a, \\ t_w &= \log(w/w_a), & t_h &= \log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, & t_y^* &= (y^* - y_a)/h_a, \\ t_w^* &= \log(w^*/w_a), & t_h^* &= \log(h^*/h_a) \end{aligned} \tag{3.22}$$

As usual, x, y, w, h represents the bounding box's mid coordinates and the width and height of the box. Three different notations; x, x^* and x_a are for the predicted bounding box, ground truth box and the anchor box respectively.

The overall architecture of the Faster-RCNN method can be viewed at Figure 3.7. We used the Faster-RCNN method with a backbone of VGG-16 and ResNet-50 that are pre-trained on the MS COCO dataset [100]. The reasons behind this choice are because the Faster-RCNN method is a popular and state-of-the-art method in terms of performance as well as speed. Also, it is applied to many domains and helped achieve good results such as; table and table structure detection [7, 8], pedestrian detection [101] and instance retrieval [102]. The model achieves 78.8% mAP on the PASCAL VOC 2007 test set [98], and 75.9% mAP on the PASCAL VOC 2012 test set [99] which surpassed previous methods and claimed the throne in the object detection research while improving on the speed with 0.2 seconds per image *including proposals* with VGG-16 as the backbone network. The Faster-RCNN architecture is adopted by previous researchers to the document analysis domain [81, 10, 75, 36]. The findings show that this network architecture adopts well to this domain and therefore, we include it in our comparisons with different backbone networks.

Mask-RCNN

He et al. (2017) [11] proposed the Mask-RCNN model just after two years from the release of the Faster-RCNN. Mask-RCNN is essentially an instance segmentation version of the previous iteration, Faster-RCNN. Instance segmentation is just a particular case of image segmentation problem. In regular image segmentation, an image is segmented into classes. That is, distinct instances of a class are not separated, but only the classes are segmented. However, in instance segmentation problem, we care about every instance of a class and this problem can be considered as an instance-aware version of the image segmentation problem. Mask-RCNN earns instance segmentation capabilities by adding a new branch to the Faster-RCNN model; a fully convolutional network which is explained

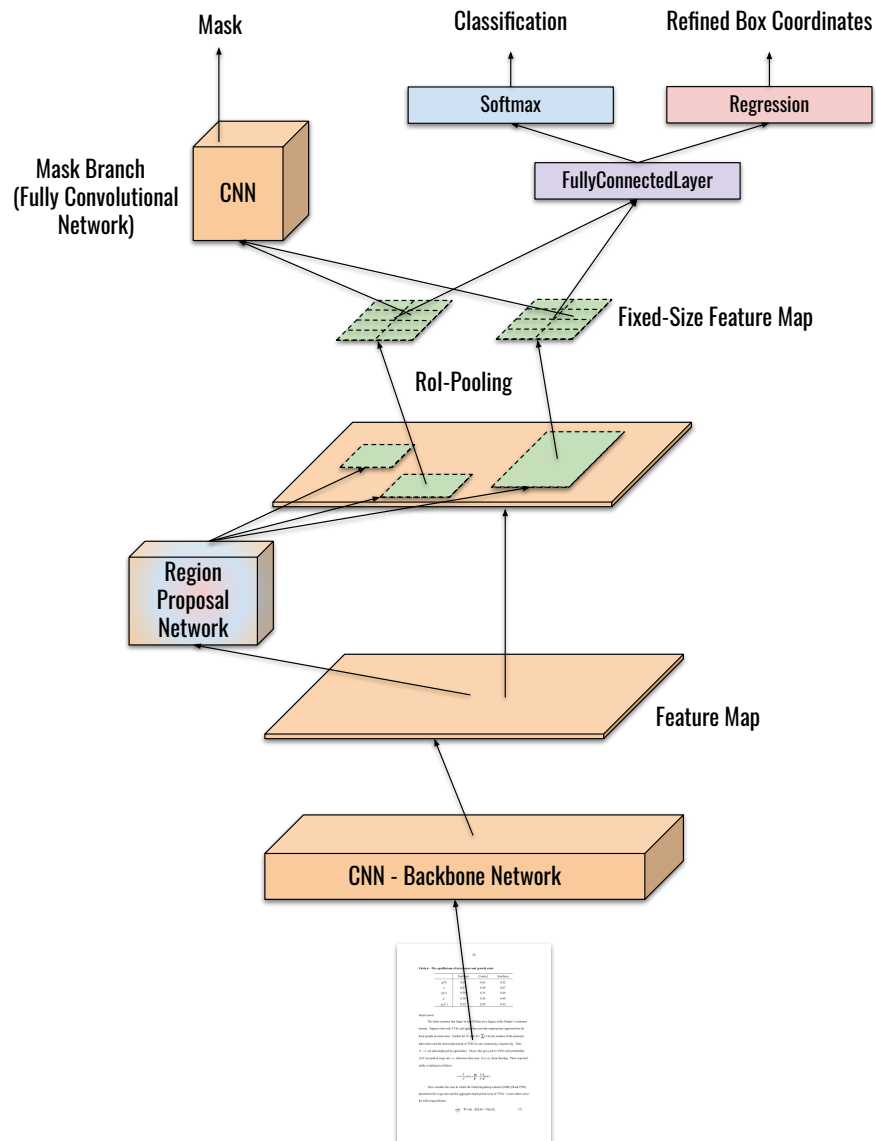


Figure 3.8: Mask-RCNN method architecture used in the thesis.

in-depth in Subsection 3.1.4. He et al. [11] adopted the Faster-RCNN where the first stage (RPN) is the same, but in the second stage in parallel to the classification and regression branches, an FCN is added to perform classification on a pixel level. As in region proposals, generated object masks are also binary with objectness scores. The classification of the masks is inferred from the classification loss of the underlying Fast-RCNN model. The overall architecture of the proposed Mask-RCNN model can be found in Figure 3.8.

Since the network extends the older methods, the multi-task loss used to train the networks are also just an extension to the previous methods. The multi-task loss L is defined as:

$$L = L_{cls} + L_{reg} + L_{mask} \quad (3.23)$$

L_{cls} is the classification loss and L_{reg} is the bounding box loss and are exactly the same as Eq. (3.17) and Eq. (3.18), respectively. Assume there are K classes and mask resolutions are $m \times m$ for each class and each RoI (in our case, $m = 28$). Therefore, there are Km^2 outputs for a single RoI. An important thing to note here is that since the masks are binary, there is no competition between classes, which makes the training easier for masks and is one of the choices that shown to improve the performance by [11]. Then, sigmoid is applied to outputs in a pixel level and L_{mask} is defined as average binary cross-entropy loss where the binary cross entropy is:

$$CrossEntropy(p, y) = \begin{cases} -\log(p), & \text{if } y = 1 \\ -\log(1 - p), & \text{otherwise} \end{cases} \quad (3.24)$$

In the above y specifies the ground truth where p stands for the predicted probability for the true class ($y = 1$). Then L_{mask} is:

$$-\frac{1}{m^2} \sum_{1 \leq i, j \leq m} [y_{ij} \log(\hat{y}_{ij}^k) + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)] \quad (3.25)$$

Where ij denotes a cell in the $m \times m$ mask and y_{ij} represents the label of this cell in the ground truth. $\hat{y}_{ij}^k$ stands for the predicted value in the same position of the predicted mask that belongs to the ground truth class k . Note that if an RoI is associated with the k -th class, L_{mask} is only defined on the k -th mask.

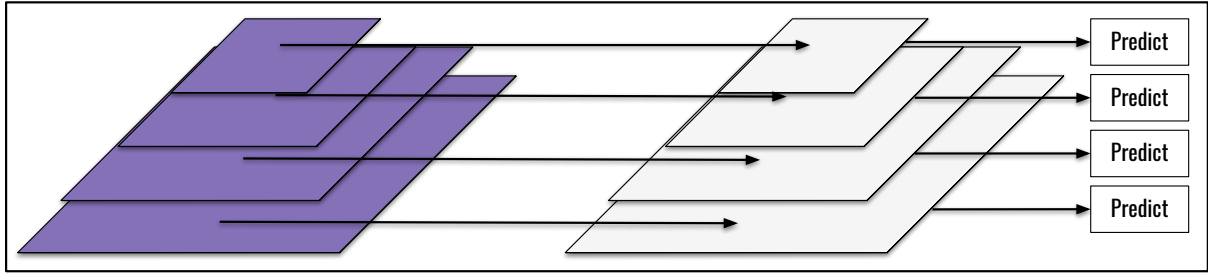
The goal of the mask branch is to detect $m \times m$ mask for each RoI. This requires to maintain the precise spatial layout of the features in the RoI. Using a fully connected layer or other operations that do not preserve spatial information is hurtful in pixel level classification, and thus, the rational choice is using fully convolutional networks. RoIPool used in the heart of the Fast and Faster-RCNN models to process the RoIs. However,

RoIPool algorithm is not suitable for the pixel-level segmentation task because, in RoIPool, the proposed region coordinates are first quantized and converted to coordinates (which are rounded to index the feature map) on the feature map which then quantized again to be used as coordinates on the RoI features. These continuous operations found to introduce misalignment when used for segmentation [11]. To eliminate this misalignment, He et al. proposed a new algorithm called RoIAlign [11]. RoIAlign is an intuitive improvement over RoIPool. The proposed algorithm does not quantize the coordinates but uses bilinear interpolation inspired by the spatial transformer networks [103] instead. With this change, relative mask accuracy is increased by 50% when compared to the older RoIPool method.

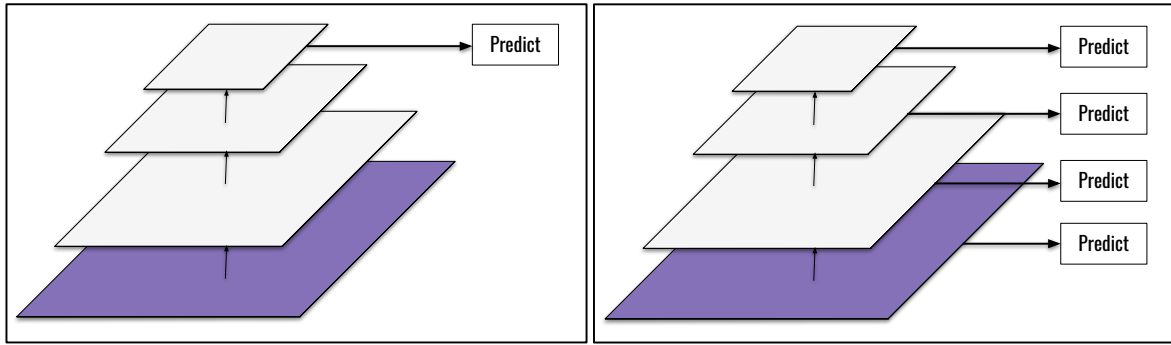
Another contribution of the Mask-RCNN model is that they have tested ResNet-101-C4 [14] (feature map is extracted from the last convolutional layer of the conv4 block of the ResNet backbone network) and ResNet-101 with FPN [14, 15] and they achieved 3-5% increase in the precision for the MS COCO test set [100] which experimentally shows the importance of the backbone network selection. Using sigmoid in the L_{mask} (class-agnostic masks) increases the precision by 5-7% in the MS COCO compared to using softmax (class dependent masks). In our previous works [8, 9], we have used Mask-RCNN model to detect internal table structure detection and table detection, which we have obtained the state-of-the-art results for the table detection task. In [9], we have concluded that FPNs and FCNs are helping the model to learn low-level features and increases the robustness of the method. In our experiments, our choice of a backbone network is the ResNet-50 because the dataset sizes are limited for tabular structure detection task, and it is challenging to train deeper networks with this amount of data. Since Faster-RCNN is widely adopted in table detection with deep learning (see Subsection ??), we know that R-CNN based object detectors works well on this domain. Mask-RCNN is an improvement made on top of the Faster-RCNN architecture and inclusion of fully convolutional network into the architecture will allow the model to learn details of the documents. The models need every bit of detail especially for row detection purposes (as shown in Chapter 5). This is the main motivation to try Mask-RCNN, for the first time in tabular structure detection, and the results are satisfactory as we will discuss in the upcoming chapters.

3.1.3 Feature Pyramid Networks

Detecting objects at different scales has always been a problem in computer vision, and many people proposed different solutions. One of them is scaling the images manually and performing detection at different scales which is rooted from [104](Figure 3.9a) and is a common practice as it can be seen in hand-crafted rule-based image processing methods [105, 106]. This method allows the methods to be scale-invariant with a sacrifice of the

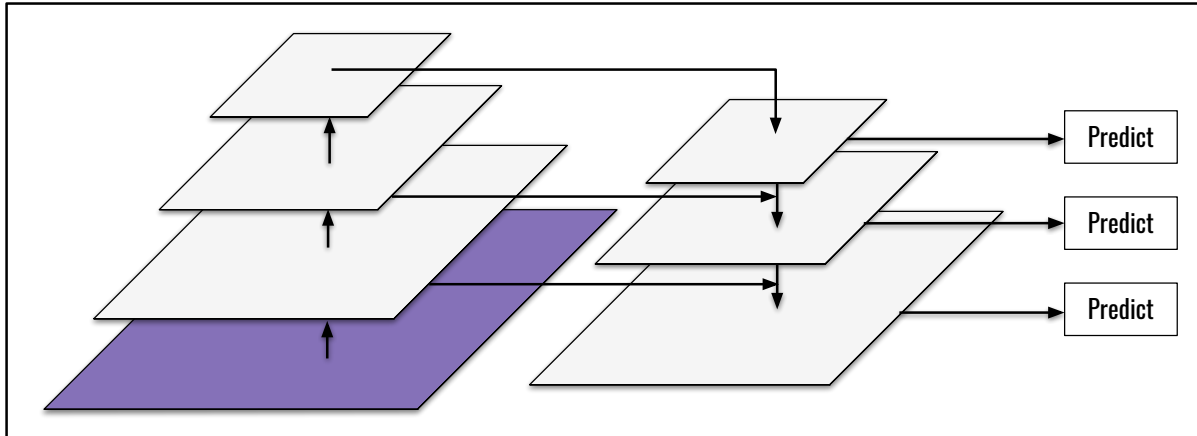


(a) Image scaling



(b) Single feature map

(c) Hierarchical features



(d) Feature pyramid network

Figure 3.9: Feature pyramid networks used in the thesis. Coloured boxes represent the images and white boxes stands for feature maps.

speed since the detection is performed more than once. Convolutional networks have shown to be more robust to the scale changes of objects [21, 12](Figure 3.9b) but more in-depth inspection shows that more performance gains can be attainable with pyramid-like structures [15]. Detection at different scales is beneficial when detecting objects at a different scale. For example, tables can be detected on most scales, whereas smaller objects like rows can only be detected on a grander scale, as [8] shows. Although it is the intuitive way to make the models scale-invariant, image pyramids are not the only way to achieve this goal. Since CNNs already calculate *feature pyramids* in their forward-pass which inherently is a pyramid-like shape with hierarchical features at each level. SSD [79] is a method that used CNN’s hierarchical features and performed detections at different levels (Figure 3.9c), but they have added new layers to complete the pyramid and avoided using lower layers. Lin et al. (2017) [15] proposed a new method called FPN that benefits from the CNNs’ inherent feature pyramids and they have combined semantically low-level features and semantically high-level features to imitate the image pyramids scheme and form a top-down architecture to perform detections on every level as in Figure 3.9d.

The tests by Lin et al. [15] shows that FPNs significantly increases the recall on object detection task. Numerically, FPN architecture increased the Average Recall (AR) by 8 and Average Precision (AP) by 2.3 or 3.8 for COCO-style [100] or PASCAL-style [96] precision calculation respectively. The FPN can be divided into two parts, the bottom-up path and the top-down path. The bottom-up pathway is the regular feed-forward operations of the convolutional networks. Since many convolutional layers are not changing the size of the input (i.e. convolution layers of the conv4 block of the ResNet [14] model), they are considered to belong to the same level in the bottom-up pathway. Then, bottom-up feature pyramid is constructed with last convolutional layers of each of these levels. For example, for the ResNet models, feature maps are extracted from the last layers of the conv2, conv3, conv4 and conv5 blocks. The top-down pathway is being constructed starting from the later layers of the convolutional network and going down in the network by extracting feature maps at selected layers as stated. These feature maps are then laterally connected to the top-down path and merged with the other feature maps. Every layer in the path is upsampled using the nearest neighbour algorithm before sent to the lower layer and then merged with the lower layer by simply adding the features. The process of the FPN architecture is shown in Figure 3.9d. On the left of the figure, the bottom-up path is shown, and on the right, the top-down path can be found.

Since in our problem, we have wide and short objects as well as tall and narrow objects (rows and columns) performing detection at different scales should help us. As per the experiments in [15], Faster-RCNN model with FPN performs marginally better on the COCO test set [100]. We have also experimented with FPN, and according to our previous

investigations in [8], the model with FPN (and FCN) improved the AP and AR by 4-6% in the public UNLV table structure dataset.

3.1.4 Fully Convolutional Networks

Object detection benefited from the CNNs in the last years [20, 12, 79]. Following the success in detection tasks, the next intuitive progression is the segmentation of objects. Being able to segment objects might help us in some cases or strike as more natural rather than a chunky rectangular bounding box for non-rectangular shaped objects (i.e. most of the objects in the world). Convolutional layers are the de facto operation in image-related methods and feature extraction from images [20, 15] but feeding these features to fully connected neural networks would result in losing the spatial information and inaccurate segmentation of objects [16]. Therefore, they should be handled with convolutional layers as well. Long et al. (2015) [16] proposed the Fully Convolutional Networks (FCNs) by adapting object detection networks. The adaptation process requires converting classification branches of these networks (i.e. Faster-RCNN [12]) to deconvolutional layers to upscale the feature map back to its original size and predict and define loss for each pixel. [16] shows that from the features there are two options to upscale the image to the original size with detections; (i) use bilinear upsampling (or similar) and (ii) use deconvolutional layers which are convolution layers where the forward and backward passes are reversed. The choice of upsampling for FCNs are usually with learnt deconvolutional layers [16, 11]. To reduce the dimensionality, 1×1 convolution is applied with channel dimension equal to the number of classes to predict.

An example segmentation method is illustrated in Figure 3.10 as described in [16]. This architecture is a modified version of VGG-16 [19] network with only a slight modification where fully connected layers are converted into convolutional ones which also allows the network to take arbitrarily sized input since there are no dense connections between layers. The networks are named as FCN-32s or FCN-8s to denote the total stride amount of the network with respect to the input image. Since FCN-32 is obtained by upsampling the conv7 features (converted to convolutional layer from the fc7 of the VGG-16) through deconvolution operations and the stride amount of conv7 layer to the original image is 32. Also, FCN-8 (or FCN-16 or FCN-4, same idea) is upsampled from the output of the 'pool3' layer which had eight strides on the original image. When connecting the output of pool4 layers to the output conv7, since they have a stride of 2 difference in between, first, conv7 is upsampled by 2 (shown as '2x' in Figure 3.10), and then the features are added together and deconvoluted. Upsampling is initialized to bilinear interpolation, but the network is allowed to refine the weights. 1×1 convolutions are used to acquire class predictions

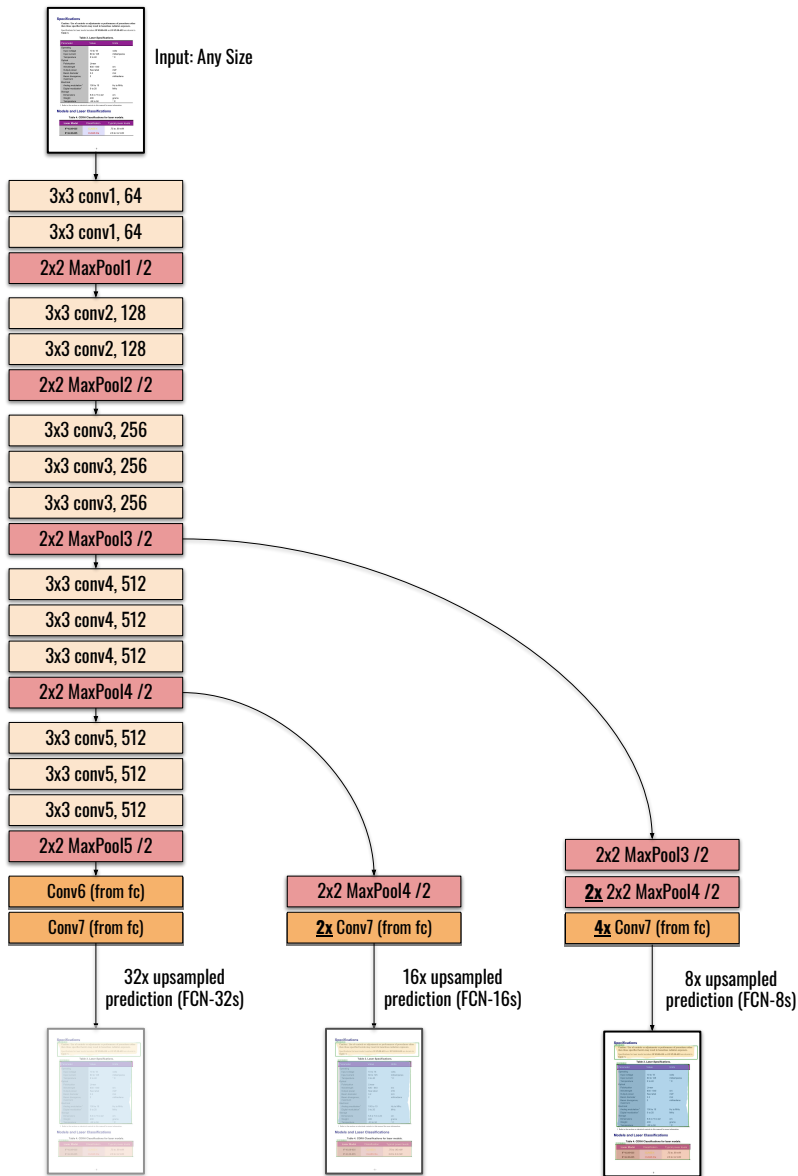


Figure 3.10: A fully convolutional architecture based on VGG-16 to classify pixels. FCN-32s upsamples the predictions made with 32 stride back to pixels. FCN-16s combines the output of both the last layer and the 'pool4' layer which has stride of 16. FCN-8s is the combination of information from pool3, pool4 and conv7. It provides further precision and allow the network to keep the high-level semantics as well as providing finer details about the input.

and before the addition operation that merges layers to reduce the dimensionality. Best results are attained with FCN-8 architecture on PASCAL VOC 2011 [107], but we believe for our case, it might be useful to try lower stride amounts as well, and we leave it as future work. The proposed FCN architecture achieved state-of-the-art on segmentation tasks and now widely adopted as the de-facto segmentation method [11, 10, 108] In our Mask-RCNN model, we also make use of FCNs to segment instances and help the model to retain information from the lower levels better. Since the FCN has to classify each pixel of an image, the network is forced to remember every detail of the input. This property is expected to increase the accuracy of our models.

3.1.5 RetinaNet

Up to now, all the previously mentioned detection models were *two-stage* detection methods where the first stage is the region proposal (selective search [97] or RPN [12]) and the second stage is the refinement and classification of these proposals. If we were to examine the weaknesses of these architectures, we would see that there are around 2000 bounding box proposals where there is only one box to predict. The time is wasted on generating these proposals as well as classifying them. Single-stage object detectors (which eliminate the first, proposal, stage) are inherently faster but less accurate. Lin et al. (2017) [13] studied this weakness of the single-stage methods and tried to eliminate what causes the weakness. Since single-stage methods do not eliminate the massive size of possible object regions in an image to couple thousands, the number of correct boxes are near to zero next to the total amount of possible bounding boxes. Even though at the time there were fast and single-stage detectors like YOLO [80] and SSD [79], they were not competent enough against two-stage networks on public test sets such as MS COCO [100].

Class imbalance problem is mostly present in the single-stage methods in object detection problem. In two-stage methods, a method like RPN decreases the search space immensely, which results in a sparse search on the input image where only 2000 bounding boxes are looked at. Although the search is sparse, this operation is time-costly due to the nature of the process, for example, the selective search takes almost 2 seconds [20] per image. In the single-stage methods, dense sampling is performed from the image to search for objects. In the end, these methods evaluate many more boxes than the two-stage methods in which most of them turn out to be negatives. This imbalance causes several problems where one of them is being the harmful effects of many detections because they are easy to classify and do not provide valuable loss to backpropagate. To overcome the class imbalance problem, [13] proposes a novel loss function called *focal loss*, which down-weights the easy examples to prevent the network from being overwhelmed by negative examples.

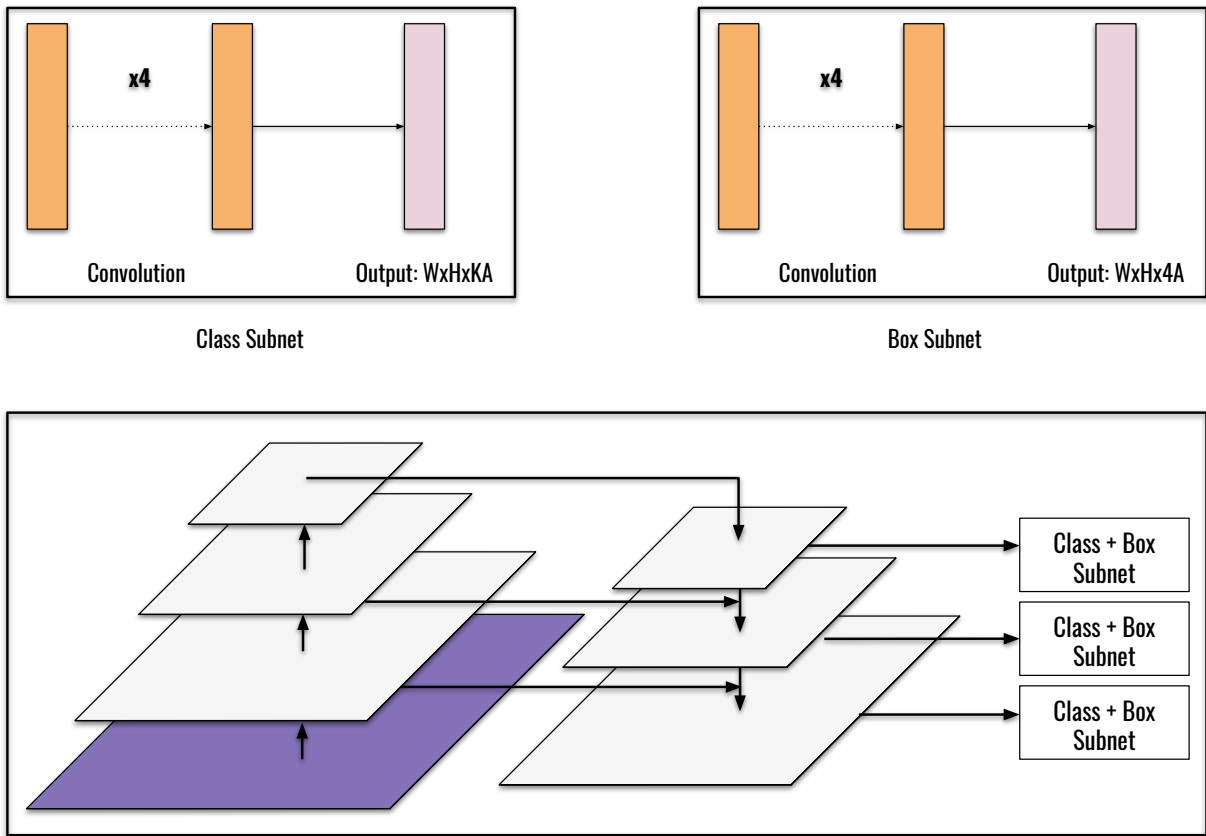


Figure 3.11: The RetinaNet architecture used in the thesis. Purple boxes represent the input image and white boxes stand for feature maps.

The focal loss is based on the cross-entropy function (Eq. (3.24)) with an addition:

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \quad (3.26)$$

Which simplifies the *CrossEntropy* function to:

$$CrossEntropy(p_t) = -\log(p_t) \quad (3.27)$$

The cross-entropy could be multiplied with a value (for example class frequency) to address the imbalance problem in the dataset (positive/negative data imbalance), but the inherent problem that comes with the single-stage methods require a distinguishing factor between easy and difficult examples. Lin et al. [13] proposed a modification to the vanilla cross-entropy, which gives less importance to the correctly classified examples. They have achieved that by proposing a factor $(1 - p_t)^\gamma$ with focusing parameter $\gamma \geq 0$. The focal loss is then defined as:

$$FocalLoss(p_t) = -\beta_t(1 - p_t)^\gamma \log(p_t) \quad (3.28)$$

with β being the scaling factor and focusing parameter is denoted as γ . From these formulations, it is clear that even though the easy examples are classified with near-perfect precision, when their quantity is considered, they do not contribute to the loss and cannot overwhelm the training. When an example is correctly classified and p_t is close to 1, the loss for that example drops dramatically. [13] shows that their proposed method RetinaNet with ResNet-101 [14] with FPN [15] outperforms the Faster-RCNN model with the same backbone network on the MS COCO [100] test set. The architecture of a RetinaNet model can be seen in Figure 3.11. RetinaNet uses a backbone feature extractor connected with FPN to extract features with different semantic levels. Anchors are then applied at each level; to the feature maps extracted from the backbone network as in [15]. Moreover, as in Mask-RCNN [11], there is a regression subnetwork to calculate the offset from each of these anchors placed. Calculating the offset from the anchors but not predicting the location of the box altogether without anchors limits the capabilities of the network. The box regression subnetwork is a fully connected network as the classification subnet. The classification subnet classifies each anchor for each class in parallel to the regression subnet. We also test the RetinaNet model to see the performance of the single-stage detectors on our task. We found it to be outperformed by the Faster-RCNN and Mask-RCNN.

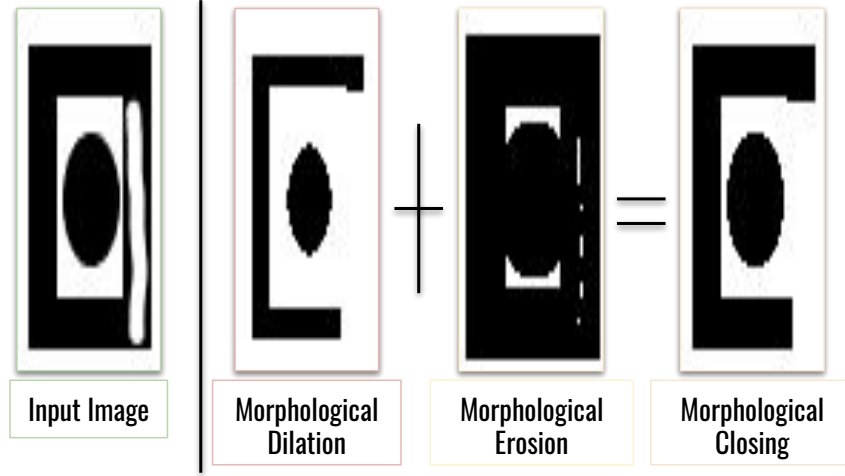


Figure 3.12: The morphological method examples. All operations are shown with a simple 5×5 square kernel.

3.2 Rule-Based Mathematical Morphology Method

Deep learning methods, especially the object detection methods, excelled on a wide variety of tasks including table and table structure detection tasks [7, 8, 9, 101]. Although deep learning methods are good at this, there are also heuristics based solutions to these problems, and they are still studied [109, 110, 76]. One popular method is to use the concept of connected components [109, 8, 111] and to form the connected components we need to connect words to create a component. To do this, we used the mathematical morphology method as defined in [22]. The morphology method is used to process digital images and analyze geometrical compositions. It has several sub-methods, but the two fundamental methods are; morphological erosion and morphological dilation. Erosion is an operation that is applied to a binary image with a kernel which then the image gets eroded according to the kernel applied. Dilation is the inverse of the erosion operation with a kernel as well. The effects of these methods can be simplified as; erosion decreases the intensity of the white pixels where dilation increases the intensity. When these two methods are applied back-to-back (first dilating then eroding), the small gaps between the structures are closed, and hence, forms '*connected components*'.

We test the mathematical morphology method because the connected components are widely used, and morphological closing method is an effective and established way to form connected components. Moreover, since they are developed to analyze geometrical

structures [22], it makes a compelling case to extend this method to contemporary problems as well. These methods have been previously applied to the table detection problem [31], and we also have proposed it to be used to detect column and rows [8]. Table detection by using the connected components strategy is less challenging because there are many rows and columns to supplement the preliminary table detection proposal with alignment and white space checks. However, structure detection is a more laborious task as sometimes the structural elements include other structures (i.e. smaller tables or figures) inside them. The effects of the erosion (Figure 3.12, middle-right), dilation (Figure 3.12, middle-left) and closing (Figure 3.12, right) is shown in Figure 3.12.

Chapter 4

Table Detection

We propose deep learning based solutions to find tables in arbitrarily styled documents be it research papers, magazines, journals, newspapers, web pages and more. Our previous findings on table detection can be found in [7]. The document format is in digital image format because every other document format can be converted into an image which allows our system to take any document as input. The proposed table detection model also works on scanned and possible tilted images that result from scanning progress, and we achieve state-of-the-art results on a number of public datasets. We also present a comparison of state-of-the-art object detector networks on our domain under public datasets. We conclude that our Mask-RCNN [11] based method is the best among the other methods we study. The detections by this model are accurate, and speed-wise it competes with other methods we present. Since the annotated dataset for the table detection task is lacking, the research community use existing public datasets and benefit from transfer learning capabilities of the deep learning methods to adapt the models to table detection domain. We propose that unlike how public datasets and their training sets annotated, annotating all table-like areas, which we call *tabular areas*, increases the overall recall and precision with a trade-off of an increased number of false-positives. Tabular regions include the table of content pages, some form of *lists* that has several columns and rows, actual tables or sometimes figures as well or in other words, *everything that looks like tables*. With this addition, we had achieved a perfect recall (1.0) with Mask-RCNN model.

In training the deep learning models, even though we use transfer learning to adapt pre-trained backbone networks in our models to the document domain, we still require more data. To enlarge the available dataset, we combine several public datasets and hand-label a private dataset which is aggregated by Lytica Inc and provided to us. Due to the privacy reasons, this private dataset is not shared with the public. In the end, the datasets

are Marmot¹, ICDAR 2013 [73], ICDAR 2017 Page Object Detection (POD) Dataset [77], UNLV² and our private dataset provided to us by Lytica Inc³. The tested methods are Faster-RCNN [12], RetinaNet [13], Mask-RCNN [11] with the backbone networks of VGG-16 [19], ResNet [14] and ResNeXt [85] variants. This chapter starts by examining the datasets we use in Section 4.1 and then followed by the methods we use and our choice of parameters and additions if any in Section 4.2. After the methods and datasets are detailed, Section 4.3 explains our training process and we concludes the table detection approach of ours by presenting our results and comparing it to others in Section 4.4. We reach to a final conclusion in Section 4.5.

4.1 Datasets

Table 4.1: Datasets and distribution of the images.

Dataset	Total Images	Training and Validation	Test
ICDAR 2013	238	-	238
ICDAR 2017	2417	2217	200
Marmot	2000	829	-
UNLV	424	350	50
Private	2323	2123	200

In this study, we primarily focus on a specific domain that is tabular information extraction from the electronic component datasheets. However, when the examples from the provided private dataset are examined (Figure 4.5), one can observe a diverse set of table styles. Hence, along labelling more than 2000 pages from this dataset which includes millions of documents with many pages, we also use public datasets because the deep learning methods are data-driven methods. With this wide range of datasets from different sources, our models can detect tables from documents from many domains. The datasets used are Marmot, ICDAR 2013, ICDAR 2017 POD, UNLV and our private dataset. The overview of all the instances of the datasets and their distribution to the training (and validation) and the test set is shown in Table 4.1. We experimented with increasing the number of test images by including some images from the training set to be used in the test

¹http://www.icst.pku.edu.cn/cpdp/data/marmot_data_bak.htm (currently not available)

²<http://www.isri.unlv.edu/ISRI/OCRtk> (currently not available)

³<https://www.lytica.com/>

sets. Since the models are only trained for a few epochs, it seems that they did not overfit to the training set and increasing the test set size with data points from the training set did not improve our results and instead, we acquired slightly worse results in some cases. In the end, we use a collection of all the datasets to train our networks and we have 171 images separated for validating the trained snapshots of our models and 5348 images for training. Test sets are not combined and we provide testing performances for each dataset separately to compare our results with the previous work.

ICDAR 2013

The International Conference on Document Analysis and Recognition (ICDAR) is a flagship conference on document analysis and holds competitions on this field every other year. At 2013 ICDAR published a dataset, ICDAR 2013 table detection dataset [73], that had tables and structures annotated in them; hence, the popularity and why this dataset quickly adopted by many research and now a benchmark dataset to compare works to each other [36, 10, 7]. The dataset includes documents from the European Union (EU) and the US Government websites. Since these sources are highly regulated, we expect the tables do not differ much between them; however, it is used to simulate the real-life performance of the models. Therefore, we use this dataset to only test our results. The dataset in its original form is composed of born-digital PDF files which we convert it to images to be used in our systems. The dataset consists of 238 images when converted. Two sample images from the ICDAR 2013 dataset can be found in Figure 4.1. When evaluating this dataset, we use Intersection over Union (IoU) of 0.5 to be compatible with the previous research.

ICDAR 2017

In a later iteration of the ICDAR competitions, at 2017 POD [77] competition is held with a new table detection dataset included. This dataset also have formulas and figures annotated, which we do not consider and believe that it causes some problems, but we leave the detection of those page objects to the future work. This dataset is considerably larger than the previous counterpart and has 2417 images in it. This dataset is comprised of research papers from academic venues which are still largely regulated and has a similar structure between the instances of the data. However, the nature of the research papers considered, there are formulas and figures which can be misclassified as tables in large quantities which makes it harder to predict tables accurately as we have observed it in

Undergraduate and Graduate Borrowing: All Bachelor's Degree Recipients

Forty-five percent of bachelor's degree recipients who enrolled in graduate or first-professional degree programs took out loans to help pay for that education, borrowing an average of \$33,200 by 2003 (table 4). Borrowing a large amount as an undergraduate does not appear to

Table 4. Among 1992-93 bachelor's degree recipients with graduate degree enrollment, percentage who borrowed for graduate education and, among borrowers, average amount and percentage distribution of amount borrowed for graduate education, by student and institutional characteristics: 2003

Student and institutional characteristics	Percent who borrowed	Average amount	Amount borrowed				
			Less than \$10,000	\$10,000-14,999	\$15,000-29,999	\$30,000-54,999	\$55,000 or more
Total	44.8	\$33,200	23.2	10.3	27.0	20.1	19.4
Type of degree-granting institution							
Public 4-year	44.4	31,200	25.5	10.2	26.9	19.3	18.1
Non-doctorate-granting	40.0	25,000	27.3	11.3	32.4	18.3	10.7
Doctorate-granting	46.5	33,900	24.7	9.7	24.6	19.8	21.2
Private not-for-profit 4-year	45.9	36,600	20.3	10.3	25.7	21.8	21.8
Non-doctorate-granting	46.1	31,300	24.8	8.9	27.5	21.6	17.1
Doctorate-granting	45.7	42,300	15.5	11.8	23.8	22.1	26.8
Other	41.3	31,900	11.2	11.2	43.2	15.7	18.8
Undergraduate major							
Business and management	34.3	28,700	28.1	7.1	28.9	19.4	16.5
Education	34.3	23,000	37.4	12.5	25.5	12.9	11.8
Engineering, mathematics, or science	47.1	47,400	19.1	8.7	19.2	19.7	33.4
Humanities or social sciences	52.8	32,800	20.0	8.5	31.3	21.5	18.7
Other	47.4	28,100	21.6	14.3	28.0	22.9	13.2
Amount borrowed (undergraduate)							
Did not borrow	36.0	36,900	19.3	7.7	26.5	22.7	23.9
Less than \$5,000	45.7	30,300	29.7	9.5	28.4	17.3	15.2
\$5,000-9,999	56.3	29,200	28.0	13.0	31.0	12.9	15.1
\$10,000-14,999	54.7	28,700	25.7	16.1	23.6	19.4	15.3
\$15,000 or more	58.4	35,100	18.7	9.9	25.3	24.9	21.3
Highest enrollment after bachelor's degree by 2003							
Master's degree	37.9	19,900	30.0	13.2	33.5	18.9	4.5
Doctoral degree	63.1	43,700	18.3	7.6	21.6	21.1	31.5
First-professional degree	71.8	69,200	4.6	2.6	9.4	23.6	59.9
Highest degree earned by 2003							
Bachelor's degree	32.3	23,300	36.2	11.6	26.6	14.4	11.4
Master's degree	45.4	22,900	23.9	12.6	34.9	22.7	5.9
Doctoral degree	64.3	44,100	21.4	9.6	14.7	19.1	35.2
First-professional degree	78.8	75,500	2.0	1.1	8.2	21.6	67.1

NOTE: Graduate includes first-professional. Detail may not sum to totals because of rounding. Estimates include students from the 50 states, DC, and Puerto Rico. Standard error tables are available at <http://nces.ed.gov/ipeds/data/library/reports.asp>. SOURCE: U.S. Department of Education, National Center for Education Statistics, 1993/03 Baccalaureate and Beyond Longitudinal Study (B&B-93/03).

(which is not confined to food retailing).

Table 6.8: Franchising across the member states

	number of franchisors		number of franchisees	
	1993	1994	1993	1994
Austria	80	170	2500	2700
Belgium/Luxembourg	90	135	3200	2495
Denmark	42	42	500	500
Finland	..	..	..	..
France	500	500	30000	30000
Germany	370	420	15500	18000
Greece	..	..	..	..
Ireland	20	..	..	..
Italy	318	361	16100	17500
Netherlands	331	340	12640	12120
Portugal	55	70	..	..
Spain	117	250	14500	20000
Sweden	200	200	900	900
UK	373	396	18100	24900
EU Total	2496	2884	113940	129115

Source: table 14, p.18, "Retailing in the European Economic Area, 1996", EUROSTAT

With the growth of the hypermarket, in particular, new opportunities for scale economies and innovation have emerged. Perhaps most significant of all, is the growing use of electronic scanning at the check out.

- The diffusion of scanning has been rapid in recent years. In all member states for which data are available, its usage at least doubled between 1991 and 1994 (Table 6.9). Assuming a further acceleration post-1994, it must by now, have become a significant feature in the operations of many of Europe's leading retailers.

Not only does this technology permit a variety of internal economies, but also it provides the retailer with a rich source of detailed information about, for example, the elasticities of demand for specific brands. Undoubtedly, this has sharpened the retailer's capabilities - both in competing with its rivals and in bargaining with its suppliers, the food manufacturers.

appear to use the term consistently.

14

(a)

48

(b)

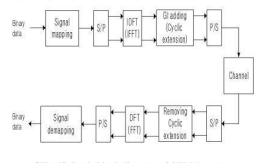
Figure 4.1: Example images from the ICDAR 2013 dataset.

[7]. The ICDAR 2017 POD dataset has 817 test images and 1600 training images pre-separated. However, we use only a partial small subset of the dataset (200 images)⁴ for testing and add the remaining 617 images to the training set. Two sample images from the ICDAR 2017 dataset can be found in Figure 4.2. In the evaluation phase, we use IoU of 0.5 to be compatible with the previous research and since it is a common threshold [96].

subchannel at time interval $[t, t+T]$, N is the number of subcarriers, and T is the OFDM symbol duration. If the k -th subcarrier from (1) is demodulated by downconverting the signal with a frequency of k/T and then integrating the signal over T seconds, the result is as written in (2)

$$\begin{aligned} & \int_t^{t+T} \exp(-j2\pi \frac{k}{T}(t-\tau)) \left(\sum_{i=0}^{\frac{N}{2}-1} d_i \exp(j2\pi \frac{i}{T}(t-\tau)) \right) d\tau \\ &= \sum_{i=0}^{\frac{N}{2}-1} d_i \int_t^{t+T} \exp(j2\pi \frac{i-k}{T}(t-\tau)) d\tau \\ &= d_i \cdot \frac{N}{2} \quad (2) \end{aligned}$$

By looking at the intermediate result of (2), it can be seen that a complex carrier is integrated over T seconds. When i is equal to k , in other words, when received OFDM signal is demodulated i -th sub-carrier, this integration gives the desired output $d_i \cdot \frac{N}{2}$, which is the QAM value for that particular subcarrier. For all other subcarriers, the integration is zero, because the frequency difference $(i-k)/T$ produces an integer number of cycles within the integration interval T , such that the integration result is always zero. In the previous concept of OFDM is analog version. The discrete version of OFDM is based on IDFT(Tx) and DFT(Rx) algorithm. And IFFT and FFT are the fast algorithm pair of IDFT and DFT.



[Fig. 2] Basic block diagram of OFDM system

This is the mathematical process of OFDM system with IDFT and DFT. The IDFT output sequence of a OFDM symbol is

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp(j2\pi k n / N) \quad (n=0,1,2,\dots,N-1)$$

$$\text{The IDFT output sequence of infinite OFDM symbol is} \\ x_{n,i} = \sum_{k=0}^{N-1} \frac{1}{N} \sum_{m=0}^{N-1} X_k \exp(j2\pi k m i / N) \quad (4)$$

where X_k is the data mapped onto QPSK or QAM. The DFT output sequence of a received OFDM symbol is

$$\begin{aligned} Y_i &= \sum_{k=0}^{N-1} x_k \exp(-j2\pi k i / N) \quad (i=0,1,2,\dots,N-1) \\ &= \sum_{k=0}^{N-1} \left(\frac{1}{N} \sum_{m=0}^{N-1} X_k \exp(j2\pi k m i / N) \right) \exp(-j2\pi k i / N) \\ &= \sum_{k=0}^{N-1} \left(\frac{1}{N} \sum_{m=0}^{N-1} X_k \right) \exp(j2\pi k (i-m) / N) \end{aligned} \quad (5)$$

i) if $i=k$,

$$\begin{aligned} Y_i &= \sum_{k=0}^{N-1} \left(\frac{1}{N} \sum_{m=0}^{N-1} X_k \right) \exp(j2\pi k (i-m) / N) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X_k = X_i \end{aligned}$$

ii) if $i \neq k$, by a geometric sequence $Y_i = 0$

3. Wideband Multipath channel modeling
The wideband multipath channel is generally modeled as a tapped delay linear filter. That is shown in eq. (6) and Figure 3.

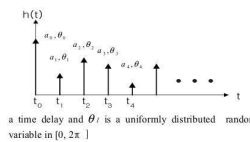
$$h(t) = \sum_{l=0}^{L-1} a_l \delta(t - \tau_l) e^{j\theta_l} \quad (6)$$

where a_l is path strength, τ_l is time delay, and θ_l is phase shift.

In this paper, we use the JTC realistic channel model for simulation. The channel impulse response of the JTC channel model is [3][4]

$$h(t) = \sum_{l=0}^{L-1} \sqrt{p_l} \alpha \delta(t - \tau_l) e^{j\theta_l} \quad (7)$$

where p_l is the amplitude of l -th component in the JTC power delay profile, α is a Rayleigh-distributed random variable, $\sqrt{p_l} \alpha$ is the signal amplitude of l -th path, τ_l is



[Fig. 3] Linear filter model

the growth of existing loops via the capture of interstitial clusters.

In the case of MgO, because we are interested in room temperature, a temperature at which vacancies are immobile, we have included no mobility for vacancies and therefore no clustering mechanisms either. All vacancies are assumed to be isolated vacancies here. We have taken $E_v = 0$, as tetra-interstitials are also immobile at this temperature on experimental time scales. In addition, while we have not explicitly calculated any properties for the penta-interstitial, we assume here that it behaves like the tetra-interstitial and that $E_v = 0$. The non-zero rates were calculated using $E_v = v_k \exp(-E_{m,k}/k_B T)$, where $E_{m,k}$ is the migration energy for clusters of size n . All rate prefactors v_k were set to $10^{13}/s$. For simplicity, we have also assumed that $Z_1 = Z_2 = Z_3 = 100$ atomic sites. We have assumed an initial production rate $P = 10^{-4}/s$. These values have not been carefully tuned to the properties of MgO, but at this point we are more interested in qualitative effects than quantitative results. Finally, we have not made any distinction between O and Mg in this model. That is, we are only modeling a single concentration of interstitials and vacancies. This seems reasonable as the results presented previously indicate that non-stoichiometric defects of differing types tend to behave very similarly. The diffusion barrier for isolated O interstitials is similar to that of Mg interstitials and here we have taken the average value, 0.36eV, as the migration barrier for single interstitials. The same was done for tri-interstitials of both types: Mg-O-Mg and O-Mg-O. A is a parameter that is on the order of unity and is taken to be unity here. Finally, we consider clusters up to size $n_{max} = 6$. These values are summarized in Table 2.

We find that most steady-state concentrations are unaffected by the presence of mobile hexa-interstitials. For example, the concentrations of vacancies C_v and interstitials in loops C_{il} remain unchanged by including mobile hexa-interstitials. The time to reach steady-state is increased significantly when mobile hexa-interstitials are introduced into the system. The time for the vacancy concentration to reach steady-state at 300K, for example, increases by over a factor of 3.

Table 2
Summary of the parameters used in the chemical rate theory model

Parameter	Value
P	$10^{-4}/s$
v_k	$10^{13}/s$
$E_{m,k}$	0.36eV
$E_{m,k}$	0.75eV
$E_{m,k}$	0.8eV
$E_{m,k}$	0.24eV
E_v, E_i	0
n_{max}	6
Z_1, Z_2, Z_3	100
A	1

Where we do see a large impact of the mobility of the hexa-interstitial is in the concentration of loops C_{il} themselves and, correspondingly, their average size, estimated as the ratio C_{il}/C_l and plotted in Fig. 12. By 300K, the average size of interstitial dislocation loops is over an order of magnitude larger when mobile hexa-interstitials are included. Measuring the size of interstitial dislocation loops as a function of temperature may be an indirect way to experimentally probe the existence of large mobile clusters in MgO. We expect there may be orientation or shape effects as well, arising from the one-dimensional diffusion of the hexa-interstitial cluster, but exploring such effects will require more sophisticated models.

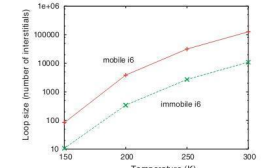


Fig. 12. Average size of interstitial dislocation loops in MgO, given as the number of interstitials in the loops, as a function of temperature for chemical rate theory models that do and do not include mobile metastable hexa-interstitials.

Figure 4.2: Example images from the ICDAR 2017 dataset.

⁴Our Faster-RCNN and RetinaNet models are tested on the full test set and trained only on the original training partition.

Marmot

One of the big public datasets on table detection is the Marmot dataset published by the Institute of Computer Science and Technology (Peking University). There are 2000 images with about half of them containing table regions. The annotations in this dataset are not clear and have some errors as stated by Schreiber et al. [10] as well. Moreover, since we aim for tabular regions instead of more strict table areas, we re-annotated some and did not use most of the dataset. In the end, we only use 829 data points from this dataset to be included only in the training set. Two sample images from the Marmot dataset can be found in Figure 4.3.

UNLV

The UNLV dataset is published by the University of Nevada, Las Vegas and consists of 2889 documents from sources like reports, business letters, magazines and so on. Only 424 of those data points included table regions; therefore, we can only use those. The UNLV dataset consists of scanned document images with possible tilt introduced by the scanning process which, we believe, introduces extra difficulty to the training since our end goal is table detection in born-digital documents and all of our other datasets are composed of born-digital documents. In this work, this dataset is not included in our Faster-RCNN and RetinaNet training sets because scanned documents are not our primary focus. Hence, for accurate detection and smooth training, we exclude some of the heavily tilted images that left us with usable 400 images to be used in the Mask-RCNN model. We select random 50 images to be used in the testing since the UNLV dataset does not have a defined test and training split. The importance of the UNLV dataset comes from the annotation which has both table and structure annotations. Two sample images from the UNLV dataset can be found in Figure 4.4.

Private

Our private dataset is provided to us by Lytica Inc. in the form of born-digital electronic component datasheets aggregated from many manufacturers. The dataset comprised initially of millions of PDF datasheets which then we converted them to document images. Since the range of sources are diverse, and the style of tables are different than observed in public datasets we also hand annotated 2323 document images from this dataset and used 2123 of them in training and the rest in the testing set. As it can be witnessed in Figure 4.5, our private dataset includes complicated documents where many figures have

衡的角度考虑，应做以下努力：

(1) 为了缩短冶炼周期，必须提高吨钢输入电功率。输入功率 $P_{\text{入}}$ 与变压器额定功率有关，变压器额定功率大，可输入电弧炉的电功率大， $P_{\text{入}}$ 也大。通常将吨钢变压器额定功率大于 $700\text{kV} \cdot \text{A}$ 的电弧炉称为超高功率电弧炉，20 世纪 80 年代末，超高功率电弧炉的变压器额定功率高达 $800 \sim 1200\text{kV} \cdot \text{A}/\text{t}$ 。

(2) 提高由化学热换算成的电功率 $P_{\text{化}}$ ，有利于缩短冶炼周期。 $P_{\text{化}}$ 主要来自钢中元素氧化的化学热及氧燃烧嘴提供的化学热。元素（主要是碳）氧化的化学热可占炼钢所需总能量的 30% 左右，当二次燃烧率高时，这一比例还会更大些。氧燃烧嘴提供的化学热一般占总能量的 5% ~ 10%。

(3) 提高由物理热换算成的电功率 $P_{\text{物}}$ ，有利于缩短冶炼周期。 $P_{\text{物}}$ 主要是由废钢预热后废钢的显热及加铁水操作时铁水的显热提供的。

(4) 提高电效率 η 、功率因数 $\cos\varphi$ ，可以缩短冶炼周期。提高 $\eta\cos\varphi$ 的措施包括供电制度合理化（优化供电曲线）、人工智能电弧炉、采用导电横臂、优化短网结构等。

(5) 增加化学热和物理热，可以收到节电效果。理论计算和实践结果表明，目前每增加 1m^3 氧，可以节约 $4\text{kW} \cdot \text{h}/\text{t}$ ；每增加 1% 铁水量，可以节约 $4 \sim 5\text{kW} \cdot \text{h}/\text{t}$ 。

(6) 增加化学热和物理热，在通电时间 τ_1 、钢液重量 W 、冶炼电耗 C 、 $\eta\cos\varphi$ 不变的条件下，可以减小变压器额定功率；在 τ_1 、 C 、 $P_{\text{入}}$ 、 $\eta\cos\varphi$ 不变的条件下，可以增加 W ，即提高生产率，提高炉子的年产量。

(7) 为缩短冶炼周期，必须缩短热停工时间 τ_2 。由于在化学热和物理热供量较大的条件下，例如在供氧强度较大的竖炉冶炼过程和加铁水的 Consteel 冶炼过程中，可以有一段停电操作，这时 $P_{\text{入}}$ 和 $P_{\text{化}}$ 不等于 0，所以 τ_2 称为较合适热停工时间（在过去的文献中均称为非通电时间）。它是指装废钢时间、加铁水时间（对于顶装铁水而言）、接电极时间、出钢时间、补炉时间等。

由于一系列电弧炉冶炼新技术的开发应用，电弧炉冶炼周期已从 1965 年的平均 180min 缩短到目前的 60min 以下，最短可达 35min，相应的电耗从原来的 $630\text{kW} \cdot \text{h}/\text{t}$ 降低到目前最低可达 $200\text{kW} \cdot \text{h}/\text{t}$ ，电极消耗从原来的 $6.5\text{kg}/\text{t}$ 降低到目前最低可达 $1\text{kg}/\text{t}$ 。

现代电弧炉冶炼技术的发展随时代进步。20 世纪 60 ~ 70 年代主要是发展超高功率供电及其相关技术，包括高压长弧操作、水冷炉壁、水冷炉盖、泡沫渣技术等，在这一阶段钢包精炼及强化用氧已开始采用。80 年代，LF 及 EBT 技术的开发使电弧炉冶炼加炉外精炼的现代电弧炉炼钢流程基本成熟。80 年代末，大型超高功率直流电弧炉问世，由于其对电网冲击小、石墨电极消耗低，以致 80 年代末到 90 年代初直流电弧炉占了绝对的优势，我国上海的几台 100 ~ 150t 的直流电弧炉就是在这一背景下建立起来的。与此同时，强化用氧技术（包括超音速氧枪、碳氧枪、氧燃烧嘴、底风口、二次燃烧技术）趋于成熟，VAI 公司称之为 K-ES 技术，Danieli 公司称之为 Danarc 技术，Demag 公司称之为 Kor-fare 技术。大量化学能源的输入，使得冶炼周期大大缩短，电耗和电极消耗不断降低，加上高阻抗交流电弧炉对电网的冲击比普通交流电弧炉低 50%，导致 90 年代中期以来，交流电弧炉与直流电弧炉形成竞争发展的趋势。值得注意的是，在这一阶段人们关注的已不是用直流供电还是用交流供电，而是废钢预热，即二次燃烧和烟气显热的利用问题，不同的废钢预热方式产生了不同类型的现代电弧炉，它们分别是用料篮废钢预热的普通电弧

(a)

表 2-8 行为科学理论主要内容

人际管理理论	工人是“社会人”而不是“经济人”；工人的工作态度与士气是影响工作效率的关键；企业中存在着“非正式组织”
激励理论	(1) 需求层次理论：动机产生行为，需要产生动机；生理-安全-社会-尊重-自我实现需要 (2) 人性假设理论：任何组织的管理者在其下属时，对其下属所持基本看法为“管理的假定”。有麦克斯韦格假定(X理论)、超Y理论、Z理论和不成熟-成熟理论 (3) 群体行为理论：团体力学理论，主要论述了非正式团体的要素、目标、凝聚力、规范、结构、领导方式、规模、行为分类、参与者、组织对变动的反应等 (4) 敏感性训练理论 (5) 强化理论 (6) 期望理论
领导效能理论	领导者追随理论、行为连续带理论、领导方式系统理论、领导四象限模式、管理方格论、领导行为权变理论

表 2-9 现代管理理论主要内容

社会系统学派	组织是一个协作的系统；非正式组织是不属于正式组织的，是不受正式组织管理的个人联系和相互作用以及有关群体的总和；任何正式组织都包括协作意愿、共同目标和信息沟通；管理者在组织中的作用和职能是建立和维护一个信息联系的系统，协调组织成员的行动，规定、解释目标并处理个人目标与组织目标的矛盾
决策理论学派	管理就是决策，决策过程包括搜集情报阶段、拟定计划、选定计划阶段；决策的准则是制定出满意的标准，达到或超过这个标准就是可行的方案；程序化与非程序化决策；决策制约着组织机构的设置，决策的技术和方法
系统管理学派	系统管理的内容包括系统哲学、系统管理和系统分析。系统是一种有组织的综合整体，企业作为一个系统进行设计和经营，按照系统的要求进行组织和运行，确定一个最优的解决方法或实际行动
经验主义学派	重点分析成功管理者实际管理的经验，并加以概括，总结出成功经验共性，使之系统化、合理化，提出建议
管理科学学派	制定和运用数学模型与程序系统来表示计划、组织和控制活动，通过定量分析，选择最优方案达到组织的目标。特征是将数学、计算机和系统观念引入管理方法中，其主要方法包括线性规划法、决策树法、计划评审及关键路线法、模拟、概率论、对策论、排队论、库存论和搜索论
权变理论学派	“如果—就要”关系理论是权变理论的思想基础，主要集中于领导效能理论和组织结构的设计

三、管理学理论应用于水资源管理

将管理学基本理论应用于水资源管理，具有广泛的前景。主要表现在以下几个方面。

1. 以人为本，挖掘水资源管理者潜力

人是水资源管理制定者和执行者，水资源管理活动必须以人为基本前提，做好水资源管理者的工作管理。为此，必须充分认识人的个体差异，采用多种手段发挥水资源管理者潜能。在分配工作的时候，要充分注意到每个人的工作与其能力相“匹配”，达到既发挥其潜力，同时让水资源管理者有信心完成本职工作。在实际工作中，充分发挥水资源管理的积极参与热情，建立与之相配套的组织文化；对每个水资源管理者建

Figure 4.3: Example images from the Marmot dataset.

Line	Compass Bearing	(β)	Rise Angle (δ) (Positive if Above Horizontal)
x'	S 50° E	-40°	55°
y'	N 50° W	140°	35°
z'	S 40° W	-130°	0

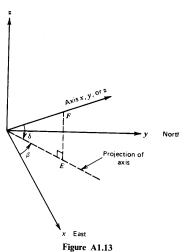


Figure A1.13

024

Stratigraphic and rock description ¹	Depth meters (feet)	Thickness of interval, meters (feet)
Paintbrush Tuff-Continued		
Tiva Canyon Member--Continued Tuff, ash-flow, pale-yellowish-brown, well compacted but essentially nonwelded, devitrified; pumice brownish to yellowish-gray, as much as 4 cm (1.6 in.) across, but mostly <0.5 cm. Phenocrysts, 3 percent, mainly sandstone and biotite. Basal contact is irregular scoured surface	757-760 (2,484-2,494)	3 (10)
Begged tuff		
Tuff, mostly massive, pale-reddish-brown to yellowish-gray and moderate-orange-pink, well indurated; small (<2 cm) pumice, sparse dark-reddish-brown volcanic lithic fragments, mostly <4 mm across. Maximum dip of weak stratification near top is 30°. Small high-angle faults ² at 760.5 m (2,495 ft) have offset of less than 3 cm. Contact at base dips 35°	760-774 (2,494-2,540)	14 (46)
Tuff, massive to reworked, crudely bedded in places, pale-reddish-brown, small (<2 cm) grayish-brown pumice; partly silicified	774-775 (2,540-2,543.5)	1 (3.5)
Begged tuff		
Tuff, ash-flow(?), massive, nonwelded, yellowish-gray, zeolitic; pumice, grayish-yellow to olive-gray, altered. Phenocrysts, 3-5 percent, quartz, sandstone, and biotite. Numerous dark-reddish-brown volcanic lithic fragments 0.5-3 cm in diameter	775-777 (2,543.5-2,549.5)	2 (6)
Tuff, bedded, reworked, reddish-brown to dark-yellowish-brown. Phenocrysts 8 percent, biotite and sandstone. Dip 20°. Basal contact is fault dipping 10°	777-779.2 (2,549.5-2,550)	0.2 (0.5)

Figure 4.4: Example images from the UNLV dataset.

table-shapes, and there are some cases where tables are located inside rows of tables. These characteristics are hard to encounter in public datasets; therefore, our primary aim is to achieve accurate information extraction from these documents in the end.

4.2 Methods

To solve the table detection from both born-digital and scanned document problems, we propose and compare three deep learning methods to each other where each of them accompanied by distinct architectures and features that lead us to a final decision on which architecture to choose. Our work follows a step-by-step and incremental approach in the selection of the methods. We build up to our latest model and clarify the reasons why that model performs the best. In this section, we first start with describing the Faster-RCNN [12] model under study, and then we test the RetinaNet [13] based method. The final object detection model we use to detect tables is based on Mask-RCNN [11] and all these models are compared with each other as well as to the literature. Architectural details of all the methods that are described in this section are given in Chapter 3.

Faster-RCNN [12] is a state-of-the-art object detection model that is built on top of an earlier iteration called R-CNN [20]. Faster-RCNN is a deep learning-based two-stage object detection method. It uses a subnetwork called RPN to propose bounding boxes and then classify each box separately. In the first stage, we set the threshold for the bounding boxes to be 2000. These 2000 boxes are classified and regressed to form the final outputs together. Due to the two-stage nature of the Faster-RCNN, we expect it to be slower than single-stage networks even though it is one of the fastest state-of-the-art object detection methods. The reasoning behind the choice of Faster-RCNN stems from the popularity of the R-CNN and its iterations including Faster-RCNN in computer vision research [101]. Faster-RCNN has been used in many different domains for instance, to detect faces [112], medical chart interpretation [113, 114], table detection as well [10, 7, 9] and many other abstract object detection challenges as shown in [115]. Since the model has been used in a wide range of domains, we trust the transfer learning capabilities of the model and use the datasets mentioned above to adapt the pre-trained backbone networks to our domain. As the backbone feature extractor, we use VGG-16 which has been proven to perform well on established datasets such as ImageNet and PASCAL VOC [92, 96]. VGG-16 is a convolutional neural network with 13 convolution layers as outlined in Figure 3.1. We use the latest convolution output to be used as a feature map in Faster-RCNN model (Figure 3.7). We chose the VGG-16 network to see if the task is easy enough and we do not need an overly complex network to solve it and to benefit from the speed advantages

COMMERCIAL PRODUCTS										
SOLID TANTALUM CAPACITORS - CONFORMAL COATED										
SERIES	500V	500V	500V	500V	500V	500V	500V	500V	500V	
PRODUCT IMAGE										
TYPE	Surface mount	Through-hole	Through-hole	Through-hole	Through-hole	Through-hole	Through-hole	Through-hole	Through-hole	
FEATURES	Low profile, conformal coated for surface mount	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	Low profile, conformal coated for through-hole	
TEMPERATURE RANGE	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	-55 °C to +125 °C	
VOLTAGE RANGE	330V to 250V (1)	470V to 250V	470V to 250V	470V to 250V	470V to 250V	470V to 250V	470V to 250V	470V to 250V	470V to 250V	
CAPACITANCE RANGE	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	0.1 µF to 100 µF	
VOLTAGE SURGE	6V to 15V	6V to 15V	6V to 15V	6V to 15V	6V to 15V	6V to 15V	6V to 15V	6V to 15V	6V to 15V	
ESR	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	≤ 25 mΩ	
CURRENT	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	0.01 C or 0.5 A, whichever is greater	
FACTORY TEST	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	4 kV to 50 kV	
CASE CODES	S, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS, AT, AU, AV, AW, AX, AY, AZ, BA, BB, BC, BD, BE, BF, BG, BH, BI, BJ, BK, BL, BM, BN, BO, BP, BQ, BR, BS, BT, BU, BV, BW, BX, BY, BZ, CA, CB, CC, CD, CE, CF, CG, CH, CI, CJ, CK, CL, CM, CN, CO, CP, CQ, CR, CS, CT, CU, CV, CW, CX, CY, CZ, DA, DB, DC, DD, DE, DF, DG, DH, DI, DJ, DK, DL, DM, DN, DO, DP, DQ, DR, DS, DT, DU, DV, DW, DX, DY, DZ, EA, EB, EC, ED, EE, EF, EG, EH, EI, EJ, EK, EL, EM, EN, EO, EP, EQ, ER, ES, ET, EU, EV, EW, EX, EY, EZ, FA, FB, FC, FD, FE, FF, FG, FH, FI, FJ, FK, FL, FM, FN, FO, FP, FQ, FR, FS, FT, FU, FV, FW, FX, FY, FZ, GA, GB, GC, GD, GE, GF, GH, GI, GJ, GK, GL, GM, GN, GO, GP, GQ, GR, GS, GT, GU, GV, GW, GX, GY, GZ, HA, HB, HC, HD, HE, HF, HG, HH, HI, HJ, HK, HL, HM, HN, HO, HP, HQ, HR, HS, HT, HU, HV, HW, HX, HY, HZ, IA, IB, IC, ID, IE, IF, IG, IH, II, IJ, IK, IL, IM, IN, IO, IP, IQ, IR, IS, IT, IU, IV, IW, IX, IY, IZ, JA, JB, JC, JD, JE, JF, JG, JH, JI, JJ, JK, JL, JM, JN, JO, JP, JQ, JR, JS, JT, JU, JV, JW, JX, JY, JZ, KA, KB, KC, KD, KE, KF, KG, KH, KI, KJ, KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT, KU, KV, KW, KX, KY, KZ, LA, LB, LC, LD, LE, LF, LG, LH, LI, LJ, LK, LL, LM, LN, LO, LP, LQ, LR, LS, LT, LU, LV, LW, LX, LY, LZ, MA, MB, MC, MD, ME, MF, MG, MH, MI, MJ, MK, ML, MM, MN, MO, MP, MQ, MR, MS, MT, MU, MV, MW, MX, MY, MZ, NA, NB, NC, ND, NE, NF, NG, NH, NI, NJ, NK, NL, NM, NN, NO, NP, NQ, NR, NS, NT, NU, NV, NW, NX, NY, NZ, OA, OB, OC, OD, OE, OF, OG, OH, OI, OJ, OK, OL, OM, ON, OO, OP, OQ, OR, OS, OT, OU, OV, OW, OX, OY, OZ, PA, PB, PC, PD, PE, PF, PG, PH, PI, PJ, PK, PL, PM, PN, PO, PP, PQ, PR, PS, PT, PU, PV, PW, PX, PY, PZ, QA, QB, QC, QD, QE, QF, QG, QH, QI, QJ, QK, QL, QM, QN, QO, QP, QQ, QR, QS, QT, QU, QV, QW, QX, QY, QZ, RA, RB, RC, RD, RE, RF, RG, RH, RI, RJ, RK, RL, RM, RN, RO, RP, RQ, RR, RS, RT, RU, RV, RW, RX, RY, RZ, SA, SB, SC, SD, SE, SF, SG, SH, SI, SJ, SK, SL, SM, SN, SO, SP, SQ, SR, SS, ST, SU, SV, SW, SX, SY, SZ, TA, TB, TC, TD, TE, TF, TG, TH, TI, TJ, TK, TL, TM, TN, TO, TP, TQ, TR, TS, TT, TU, TV, TW, TX, TY, TZ, UA, UB, UC, UD, UE, UF, UG, UH, UI, UJ, UK, UL, UM, UN, UO, UP, UQ, UR, US, UT, UY, UV, UW, UX, UY, UZ, VA, VB, VC, VD, VE, VF, VG, VH, VI, VJ, VK, VL, VM, VN, VO, VP, VQ, VR, VS, VT, VU, VV, VW, VX, VY, VZ, WA, WB, WC, WD, WE, WF, WG, WH, WI, WJ, WK, WL, WM, WN, WO, WP, WQ, WR, WS, WT, WU, WV, WW, WX, WY, WZ, XA, XB, XC, XD, XE, XF, XG, XH, XI, XJ, XK, XL, XM, XN, XO, XP, XQ, XR, XS, XT, XU, XV, XW, XX, XY, XZ, YA, YB, YC, YD, YE, YF, YG, YH, YI, YJ, YK, YL, YM, YN, YO, YP, YQ, YR, YS, YT, YU, YV, YW, YX, YY, YZ, ZA, ZB, ZC, ZD, ZE, ZF, ZG, ZH, ZI, ZJ, ZK, ZL, ZM, ZN, ZO, ZP, ZQ, ZR, ZS, ZT, ZU, ZV, ZW, ZX, ZY, ZZ	S, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS, AT, AU, AV, AW, AX, AY, AZ, BA, BB, BC, BD, BE, BF, BG, BH, BI, BJ, BK, BL, BM, BN, BO, BP, BQ, BR, BS, BT, BU, BV, BW, BX, BY, BZ, CA, CB, CC, CD, CE, CF, CG, CH, CI, CJ, CK, CL, CM, CN, CO, CP, CQ, CR, CS, CT, CU, CV, CW, CX, CY, CZ, DA, DB, DC, DD, DE, DF, DG, DH, DI, DJ, DK, DL, DM, DN, DO, DP, DQ, DR, DS, DT, DU, DV, DW, DX, DY, DZ, EA, EB, EC, ED, EE, EF, EG, EH, EI, EJ, EK, EL, EM, EN, EO, EP, EQ, ER, ES, ET, EU, EV, EW, EX, EY, EZ, FA, FB, FC, FD, FE, FF, FG, FH, FI, FJ, FK, FL, FM, FN, FO, FP, FQ, FR, FS, FT, FU, FV, FW, FX, FY, FZ, GA, GB, GC, GD, GE, GF, GH, GI, GJ, GK, GL, GM, GN, GO, GP, GQ, GR, GS, GT, GU, GV, GW, GX, GY, GZ, HA, HB, HC, HD, HE, HF, HG, HH, HI, HJ, HK, HL, HM, HN, HO, HP, HQ, HR, HS, HT, HU, HV, HW, HX, HY, HZ, IA, IB, IC, ID, IE, IF, IG, IH, II, IJ, IK, IL, IM, IN, IO, IP, IQ, IR, IS, IT, IU, IV, IW, IX, IY, IZ, JA, JB, JC, JD, JE, JF, JG, JH, JI, JJ, JK, JL, JM, JN, JO, JP, JQ, JR, JS, JT, JU, JV, JW, JX, JY, JZ, KA, KB, KC, KD, KE, KF, KG, KH, KI, KJ, KK, KL, KM, KN, KO, KP, KQ, KR, KS, KT, KU, KV, KW, KX, KY, KZ, LA, LB, LC, LD, LE, LF, LG, LH, LI, LJ, LK, LL, LM, LN, LO, LP, LQ, LR, LS, LT, LU, LV, LW, LX, LY, LZ, MA, MB, MC, MD, ME, MF, MG, MH, MI, MJ, MK, ML, MM, MN, MO, MP, MQ, MR, MS, MT, MU, MV, MW, MX, MY, MZ, NA, NB, NC, ND, NE, NF, NG, NH, NI, NJ, NK, NL, NM, NN, NO, NP, NQ, NR, NS, NT, NU, NV, NW, NX, NY, NZ, OA, OB, OC, OD, OE, OF, OG, OH, OI, OJ, OK, OL, OM, ON, OO, OP, OQ, OR, OS, OT, OU, OV, OW, OX, OY, OZ, PA, PB, PC, PD, PE, PF, PG, PH, PI, PJ, PK, PL, PM, PN, PO, PP, PQ, PR, PS, PT, PU, PV, PW, PX, PY, PZ, QA, QB, QC, QD, QE, QF, QG, QH, QI, QJ, QK, QL, QM, QN, QO, QP, QQ, QR, QS, QT, QU, QV, QW, QX, QY, QZ, RA, RB, RC, RD, RE, RF, RG, RH, RI, RJ, RK, RL, RM, RN, RO, RP, RQ, RR, RS, RT, RU, RV, RW, RX, RY, RZ, SA, SB, SC, SD, SE, SF, SG, SH, SI, SJ, SK, SL, SM, SN, SO, SP, SQ, SR, SS, ST, SU, SV, SW, SX, SY, SZ, TA, TB, TC, TD, TE, TF, TG, TH, TI, TJ, TK, TL, TM, TN, TO, TP, TQ, TR, TS, TT, TU, TV, TW, TX, TY, TZ, UA, UB, UC, UD, UE, UF, UG, UH, UI, UJ, UK, UL, UM, UN, UO, UP, UQ, UR, US, UT, UY, UV, UW, UX, UY, UZ, VA, VB, VC, VD, VE, VF, VG, VH, VI, VJ, VK, VL, VM, VN, VO, VP, VQ, VR, VS, VT, VU, VV, VW, VX, VY, VZ, WA, WB, WC, WD, WE, WF, WG, WH, WI, WJ, WK, WL, WM, WN, WO, WP, WQ, WR, WS, WT, WU, WV, WW, WX, WY, WZ, XA, XB, XC, XD, XE, XF, XG, XH, XI, XJ, XK, XL, XM, XN, XO, XP, XQ, XR, XS, XT, XU, XV, XW, XX, XY, XZ, YA, YB, YC, YD, YE, YF, YG, YH, YI, YJ, YK, YL, YM, YN, YO, YP, YQ, YR, YS, YT, YU, YV, YW, YX, YY, YZ, ZA, ZB, ZC, ZD, ZE, ZF, ZG, ZH, ZI, ZJ, ZK, ZL, ZM, ZN, ZO, ZP, ZQ, ZR, ZS, ZT, ZU, ZV, ZW, ZX, ZY, ZZ								

Revision: 12-Sep-17

3

For technical questions, contact: tantalum@nslady.com

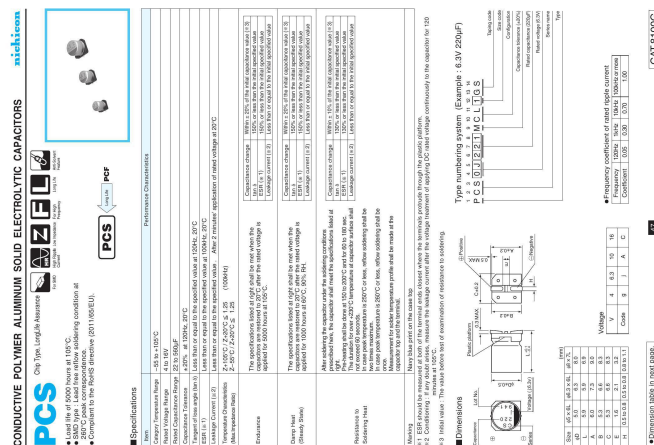
THIS DOCUMENT IS SUBJECT TO SPECIFIC WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO CHANGE WITHOUT NOTICE. THE FOLLOWING DISCLAIMERS SET FORTH AT: www.vishay.com/doc/91006

Document Number: 40150

[illegible]

Revision: 14-Sep-17
 For technical questions within your region: DoDnetAmerica@vishay.com, DoDnetEurope@vishay.com
 THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT
 ARE SUBJECT TO SPECIFIC DISCLAIMERS. SET FORTH AT: www.vishay.com/doc/7911000
 Document Number: 946588

Figure 4.5: Example images from our private dataset.



of the VGG-16 network.

Our second model is RetinaNet (Figure 3.11) which is a single-stage object detector proposed by Lin et al. [13]. RetinaNet is another method that is influenced by the incremental breakthroughs by the R-CNN family. Instead of the first stage of R-CNN and Faster-RCNN, 'RPN', RetinaNet model borrows the idea of anchors from the Faster-RCNN and eliminates the candidate bounding box selection phase. Instead, each anchor for each class are classified and regressed, and then a novel *focal loss* (Eq. (3.28)) is applied to solve a problem called class imbalance problem. The class imbalance problem is caused by the lack of a subnetwork like RPN in single-stage detectors. RPN reduces the number of possible locations for a box to a couple thousand which eliminates the imbalance between classes greatly, whereas RetinaNet deals with possibly hundreds of thousands of possible boxes. We use the backbone networks of ResNet-50 [14] and ResNeXt-101 [85] with FPN [15] included in both of the backbone networks to strengthen the stance of our single-stage detector compared to the Faster-RCNN model with VGG-16 backbone network. We know that single-stage detectors lack in performance (recall and precision), but they have a trade-off and gain a speed advantage over the two-stage networks [13]. RetinaNet is also used widely in a range of topics; including table detection as we have shown in our previous work [7] or road damage detection [116]. We also test ResNet-101, but the performance was in between the two backbones we are using. Hence, we do not include it when presenting our results.

The third and final table detection method is based on a Mask-RCNN [11] model. Mask-RCNN method is merely an addition on top of the already successful Faster-RCNN model. It is a two-stage object detection network where the first stage is same as Faster-RCNN, with an RPN and the second stage is an extended version of the Faster-RCNN method. This extension is an FCN branch which was not present in the original Faster-RCNN model. The new FCN branch, allowed the Mask-RCNN model to predict bounding boxes as usual and become an instance segmentation model in parallel by predicting pixel-level classification. Pixel-level classification is done by deconvolutions where the features in an RoI is upsampled with learned convolutional layers to represent the image and classify each pixel accurately. Although the model has a new branch, the runtime is almost the same as Faster-RCNN because new FCN branch is operating in parallel to the Faster-RCNN branches (classification and regression). Mask-RCNN architecture can be seen in Figure 3.8. RPN in our experiments proposes 2000 areas (RoI) and we have also attempted to set the threshold at 3000 and 4000 proposals, but we did not see a contribution. This phenomenon is also observed in [12]. The authors have observed that the lower number of proposals prevents the class imbalance problem from occurring and large values hurt the performance instead of improving it further. In the Mask-RCNN based model, we use

large input images (maximum size of 2000x1200) and hence, due to the video memory limitations, our mini-batch size is limited to 1. Every deep learning model we make use of includes BatchNorm [18] which is used to solve the internal covariate shift problem by regularizing each mini-batch. Naturally, if the mini-batch includes only one image, there is nothing to regularize for. Hence, BatchNorm becomes ineffective in this case. However, Yuxin et al. (2018) [17] proposed Group Normalization (GroupNorm) which operates by regularizing the channels of the input. The channels of the input to a layer are divided into a fixed number of groups (we use *GroupNumber* = 32) to be normalized. Yuxin et al. [17] demonstrated that GroupNorm achieves higher precision than BatchNorm under training with small mini-batch sizes. Mask-RCNN is a state-of-the-art method both on instance segmentation, on object detection problems and keypoint detection problem. Therefore, it is widely used and became one of the de-facto models to use in various computer vision problems including nucleus segmentation [117], human pose estimation [118] and augmenting object detection with instance segmentation [119]. From our previous experience on Faster-RCNN and RetinaNet and their backbone networks which represents a wide range of choices, we decide on ResNet-50 [14] as our backbone in this network instead of VGG-16. This decision is based on our results we observe in Section 4.4: large ResNet networks seem to attain better recall values. We also designed additional improvements on top of the detections of the Mask-RCNN based table detection model, but they are focused on in Chapter 6.

4.3 Training Details

Our implementation language is Python with deep learning libraries; Tensorflow and PyTorch [120, 121]. Since our dataset size (5519) is not enough to train a deep neural network from scratch, we benefit from transfer learning capabilities of these networks since it is shown that deep learning solutions can transfer the knowledge to the domain of table detection from documents [7, 9, 10]. The transfer learning is applied to the regression, classification and FCN branches of our models as well as to the pre-trained VGG-16, ResNet and ResNeXt backbone feature extractors that are trained on ImageNet [92] and MS COCO [100]. We do not apply any preprocessing or postprocessing that might help improve the performance since our primary focus is to find a suitable combination of architectural decision that works well on this task and leave the rest to future work. All the training is carried out with the initial learning rate of 0.001 and we gradually decrease the learning rate throughout the training. Different learning rates as well as other simple parameters such as FPN connection points are experimented but a more broad search

(grid-search like approach) should be tried in the future work. Our training dataset (5348 images) is the merged version of all the datasets mentioned and is shuffled before training to prevent the networks to overfit to dataset specific structures, if there are any. When shuffled, every mini-batch is possibly fetched from a different dataset.

We utilize a Faster-RCNN model as described in [12]. Faster-RCNN model is powered with a pre-trained backbone feature extractor, VGG-16. Input images to the network are resized to make their maximum possible length of an edge to 1000 and their shortest side to 600 (1000×600 max. resolution). Faster-RCNN model is trained for eight epochs with the batch size of 1 and 128 RoIs per batch. During training, we constantly monitored the validation loss and cut the training when the validation loss stopped improving to prevent overfitting problem. The initial learning rate of the model is 0.001, but we reduce it by a factor of 10 after 30000 steps.

The RetinaNet architecture is described in [13]. We use the described architecture as is with our two backbone networks to experiment with. Both the backbone networks include an FPN attached to them. FPN is connected to the outputs of every convolutional block of the underlying feature extractors. The input size is the same as our Faster-RCNN model with a maximum possible length of a side is 1000 where their shortest side is 600 (1000×600 max. resolution). The RetinaNet network with ResNet-50 backbone network is trained with a mini-batch size of 6 and 64 RoIs per batch for nine epochs, and RetinNet model with ResNeXt-101 backbone network is trained with a mini-batch size of 2 and 64 RoIs per batch for only five epochs. The validation loss is also observed for these methods, and a suitable cut-off position is obtained.

Finally, we train a Mask-RCNN model with pre-trained ResNet-50 backbone network as described in [11]. Additionally, to make a comparison and show that pre-training is necessary, we also train and test the same network (Mask-RCNN with ResNet-50) without pre-trained weights. The Mask-RCNN model is an improvement on top of Faster-RCNN and includes novel architectures such as GroupNorm, FPN and FCN as described earlier. Segmentation, classification and regression branches are calculated at the same time in parallel to prevent classes from competing from each other. Which means segmentation and bounding boxes are detected on the basis of 'objectness'. Inputs to our model are resized to make their shortest side equal to 1200 pixels, and their side can be 2000 pixels long at most (1200×2000 max. resolution). Therefore, we have effectively doubled the input size of our Faster-RCNN model. We also trained this model with lower resolutions with larger mini-batches as we did with Faster-RCNN, but the performance gains at those settings were minimal when compared to this resolution. In this model, we did not test more complex backbone networks such as ResNet-101 because as we present in the next section, the ResNet-50 network is capable of extracting meaningful features to allow the

network to detect tables. The network with pre-trained backbone network is trained for 12 epochs while validation loss under supervision to prevent overfitting. The best weights are observed at 7 epochs. The network which is trained from scratch is also trained for 12 epochs while validation loss under supervision to prevent overfitting. The best weights are observed at 7 epochs as well. The selection of the final weights is as follows; The training is conducted for 10 hours on 4 GPUs and the loss graph for both with pre-training and without pre-training can be seen in Figure 4.6. Every thousandth step (every step is a feed-forward processing of a mini-batch), validation set accuracy has been tested. The snapshot of weights with the highest validation accuracy has been selected and used for testing. As it can be seen in Figure 4.6, using the pre-trained weights for feature extractors helps tremendously in reaching to a plausible solution faster while achieving also higher performance as it will be shown in the next section. Although the loss values are close, the figure shows that even after 50000 steps, the model that is trained from scratch has a slightly higher loss on average (0.01) and fails to converge to the loss value pre-trained network can achieve. The same phenomenon is also observed in structure detection system where it shows the importance of pre-training on a larger dataset. Learning rate schedule is also shown in the figure. We start with a learning rate of 0.01 (distributed over 4 GPUs), and scheduling is used to decrease the learning rate on steps 26000 and 40000. We have tested different learning rates as well (including 0.0001 starting learning rate) and found that higher learning rates work better. Further investigation can be conducted to find optimal parameters. Learning rate is divided onto 4 GPUs we use for training. This learning rate is set as described in [11] according to the GPU number.

Counting of true-positives, false-positives and others are accomplished by calculation of Intersection over Union (IoU) and checking the overlap percentage of predicted bounding boxes (P) and the ground truth bounding boxes (Gt). Then, IoU can be described as;

$$IoU(P, Gt) = \frac{P \cap Gt}{P \cup Gt} \quad (4.1)$$

IoU score denotes the correctness of the detected bounding box. Higher IoU score denotes a better overlap with the actual bounding boxes (tables). We use IoU threshold of 0.5 which means if the overlap of the predicted box and the ground truth annotation is less than 0.5, the detection is considered wrong. After the calculation of true-positives, false-positives and false-negatives, we calculate *recall*, *precision* and *F1 scores*. A true-positive detection is a correct detection, a false-positive detection is a false detection and a false-negative is when we cannot detect an object in the image. After these numbers are found, recall is calculated as;

$$Recall = \frac{truepositives}{truepositives + falsenegatives} \quad (4.2)$$



Figure 4.6: Training loss over time for the Mask-RCNN models.

Precision calculation is as follows;

$$Precision = \frac{truepositives}{truepositives + falsepositives} \quad (4.3)$$

After calculating these two values, F1 score is a function of these two and defined as;

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (4.4)$$

VOC-style Average Precision (AP) computes the average precision value for recall value 0 to 1. AP is calculated by sorting the detections with their confidence scores and drawing the precision-recall curve. The area under the curve is AP and is used by many researchers to assess the effectiveness of the object detection networks [11, 12].

4.4 Results

The first thing to note is that an image takes 0.4 seconds for our Faster-RCNN method, which only has 16 layers of convolution in the backbone feature extractor, the exact timing depends on the size of the input image. The timing of RetinaNet networks, vary between 0.2 and 0.5 seconds for the models with ResNet-50 and ResNeXt-101 backbones respectively. Lastly, the Mask-RCNN model with ResNet-50 backbone network takes around 0.5 seconds for regular sized images (images that are smaller than the max. possible resolution, 1200×2000), but can take up to 1.5 seconds per large images.

In Table 4.2 we present our results on public ICDAR 2013 and ICDAR 2017 POD datasets and compare our models to the previous work in the literature. As it can be observed in Table 4.2, ICDAR 2017 presents a higher difficulty than ICDAR 2013 especially when looked at the Average Precision (AP, calculated with the 11-point average method introduced by the PASCAL VOC [96]). We believe this is because ICDAR 2017 is a dataset from mostly research papers which includes formulas and figures and they are easy for the networks to misclassify. A brief look at the RetinaNet models shows that ResNeXt-101 backbone attains more AP than the model with ResNet-50 backbone. However, ResNet-50 backbone achieves a higher recall value which may mean, ResNet-50 learned simpler representations of tables whereas ResNeXt-101 learns to distinguish fine details and acquires higher AP. Faster-RCNN achieves better F-Score than the RetinaNet models, and as a two-stage method, it is again performing better while being slightly slower as expected. Mask-RCNN, on the other hand, achieves perfect recall on both of the datasets with significantly higher AP (about 5% between the closest model in ICDAR 2017).

Table 4.2: Table detection performance on public ICDAR datasets.

Methods	ICDAR 2013			ICDAR 2017		
	Recall	AP	F1-Score	Recall	AP	F1-Score
Proposed Faster-RCNN	0.9808	0.9738	0.9773	0.9685	0.9385	0.9535
Proposed RetinaNet w/ ResNet-50	0.9872	0.9440	0.9656	0.9937	0.9025	0.9481
Proposed RetinaNet w/ ResNeXt-101	0.9865	0.9617	0.9742	0.9748	0.9241	0.9495
Proposed Mask-RCNN	1.0000	0.9900	0.9950	1.0000	0.9868	0.9933
Proposed Mask-RCNN Scratch	1.0000	0.9662	0.9828	1.0000	0.9721	0.9858
Siddiqui [81]	0.9960	0.9960	0.9960	0.971	0.965	0.968
Kavasidis [82]	0.9810	0.9750	0.9780	-	-	-
Schreiber [10]	0.9615	0.9740	0.9677	-	-	-
Tran [31]	0.9636	0.9521	0.9578	-	-	-
Hao [72]	0.9215	0.9724	0.9463	-	-	-
Fan [69]	0.8957	0.7464	0.8142	-	-	-

Table 4.3: Table detection performance on public UNLV dataset and our private dataset.

Methods	UNLV			Private		
	Recall	AP	F1-Score	Recall	AP	F1-Score
Proposed Faster-RCNN	0.6486	0.5227	0.5789	0.9528	0.9112	0.9320
Proposed RetinaNet w/ ResNet-50	0.8784	0.3751	0.5257	0.9442	0.7649	0.8546
Proposed RetinaNet w/ ResNeXt-101	0.9324	0.7257	0.8161	0.9099	0.7773	0.8436
Proposed Mask-RCNN	1.0000	0.9819	0.9909	1.0000	0.9980	0.9990
Proposed Mask-RCNN Scratch	1.0000	0.9797	0.9897	1.0000	0.9854	0.9926
Siddiqui [81]	0.7490	0.7860	0.7670	-	-	-
Gilani [75]	0.9067	0.8230	0.8629	-	-	-
Arif [83]	0.9321	0.8633	0.8964	-	-	-
Shafait [27]	0.7900	0.8600	0.8200	-	-	-

In Table 4.3 the results on our private dataset and the public UNLV datasets obtained from our models are shown. At this point, it is crucial to remind that our models, except Mask-RCNN based model, are trained on the vanilla versions of our datasets. However, before training the Mask-RCNN, we refactor and relabel the entire dataset to encapsulate all tables and *tabular regions*. Moreover, our Faster-RCNN and RetinaNet models did not have any images from the UNLV dataset in their training set. Therefore, the documents from UNLV are brand new for these models and we can see that Faster-RCNN model with a simpler VGG-16 backbone network could not generalize well to this new dataset. However, RetinaNet model with ResNeXt backbone has achieved exceptionally well AP when compared to the RetinaNet model with ResNet backbone which confirms our deductions as we said ResNeXt backbone allowed the network to have better understanding of what a table is. As in other methods, our final Mask-RCNN model which trained on aggregated data from all of our datasets achieves the state-of-the-art performance.

Results in Table 4.2 and Table 4.3 shows us that ResNet-50 is a acceptable choice as a backbone for this task when speed and f-score is considered. Since the main focus is the optimization of the supply chains, there are millions of documents that need to undergo this operation, and this fact makes the speed as one of the key factors in the decision process. The 'Scratch' denotes when the network is trained from scratch (randomly initialized weights) and it is apparent from the results that using pre-trained feature extractors help achieve better detection results. However, the results are close, and with further improvements to the training dataset (such as increasing the number of data points), there might be no need to use pre-trained weights. The experiments are conducted with ResNet-50 backbone feature extractor in Mask-RCNN model. Example table detections from a public dataset (left) and an example from our private dataset where there are more than 4 tables with table in table special case and where figures are also present (right) can be found in; Figure 4.7 (Faster-RCNN with VGG-16), Figure 4.8 (RetinaNet with ResNet-50), Figure 4.9 (RetinaNet with ResNeXt-101) and Figure 4.10 (Mask-RCNN with ResNet-50).

4.5 Conclusion

In this chapter, we compared three state-of-the-art object detectors on table detection task from the documents acquired from a wide range of fields and domains. The table extraction task can save thousands of valuable human hours which spent to extract information from the tables. Even though we do not employ any pre or post-processing, our models achieved state-of-the-art performance on many public datasets. The best performing model is the Mask-RCNN model with a successful performance on every table detection

表 2-1 行为科学理论主要内容

人际关系理论	存在着“非正式组织”
激励理论	(1)需求层次理论：动机产生行为，需要产生动机，生理-安全-社会-尊重-自我实现需求。 (2)人性假设理论：任何组织的管理者在管理其下属时，对其下属所持基本看法为“管理的假定”。有变化需求假定(X理论/Y理论)、超Y理论、Z理论和成熟-不成熟理论。 (3)群体行为理论：团体行为理论，主要论述了非正式团体的要素、目标、凝聚力、规范、结构、领导方式、规模、行为分类、参与者、组织对变动的反应等。 (4)敏感性训练理论 (5)强化理论 (6)期望理论
领导效能理论	领导者地理论、行为连续带理论、领导方式系统理论、领导四象限模式、管理方格论、领导行为权变理论

表 2-9 现代管理理论主要内容

社会系统学派	作用以及有关集体的总和，任何正式组织都包括协作意愿、共同目标和信息沟通；管理者在组织中的作用和职能是建立和维持一个信息联系的系统，协调组织成员的行动，规定、解释目标并处理个人目标与组织目标的矛盾
决策理论学派	管理就是决策，决策过程包括搜集情报阶段、拟定计划、选定计划阶段；决策的原则是制定出满意的标准，达到或超过这个标准就是可行的方案，程序化与非程序化决策；决策制约着组织机构的设置、决策的技术和方法
系统管理学派	系统管理的内容包括系统哲学、系统管理和系统分析。系统是一种有组织的综合整体，企业作为一个系统进行设计和经营，按照系统的要求进行组织和运行，确定一个最优的解决方法或实际行动
经验主义学派	重点分析成功管理者的实际管理的经验，并加以概括，总结出成功规律性，使之系统化、合理化，提出建议
管理科学学派	制定和运用数学模型与程序系统来表示计划、组织和控制活动，通过定量分析，选择最优方案达到组织的目标。特征是将数学、计算机和系统概念引入管理方法中，其主要方法包括线性规划法、决策树法、计划评审及关键路线法、模拟、概率论、对策论、排队论、库存论和搜索论
权变理论学派	“如果-就要”关系理论是权变理论的思想基础，主要集中在领导效能理论和组织结构的设计

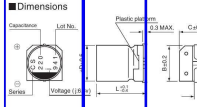
三、管理学理论应用于水资源管理

将管理学基本理论应用于水资源管理，具有广泛的前景。主要表现在以下几个方面。

1. 以人为本，挖掘水资源管理者潜力

人是水资源管理制定者和执行者，水资源管理活动必须以人为基本前提，做好水资源管理者的工作管理。为此，必须充分认识人的个体差异，采用多种手段发挥水资源管理者潜能。在分配工作的时候，要充分注意到每个人的工作与其能力相“匹配”，达到既发挥其潜力，同时让水资源管理者有信心完成本职工作。在实际工作中，充分发挥水资源管理的积极参与热情，建立与之相配套的组织文化；对每个水资源管理者建

(a)

CONDUCTIVE POLYMER ALUMINUM SOLID ELECTROLYTIC CAPACITORS		nichicon
PCS	Chip Type, Long Life Assurance	
- Load life of 5000 hours at 105°C. - SMD type: Lead free reflow soldering condition at 260°C peak corresponds. - Compliant to the RoHS directive (2011/65/EU).		
Specifications Operating Temperature Range: -55 to +105°C Rated Voltage Range: 4 to 16V Rated Capacitance Range: 22 to 560µF Capacitance Tolerance: ±20% at 120Hz, 20°C Tangent of loss angle (tan δ): Less than or equal to the specified value at 120Hz, 20°C ESR (±1): Less than or equal to the specified value at 100kHz, 20°C Leakage Current (±2): Less than or equal to the specified value. After 2 minutes of application of rated voltage at 20°C Temperature Characteristic (Max Impedance Ratio): Z+105°C / Z+20°C ≤ 1.25 (100kHz) Z-55°C / Z+20°C ≤ 1.25		
Endurance	The specifications listed at right shall be met when the capacitors are subjected to 20°C after the rated voltage is applied for 500 hours at 105°C.	Capacitance change: Within ±20% of the initial capacitance value (P/B) ESR (±1): 150% or less than the initial specified value Leakage current (±2): Less than or equal to the initial specified value
Damp Heat (Steady State)	The specifications listed at right shall be met when the capacitors are subjected to 85°C after the rated voltage is applied for 100 hours at 85°C, 90% RH.	Capacitance change: Within ±20% of the initial capacitance value (P/B) ESR (±1): 150% or less than the initial specified value Leakage current (±2): Less than or equal to the initial specified value
Resistance to Soldering Heat	After soldering the capacitor under the soldering conditions prescribed herein, the capacitor shall meet the specifications listed at right. The heating shall be done at 150 to 200°C and for 60 to 180 sec. The duration for 150 to 200°C temperature of capacitor surface shall not exceed 60 seconds. In case peak temperature is 250°C or less, reflow soldering shall be two times maximum. Measurement for solder temperature profile shall be made at the capacitor top and the terminal.	Capacitance change: Within ±10% of the initial capacitance value (P/B) ESR (±1): 150% or less than the initial specified value Leakage current (±2): Less than or equal to the initial specified value
Marking	Navay blue print on the case top	
*1 ESR should be measured at both of 10 terminal ends closest where the terminals preclude through the plastic platform. *2 Conditioning: If an doubt arises, measure the leakage current after the voltage treatment of applying DC rated voltage continuously to the capacitor for 120 minutes at 105°C. *3 Initial value: The value before test of examination of resistance to soldering.		
Dimensions 		
Type numbering system (Example: 6.3V 220µF) PCS 0J1221M C L T G S Taping code Size code Configuration Capacitance tolerance (±20%) Rated capacitance (220µF) Rated voltage (6.3V) Series name Type		Frequency: 100Hz 1kHz 10kHz 100kHz or more Coefficient: 0.05 0.30 0.70 1.00
*Dimension table in next page.		

(b)

Figure 4.7: Example detection results by our Faster-RCNN based model with VGG-16 backbone network. Detections are not accurate, however it shows promise on these hard examples and even though it has only 16 layers CNN, captures the idea of tables.

Table 2-8 行为科学理论主要内容

人和管理理论	存在着“非正式组织”
激励理论	(1)需求层次理论：动机产生行为，需要产生动机，生理-安全-社会-尊重-自我实现需要 (2)人性假设理论：任何组织的管理者在其下属时，对其下属所持基本看法为“管理的假定”，有麦克雷格假设(X理论/Y理论)，超Y理论、Z理论和成熟-成熟理论 (3)群体行为理论：团体力学理论，主要论述了非正式团体的要素、目标、凝聚力、规范、结构、领导方式、规模、行为分类、参与者、组织者对变动的反应等 (4)敏感性训练理论 (5)强化理论 (6)期望理论
领导效能理论	领导者特质理论、行为连续带理论、领导方式系统理论、领导四象限模式、管理方格论、领导行为权变理论

Table 2-9 现代管理理论主要内容

社会系统学派	组织是一个协作的系统；非正式组织是不属于正式组织的，是不受正式组织管理的个人联系和相互作用以及有关集体的总和；任何正式组织都包括协作意愿、共同目标和信息沟通；管理者在组织中的作用和职能是建立和维持一个信息联系的系统，协调组织成员的行动、规定、解释目标并处理个人目标与组织目标的矛盾
决策理论学派	管理就是决策；决策过程包括搜集情报阶段、拟定计划、选定计划阶段；决策的准则是制定出满意的标准，达到或超过这个标准就是可行的方案；程序化与非程序化决策；决策影响着组织机构的设置、决策的技术和方法
系统管理学派	系统管理的内容包括系统哲学、系统管理和系统分析。系统是一种有组织的综合整体，企业作为一个系统进行设计和经营，按照系统的要求进行组织和运行，确定一个最优的解决方法或实际行动
经验主义学派	重点分析成功管理者实际管理的经验，并加以概括，总结出成功经验共性，使之系统化、合理化，提出建议
管理科学学派	制定和运用数学模型与程序系统来表示计划、组织和控制活动，通过定量分析，选择最优方案达到组织的目标。特征是将数学、计算机和系统观念引入管理方法中，其主要方法包括线性规划法、决策树法、计划评审及关键路线法、模拟、概率论、对策论、排队论、库存论和搜索论
权变理论学派	“如果...那么”关系理论权变理论的面相展线，主要在于领导效能理论和组织结构的设计

三、管理学理论应用于水资源管理

将管理学基本理论应用于水资源管理，具有积极的意义。主要表现在以下几个方面。

1. 以人为本，挖掘水资源管理者潜能。人是水资源管理制定者和执行者，水资源管理活动必须以人为基本前提，做好水资源管理者的工作管理。为此，必须充分认识人的个体差异，采用多种手段发挥水资源管理者潜能。在分配工作的时候，要充分注意到每个人的工作与其能力相“匹配”，真正做到发挥其潜力，同时让水资源管理者有信心完成本职工作。在实际工作中，充分发挥水资源管理的积极参与热情，建立与之相配套的组织文化，对每个水资源管理者在

CONDUCTIVE POLYMER ALUMINUM SOLID ELECTROLYTIC CAPACITORS **nichicon**

PCS Chip Type, LongLife Assurance

- Load life of 5000 hours at 105°C.
- SMD type: Lead free reflow soldering condition at 260°C peak correspondence.
- Compliant to the RoHS directive (2011/65/EU).

PCS LongLife PCF

Item	Specifications (25°C ± 5°C)	Performance Characteristics
Category	Temperature Range: -55 to +105°C	
Rated Voltage Range	4 to 16V	
Rated Capacitance Range	22 to 560µF	
Capacitance Tolerance	±20% at 120Hz, 20°C	
Tan δ (loss angle tan δ)	Less than or equal to the specified value at 100Hz, 20°C	
ESR (Ω)	Less than or equal to the specified value at 100Hz, 20°C	
Leakage Current (α=2)	Less than or equal to the specified value. After 2 minutes' application of rated voltage at 20°C	
Temperature Characteristics (Max. Impedance Ratio)	Z+105°C / Z+20°C ≤ 1.25 (100kHz) Z-55°C / Z+20°C ≤ 1.25	
Endurance	The specifications listed at right shall be met when the capacitors are restored to 20°C after the rated voltage is applied for 1000 hours at 105°C.	Capacitance change: Within ± 10% of the initial capacitance value (α=3) tan δ: 10% or less than the initial specified value ESR (α=1): 10% or less than the initial specified value Leakage current (α=2): Less than or equal to the initial specified value
Damp Heat (Steady State)	The specifications listed at right shall be met when the capacitors are restored to 20°C after the rated voltage is applied for 1000 hours at 60°C, 90% RH.	Capacitance change: Within ± 10% of the initial capacitance value (α=3) tan δ: 10% or less than the initial specified value ESR (α=1): 10% or less than the initial specified value Leakage current (α=2): Less than or equal to the initial specified value
Resistance to Soldering Heat	After soldering the capacitor under the soldering conditions prescribed here, the capacitor shall meet the specifications listed at right. Pne heating shall be done at 150 to 200°C and for 60 to 180 sec. The duration for over 200°C temperature at capacitor surface shall not exceed 60 seconds. If case peak temperature is 250°C or less, reflow soldering shall be two times maximum. Measurement for solder temperature profile shall be made at the capacitor top and the terminal.	Capacitance change: Within ± 10% of the initial capacitance value (α=3) tan δ: 10% or less than the initial specified value ESR (α=1): 10% or less than the initial specified value Leakage current (α=2): Less than or equal to the initial specified value
Marking	White blue print on the case top	

■ Dimensions

Case size: 6.3 x 6.3 x 6.3 (mm)

Terminal spacing: 2.54 (mm)

Mounting dimensions: L=5.0, W=5.0, H=5.0 (mm)

■ Type numbering system (Example: 6.3V 220µF)

PCS 6.3V 220µF

■ Frequency coefficient of rated ripple current

Frequency	10Hz	1kHz	10kHz	100kHz or more
Coefficient	0.05	0.30	0.70	1.00

■ Dimension table in next page.

CAT.8100G

(a)

(b)

Figure 4.8: Example detection results by our RetinaNet based model with ResNet-50 backbone network.

table 0.99

表 2-8 行为科学理论主要内容

人际管理理论	工人是“社会人”而不是“经济人”；工人的工作态度与士气是影响工作效率的关键；企业中存在着“非正式组织”
激励理论	(1)需求层次理论：动机产生行为，需要产生动机，生理-安全-社会-尊重-自我实现需要 (2)人性假设理论：任何组织的管理者在其管理其下属时，对其下属所持基本看法为“管理的假定”，有麦克雷格假设(X理论/Y理论)、超Y理论、Z理论和不成熟-成熟理论 (3)群体行为理论：团体力学理论，主要论述了非正式团体的要素、目标、凝聚力、规范、结构、领导方式、规模、行为分类、参与者、组织者对变动的反应等 (4)敏感性训练理论 (5)强化理论 (6)期望理论
领导效能理论	领导者特质理论、行为连续变理论、领导方式系统理论、领导四条限制模式、管理方格论、领导行为权变理论

table 0.99

表 2-9 现代管理理论主要内容

社会系统学派	组织是一个协作的系统；非正式组织是不属于正式组织的，是不受正式组织管理的个人联系和相互作用以及有关集体的总和；任何正式组织都包括协作意愿、共同目标和信息沟通；管理者在组织中的作用和职能是建立和维持一个信息联系的系统，协调组织成员的行动，规定、解释目标并处理个人目标与组织目标的矛盾
决策理论学派	管理就是决策；决策过程包括情报阶段、拟定计划、选定计划阶段；决策的准则是制定出满足的标准，达到或超过这个标准就是可行的方案；程序化与非程序化决策；决策制约着组织机构的设置、决策的技术和方法
系统管理学派	系统管理的内容包括系统哲学、系统管理和系统分析。系统是一种有组织的综合整体，企业作为一个系统进行设计和经营，按照系统的要求进行组织和运行，确定一个最优的解决方法或实际行动
经验主义学派	重点分析成功管理者实际管理的经验，并加以概括，总结出成功经验共性，使之系统化、合理化，提出建议
管理科学学派	制定和运用数学模型与程序系统来表示计划、组织和控制活动，通过定量分析，选择最优方案达到组织的目标。特征是将数学、计算机和系统观念引入管理方法中，其主要方法包括线性规划法、决策树法、计划评审及关键路线法、模拟、概率论、排队论、库存论和搜索论
权变理论学派	“如果-就要”关系理论是权变理论的思想基础。主要集中于领导效能理论和组织结构的设计

三、管理学理论应用于水资源管理

将管理学基本理论应用于水资源管理，具有广泛的前景。主要表现在以下几个方面。

1. 以人为本，挖掘水资源管理者潜力

人是水资源管理制定者和执行者，水资源管理活动必须以人为基本前提，做好水资源管理者的工作管理。为此，必须充分认识人的个体差异，采用多种手段发挥水资源管理者潜能。在分配工作的时候，要充分注意到每个人的工作与其能力相“匹配”，达到既发挥其潜力，同时让水资源管理者有信心完成本职工作。在实际工作中，充分发挥水资源管理的积极参与热情，建立与之相配套的组织文化；对每个水资源管理者建

45

(a)

CONDUCTIVE POLYMER ALUMINUM SOLID ELECTROLYTIC CAPACITORS

nichicon

PCS Chip Type, LongLife Assurance

● Load life of 5000 hours at 105°C.
 ● SMD type : Lead free reflow soldering condition at 260°C peak correspondence.
 ● Compliant to the RoHS directive (2011/65/EU).

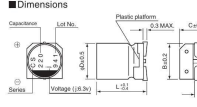
PCS LongLife PCF



Specifications	
Item	Performance Characteristics
Category Temperature Range	-55 to +105°C
Rated Voltage Range	4 to 16V
Rated Capacitance Range	22 to 560µF
Capacitance Tolerance	±20% at 120Hz, 20°C
Tangent of loss angle (tan δ)	Less than or equal to the specified value at 100kHz, 20°C
ESR (α 1)	Less than or equal to the specified value at 100kHz, 20°C
Leakage Current (α 2)	Less than or equal to the specified value After 2 minutes' application of rated voltage at 20°C
Temperature Characteristics (Max Impedance Ratio)	Z+105°C / Z+20°C ≤ 1.25 (100kHz) Z-55°C / Z+20°C ≤ 1.25
Endurance	The specifications listed at right shall be met when the capacitors are restored to 20°C after the rated voltage is applied for 1000 hours at 105°C. Capacitance change Within ± 20% of the initial capacitance value (α 3) tan δ 150% or less than the initial specified value ESR (α 1) 150% or less than the initial specified value Leakage current (α 2) Less than or equal to the initial specified value
Damp Heat (Steady State)	The specifications listed at right shall be met when the capacitors are restored to 20°C after the rated voltage is applied for 1000 hours at 60°C, 90% RH. Capacitance change Within ± 20% of the initial capacitance value (α 3) tan δ 150% or less than the initial specified value ESR (α 1) 150% or less than the initial specified value Leakage current (α 2) Less than or equal to the initial specified value
Resistance to Soldering Heat	After soldering the capacitor under the soldering conditions prescribed here, the capacitor shall meet the specifications listed at right. Pre-heating shall be done at 150 to 200°C and for 60 to 180 sec. The duration for over 200°C temperature at capacitor surface shall not exceed 60 seconds. In case peak temperature is 250°C or less, reflow soldering shall be two times maximum. Measurement for solder temperature profile shall be made at the capacitor top and the terminal. Capacitance change Within ± 10% of the initial capacitance value (α 3) tan δ 150% or less than the initial specified value ESR (α 1) 150% or less than the initial specified value Leakage current (α 2) Less than or equal to the initial specified value
Marking	Navvy blue print on the case top

① ESR should be measured at both of the terminal ends closest where the terminals protrude through the plastic platform.
 ② Conditioning : If any doubt arises, measure the leakage current after the voltage treatment of applying DC rated voltage continuously to the capacitor for 120 minutes at 105°C.
 ③ Initial value : The value before test of examination of resistance to soldering.

Dimensions



Type numbering system (Example : 6.3V 220µF)

1 2 3 4 5 6 7 8 9 10 11 12 13 14

PCS 0 J 2 2 1 M C L T G S

Taping code
 Size code
 Configuration
 Capacitance tolerance (±20%)
 Rated capacitance (220µF)
 Rated voltage (6.3V)
 Series name
 Type

Frequency coefficient of rated ripple current

Frequency	150Hz	1kHz	10kHz	100kHz or more
Coefficient	0.05	0.30	0.70	1.00

● Dimension table in next page.

CAT.8100G

47

(b)

Figure 4.9: Example detection results by our RetinaNet-based model with ResNeXt-101 backbone network. Detection is precise but not all tables are detected.

table 1.00

表 2-8 行为科学理论主要内容

Table 1.00	
人际管理理论	工人是“社会人”而不是“经济人”；工人的工作态度与士气是影响工作效率的关键；企业中存在着“非正式组织”
激励理论	(1) 需求层次理论：动机产生行为，需要产生动机，生理-安全-社会-尊重-自我实现需要 (2) 人本假说理论：任何组织的管理者在其管理其下属时，对其下属所持基本看法为“管理的假说”，有艾克索弗假说(X理论)、超Y理论、Z理论和不成熟、成熟理论 (3) 群体行为理论：团体心理学说，主要论述了非正式团体的要素、目标、凝聚力、规范、结构、领导方式、规模、行为分类、参与者、组织者对变动的反应等 (4) 敏感性训练理论 (5) 强化理论 (6) 期望理论
领导效能理论	领导者素质理论、行为连续统理论、领导方式系统理论、领导四象限理论、管理方格论、领导行为权变理论

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
Population	1.00	1.01	1.02	1.03	1.04	1.05	1.06	1.07	1.08	1.09	1.10	1.11	1.12	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.20	1.21	1.22	1.23	1.24	1.25	1.26	1.27	1.28	1.29	1.30	1.31	1.32	1.33	1.34	1.35	1.36	1.37	1.38	1.39	1.40	1.41	1.42	1.43	1.44	1.45	1.46	1.47	1.48	1.49	1.50	1.51	1.52	1.53	1.54	1.55	1.56	1.57	1.58	1.59	1.60	1.61	1.62	1.63	1.64	1.65	1.66	1.67	1.68	1.69	1.70	1.71	1.72	1.73	1.74	1.75	1.76	1.77	1.78	1.79	1.80	1.81	1.82	1.83	1.84	1.85	1.86	1.87	1.88	1.89	1.90	1.91	1.92	1.93	1.94	1.95	1.96	1.97	1.98	1.99	2.00

表 2-9 现代管理理论主要内容

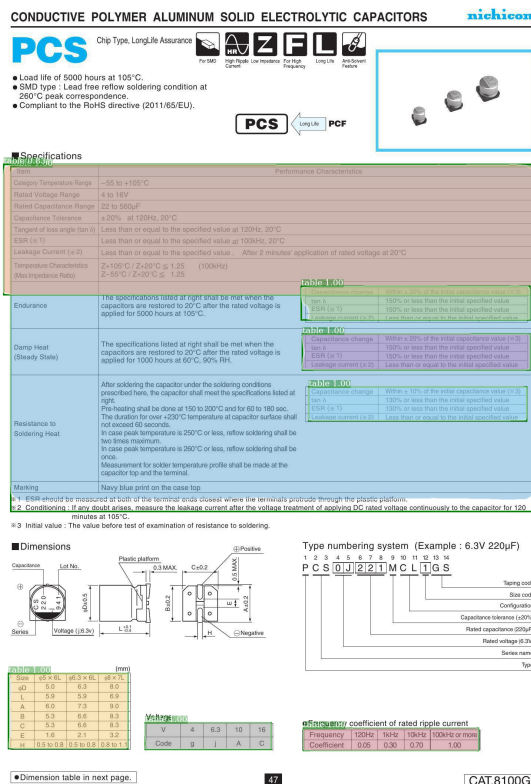
Table 1-9-6	
社会学学派	组织是一个协作的系统,非正式组织不属于正式组织,是不受正式组织管理的个人联系和相互作用以及有天然体的总和,任何正式组织都包括协作意愿、共同目标和信息沟通;管理者在组织中的作用和职能是建立和维护一个信息联系的系统,协调组织成员的行动,规定、解释目标并处理个人目标与组织目标的矛盾
决策理论学派	管理就是决策;决策过程包括搜集情报阶段,拟定计划,选定计划阶段;决策的准则是制定出满意的标准,达到或超过这个标准就是可行的方案;程序化与非程序化决策,决策的制定者包括组织机构的设置,决策的技术和方法
系统管理学派	系统管理这个名称包括系统学、系统管理学和系统分析。系统是一个有组织的综合整体,企业作为一个系统进行设计和经营,按照系统的要求进行组织和运行,确定一个最优的解决方法或实现行动
经验主义学派	重点分析成功管理者实际管理的经验,并加以概括,总结出成功经验共性,使之系统化、合理化,提出建议
管理科学学派	制定和运用数学模型与程序系统来表示计划、组织和控制活动,通过定量化分析,选择最优方案达到组织的目标。特点是将以数学、计算机和系统观念引入管理方法中,其主要方法包括线性规划法、决策树法、计划评审法及关键路线法、排队论、概率论、对策论、排队论、库存论和理论
权变理论学派	“加絮一铁要”是关键理论是权变理论的思想基础,主要集中在导致效能理论和组织结构的设计

三、管理学理论应用于水资源管理

将管理学基本理论应用于水资源管理,具有广泛的前景。主要表现在以下几个方面。

1. 以人为本, 挖掘水资源管理者潜力

人是水资源管理制定者和执行者,水资源管理活动必须以人为基本前提,做好水资源管理者的工作管理。为此,必须充分认识人的个体差异,采用多种手段发挥水资源管理者潜能。在分配工作的时候,要充分注意到每个人的工作与其能力相“匹配”,达到既发挥其潜力,同时让水资源管理者有信心完成本职工作。在实际工作中,充分发挥水资源管理的积极性与热情,建立与之相配合的组织文化;对每个水资源管理者建



(a) Successful detection results on a non-english document

(b) Near perfect detection with one overlapping detection. Nested table structure is detected.

Figure 4.10: Example detection results by our Mask-RCNN based model with ResNet-50 backbone feature extractor.

dataset we have. We observe an incremental increase in f-score that is correlated with the size of the backbone network and the architectural decisions made in building the model itself. Since we can only use small mini-batches, GroupNorm aids us. FPN and FCN are included in our model with a considerably deep neural network, ResNet-50 [14], as our backbone feature extractor. Detections on ICDAR datasets are of higher quality and we believe this is due to the difficulty of the tables inside the datasets. All our models achieve satisfactory f-scores on ICDAR but on more challenging datasets such as UNLV and our private dataset, the models start to give inconsistent results, except for our Mask-RCNN model. UNLV dataset includes tilted examples and a unique composition of tables and the results are the lowest in all the datasets for all the methods. RetinaNet models despite their lower performance, promise a exceptional value if the speed is one of the main considerations. However, Mask-RCNN based method is the clear winner for table extraction. However, it is essential to note that except the ICDAR 2013 dataset, we use some portion of the dataset for training. Also, some of the previous work uses a single dataset to train and test [10], but we use many public and private dataset to train our networks. Hence, although Mask-RCNN method achieves higher F1 score than the previous work with more data [74], we conclude that the increase in precision and recall of ours is supported by increased variety and quantity of the data. In the next chapter, we extract crops of tables from the UNLV dataset and train a network to detect table structure from extracted table crops while the final chapter bridges these two works together and form our E2E solution.

Chapter 5

Table Structure Detection

In the previous chapter, we present our solutions for detecting tables from arbitrary types of documents. However, the need is more than just extracting the tables. Supply chains and other organizations require the information contained in the tables. To extract the information from the tables, we have to identify and extract the structure of the table since tables are a structured data format but only not suitable for computers to understand. Detection of rows and columns is a crucial step to find the semantic meaning of the content further. However, we leave the semantic meaning extraction to future work. We propose both deep learning based and hand-crafted rule-based solutions to the table structure detection problem. In the second phase of our E2E system design, we take an arbitrary table as input (not the image of a document in full) and output the detected structural elements, *rows* and *columns*. The text (or what is contained in the cells. i.e. text or figures or formulas or tables) is located in the intersection of these structural elements which we call *cells*. We aim to make our methods to operate on a variety of domains that include but not limited to research papers, magazines, journals, newspapers, web pages. We have published our findings on table structure detection task which can be found in [8]. In this section when we propose our solutions, we make the assumption that the first stage produces perfect table detection and those detection instances are cropped to be used in this stage, the second stage. Therefore, inputs to our table structure detection systems are table crops in an image format. Table structure detection solutions that we propose can operate on both tables from born-digital (which we convert to an image format) and scanned documents. We compare our methods to each other as well as to the literature, but the general approach to this task is to detect structure directly from the full document image as seen in [10, 64], and hence, our work cannot be directly compared. However, we provide a comparison to be complete and conclude that out of the models that we

proposed, Mask-RCNN based method outperforms other tested methods. In the table detection problem, we mention that the dataset is lacking the numbers required to train a deep learning model. That would be an understatement for the table structure dataset. Most of the datasets (public and private) we benefit from in the table detection task, do not have table structure annotations. Thus, the data is overwhelmingly limited, and a possible future work should include labelling more data to achieve possibly more robust solutions. Since every table generally has more than 10 rows and column structures, hand-labelling a new dataset is a costly operation and in this thesis, we focus on training models with a few data. Another point is that as Chapter 2 shows, this problem is not as much studied as the table detection task. Since we propose deep learning methods, we have to make use of transfer learning to accommodate the amount of data available. To increase the amount of data in the training phase, we also add flipped (mirrored) images to the training set which effectively doubles the training set size but with less variation. Since the deep learning models are data hungry and learn better representations with more data to give better outputs, incrementing the dataset size with augmentations are useful [122]. Different data augmentations include traditional augmentations (flipping, cropping), using Generative Adversarial Networks (GANs) [123] or training a network to augment images [122]. These methods can increase the classification accuracy up to 7% [122], and the future extension to this work could also use different augmentations to increase the amount of data.

We propose solutions with deep learning methods; Faster-RCNN [12], RetinaNet [13] and Mask-RCNN [11] with the backbones VGG-16 [19], ResNet-50 [14] and ResNet-50-C4 [14] which means the conv4 output of ResNet-50 subnetwork used as the feature map. We also propose a heuristics-based method called Mathematical Morphology. We study a rule-based method to form a basis and see the behaviour of this method and compare it to the deep learning based methods. We expected our rule-based model to perform equally well on both column and row detection. However, similar to our deep learning models, it performed substantially better on columns and had more difficulty detecting rows. Our dataset for this task is limited due to the lack of public datasets, and we only use UNLV dataset as it provides a reliable and precise set of structure annotations. This chapter starts by examining the datasets we use in Section 5.1 and then followed by the methods we use and our choice of parameters and additions if any in Section 5.2. After the methods and datasets are detailed, Section 5.3 explains our training process and heuristics, and we conclude the table structure detection approach of ours by presenting our results and comparing it to others in Section 5.4. We reach to a final conclusion in Section 5.5.

5.1 Datasets

As previously mentioned, our focus is on tabular structure detection from the electronic component datasheets. However, no dataset specifically targets this domain or a similar domain. Only datasets that provide annotations for the table structures we have is the ICDAR 2013 [73] and UNLV datasets. We use publicly available UNLV dataset which is one of the two public datasets that provide annotations for table structures. We only use UNLV dataset because we believe the difficulty of this dataset is more similar to our end goal, tabular structure detection from electronic component datasheets. Although we use UNLV, an extension of this work must try ICDAR 2013 dataset in addition to labelling more data, and we acknowledge that there are more improvements to this task in terms of the dataset.

UNLV

The UNLV dataset initially included 2889 scanned documents from various sources. Out of those, only 424 images include annotations, which we had further removed some (24) of the images due to the extreme tilt introduced by scanning the documents. This first elimination phase is the same as it is introduced in Chapter 4. The resulting 400 images from the UNLV dataset included 537 tables and we crop them since this section focuses on table structure detection on table crops only. Even with the eliminated set, we observe tilt in some of the tables. The tilt is not as big of a problem in table detection as it is in structure detection. The increased difficulty in this new task is because table structures are smaller and narrower in size and our deep learning models require more precision on the dataset to backpropagate meaningful errors. Therefore we conduct another run over the entire dataset to eliminate extreme tilt and fix the annotations to minimize errors in the labels. Finally, we have left with only 513 tables which are divided into only training and test sets. We did not use a validation set to maximize the available training data; instead, we train the models for a fixed number of epochs and selected one model after training by using a small set of images (40 images) extracted from the training set itself. Two sample images from our UNLV dataset can be found in Figure 5.1.

After we obtain the initial results from our models (Table 5.1), a pattern revealed itself. A pattern that clearly showed that row detection is problematic and there is a need for further adjustments to get better detections. We developed a pre-processing step to overcome the issue. This step goes over each annotation and ensures that each side of the labelled rectangles, be it either row or column, has at least Ω pixels in length. We experimented with different values of Ω . Our findings denote that 50 is enough and when

we make it 100, the stretching goes out of hand (we ended up with images with more than 65536 pixels in height). When Ω is set to 50, we find and propose that a table crop (and possible a full document image) should be vertically stretched by 2.1x to its original size and should be horizontally scaled by 4% of its size. Hence, horizontal scaling is not necessary and in fact when we try horizontally stretching the image to its double in size, detections are observed to get worse. Beforehand, the network could not see the rows and the detecting them were virtually impossible because of the rows' small visual areas in the image, but the models were still being punished for this. Consequently, column detection performance had improved when we stretched the dataset vertically, and this is because backpropagated loss started to be more meaningful for the row detection.

NWSSI screening objectives	Comparable national criteria at time of screening			Current national criteria			
	NWTS 33(1) (DOE, 1982b)	NWTS 33(2) (DOE, 1981a)	10 CFR Part 60 (July 1981 NRC proposed rule)	10 CFR Part 960 (1984)	No.	Title	Page
1.0 CONTAINMENT	3.1.2, 3.2.2(1), 4.2	3.2(par. 1), 3.3(par. 1), 3.4(par. 1)	60.111(b)(2)(i), 60.111(b)(2)(ii)(A), 60.111(b)(3)(i)	960.4-1(a)			
1.1 Processes		3.4(2)					
1.1.1 Chemical release		3.3(1), 3.4(2), 3.2(1), 3.2(4)	60.123(b)(5), 60.123(b)(13-14)	960.4-2-2(a), 960.4-2-2(b)(4), 960.4-2-2(c)(1,3)	1.	pH Trends Observed for SRL A and SRL U Glass	18
1.1.2 Mechanical release		3.4(2)	60.123(b)(15), 60.132(k)(1)	960.4-2-3(a), 960.4-2-3(b)(1,2)	2.	Average Cation Trends Observed for SRL Glass	19
1.2 Events		3.5(par. 1), 3.5(1)	60.123(a)(7), 60.123(b)(6,7,10)		3.	Total Am, Pu, and U Release from SRL A Glass, Disk-Only Test	21
1.2.1 Seismic		3.5(2), 3.5(5)	60.112(a), 60.123(a)(5), 60.123(b)(9)	960.4-2-7(a), 960.4-2-7(c)(1-4)			
1.2.2 Erosion		3.5(4)	60.112(b), 60.122(i), 60.123(b)(4)	960.4-2-5(a), 960.4-2-5(b)(1,3), 960.4-2-5(c)(1), 960.4-2-5(d)	4.	Micrographs of the Surfaces of SRL U Glass	25
1.2.3 Volcanic		3.5(3)	60.112(a), 60.123(b)(11)	960.4-2-7(a), 960.4-2-7(b)(1), 960.4-2-7(c)(1)	5.	Micrograph of the Polished Cross Section of Sample 15	27
1.2.4 Human intrusion	3.2.2(3), 3.3.2(4)	3.6(par. 1), 3.6(2)	60.123(b)(1-3)	960.4-2-8(a), 960.4-2-8(b)(1,2), 960.4-2-8(c)(1-4), 960.4-2-8(d)(1,2)	6.	EDS Spectra of SRL U Glass Surfaces	29
1.2.5 Miscellaneous	2.3		60.122(j)	960.4-2-6(a), 960.4-2-6(b)(1)	7.	SIMS Profiles for Selected Elements in Reacted SRL U Glass . .	30
					8.	(NL) _{Li} as a Function of Time ^{1/2}	33

Figure 5.1: Example images from the UNLV table structure dataset. As it can be seen, some tables have dense rows and columns (left) and a table of content example (right).

5.2 Methods

In this work, the end goal is to extract information from both born-digital and scanned documents through tabular structure detection with the proposed two-stage process. To

solve the first stage, table detection stage, we proposed three different deep learning based methods and concluded that Mask-RCNN is equipped with right setup and performs the best. In this work, we follow the same incremental, step-by-step process to find the best suitable set of deep learning architectures to detect structures. This section starts by describing the Faster-RCNN model we employ, and for each deep learning method, we explain exceptional cases (if any) as well as backbone networks under study. Then, we give details on our version of RetinaNet and Mask-RCNN methods we use in solving table structure detection task. The section ends by describing the mathematical morphology method we employ and explain our hand-crafted heuristics to detect rows and columns. Architectures of the deep learning models and the morphology-based method are explained in Chapter 3.

Faster-RCNN [12] is a prominent two-stage object detection network of which still competing to be state-of-the-art amongst newer models. The first of the two stages is RPN where a number of proposals are made based on the feature map extracted from the underlying feature extractor network called *backbone network*. Classification and regression of the proposals mentioned above are completed in the second stage which classifies and refines the bounding boxes for each input image. In the first stage, we generate 2000 proposals which is the default value proposed by [12]. They also showed that increasing this number is not always improves the output of the model. We are testing Faster-RCNN to be complete and provide a more sound comparison with our table detection methods and to have base grounds to discuss the inherent differences of the two tasks. More detailed explanation and different use cases to attest the reputation of the Faster-RCNN is shown previously, and are not repeated here. Since our dataset for this task is severely limited, transfer learning is a must, and we heavily rely on it. We use the redefined version of our UNLV structure dataset to fine-tune the backbone networks of VGG-16 [19] and the ResNet-50 [14]. We use the VGG-16 as our backbone as in table detection, and we add the ResNet-50 network to our Faster-RCNN model as it is shown to be optimal in our previous works [7, 8, 9]. Regularly, when we say ResNet-50, we mean that the feature map is extracted from the final convolutional layer (at the end of *conv5*, refer to Figure 3.2). However, in Faster-RCNN, we do not use ResNet-50 *conv5* output since the Faster-RCNN model is not benefiting from an FPN-like architecture [15], every layer of convolution is a disadvantage that causes information loss. Therefore, we use the final convolutional layer of the *conv4* block of the ResNet-50 (Figure 3.2), namely *ResNet-50-C4*. As it can be seen in Figure 5.1, rows and columns can be remarkably narrow, which makes them vulnerable to the operations of convolution, specifically max pooling operation. Therefore, we train our Faster-RCNN models on the stretched version of our dataset as well. In the end, the Faster-RCNN based method with VGG-16 backbone feature extractor is demolished

under the performance of the same network with ResNet-50-C4 backbone trained on the stretched dataset. The model with ResNet-50-C4 backbone achieved an F-Score of 91% on the stretched dataset whereas VGG-16 based model could only achieve F-Score of 25% on the initial dataset. We observed lesser improvements on the column detection (67% to 96%), but this was expected since the main problem was with the row detection and our solution addressed that problem.

The RetinaNet used in this task is summarized in Figure 3.11 and essentially the same as the proposal in [13]. Instead of two different backbones as in table detection, here we only use backbone network of ResNet-50 where feature maps extracted from the conv5 body output. The input to the system is table crops in maximum size of 800×1333 and mini-batch size of 2 for nine epochs on the initial UNLV dataset. The epoch number is high compared to our dataset size, but our experiments have shown that this training leads to optimal results. After training for nine epochs, the UNLV dataset is stretched, as mentioned, to make the rows visually more salient because initial models performed horribly. This degradation in performance on row detection can easily be explained by the max pools and large filter sizes of backbone networks (i.e. ResNet Figure 3.2). Hence, stretching the images *vertically* allowed us to improve the Average Precision (AP) of our methods by up to 60%. However, when one investigates the results, we see that RetinaNet gains recall with the stretched training, but AP degrades. Which we believe caused by the absence of an RPN-like structure to limit the search space. That means stretched data help the model find rows and columns better, but it is overwhelmed by large images and the increasing amount of anchor boxes; thus, cannot make accurate predictions. After we obtain the stretched dataset, we fine-tuned the RetinaNet network for one more epoch. We observed that training this dataset from scratch with the stretched version of our dataset prevents the network from converging onto an equivalent AP. This is due to the sizes of the images. Since the image sizes grow, the network starts to has a hard time detecting objects.

Our final deep learning method under study is Mask-RCNN [11]. In essence, Mask-RCNN (Figure 3.8) is a two-stage instance segmentation network based on the previous iteration of the R-CNN family, Faster-RCNN [12]. Mask-RCNN utilizes the Faster-RCNN architecture as it is, the only addition made is the FCN branch added in parallel to the classification and regression branches. Where regression and classification branches predict refined bounding boxes and associated classes from the object area proposals, the newly implemented FCN branch generates pixel-level class-agnostic classification outputs in parallel to these branches. Pixel-level classifications are realized with the deconvolutional layers where the forward and backward processes of convolutional layers reversed to upsample the features inputted by the previous layer. Upsampling through convolutional layers in-

stead of fully connected layers allows the network to retain spatial information extracted through the backbone feature extractor and is able to reconstruct the image correctly and classify each pixel [9]. Another improvement that helped for accurate reconstruction is RoIAlign algorithm which allowed the network to separate RoIs from the extracted feature map correctly. Our backbone in Mask-RCNN model is the ResNet-50 as in RetinaNet and Faster-RCNN. However, in Faster-RCNN, ResNet-50 is not constructed with an FPN to use both low-level and semantically strong high-level features, and the best results are attained with when we extracted features from the conv4 block. In Mask-RCNN, ResNet-50 is constructed with FPN, where each convolutional block output (conv1, ..., conv5) is connected to the top-down pathway of FPN (Figure 3.9). We, again, generate 2000 proposals in the first stage with RPN. In this set of experiments, the input size is set to be the same as RetinaNet, and input images are resized (if needed) to fit our maximum input size constraint, 800×1333 . We use a ResNet-50 backbone network that is pre-trained on the ImageNet [92] dataset. Then, mini-batch size is set to 2, and the network is trained (fine-tuned) for 30 epochs on the non-stretched UNLV dataset with GroupNorm layer. Following the initial training, we train for 2 more epochs with the stretched UNLV dataset. We observe more than 19% increase in the F-Score of row detection and more than 10% increase in the F-Score of column detection performance contributed by the extra training with stretched dataset. These observations are in alignment with what we have seen in RetinaNet, and this is due to the loss to the network being more meaningful with the stretched dataset. When trained only on the stretched dataset from scratch, the model could not achieve the same performance. We believe large input sizes (images are at least doubled in size) turns the problem into more difficult one that the network cannot learn with little data.

Our final tested method, Mathematical Morphology is developed to analyze the geometrical structures based on strong mathematical foundations [22]. Moreover, in the past, it is used to detect tables in a bottom-up manner by detecting columns first [31]. Inspired by this work and the success of the heuristics based solutions used to detect tabular structures (Subsection 2.1.1), it is explicit that tabular structures are either aligned or can be inferred with the white space checks without the need for complex operations such as object detection methods backed with deep learning which suffers from the long training times. We employ mathematical morphology method with a set of basic heuristics to detect table structure. We previously presented that rows and columns are harder to detect than tables because many subtle details need to be addressed such as when there are nested tables as shown in Figure 4.5c. The method to find structures operate on the concept of the connected components as many other previous works did [109, 8, 111]. Finding connected components has to be carried out carefully because if a mistake is made when

obtaining them, there is no easy way to fix it afterwards. To find connected components, we first apply morphological closing (Figure 3.12). The closing operation first dilates the white pixels in the image and then erodes which connects white pixels even if there are black spots in between if they are close to each other. We choose a rectangular filter of size 2×8 because letters are usually longer than they are wider. Closing operation connects the words into single connected components, but with minor errors. Therefore, we are obligated to conduct post-processing step to check IoU of these connected components and then merge the connected components that may belong to the same word. Another problem was that in the left-alignment case, where some words are connected with the lines of the tables which creates a problem. Therefore, we limited the maximum size of the connected components and applied *line deletion* operation before finding connected components to prevent this issue. First, we detect the lines by first finding the edges with the Canny Edge Detector [124] followed with Probabilistic Hough Transform [125]. Experiments are conducted to find the best parameters. After straight and continuous lines are found, they are replaced with black pixels (image is black-and-white where the text is white).

When the connected components are finalized, they are checked only against the alignments between them, and we acknowledge that this method can be further improved with additional checks. The threshold for horizontal alignments is set to be the average height of all connected components which turns out to be a value that is close to the font size of the document. The vertical alignment threshold is set to be the average width of all connected components. Which gives the average length of a word and therefore, every column is expected to have similar widths for perfect results. Extracted connected components are checked against horizontal alignment with the alignment threshold to be potentially classified as rows. If any two connected components align vertically on the left, right or in the middle, we mark these connected components to possibly belong to the same columns. These operations are done on every pair of connected components which then concludes the structure detection by finally eliminating wrong detections with IoU checks.

5.3 Training Details

Since the data is limited, whenever we refer to a backbone network, we refer to a pre-trained network that is either initially trained on MS COCO [100] or ImageNet [92] natural image datasets. On top of these networks, we apply transfer learning to fine-tune the feature extraction layers of the CNNs. Mathematical morphology based heuristics approach do not require training and creation of the heuristics to detect rows and columns

are conducted based on a couple of images sampled from the training set with distinct characteristics. Although it is not a trainable approach, constructing accurate and meaningful heuristics are troublesome and time-consuming, and we observed that for images with many connected components, it takes longer than deep learning based models (up to 30 seconds). After the prediction are made, non-maximum suppression is applied to eliminate overlapping classifications. Training the backbone networks and object detection models includes training of the backbone network, RPN (if included) and detection branches; classification, regression and FCN, where applicable. During training, we did not use a dedicated validation set to maximize the available training set. Hence, we validate the models by performing detections on other datasets where we do not have the annotations. Therefore, validation is done by visually observing the outputs of the trained models on the data they have never seen before.

The Faster-RCNN model with VGG-16 backbone subnetwork is trained with a mini-batch size of 1 image with maximum 128 RoIs for 15 epochs with the maximum input size to model is set to 800×1333 . All the training is carried out with the initial learning rate of 0.001, and we gradually decrease the learning rate throughout the training. Training the same network with the stretched dataset more than doubled the recall and tripled the AP of the network. Since we already know the non-stretched version of the dataset is not worth trying in default settings, the Faster-RCNN model with ResNet-50-C4 backbone is trained on the stretched dataset with the same input size but with the mini-batch size of 2 images with 128 RoIs per image for 20 epochs. ResNet based model is trained with an initial learning rate of 0.01, and the learning rate is gradually decreased as others. Both modules scale the shortest side of the image to be 800 pixels long.

RetinaNet model is trained with ResNet-50 backbone with an initial learning rate of 0.001 with the scheduled decrease on specific points in training. We used 3 of the conv body outputs of the ResNet model and connect them to form the FPN. The inputs are scaled to make their smallest side 800 pixels in length whereas maximum allowed input resolution is 800×1333 . The training of the network is conducted with the mini-batch size of 2 images for nine epochs on the non-stretched version of the UNLV dataset. The performance of the model in this stage was competitive for column detection. However, two-stage networks compare better for row detection. Following the initial training, instead of re-training the network from the same pre-trained foundations, we continued training for one more epoch on the stretched version of the UNLV dataset with the learning rate of 0.001 as we found that this process trains the model better.

Our last deep learning method Mask-RCNN is equipped with the pre-trained ResNet-50 backbone with FPN as in the RetinaNet model. Although it has similar foundations, Mask-RCNN employs an RPN and FCN in addition to the RetinaNet model. The input

constraints for Mask-RCNN based model is the same as our previous models; where the maximum input image size is 800×1333 . The initial learning rate of the model was 0.01, and a scheduled decrease is applied during training. FPN is created with the underlying ResNet-50 network from the convolutional building blocks. Mini-batch size was set to 2 images but with GroupNorm layer in the ResNet network instead of BatchNorm. This model is also trained first on the non-stretched UNLV dataset for 30 epochs, which is then followed by training on the stretched dataset for two more epochs. Another interesting note here that when we stretched the dataset, even before training the model on this new dataset, we observed 13% improvement on the F-Score for rows. Further training increased the F-Score further which totalled to 20% improvement from the non-stretched score. Our experiments show that, for all our models, fine-tuning the pre-trained backbone networks on the stretched dataset prevents the models from converging to acceptable detections (empirically and F-Score wise). Also, it is noteworthy to mention that when the images are stretched, they take significantly longer to test. For example, although Mask-RCNN model takes 0.4 seconds per image whereas for some stretched images, it takes more than 1.5 seconds. We conclude that since they take longer and the image is more distorted as we stretch, the dataset becomes harder to learn for models when they are complete strangers to the domain. However, when they are fine-tuned to the document detection domain, this extra training means that we perform another transfer learning and fine-tune these networks to the stretched document domain. The selection of the final weights is as follows; Every thousandth step during the training (every step is a feed-forward processing of a mini-batch), validation set accuracy has been tested and the most successful weights are selected to have the final model. To make a comparison and show that pre-training is necessary, we also train and test the same network (Mask-RCNN with ResNet-50) without pre-trained weights. Training times for Mask-RCNN models are 11 hours and 30 minutes for both pre-trained and not pre-trained models on the non-stretched version of our dataset. Then, the best weights are selected as described above and the training continued with them for 6 more hours on the stretched dataset. Final weights of the model is then selected by validating them again with our validation set. The model with randomly initialized weights (trained from scratch) achieves an optimal validation accuracy on the 20th epoch. Although it reaches its maximum capacity earlier (after the 20th epoch the model did not improve itself and started overfitting), we show in the next section that, it cannot perform as good as the pre-trained version of the same model.

Table 5.1: Comparison of considered methods.

Models	UNLV[126]					
	Row			Column		
	Recall	AP	F1	Recall	AP	F1
Faster-RCNN-VGG16	0.3517	0.1976	0.2530	0.6891	0.6630	0.6758
Mask-RCNN	0.7840	0.7750	0.7794	0.8571	0.8324	0.8446
RetinaNet	0.6975	0.5856	0.6367	0.9221	0.8701	0.8953
Faster-RCNN-VGG16-S	0.8191	0.7642	0.7907	0.8480	0.8047	0.8258
Faster-RCNN-R50-C4-S	0.9422	0.8903	0.9156	0.9657	0.9569	0.9613
Mask-RCNN-S	0.9802	0.9573	0.9686	0.9853	0.9828	0.9840
Mask-RCNN-S-Scratch	0.9573	0.9393	0.9482	0.9498	0.9447	0.9472
RetinaNet-S	0.8359	0.4498	0.5848	0.9216	0.7117	0.8031
Morphology	0.4650	0.2389	0.3157	0.7059	0.5086	0.5912

5.4 Results

In Table 5.1 we show the recall, AP and the F1-Scores of our methods on our test split of the UNLV dataset. The results are formatted in order of method name, backbone network name and S denotes that the network is trained on the stretched dataset.

However, before getting into the results, speed comparisons of the methods are presented. Tables include critical pieces of information for supply chains and to many other organizations and optimizing and digitizing the operations require the information extraction from the tables to be accurate as well as fast too. Because tables can be in terms of millions or even billions or sometimes processing the documents may be required to be completed in near real time (i.e. hospitals). The deep learning models are tested on a single GPU whereas the morphological method runs on a modern CPU with no hardware acceleration. The Faster-RCNN model with VGG-16 backbone takes 0.3 seconds on average and the Faster-RCNN with ResNet-50-C4 feature extractor averages to 0.4 seconds per image. The Mask-RCNN model with ResNet-50 backbone sometimes hits the memory limit of our GPU and triggers the re-caching and averages to 0.6 seconds per image. However, some of the stretched images can take up to two seconds. While Mask-RCNN is comparably slower than the Faster-RCNN and RetinaNet, it outperforms these models. RetinaNet model, on the other hand, averages out to 0.2 seconds per image which is the lowest average time per image amongst our proposals. However, the metrics are not on par with the two-stage networks. Finally, our morphology based heuristical method averages to 4.2 seconds per image but it is important to note that it does not require training. Training of the deep learning methods took about 12 hours each.

As we can observe in Table 5.1, before stretching the datasets the performance of all the models were mediocre. Exceptionally, Mask-RCNN performed remarkably even without the stretching is performed and we conclude that this is not strictly due to FPN architecture but FCN because FCN reconstructs the input image by upsampling the extracted features with the deconvolution layers. To reconstruct the image and classify each pixel, FCN should possess features that have semantically strong information to retain the details in the image. Therefore, the features that represent the rows are getting stronger signals hence the detection performance and we conclude that lower-level information is needed for better detection of rows. The same phenomenon is seen in Mask-RCNN model even when it is trained from scratch. When the Mask-RCNN is trained from randomly initialized weights, it is denoted as 'Scratch' in the table. Even when it is trained from scratch instead of using pre-trained weights, it outperforms all other methods in terms of row detection performance. It is behind the pre-trained version of the network but still achieves a competitive performance. The reason behind the good performance of the scratch training

proves that actual benefit of the Mask-RCNN method is the architectures that makes it stand out such as FPN and FCN. Both Mask-RCNN models use ResNet-50 backbone architecture. When the models that are trained on non-stretched UNLV dataset and the stretched dataset (denoted by 'S') are compared, we observe that stretching helps immensely and is proposed as a quick fix for this issue. A more robust and intrinsic fix would be re-training the ResNet networks by changing (or eliminating) max pooling layers and altering the filter sizes. However, it is not possible to train these networks from scratch with the amount of data for this task and processing power available. An interesting attempt at this would be increasing the amount of data by using *weakly supervised learning* which is a method of training where the training data is growing as the classifiers improve [127]. Labelling structures of the tables are labour intensive task due to the number of objects in a single table. However, weakly supervised learning can be initiated with the amount of data we have and it would hypothetically boost the dataset on the go. We leave further investigation on this for future work.

As we observe in Table 5.1, stretching the dataset and fine-tuning on this dataset noticeably improved the detection metrics, specifically for the rows. Column detection metrics is also improved, and this is due to the signals becoming more meaningful to the networks. When the data is not stretched, the networks are clueless as to why they are being punished since feature map do not hold enough information regarding the rows. Whereas when the images are stretched, they retain more information on rows since they encapsulate more pixels in the image. We observed that F-Score for row detection improved by 19% for Mask-RCNN, 54% for Faster-RCNN with VGG-16 but it decreased by about 5% for RetinaNet. The RetinaNet model attained a lower F-Score, except a higher recall value is observed with lesser AP. This phenomenon may mean that RetinaNet model became braver and predicted more rows with less accuracy. Predictions made by the RetinaNet model can be observed in Figure 5.4. Moreover, we can observe that the Faster-RCNN model with simple ResNet-50-C4 backbone with no FPN, FCN or GroupNorm layer performs almost as good as the Mask-RCNN model where the difference is only 5% in F-Score. It is shown by our experiments that lower-level features are useful in detecting rows because in some tables there are many rows (up to 40) and they are narrow in height in an image (less than 20 pixels). Adding to this, ResNet networks have 7x7 convolutional filter followed by 3x3 max-pooling layer as their first 2 layers. In turn, at the second layer, 21 pixels of the raw image is shrunked to only 1. This shrinking is further continued on every following layer and semantic information regarding rows are lost. This is why FPNs are useful. Example predictions for both of the Faster-RCNN based models can be seen in Figure 5.2 and Figure 5.3. However, Mask-RCNN is the clear winner, and some example predictions can be viewed in Figure 5.5. Lastly, we find it intriguing that handcrafted rule-based morphology

method, although inaccurate, have the same phenomenon where columns have higher AP which supports the claim that rows are harder to detect. Moreover, it shows how innately deep learning methods behave in deciding on a prediction and are parallel to our heuristics based method. Example predictions from our morphology-based method can be found in Figure 5.6.

It is worth noting that we do not benefit from the metadata extracted from either the image or the PDF as in [74]. However, adding the extracted metadata (i.e. text from the detected table locations) to one of the later CNN layers is compelling and may worth investigating.

5.5 Conclusion

In this chapter, three state-of-the-art object detectors are compared on the table structure detection task; Faster-RCNN, RetinaNet and Mask-RCNN. We train and test our methods with the public UNLV dataset with some heavily tilted examples removed, and remaining ones are stretched. Our selection includes two-stage networks as well as a single-stage deep learning method. Our findings are matching with our expectations where the single-stage method, RetinaNet, performed worse than the others but while being considerably faster (three times faster than the Mask-RCNN model). Another tested method, which is not a deep learning method, was the mathematical morphology and connected component based method which detected rows and columns based on the handcrafted heuristics. We tried this method because as it can be seen in Chapter 2, the concept of connected components and heuristics are popular in table detection problem although there are not many works focused on table structure detection. This is one of the novel contributions of ours. The best performing model is Mask-RCNN with better detections for both rows and columns. Mask-RCNN model was also the best model on table detection, which was the first stage of our proposed E2E system. The primary focus of this work is to systematically extract structured information from structured analog data such as tables from arbitrary documents. Related works chapter is enough to show the popularity of the task in the research community. Moreover, tech giants¹ are also proposing methods to extract information from tables. However, their operation is mostly limited to simple forms and documents and is expensive in the long run. Now that we have the first stage (table detection with Mask-RCNN) and the second stage (table structure detection with Mask-RCNN) ready, the next chapter, bridges them together and creates the Stitched

¹<https://aws.amazon.com/textextract/>

[illegible]

	Fields		Lubricants								
For the Year Ended December 31, 1993	Wholesale Fuels	Branded Marketing	Lubes	Related Fuels	Chemicals	Logistics	International Production	Canada (Sunoco)*	Corporate*	Consolidated	
Cash flows from operating activities:											
Income (loss)	\$ (49)	\$ 92	\$ 50	\$ (32)	\$ 13	\$ 56	\$ 73	\$ 33	\$ 52	\$ 298	
Noncash items included in income (loss)	80	95	7	22	—	4	64	178	(138)	312	
Net cash provided by operating activities:											
before working capital changes	\$ 31	\$ 187	\$ 57	\$ (10)	\$ 13	\$ 60	\$ 137	\$ 211	\$ (86)	600	
Changes in working capital pertaining:											
to operating activities										(187)	
Net cash provided by operating activities										413	
Cash flows from investing activities:											
Capital expenditures										(612)	
Cash provided by operations held for sale										154	
Proceeds from divestments										370	
Other										(26)	
Net cash used in investing activities										(114)	
Cash flows from financing activities:											
Cash used for debt repayments										(176)	
net of borrowings										(192)	
Cash dividend payments										8	
Other										(360)	
Net cash used in financing activities										(618)	
Decrease in cash and cash equivalents										\$ (61)	

(a)

Stratigraphic and rock description ¹	Depth meters (feet)	Thickness of interval, meters (feet)
Crater Flat Tuff-Continued Prow Pass Member--Continued Tuff, ash-flow, pale-yellowish-brown, moderately to densely welded, devitrified; pumice, zeolitized, grayish-orange to yellowish-gray. Phenocrysts and lithic fragments same as in unit above	1,196.5-1,208 (3,925-3,963)	11.5 (38)
Tuff, ash-flow, greenish-gray, pale grayish-yellow-green, zeolitic, moderately welded or compacted, devitrified; pumice, small (1-2 cm). Phenocrysts, 5 percent, mostly feldspar; lithic fragments, mainly 2-4 mm across, are volcanic rock and dark-red mudstone	1,208-1,212.7 (3,963-3,978.5)	4.7 (15.5)
Bedded Tuff Tuff, ash-fall, pale-pink and grayish-yellow-green, zeolitized, well-sorted; bedding dips 15°	1,212.7-1,212.8 (3,978.5-3,979)	0.1 (0.5)
Tuff, crudely bedded, reworked, moderate-reddish-brown	1,212.8-1,214.6 (3,979-3,985)	1.8 (6)
Crater Flat Tuff Bullfrog Member Tuff, ash-flow, grayish-red to pale-reddish-brown, moderately to densely welded, devitrified; pumice, brownish-gray to red-purple, argillized, 1-6 cm (0.4-2.4 in.) long. Sparse grayish-red volcanic lithic fragments. Phenocrysts 10-15 percent, sanidine, plagioclase, quartz, biotite, and hornblende	1,214.6-1,218.6 (3,985-3,998)	4 (13)

(b)

Figure 5.3: Example detection results by our Faster-RCNN based model with ResNet-50-C4 backbone network.

表 2-8 行为科学理论主要内容

人格管理理论	存在着“非正式组织”
激励理论	(1)需求层次理论：动机产生行为，需要产生动机，生理-安全-社会-尊重-自我实现需要 (2)人性假设理论：任何组织的管理者在其下属时，对其下属所持基本看法为“管理的假定”，有麦克雷格假设(X理论/Y理论)，超Y理论、Z理论和成熟-成熟理论 (3)群体行为理论：团体力学理论，主要论述了非正式团体的要素、目标、凝聚力、规范、结构、领导方式、规模、行为分类、参与者、组织者对变动的反应等 (4)敏感性训练理论 (5)强化理论 (6)期望理论
领导效能理论	领导者特质理论、行为连续带理论、领导方式系统理论、领导四象限模式、管理方格论、领导行为权变理论

表 2-9 现代管理理论主要内容

社会系统学派	组织是一个协作的系统；非正式组织是不属于正式组织的，是不受正式组织管理的个人联系和相互作用以及有关集体的总和；任何正式组织都包括协作意愿、共同目标和信息沟通；管理者在组织中的作用和职能是建立和维持一个信息联系的系统，协调组织成员的行动、规定、解释目标并处理个人目标与组织目标的矛盾
决策理论学派	管理就是决策；决策过程包括搜集情报阶段、拟定计划、选定计划阶段；决策的准则是制定出满意的标准，达到或超过这个标准就是可行的方案；程序化与非程序化决策；决策影响着组织机构的设置、决策的技术和方法
系统管理学派	系统管理的内容包括系统哲学、系统管理和系统分析。系统是一种有组织的综合整体，企业作为一个系统进行设计和经营，按照系统的要求进行组织和运行，确定一个最优的解决方法或实际行动
经验主义学派	重点分析成功管理者实际管理的经验，并加以概括，总结出成功经验共性，使之系统化、合理化，提出建议
管理科学学派	制定和运用数学模型与程序系统来表示计划、组织和控制活动，通过定量分析，选择最优方案达到组织的目标。特征是将数学、计算机和系统观念引入管理方法中，其主要方法包括线性规划法、决策树法、计划评审及关键路线法、模拟、概率论、对策论、排队论、库存论和搜索论
权变理论学派	“如果...那么”关系理论权变理论的面相展线：主要基于主观领导效能理论和组织结构的设计

三、管理学理论应用于水资源管理

将管理学基本理论应用于水资源管理，具有积极的意义。主要表现在以下几个方面。

1. 以人为本，挖掘水资源管理者潜能。人是水资源管理制定者和执行者，水资源管理活动必须以人为基本前提，做好水资源管理者的工作管理。为此，必须充分认识人的个体差异，采用多种手段发挥水资源管理者潜能。在分配工作的时候，要充分注意到每个人的工作与其能力相“匹配”，真正做到发挥其潜力，同时让水资源管理者有信心完成本职工作。在实际工作中，充分发挥水资源管理的积极参与热情，建立与之相配套的组织文化，对每个水资源管理者在

CONDUCTIVE POLYMER ALUMINUM SOLID ELECTROLYTIC CAPACITORS

PCS Chip Type, LongLife Assurance

- Load life of 5000 hours at 105°C.
- SMD type: Lead free reflow soldering condition at 260°C peak correspondence.
- Compliant to the RoHS directive (2011/65/EU).

PCS LongLife **PCF**

Specifications		Performance Characteristics	
Category	Temperature Range: -55 to +105°C	Rated Voltage Range	4 to 16V
Rated Capacitance Range	22 to 560µF	Capacitance Tolerance	±20% at 120Hz, 20°C
Target of loss angle (tan δ)	Less than or equal to the specified value at 100Hz, 20°C	ESR (Ω)	Less than or equal to the specified value at 100Hz, 20°C
Leakage Current (i _l)	Less than or equal to the specified value. After 2 minutes' application of rated voltage at 20°C	Temperature Characteristics (Max. Impedance Ratio)	Z _{105°C} /Z _{20°C} ≤ 1.25 (100kHz) Z _{-55°C} /Z _{20°C} ≤ 1.25
Endurance	The specifications listed at right shall be met when the capacitors are restored to 20°C after the rated voltage is applied for 5000 hours at 105°C.	Capacitance change (tan δ)	Within ±10% of the initial capacitance value (i _l)
Damp Heat (Steady State)	The specifications listed at right shall be met when the capacitors are restored to 20°C after the rated voltage is applied for 1000 hours at 60°C, 90% RH.	ESR (i _l)	10% or less than the initial specified value
Resistance to Soldering Heat	After soldering the capacitor under the soldering conditions prescribed here, the capacitor shall meet the specifications listed at right. The heating shall be done at 150 to 200°C and for 60 to 180 sec. The duration for over 200°C temperature at capacitor surface shall not exceed 60 seconds. In case peak temperature is 250°C or less, reflow soldering shall be two times maximum. Measurement for solder temperature profile shall be made at the capacitor top and the terminal.	Capacitance change (tan δ)	Within ±10% of the initial capacitance value (i _l)
Marking	NAVY BLUE PRINT ON THE CASE TOP	ESR (i _l)	10% or less than the initial specified value

*1 ESR should be measured at both of the terminal ends closest where the terminals protrude through the plastic platform.
 *2 Conditioning: If any doubt arises, measure the leakage current after the voltage treatment of applying DC rated voltage continuously to the capacitor for 120 minutes at 105°C.
 *3 Initial value: The value before test of examination of resistance to soldering.

Dimensions

Type numbering system (Example: 6.3V 220µF)

PCS 6.3V 220µF

Taping code
 Size code
 Configuration
 Capacitance tolerance (±20%)
 Rated capacitance (220µF)
 Rated voltage (6.3V)
 Series name

Frequency coefficient of rated ripple current

Frequency	10Hz	1kHz	10kHz	100kHz or more
Coefficient	0.05	0.30	0.70	1.00

•Dimension table in next page.

(a)

(b)

Figure 5.4: Example detection results by our RetinaNet based model with ResNet-50 backbone network.

										Stratigraphic and rock description ¹			Depth meters (feet)	Thickness of interval, meters (feet)
										Crater Flat Tuff-Continued Prow Pass Member--Continued Tuff, ash-flow, pale-yellowish-brown, moderately to densely welded, devitrified; pumice, zeolitized, grayish-orange to yellowish-gray. Phenocrysts and lithic fragments same as in unit above			1,196.5-1,208 (3,925-3,963)	11.5 (38)
										Tuff, ash-flow, greenish-gray, pale grayish-yellow-green, zeolitic, moderately welded or compacted, devitrified; pumice, small (1-2 cm). Phenocrysts, 5 percent, mostly feldspar; lithic fragments, mainly 2-4 mm across, are volcanic rock and dark-red mudstone			1,208-1,212.7 (3,963-3,978.5)	4.7 (15.5)
										Bedded Tuff Tuff, ash-fall, pale-pink and grayish-yellow-green, zeolitized, well-sorted; bedding dips 15°			1,212.7-1,212.8 (3,978.5-3,979)	0.1 (0.5)
										Tuff, crudely bedded, reworked, moderate-reddish-brown			1,212.8-1,214.6 (3,979-3,985)	1.8 (6)
										Crater Flat Tuff Bullfrog Member Tuff, ash-flow, grayish-red to pale-reddish-brown, moderately to densely welded, devitrified; pumice, brownish-gray to red-purple, argillized, 1-6 cm (0.4-2.4 in.) long. Sparse grayish-red volcanic lithic fragments. Phenocrysts 10-15 percent, sanidine, plagioclase, quartz, biotite, and hornblende			1,214.6-1,218.6 (3,985-3,998)	4 (13)

	Fuels		Lubricants																
	Wholesale Fuels	Branded Marketing	Lubes	Related Fuels	Chemicals	Logistics	International Production	Canada (Sunco)*	Corporate**	Consolidated									
For the Year Ended December 31, 1993																			
Cash flows from operating activities:																			
Income (loss)	\$(49)	\$ 92	\$50	\$(32)	\$13	\$56	\$ 73	\$ 33	\$ 52	\$286									
Noncash items included in income (loss)	80	95	7	22	—	4	64	178	(138)	312									
Net cash provided by operating activities:																			
before working capital changes	\$ 31	\$187	\$57	\$(10)	\$13	\$60	\$137	\$211	\$ (86)	600									
Changes in working capital pertaining to operating activities										(187)									
Net cash provided by operating activities										413									
Cash flows from investing activities:																			
Capital expenditures										(612)									
Cash provided by operations held for sale										154									
Proceeds from divestments										370									
Other										(26)									
Net cash used in investing activities										(114)									
Cash flows from financing activities:																			
Cash used for debt repayments net of borrowings										(176)									
Cash dividend payments										(192)									
Other										8									
Net cash used in financing activities										(360)									
Decrease in cash and cash equivalents										\$ (61)									

Figure 5.5: Example detection results by our Mask-RCNN based model with ResNet-50 backbone feature extractor.

(a)

(b)

99

E2E system. Alongside our Stitched E2E system, we propose two additional networks to detect everything at once, unlike the two-stage approach and we present the details and comparisons for the proposed E2E systems in the next chapter.

Chapter 6

End-to-End Tabular Structure Detection

In this thesis, we focus on the digitization of documents and detection of structured areas where information is stored densely. Structured areas include formulas, tables, figures and can be extended even more. We focus on information extraction from the tables and table-like areas and propose a two-stage method. The first-stage is table detection Chapter (4) where we propose to use state-of-the-art object detectors and conclude that the Mask-RCNN [11] based model performs best on this task. The first-stage did not include any pre-processing or post-processing. The second-stage was table structure detection (Chapter 5) where we propose solutions with deep learning based methods and a morphology-based method. In the second stage, we apply a pre-processing to the dataset which is calculated from the training set itself. We stretch the dataset vertically according to our calculations to make rows visually more salient. Although the same calculations presented that we should stretch columns by 4%, we did not apply this, and we think this is insignificant and would not affect the results. The second stage showed us that the Mask-RCNN based model is also suitable for table structure detection task. Now, what is left is we need to combine these two stages and form our E2E solution for the table, row and column detection from the document, which we call Stitched E2E system. The Stitched E2E system includes some post-processing to the output of the first-stage, unlike the previous chapters. In addition to the Stitched version, this chapter proposes additional E2E methods to detect rows and columns E2E.

The Stitched E2E system makes use of two previous Mask-RCNN models; table detection model and table structure detection model as shown in Figure 6.1a. We introduce three post-processing steps; (i) Agreement module, (ii) Judge module and (iii) Table classifier.

Agreement module has two components where tables from the first-stage are compared to each other and the lines of in the document that may belong to the table. The flow of our table agreement module can be seen in Figure 6.3. A table is compared with other tables from the same image to check for IoU between them. As a result of this check, either some tables are eliminated (false-positives), or they are merged or extended to cover the region better. Then, a line detection algorithm is applied (similar to the one we applied in the morphology method), and a tight bounding box is formed to encapsulate intersecting lines. These lines are then used to amend the tables, where the overall algorithm can be seen in Figure 6.4. Another module we introduce in this chapter is a table classifier. This module is a binary classifier which is applied to the outputs of the agreement module to approve the tables and differentiate the false-positives. It is a CNN and is similar to the backbone networks mentioned previously. One difference is that we do not cut the fully connected layers, but we replace them. We also propose two Pure E2E Mask-RCNN based methods to either detect tables, columns and rows in a single pass or rows and columns. The two E2E models perform similarly whereas the full detection model with tables and structures performed better and achieved the best Mean Average Precision (mAP) for E2E detection. mAP is calculated with 11-point average algorithm as proposed by PASCAL VOC [96]. The purpose of this algorithm is to draw precision-recall curve and calculate the AP from this curve which gives a better idea about the performance of the model. Then the AP is averaged over all classes to acquire mAP. We conclude that this difference in performance stems from the additional table class which allows the model to learn better representations and hierarchical features. The new models are trained on the UNLV dataset. Summary of the flow of the Pure E2E model can be seen in Figure 6.1b. The judge model takes one of the Pure E2E trained models and follows a procedure that is similar to that of agreement module, and the formula can be seen in Eq. 6.2. In short, if a table has not enough IoU with detected structures, it is rejected in this stage. Since our E2E structure detection networks are not entirely accurate and are prone to errors, judge module can only eliminate a few false-positives. Even a single false-positive could possibly mean more than 15 false-positive table structure detections which is why we developed this step. We start this chapter by first examining the dataset in Section 6.1 followed by the introduction to the deep learning models and post-processing methods in Section 6.2. Then, briefly, we explain the training steps in Section 6.3 and we present our results in section 6.4 and conclude this chapter Section 6.5.

6.1 Dataset

We proposed three different approaches to achieve the E2E detection of structures and tables. The proposed Stitched E2E system relies on previously developed models. Hence, for the table detection model (Model A) and table structure detection model (Model B), no further training is required, and therefore, we do not require additional datasets. The used datasets were; Marmot, ICDAR 2013, ICDAR 2017, UNLV and our private dataset. Where the table detection methods trained on all of these datasets, but the table structure detection model only trained with UNLV dataset, and we acknowledge this problem and leave the rest for the future work (Chapter 5). Since the UNLV is such a small dataset (less than 500 data points), our proposed structure detection model which predicts rows and columns from the table crops is susceptible to variations and can result in predictions even though there is *nothing* in the image as in Figure 6.10d where many false-positive detections are generated. This being the cause, we propose a binary classifier on top of the table detection model to eliminate easy false-positives. In our Pure E2E detection models of which we trained from scratch in this chapter, we used UNLV dataset only. The annotations for rows and columns are used from the stretched UNLV dataset to train the first Pure E2E Mask-RCNN model (Model C) which outputs *only* row and column detections from the full document images. In another round, we use the annotations for both tables and table structures from the stretched UNLV dataset to train the second Pure E2E Mask-RCNN model (Model D). Model D output includes table structures as well as tables, and this addition has increased the prediction capabilities (7% increase in mAP) of the model.

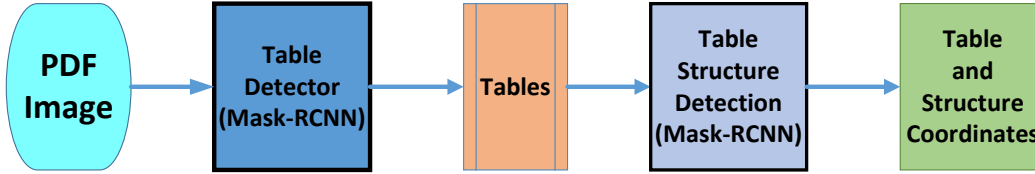
In our first experiments with the stitched E2E system, we achieved high recall values where we observed more than 96% recall. Since the aggregated recall for the structure detection system is 98% with 100% recall on the table detection, this results mean that with the current table detection outputs, the structure detection model performs nearly up to its limitations. However, there is undoubtedly room for improvement in precision. The precision is inadequate as a result of false-positives generated by the structure detection model. The false-positives mostly stems from partially correct table detections and the incapability of the structure detection model to notice false-positive tables which correlate to the limitations in the dataset. Example false-positive table detections that could lead to false-positive structure detections are demonstrated in Figure 6.9. Motivated by this observation, we introduced a binary classifier which is applied after the table detection model to eliminate some of the false-positive table crops. Proposed binary classifier needed a table dataset to learn to classify tables, and for this reason, we use the proposed table detection model (Model A). The datasets were used as is from Chapter 4 without any

pre-processing applied. Using the table detection model, true and false data points are generated and labelled, then used to train the classifiers. The dataset for this task is created from scratch. First, we analyzed the detections performed on the table detection training set. After performing table detection on all the images in the training set, we observed that the true-positive tables have the confidence score of more than 0.9997 on average and the false-positive detections averages to 0.54 confidence score. Therefore, to keep as much true-positive as possible, we do not perform the classification operation if the table detection has the confidence of 0.99 or higher. To create the dataset, the detections that have 0.8 or more IoU with ground truths are labelled as *table* and if they have a lower IoU, they are labelled as *not table*. The IoU threshold is set to 0.8 to allow the classifiers to grasp the idea of the tables more accurately when compared with the 0.5 which we use in our final models. That being said, setting the IoU threshold to a high value such as 80% did not hurt our vanilla results dramatically, and we still obtained a recall of 99.5% and AP of 95.2% which shows the strength of our table detection model. One downside of setting the threshold to a value as high as 0.8 is that it creates a class imbalance in the dataset. To solve this problem, similar to the RetinaNet, we employed the Focal Loss [13] function and trained both our models with this loss function. The focal loss allowed us to acquire additional 2% accuracy. In addition to training VGG-16 and our custom model, we also tested the ResNet-50 [14] and Inception [46]. However, they are excessively complex for this problem and could not converge to acceptable accuracy levels.

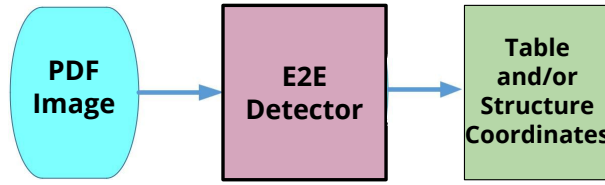
6.2 Methods

In this thesis, we propose three E2E tabular structure systems; (i) Stitched E2E system, (ii) Pure E2E table structure detection model and (iii) Pure E2E tabular structure detection model. The Stitched E2E system is based on the Mask-RCNN models proposed in Chapter 4 and Chapter 5. Both networks achieve state-of-the-art in their respective fields. However, as mentioned before, the structure detection network has limitations that resulted from the limited size of the dataset and the way the problem is formulated. This limitation is the false-positive detections from this model when the table detection is not accurate. Table structure detection dataset includes only perfect table crops where every table has a row or column class for each pixel. Hence, even when the table detection is entirely wrong, structures are detected, as demonstrated in Figure 6.10d. This problem is the downside of the first, Stitched E2E system.

The first model we propose solely for E2E structure detection purposes is Pure E2E table structure detection model, denoted as Model C. This model is a single Mask-RCNN



(a) Stitched E2E System Flow Diagram



(b) Flow Diagram of the Pure E2E Systems

Figure 6.1: System flow diagrams for proposed E2E systems.

model that is trained on the UNLV dataset to detect rows and columns. The main problem in this system was that rows are usually thin, less than 50 pixels in width and therefore, the detection results had severely impacted by this. To handle this problem, we applied the same stretching operation to make each row taller than 50 pixels and concluded that stretching to 250% of the original size, solve this problem on average and gave us around 20% increase in both recall and AP. The false-positive problem we mentioned above does not affect this model since this method relies on only one model and has no distinct components. The detection examples of the Model C can be found in Figure 6.6a.

Apart from the table structure detection, we trained the same model to detect all tabular structures which is similar to the Model C but includes table detections as well which we call Model D. The same width problem of the rows applies to this method as well, and the same stretching is applied. Model D achieved better F-Score over Model C which is caused by the inclusion of table annotations. This outcome shows that the addition of table detection improves the detection performance because learning what a table also improves the detection of rows and columns. Moreover, considering the design of this system, the method performs only a single pass over an image and therefore, is faster than the stitched model. The detection examples of the Model D can be found in Figure 6.2.

Although the systems that we call Pure E2E systems are faster and do not get influenced by any other model than themselves like the Stitched E2E system, the recall and mAP values attained by those models are not as high as we desire. Because, as we have discussed

in previous chapters, our primary aim is to detect and extract information from the tables with high accuracy and speed is not our main concern. The stitched E2E system consists of two-stages each of which is a Mask-RCNN model where the output of the first stage is fed into the second stage as input and is aimed to increase the detection recall. As can be seen in Figure 6.1a, first a PDF is converted to an image which is then inputted to the Model A to detect tables. Detected table areas are then cropped from the image, and these crops are fed into the Model B where the table structures are detected. The main concern with this method is that errors propagate and increase throughout the system. Hence, we also propose different post-processing techniques to solve the limitations of the proposed Stitched E2E system design. The limitation was the false-positive table structure detections originating from the faulty table detections. Our first proposal that targets this weakness is the Agreement module. The agreement module has two parts; (i) line agreement and (ii) table agreement. Line agreement aims to refine the edges of the bounding box to obtain better detections and since nothing is added or removed, this improvement is not aimed to increase precision of the table detection but to send better tables to the later stages. The table agreement algorithm looks at the tables in pairs and eliminates or refines tables according to empirically set thresholds. The second proposal is the judge model. Judge model takes the table detections from Model A and tabular structure detections from Model D which then compares them to decide if to keep the table or eliminate it. Lastly, we propose a table classifier on top of these to finalize our post-processing steps. The classifier aims to eliminate false-positive table detections before feeding them into the structure detection model.

Agreement Module

The Stitched E2E system works best when optimal table detections are fed into the table structure detection component. Heuristics are developed based on empirical observations which make up the agreement module to address the elimination, merging or expansion of detected tables. The agreement module takes in table coordinates and confidence scores as input to evaluate a final refined location. Based on our observations on the outputs of the table detection model, we found that the network seldom generated overlapping table predictions with varying confidence levels. The agreement module operates based on the IoU scores between tables and either tables or line bounding boxes. Line bounding boxes are defined as rectangular areas formed to cover intersecting lines tightly. The first module, table agreement module, checks the IoU of any two tables and if it is higher than 80% (an empirical value) and the larger table has lower confidence than the smaller one, we delete

the larger table and the new coordinates for the smaller table is determined by:

$$C_i = C_i + C_{ij}(1 - (Conf_i - Conf_j)) \quad (6.1)$$

where i is the smaller table and j is the larger table and C_{ij} denotes the difference of coordinates, and C_i is the coordinates of the smaller table. This formula takes the confidence levels of these boxes into account when expanding the smaller table. For instance, if the confidence difference between the smaller and the larger table is 50% (in favour of the smaller table), the small table will be expanded by 50% of the difference in coordinates. If the IoU is less than 80% but higher than 50% (an empirical value), there is no change. However, if the IoU is between 50% and 80% and the confidence of both of the tables are higher than 80%, the two table detections are merged into a single one. Hence, as Figure 6.3 shows, we never shrink a table, and they are always either vertically or horizontally expanded, aimed at solving partial detections.

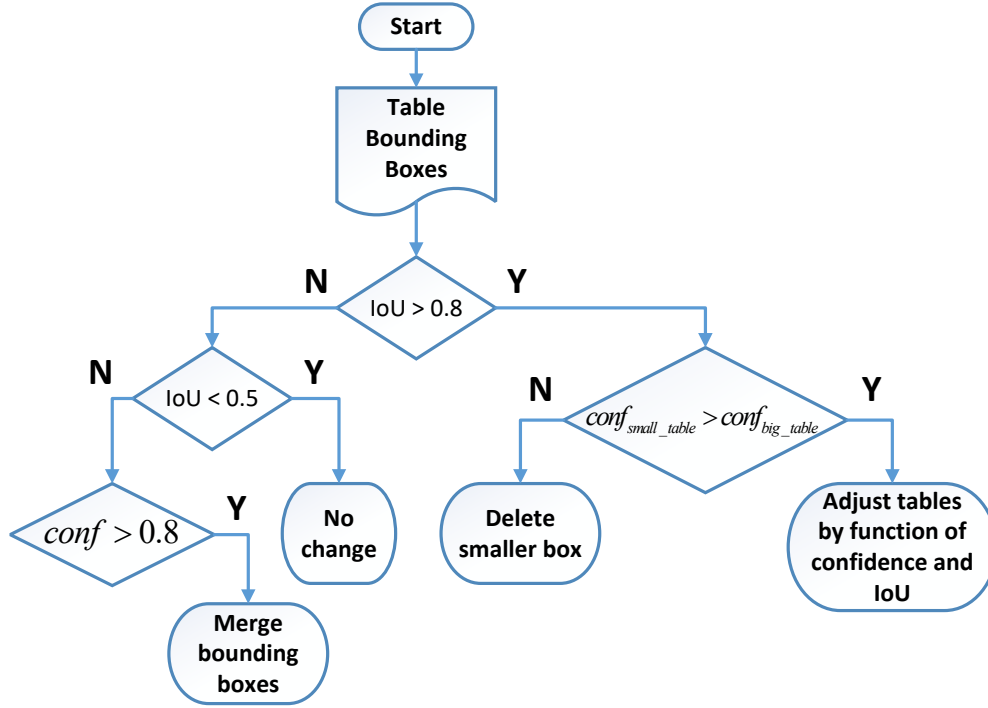


Figure 6.3: Agreement module - table agreement algorithm flow

An additional module to the table agreement module is called line agreement model

which is aimed at expanding (refining) table bounding boxes. First, tight bounding boxes that encapsulates intersecting lines are found which is created following the detection of continuous lines and the discovery of the intersections of these lines. Figure 6.4 shows the flow of this algorithm. Although not every table has lines that either on the table boundaries or inside the table area, we include the line agreement module to benefit from the information we can deduct from the lines which highlights where the tables are, whenever they are present. We set the IoU threshold between line boxes and table boxes to 80% which means when a table is correctly classified, but there are minor errors in coordinates we use the line agreement module to align the detection with the lines correctly. In Figure 6.5 we present an example that shows the line agreement module refining a table detection. Since in this method nothing is eliminated and there is no reduction in the number of false-positives, the effects are not easily measured by our metrics in the first stage. However, the second stage of our Stitched E2E system relies on accurate table detections from the first stage. Therefore, this module could be useful in some cases to eliminate false-positives in the second stage. The line detection algorithm is the same as we use in the mathematical morphology method. First, Canny Edge Detection [124] is used which is followed by the Probabilistic Hough Transform [125] with the parameters set to detect lines of tables and found empirically and by visual inspection. After the revelation of the lines, intersection points are found, if any, with Depth First Search (DFS) algorithm. Afterwards, the coordinates at the edges of the chunks of the intersecting lines are used to form bounding boxes from those lines. Then, these boxes are compared against the detections from the first stage and checked for IoU of 80% or higher. If that is the case, the table extended (but not moved) to cover the line bounding box and this concludes the agreement module. If IoU is less than 80%, no change is made. The thresholds are found empirically and open to improvements with careful investigation.

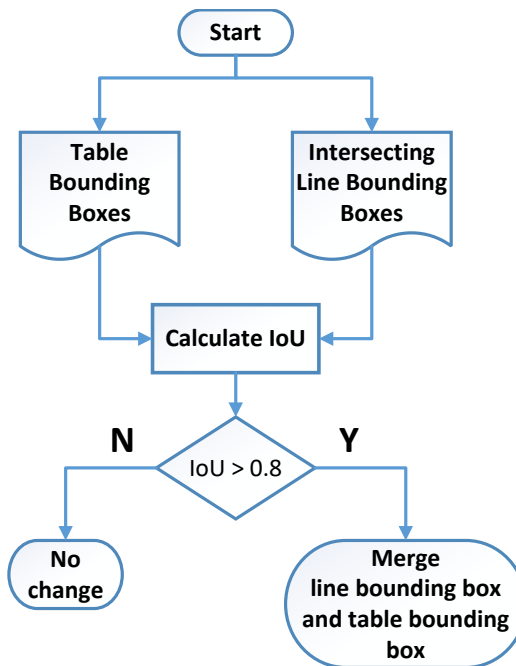


Figure 6.4: Agreement module - line agreement algorithm flow

Table 2.
Age-Adjusted and Unadjusted Disability Rates by Gender, Race, Hispanic Origin:
2005 and 2010

Category	Age-adjusted disability rate				Unadjusted disability rate			
	2005		2010		2005		2010	
	Estimate	Margin of error (s)	Estimate	Margin of error (s)	Estimate	Margin of error (s)	Estimate	Margin of error (s)
All people	18.6	0.3	18.1	0.3	16.7	0.3	18.7	0.3
Male	17.9	0.4	17.6	0.4	17.3	0.4	17.4	0.4
Female	19.0	0.3	18.3	0.4	20.1	0.3	19.8	0.4
White alone	17.9	0.3	17.4	0.3	16.6	0.3	18.5	0.3
Not Hispanic	18.1	0.4	17.6	0.4	19.7	0.4	19.8	0.4
Black alone	23.2	0.7	22.2	0.7	20.4	0.7	20.3	0.7
Not Hispanic	23.3	0.7	22.3	0.7	20.7	0.7	20.7	0.7
Asian Alone	14.5	1.3	14.5	1.1	12.4	1.2	13.0	1.0
Not Hispanic	14.6	1.3	14.4	1.1	12.5	1.2	13.0	1.1
Hispanic or Latino	18.4	0.9	17.8	0.7	13.1	0.7	13.2	0.6

— Represents or rounds to zero.
 * Denotes a statistically significant difference at the 90 percent confidence level.
 † Age-adjustments followed the methodology described in Anderson and Rosenberg (1998) using the year 2000 standard population by 5-year age groups from Day (1996).
 ‡ A margin of error is a measure of an estimate's variability. The larger the margin of error in relation to the size of the estimate, the less reliable the estimate. The margins of error shown in this table are for the 90 percent confidence level. For more information about the source and accuracy of the estimates, including margins of error, standard errors, and confidence intervals, see the Source and Accuracy Statement at <http://www.census.gov/ipeds/data/saas/2005-2010/SAAS-13.pdf>.
 Source: U.S. Census Bureau, Survey of Income and Program Participation, June–September 2005 and May–August 2010.

oldest group, more than half (55.8 percent) had a severe disability. Of individuals 55 to 64 years old and nearing retirement, about 6.0 percent needed assistance with one or more ADLs or IADLs. The percentage needing assistance for those aged 80 and older was about

rates for older age groups, and for people overall, would likely be higher.

Because age plays such an important factor in health and disability measurement, comparisons across other demographic groups

and females, and for different race and Hispanic origin groups.

At 18.7 percent, the unadjusted disability rate in 2010 was statistically unchanged from the rate in 2005, however, the aging of the population was a contributing fac-

Table 2.
Age-Adjusted and Unadjusted Disability Rates by Gender, Race, Hispanic Origin:
2005 and 2010

Category	Age-adjusted disability rate				Unadjusted disability rate			
	2005		2010		2005		2010	
	Estimate	Margin of error (s)	Estimate	Margin of error (s)	Estimate	Margin of error (s)	Estimate	Margin of error (s)
All people	18.6	0.3	18.1	0.3	16.7	0.3	18.7	0.3
Male	17.9	0.4	17.6	0.4	17.3	0.4	17.4	0.4
Female	19.0	0.3	18.3	0.4	20.1	0.3	19.8	0.4
White alone	17.9	0.3	17.4	0.3	16.6	0.3	18.5	0.3
Not Hispanic	18.1	0.4	17.6	0.4	19.7	0.4	19.8	0.4
Black alone	23.2	0.7	22.2	0.7	20.4	0.7	20.3	0.7
Not Hispanic	23.3	0.7	22.3	0.7	20.7	0.7	20.7	0.7
Asian Alone	14.5	1.3	14.5	1.1	12.4	1.2	13.0	1.0
Not Hispanic	14.6	1.3	14.4	1.1	12.5	1.2	13.0	1.1
Hispanic or Latino	18.4	0.9	17.8	0.7	13.1	0.7	13.2	0.6

— Represents or rounds to zero.
 * Denotes a statistically significant difference at the 90 percent confidence level.
 † Age-adjustments followed the methodology described in Anderson and Rosenberg (1998) using the year 2000 standard population by 5-year age groups from Day (1996).
 ‡ A margin of error is a measure of an estimate's variability. The larger the margin of error in relation to the size of the estimate, the less reliable the estimate. The margins of error shown in this table are for the 90 percent confidence level. For more information about the source and accuracy of the estimates, including margins of error, standard errors, and confidence intervals, see the Source and Accuracy Statement at <http://www.census.gov/ipeds/data/saas/2005-2010/SAAS-13.pdf>.
 Source: U.S. Census Bureau, Survey of Income and Program Participation, June–September 2005 and May–August 2010.

oldest group, more than half (55.8 percent) had a severe disability. Of individuals 55 to 64 years old and nearing retirement, about 6.0 percent needed assistance with one or more ADLs or IADLs. The percentage needing assistance for those aged 80 and older was about 5 times as large (30.2 percent).

rates for older age groups, and for people overall, would likely be higher.

Because age plays such an important factor in health and disability measurement, comparisons across other demographic groups and across time should take into

and females, and for different race and Hispanic origin groups.

At 18.7 percent, the unadjusted disability rate in 2010 was statistically unchanged from the rate in 2005, however, the aging of the population was a contributing factor in holding the disability rate

(a) Table Detection

(b) Line Bounding Box

Figure 6.5: Example case for the Line Agreement module steps.

Judge Module

We mentioned the shortcoming of our Stitched E2E system which is caused by the nature of our approach and presented one correction, agreement module. Similar to the agreement module, the judge module is developed to address the false-positives introduced by the table detection model and to prevent the snowballing with the table structure detection model. It is developed to eliminate some of the remaining misclassifications to improve the overall performance of our E2E system. As presented in Figure 6.8a, images of documents are sent into both Model A (table detection) and Model D (or Model C, Pure E2E models) to assess the tables by the table detector whereas, previously we had only used table detection output as is. Both Model A and Model D takes whole page images as input and Model A outputs only table detections where Model D predicts only the structure of tables. What exactly the *Judge System* sees is illustrated in Figure 6.6a and Figure 6.6b. Upon these predictions, the judge compares the IoU and checks if a detected table overlaps for more than a threshold with any of the rows and columns. IoU is previously defined in Eq. 4.1. The operations of the judge system can be summarized as:

$$JudgeApprove_i = \begin{cases} True, & IoU_{i,j} > 1 - Conf_i \\ False, & \text{otherwise} \end{cases} \quad (6.2)$$

Where the algorithm compares the overlapping areas (IoU) to the threshold $1 - Conf_i$ where i denotes the i -th table detection in the image and j is the j -th table structure detection, either row or column. $Conf_i$ denotes the confidence of the model for the table i . This threshold means that if the table confidence is 100%, that is the Model A is 100% sure that bounding box is a table, we do not have to work through the judging mechanism. However, as the confidence score drops, more and more overlaps might be required, or else, the table is rejected and deleted. Surviving tables proceeds to the second stage, table structure detection. The most significant problem with the judge system is that it needs to perform three detections for a single image and helper models (Model C or Model D) are not as accurate as Model A (table detection). Hence, it slows down the E2E system, and it might not be stable in some cases.

Binary Table Classifier

Previously mentioned post-processing steps applied to table detection output could only remove a few false-positives. The numerical results show that each false detected table region (false-positive) results in 15 false-positive structure detection on average. Hence, even though we achieve state-of-the-art metrics on table detection, E2E structure detection

TABLE IV			
¹³¹ I MEASURED IN J-13/HBR SOLUTION SAMPLES			
Test	Days	¹³¹ I (pCi/ml)*	±10 ⁻⁴ Inventory**
Bare Fuel	63	0.52	5.6
	223	0.72	7.5
Slit Defect	63	0.29	3.0
	223	0.38	4.0
Hole Defects	63	0.0019	0.020
	223	0.0050	0.053
Undelected	63	0.0060	0.064
	223	0.0054	0.057

*Average for 2 replicate samples measured by neutron activation analysis.
**(pCi/ml)(250 ml)/(10⁻⁴ of specimen inventory).

TABLE V				
H. B. ROBINSON FUEL IN J-13 WATER URANIUM RELEASE DATA (μg)				
	Bare Fuel	Slit Defect	Hole Defects	Undelected
[Solution Samples]	211	4.3	0.50	1.26
[Rod Samples]	34	0.3	<0.17	<0.19
Final Solution	300	23.8	1.50	3.25
[U (μg/ml)]*	(1.2)	(0.095)	(0.006)	(0.013)
Strip	2700	1.5	0.60	0.60
Rinse**	550	1.8	0.60	0.27
[Above]	3795	31.7	<3.37	<5.57
Divided by 10 ⁻³ Inv.	5.42	0.044	<0.0047	<0.008

*Unfiltered 223-day final solution uranium content in μg/ml given in parentheses.
**Bare fuel rinse solution was 0.4 μm filtered before analysis.

(a) Pure E2E Table Structure Detection (Model C)

TABLE IV			
¹³¹ I MEASURED IN J-13/HBR SOLUTION SAMPLES			
Test	Days	¹³¹ I (pCi/ml)*	±10 ⁻³ Inventory**
Bare Fuel	63	0.52	5.6
	223	0.72	7.5
Slit Defect	63	0.29	3.0
	223	0.38	4.0
Hole Defects	63	0.0019	0.020
	223	0.0050	0.053
Undelected	63	0.0060	0.064
	223	0.0054	0.057

*Average for 2 replicate samples measured by neutron activation analysis.
** $(\text{pCi/ml})(250 \text{ ml})/(10^{-3} \text{ of specimen inventory})$.

TABLE V				
H. B. ROBINSON FUEL IN J-13 WATER URANIUM RELEASE DATA (μg)				
	Bare Fuel	Slit Defect	Hole Defects	Undelected
[Solution Samples]	211	4.3	0.50	1.26
[Rod Samples]	34	0.3	<0.17	<0.19
Final Solution [U (μg/ml)]*	300 (1.2)	23.8 (0.095)	1.50 (0.006)	3.25 (0.013)
Strip	2700	1.5	0.60	0.60
Rinse**	550	1.8	0.60	0.27
[Above]	3795	31.7	<3.37	<5.57
Divided by 10 ⁻³ Inv.	5.42	0.044	<0.0047	<0.008

*Unfiltered 223-day final solution uranium content in μg/ml given in parentheses.
**Bare fuel rinse solution was 0.4 μm filtered before analysis.

(b) Table Detection

Figure 6.6: Example cases for the Judge mechanism . Different colors stand for segmentation of different instances of the same class.

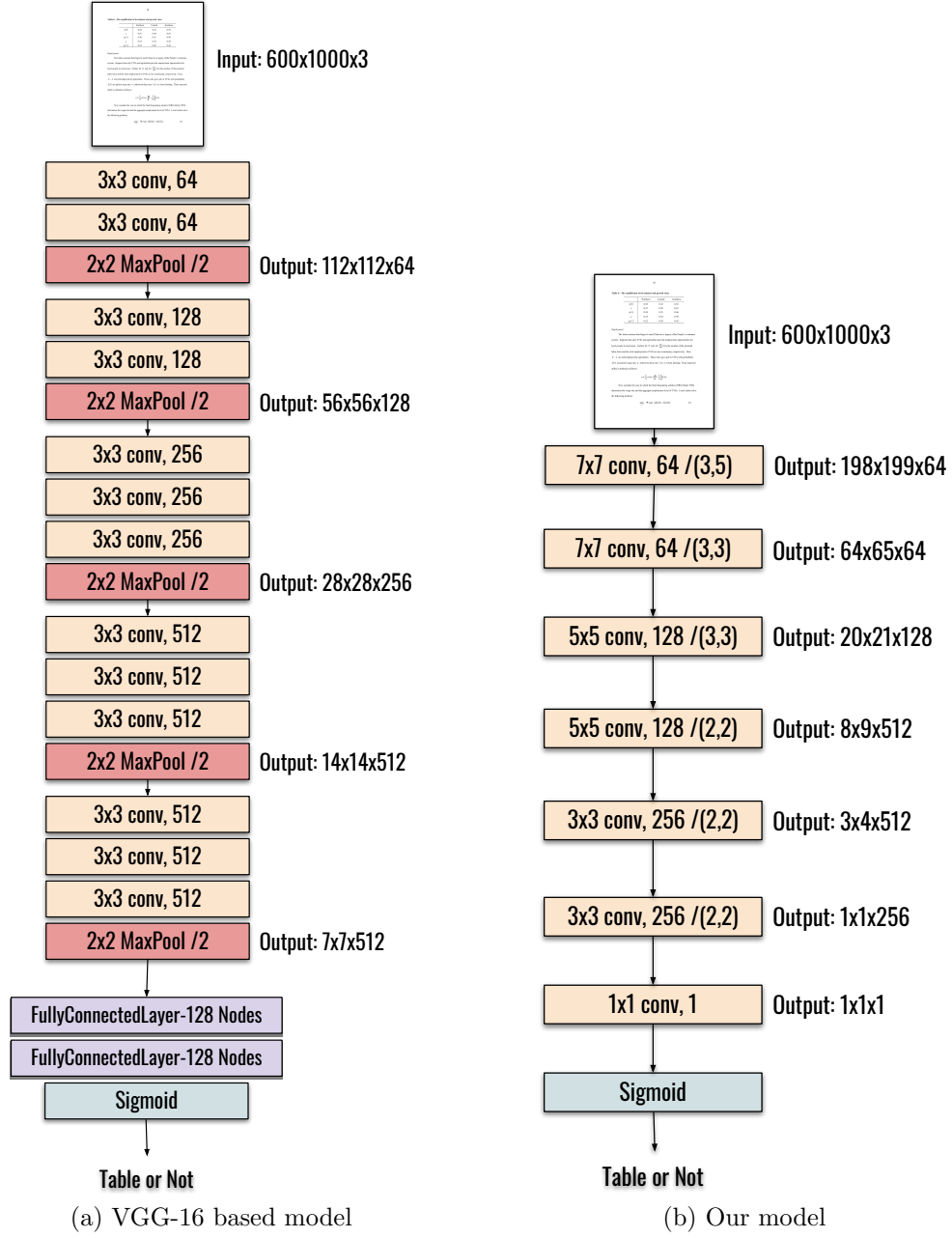


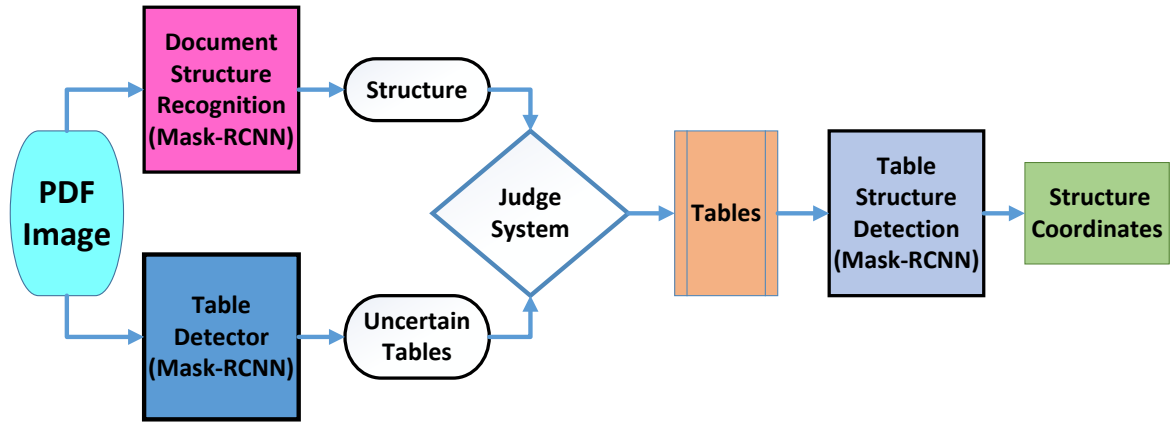
Figure 6.7: The architectures of the proposed binary table classifiers.

lacks in AP metric due to the false-positives. To further remove false-positives without

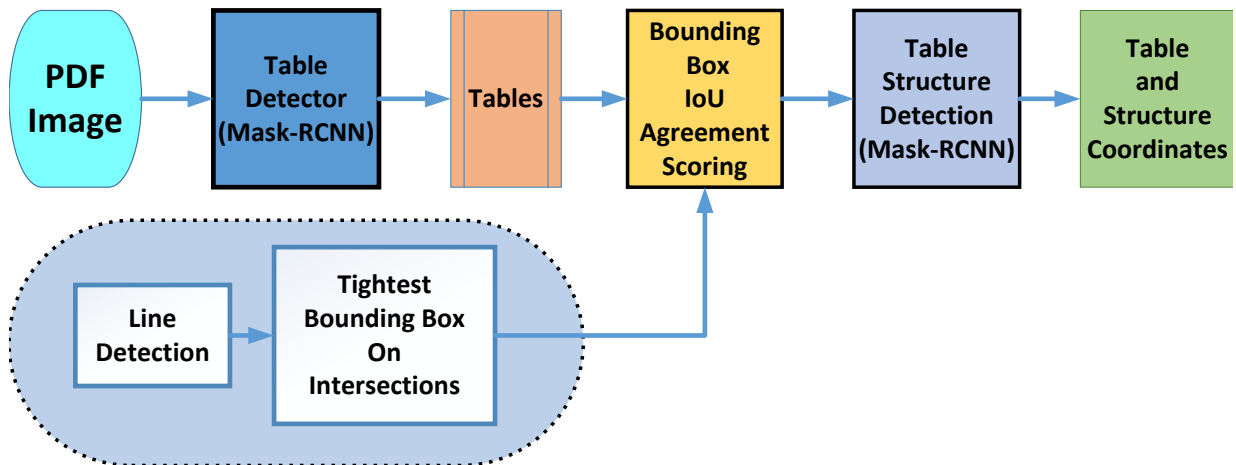
dropping many true-positives, we developed a binary table classifier based on deep learning. Overall flow of the Stitched E2E model can be seen in Figure 6.8c. Since this is a classification task and not an object detection task, the approaches were different. We propose two different binary classifiers based on CNNs. First is pre-trained VGG-16 [19] based classifier as shown in Figure 6.7a. We use the feature extraction layers and replace the fully connected layers at the end to add our layers to smaller ones. Our empirical tests show that adding layers with many nodes as in original VGG-16 model (1000 nodes per fully connected layer) hurts the performance since we only classify images to one class and not thousand. The second proposal is our custom CNN-based classifier, which the architecture is shown in Figure 6.7b where $/(3, 5)$ denotes the respective strides on width and height on the input. This custom model cannot benefit from pre-trained weights in the network, but it also does not have a fully-connected layer. Instead, the detection is performed by a 1×1 convolution at the end, which reduces the dimensionality into only one node. Fully-convolutional classification approach gives the benefit of having fewer parameters wherein the custom model we have 1.7 million parameters with 7 layers; however, the VGG-16 based model contains more than 14 million weights with 15 layers in total. Inner layers in both networks are equipped with ReLu activation function and the final layer with 1×1 convolution is accompanied by the Sigmoid activation function to perform the binary classification.

The dataset for this task is created from scratch. First, we analyzed the detections performed on the table detection training set. After performing table detection on all the images in the training set, we observed that the true-positive tables have the confidence score of more than 0.9997 on average and the false-positive detections averages to 0.54 confidence score. Therefore, to keep as much true-positive as possible, we do not perform the classification operation if a table detection has the confidence of 0.99 or higher. To create the dataset, the detections that have 0.8 or more IoU with ground truths are labelled as *table* and if they have a lower IoU, they are labelled as *not table*. The IoU threshold is set to 0.8 to allow the classifiers to grasp the idea of the tables more accurately when compared with the 0.5 which we use in our final models. One downside of setting the threshold to a value as high as 0.8 creates an imbalance in the dataset. To solve this problem, similar to the RetinaNet, we employed the Focal Loss [13] function and trained both our models with this loss function. The focal loss allowed us to acquire additional 2% accuracy. In addition to training VGG-16 and our custom model, we also tested the ResNet-50 [14] and Inception [46]. However, they are excessively complex for this problem and could not converge to acceptable accuracy levels.

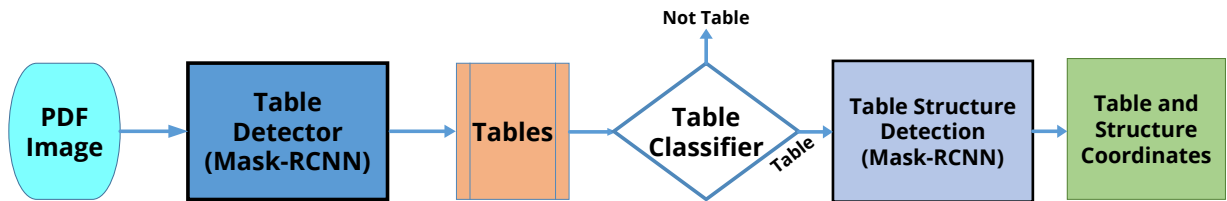
The validation set is split from the original training set, and we did not use the test set in developing the table classification model to avoid being influenced by them in the selection



(a) Judge System Flow Diagram



(b) Agreement Model Flow Diagram



(c) Table Classifier Model Flow Diagram

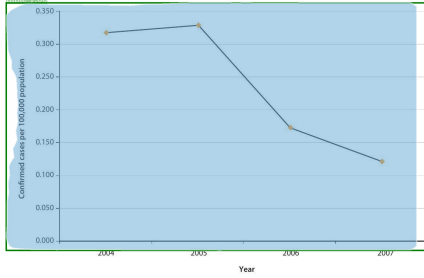
Figure 6.8: System flow diagrams for proposed post-processing steps.

of the thresholds and design of the models. When we employ the 0.99 confidence score threshold instead of classifying every detection, the VGG-16 based model can classify 650 false-positive table detection with only 1 misclassified true-positive. On the other hand, the custom model with the confidence threshold can only classify 458 false-positives as *not table* but with no sacrifice in the true-positives. The results show that both models are successful, and even further improvements can be made. When we studied the classification results, we have seen that figures and table-like structures (i.e. matrices or formulas) are accurately classified as *not table*. Previously, we mention that the lack of figure annotations in the table detection model causes misdetections. However, the introduced table classifier have solved this issue with a minimal overhead since the classification takes less than 0.2 seconds on a GPU. We leave the experimentation with different architectures to future work. CNNs are designed with vision and image tasks are in mind. However, there are previous findings that show SVMs work fine in some binary classification tasks [128]. Besides, ResNet-like architectures and many other creative things can be used to experiment and eliminate more false-positives. In designing this module, some architectural and performance decisions should be made. Images are simple documents, but if the requirements are precision, more complicated networks can eliminate better. However, if the speed of the overall E2E system is favoured, current models or maybe even simpler models (such as SVMs) should be used.

6.3 Training Details

The dataset used to train the introduced Pure E2E models is UNLV as in structure detection chapter. Hence, the same limitations and conditions apply; we have a minimal number of data points, there is tilt in the images, and the same stretching strategy (with the threshold Ω) is applied. The stretching algorithm had concluded that we should stretch to 210% whereas, in this chapter, we found that stretching to 250% of the original size is better. This is due to the type of images we have. Pure E2E models are based on Mask-RCNN [11] architecture and are trained on full document pages to detect structures, but in table structure detection (second stage of our proposed solution) the input was only table crops. Consequently, we, again, get help from the transfer learning capabilities of the deep learning based feature extractors such as ResNet [14] pre-trained on MS COCO [100]. The Mask-RCNN model is similar to the previous models where it includes an RPN and three branches; classification, regression and FCN.

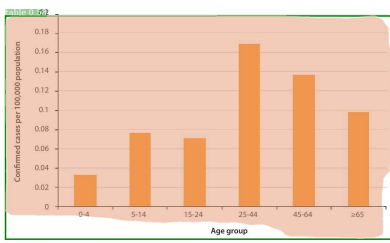
Pure E2E table structure detection model takes in the full document image and outputs rows and columns based on the features extracted from the input image. Our other proposal, Pure E2E tabular structure detection model takes in the full document image as

Figure BR1. | Notification rate of reported confirmed cases of human brucellosis in the EU², 2004-2007

1. Includes total cases for 2004 and confirmed cases from 2005-2007

2. Includes data from: AT, BE, CY, EE, FI, FR, DE, GR, IE, IT, LT, NL, PL, PT, ES, SE, UK

The highest notification rate of human brucellosis was noted in the age group 25-44 followed by the age group 45-64, (36.3% and 31.2% of confirmed cases, respectively) (Figure BR2). Brucellosis exhibited a slight seasonal pattern in 2007 with more cases occurring in the summer (Figure BR3).

Figure BR2. | Age-specific notification rate of reported confirmed human cases of brucellosis, TESSy data for reporting MSS¹, 2007

1. Includes data from all EU MSs, except CY, CZ, DK, EE, LV, LT, LU, MT, SE (N=526)

Table 14. | Actual and projected numbers for public high school graduates, by region and state: School years 2003-04 through 2021-22 – Continued

Region and state	Projected – Continued										
	2013-14	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	2021-22	2022-23	2023-24
United States	3,037,040	3,043,290	3,068,000	3,096,730	3,148,670	3,165,320	3,163,780	3,163,290	3,163,260	3,163,260	3,163,260
Northeast	548,290	548,450	550,630	550,980	554,700	550,820	546,620	551,920	552,550	552,550	552,550
Connecticut	35,540	34,960	35,170	34,730	34,410	34,030	33,190	33,910	33,910	33,910	33,910
Maine	12,640	12,530	12,600	12,360	12,300	12,250	12,000	12,000	12,000	12,000	12,000
Massachusetts	62,690	62,050	63,010	62,230	62,430	62,130	61,620	62,090	62,010	62,010	62,010
New Hampshire	13,860	13,710	13,530	13,270	13,170	12,930	12,880	12,640	12,640	12,640	12,640
New Jersey	92,220	91,260	91,330	92,060	91,940	91,460	90,390	91,400	91,400	91,400	91,400
New York	193,480	197,730	198,150	198,570	200,720	198,170	198,050	198,800	197,120	197,120	197,120
Pennsylvania	122,330	120,950	121,560	123,540	125,360	124,920	123,700	126,440	129,430	129,430	129,430
Rhode Island	9,460	9,150	9,270	9,270	9,580	9,150	9,040	8,960	9,200	9,200	9,200
Vermont	6,070	6,110	6,030	5,920	5,770	5,780	5,730	5,710	5,680	5,680	5,680
Midwest	672,600	670,100	674,720	674,480	685,020	687,300	678,690	679,440	685,520	685,520	685,520
Illinois	130,340	129,730	130,450	129,400	132,600	133,860	132,270	132,160	131,130	131,130	131,130
Indiana	69,940	64,820	64,980	65,380	66,330	66,240	65,220	65,220	66,520	66,520	66,520
Iowa	32,310	32,350	32,670	32,850	33,240	33,040	33,000	33,290	33,540	33,540	33,540
Kansas	31,040	30,740	31,780	32,210	32,860	33,140	33,010	33,760	33,920	33,920	33,920
Michigan	100,060	100,910	99,920	98,670	99,620	98,160	95,110	93,710	94,400	94,400	94,400
Minnesota	56,320	56,520	56,570	57,270	58,380	59,470	59,210	61,240	62,200	62,200	62,200
Missouri	60,340	59,930	61,740	61,110	61,970	61,540	60,860	61,270	62,210	62,210	62,210
Nebraska	19,500	19,490	19,710	20,010	20,660	20,970	21,280	21,570	22,280	22,280	22,280
North Dakota	6,980	6,930	7,050	7,010	6,700	7,110	7,130	7,570	7,950	7,950	7,950
Ohio	101,000	100,270	101,400	101,510	102,560	102,320	100,250	99,790	99,990	99,990	99,990
South Dakota	8,300	8,270	8,140	8,300	8,340	8,170	8,320	8,520	8,860	8,860	8,860
Wisconsin	60,460	60,160	60,310	60,770	61,760	61,480	60,740	61,310	62,540	62,540	62,540
South	1,103,910	1,110,100	1,126,810	1,142,200	1,174,920	1,182,420	1,178,490	1,180,140	1,186,400	1,186,400	1,186,400
Alabama	42,920	43,440	43,980	44,720	45,840	45,440	44,260	44,090	44,660	44,660	44,660
Arkansas	28,540	28,880	28,980	29,500	29,720	30,050	29,880	29,880	29,990	29,990	29,990
Delaware	8,120	7,680	7,990	8,310	8,550	8,570	8,590	8,690	8,690	8,690	8,690
District of Columbia	2,970	2,860	2,790	2,710	2,850	2,800	2,800	2,470	2,520	2,520	2,520
Florida	160,580	162,940	161,020	163,780	165,240	165,980	163,000	161,340	164,090	164,090	164,090
Georgia	92,010	91,560	94,320	95,340	97,270	98,230	96,250	95,490	96,780	96,780	96,780
Kentucky	38,480	38,790	39,610	40,160	41,370	41,490	40,080	40,300	40,480	40,480	40,480
Louisiana	35,720	33,340	33,050	33,420	37,880	38,830	38,840	38,360	38,130	38,130	38,130
Maryland	56,990	56,390	56,410	55,620	56,880	56,180	55,350	59,010	60,120	60,120	60,120
Mississippi	29,720	29,330	29,300	29,840	29,800	29,010	29,620	29,210	29,690	29,690	29,690
North Carolina	88,040	88,870	90,870	92,910	95,590	97,550	96,500	96,670	98,260	98,260	98,260
Oklahoma	37,300	37,770	38,130	39,350	40,270	40,450	40,760	41,540	41,650	41,650	41,650
South Carolina	39,450	39,520	40,350	41,180	42,880	43,000	41,530	41,530	42,680	42,680	42,680
Tennessee	58,600	58,920	60,110	61,570	62,040	62,040	61,940	61,870	62,340	62,340	62,340
Texas	291,830	297,630	303,120	311,510	320,960	326,770	328,560	334,040	338,920	338,920	338,920
Virginia	79,800	79,520	80,780	81,390	83,480	83,660	83,680	84,220	85,760	85,760	85,760
West Virginia	16,740	16,650	17,020	16,850	17,270	17,020	17,340	16,840	17,240	17,240	17,240
West	712,240	714,640	713,840	725,080	734,030	734,790	736,960	751,650	758,680	758,680	758,680
Alaska	7,380	7,360	7,370	7,670	7,710	7,750	7,690	7,990	7,990	7,990	7,990
Arizona	59,850	58,910	59,050	61,440	62,730	63,940	64,500	66,130	66,370	66,370	66,370
California	376,610	378,640	371,530	375,070	377,960	375,740	377,290	385,060	388,600	388,600	388,600
Colorado	61,330	61,790	63,030	64,300	66,210	67,010	67,590	69,290	69,110	69,110	69,110
Hawaii	10,530	10,300	10,390	10,250	10,640	10,030	10,030	10,020	10,700	10,700	10,700
Idaho	17,170	16,800	17,090	17,750	17,790	17,950	17,850	17,790	17,830	17,830	17,830
Montana	9,220	9,040	9,060	9,140	9,040	9,270	9,260	9,360	9,360	9,360	9,360
Nevada	24,580	24,360	25,470	25,810	26,240	26,710	26,440	26,250	26,380	26,380	26,380
New Mexico	18,480	18,680	18,850	19,540	19,690	20,020	20,050	20,130	20,550	20,550	20,550
Oregon	34,450	34,210	34,960	35,360	35,300	35,220	34,690	35,190	35,360	35,360	35,360
Utah	31,860	32,870	34,510	35,960	36,650	36,990	37,700	39,040	39,830	39,830	39,830
Washington	65,310	66,150	66,860	67,520	68,260	68,320	67,430	68,680	70,300	70,300	70,300
Wyoming	5,430	5,510	5,670	5,760	5,800	5,830	5,840	5,910	6,090	6,090	6,090

NOTE: Some data have been revised from previously published figures. Detail may not sum to totals because of rounding. Mean absolute percentage errors of public high school graduates by state and region can be found in table A-10, appendix A.

SOURCE: U.S. Department of Education, National Center for Education Statistics, Common Core of Data (CCD), "State Nonfiscal Survey of Public Elementary/Secondary Education," 2004-05 through 2009-10; and State Public High School Graduates Model, 1980-81 through 2008-09. (This table was prepared January 2012.)

(a) False-positives - Figures detected

(b) Missed parts of the table

Figure 6.9: False-positive detection examples from the proposed table detection approach.

well but outputs rows, columns and tables. We claim that the inclusion of tables in the model should improve the performance since learning what makes an area a table would help what makes it a row or a column as well. Pure E2E models are Mask-RCNN based deep learning models with pre-trained ResNet-50 backbone network where we fine-tune the network and apply domain adaptation. Mask-RCNN architecture includes sophisticated additions such as FPN, GroupNorm and FCN on top of a Faster-RCNN [12] architecture. In its essence, Mask-RCNN is an instance segmentation method, but since it is Faster-RCNN with an additional segmentation branch, it can also detect objects with bounding boxes. The semantic segmentation branch and the object detection branch is computed in parallel, and it becomes an instance segmentation method. Our Mask-RCNN model takes inputs of size up to 1200×2000 with a batch size of 1 image and 256 RoIs per

batch. Increasing the batch size and using BatchNorm or switching to the ResNet-101 or ResNeXt-101 [85] can be tried with a better GPU. The E2E models are trained for 20 epochs on the stretched UNLV dataset with an initial learning rate is 0.001. Scheduling is applied to reduce the learning rate after a certain number of epochs.

Another proposal of ours is the Stitched E2E method where we use previously trained table detection model and table structure detection model with no further training. Additionally, we propose improvements, in the form of post-processing steps. These improvements are applied after the table detections are made since we believe the connection between the two networks is the weakest point in this design. The proposed agreement module compares the tables and lines adversarially fronting each other and tables are either refined or rejected. Another proposed post-processing method, judge system compares the tables and structure detections from one of the Pure E2E models and adversarial processing is applied here as well. A group consensus is assessed, and then, tables are either kept or rejected. Our final post-processing proposal is the binary table classifier where we input the remaining tables after the initial eliminations by the agreement and judge models, and the classifier either accepts or rejects the table.

The table classifier is, in fact, a fine-tuned VGG-16 network, the same as in Mask-RCNN models, but in this case, we do not remove the classification branches. Instead, the classification layers are replaced with new ones and learned from scratch, whereas the feature extraction layers are only fine-tuned with focal loss from the RetinaNet is borrowed. The fully connected layers are trained for 2 epochs where the pre-trained feature extraction layers were frozen. Then, all layers in the network is trained for an additional 10 more epochs. The validation accuracy was 86.6% after all the training is complete. On the other hand, we trained our custom model from scratch for 50 epochs and the final validation accuracy was stuck at around 79%. The input image size is set to 600×1000 for both of the networks. At first, we trained with 224×224 as the original VGG models use [19]. However, we observed better results by increasing the input image. Batch sizes are set to 8 in both of the networks. In our custom model, we use BatchNorm and ReLu activation functions between the layers and large strides are employed to downsize the input image from $600 \times 1000 \times 3$ to $1 \times 1 \times 1$ and use sigmoid loss function to calculate the loss.

6.4 Results

In this section, we present our results and conduct a comparison of proposed E2E systems as well as post-processing techniques. We start by presenting and comparing outputs of our base table detection model from Chapter 4 and the table detection model

Table 6.1: Table detection performance on public ICDAR datasets.

Methods	ICDAR 2013			ICDAR 2017		
	Recall	AP	F1-Score	Recall	AP	F1-Score
Proposed (A)	1.0000	0.9900	0.9950	1.0000	0.9868	0.9933
Proposed (A) + Judge & Agreement	1.0000	0.9905	0.9952	1.0000	0.9876	0.9938
Proposed (A) + ALL	1.0000	0.9915	0.9957	1.0000	0.9878	0.9939

Table 6.2: Table detection on private dataset and on public UNLV dataset.

Methods	Private			UNLV		
	Recall	AP	F1-Score	Recall	AP	F1-Score
Proposed (A)	1.0000	0.9980	0.9990	1.0000	0.9819	0.9909
Proposed (A)	1.0000	0.9980	0.9990	1.0000	0.9820	0.9909
+ Judge & Agreement						
Proposed (A) + ALL	1.0000	0.9989	0.9994	1.0000	0.9852	0.9925

Table 6.3: E2E performance of the proposed methods

Methods	UNLV		
	Recall	mAP	F1-Score
Pure E2E(C)	0.8675	0.7913	0.8276
Pure E2E(D)	0.9224	0.8660	0.8933
Stitched E2E (A+B)	0.9615	0.7001	0.8101
Stitched E2E (A+B) + Judge & Agreement	0.9615	0.7140	0.8195
Stitched E2E (A+B) + ALL	0.9650	0.7732	0.8585

output after post-processing is applied in Table 6.1 and in Table 6.2. The comparison is made on our table datasets; ICDAR 2013 [73], ICDAR 2017 [77], UNLV and our private dataset. In all the tables, given metrics denotes the mean average calculations such as mean Average Precision (mAP). In all our experiments, the IoU threshold of detections with the ground truth is set to 50% to be compliant with the previous research and since it is a common choice [100, 96]. If detection has higher IoU than this threshold, it is considered a correct prediction, else a false detection.

Model A denotes the table detection model proposed in Chapter 4, Model B represents the table structure detection model from Chapter 5, and Model A+B represent the Stitched E2E system we propose here. If in the results, a post-processing method is mentioned, it means it is applied to the output of the Model A. The proposed Pure E2E table structure detection model which detects only rows and columns from the full document image is called Model C in the tables, and Pure E2E tabular structure detection model that gives all the detections is denoted as Model D. Since all the Models (A to D) are based on the Mask-RCNN [11] architecture with ResNet-50 [14] backbone, it is understandable that they use the same decision paths and hence, agreement and judge post-processing steps have only minimal effect and remove a few false-positives. A visual inspection of the results manifests that false-positives are mostly the figures in the documents as it is shown in Figure 6.9a. As we discussed before, the inclusion of figures, formulas and other structures alongside table detection is left for future work. However, as shown in Table 6.3, with the binary table classifier added, we were able to remove up to 500 false-positive row detections and up to 100 false-positive column detections. This is achieved by eliminating 45 false-positive table detections. The elimination of false-positives causes a significant leap in precision. When we visually inspect the eliminations, we see that they are mostly paragraphs and figures which were mistaken as a table. Therefore, we conclude that the binary classification is helpful in our Stitched E2E model and is open for further investigation.

The workflow of the Stitched E2E model is presented in Figure 6.1a. Since the output of the table detection model is used to detect structures in the second stage, the performance of this system is solely conditioned on the performance of the first stage. As presented in Figure 6.10d, one false-positives that is generated by the first stage, cumulates and tenfolds the false-positives. The effects of this phenomenon are presented in Table 6.3 where the recall is converging to the limitations of our models, but the system attains a lower mAP value since the precision is a function of false-positives and true-positives. The limitation of the performance of the Stitched E2E model is calculated by the cumulative successes of the two stages; table detection and structure detection. For example, since the structure detection recall is around 98% which is achieved with perfect table detections assumed, the Stitched E2E model cannot easily reach the same levels. To present a pure model that is

easier to train, test and faster to use, we introduced two models; Pure E2E table structure detection model and Pure E2E tabular structure detection model. Model C takes the whole page image as input, and the output is the rows and columns. It is fast since it does only one pass over the image; however, it is not up to par with other methods in terms of metrics presented, especially the row detection performance is only mediocre. On the other hand, Model D takes in the same input as Model C, but the model is extended to detect tables as well. Addition of tables improved the mAP by more than 7% which allowed this model to surpass others and achieve the best AP between all proposed methods with a slightly lower recall than the Stitched E2E system. When calculating the mAP for Model C, we counted the tables from the Model A to achieve mAP over three classes. Both Models C and D are trained with the stretched UNLV dataset.

The Stitched E2E System (Model A+B) takes in the full document image and then outputs a table detection with segmentation results alongside each other. To this detections, either some improvements are made, or nothing is changed, and the vanilla output is used. After the table detection is finalized one way or the other, the coordinates are used to crop the table region image, and this is sent to Model B to detect rows and columns. Therefore, Model B’s output is strongly correlated with the initial table detection, and any false-positive table detection causes the precision to drop notably. Model B is not capable of discriminating incorrect tables, and it always assumes the input is a perfect table crop. This is introduced to the method through how the dataset is formed to train the network. Because in every step of training, Model B learned that a table is formed of multiple rows and columns which are next to each other and they *always* start from left side extending up to the right side for rows (or up and down for columns). Even though the input to the model is a blank page (Figure 6.10d), the model predicts rows and columns, though the confidence scores are not high. Hence, false-positive elimination is crucial. However, as evidenced in Figure 4.5c, our private dataset includes challenging cases such as nested tables which makes the false-positive elimination even more difficult.

In Table 6.1 we can see that the proposed improvements allowed us to improve our precision on the public ICDAR datasets. Since we already attain adequate precision, further improvements are not easy to achieve. The agreement module and judge module helped us to see minor improvements in precision where they managed to remove some false-positives. With table classifier, we observe further improvements. Since ICDAR 2017 is a slightly more complex dataset, the improvements are less significant. However, our findings support the importance of the binary classification applied at the end. In Table 6.2 we see less improvement from the agreement and judge modules when compared to ICDAR datasets. This is due to that we already achieve fantastic results on our private dataset, and the false-positives in UNLV dataset is mostly figures or caused by nested

table limitations which is not handled effectively by these methods. The Mask-RCNN table detector achieves a perfect recall and a high AP. However, the table classifier was able to remove even more false-positives, and the AP jumps to 0.9989. In the UNLV table dataset, an improvement of 0.003 AP was observed, and further impacts of this change are observed in E2E detection.

(a) Accurate table and structure detection

(b) Incomplete table but structure is still detected

(c) Complete version of the table in Figure 6.10b

(d) Structures are detected in a false-positive table crop.

6.5 Conclusion

With the wide-scale transition from the analogous world to the digital world, supply chains or any organization that handles data have to adapt as businesses are moving to digital mediums. Since everything is shifting to digital; discovering suppliers, retailers or manufacturers and connecting different ends of a chain as it was never possible before is becoming a reality. With this growth, information and the flow of it grows unearthly, which brings new requirements for the data processing. We now need faster and more efficient processing methods to extract vital information from documents. Since the information is mostly in the form of reports, handbooks and such, extracting information from such documents are in demand. Since the most vital information is contained in tabular structures like tables, standard utilities that are used to read documents such as OCR methods cannot be accurately applied for this purpose. The importance of this task is well-known, and valuable prior research has looked into the table detection [7, 9, 29, 72] and structure detection as well [8, 10, 67] which provided invaluable knowledge to build upon. With the power of deep learning on our back, we proposed three E2E systems that can be used for tabular structure detection purposes. Our first method is called Stitched E2E method which we combine two of our previous works [7, 8] and propose post-processing improvements to be applied before the link is made. We propose a table classifier as well as a judge module which checks the consensus of the table detections and structure detections and an agreement module which checks the consensus between different table detections in the same image. The proposed consensus-based algorithms and the table classifier model is included to prevent the error from the table detections (mostly in the form of false-positives) to cumulating in the second stage, table structure detection. Every single false-positive table detection results in more than ten false-positives from the structure detection model and hence lowers the precision for the E2E system. According to our experiments, even though agreement and judge modules increase the precision a small amount, line agreement module refines the location and possibly makes detections perfect in some cases. In the end, E2E mAP is increased for more than 7.3% with these augmentations. When the table classifier is included, a massive removal of easy false-positives (i.e. figures, formulas and paragraphs detected as table) is observed which resulted in the removal of hundreds of false-positive row detections.

Chapter 7

Conclusion

The widespread use of Internet has led to advancements in the processing and analysis capabilities for digitally stored data. Extracting the contents of the conventional document formats of which either in printed or born-digital format (i.e. Portable Digital Format (PDF)), is not straightforward. Printed documents are images with pixels and PDF does not save structure information from the documents. Therefore, to convert millions of documents into a computer-understandable format, an automated system which can extract information is needed. The source of these documents can be the governmental agencies, academic institutions and supply chains. For instance, the supply chain optimization case involves thousands of suppliers and other entities up to the end users. Since the source is significantly varied, it is not feasible to develop source-specific solutions. Hence, the state of the art seeks a source-transparent and generalized solutions. As a widely accepted and compact information container, tables and more generally, tabular data is the main focus of this thesis. Table detection and extraction cannot be successfully achieved with standard Optical Character Recognition (OCR) utilities since they contain complex structures and lines that determine the table region.

Table detection from various document types is a popular and well-studied problem by the researchers in this area [36, 10, 74], as well as ourselves [7, 9]. On the other hand, since extraction of the information inside the tables remains an open issue, table detection turns out to be an intermediary process that feeds into the information extraction operation by either a human or a computer. Due to the fact that finding rows and associated columns is a time-consuming process, we argue that automating the second part is also essential. Related work also contains several studies focusing on this particular problem [27, 86, 10] prior to our own contributions to the solution of this problem [8, 9]. Indeed, table structure detection is an inherently more complex problem since the rows and columns differ from

the most common square-shaped objects. More specifically, the rows and columns are thin at the edges, which makes the detection process further challenging.

This thesis has introduced two deep learning models to table detection and table structure detection problems. Both models leverage the renowned Mask-RCNN [11], and achieve the state-of-the-art performance in the solution of these individual problems. Mask-RCNN performs exceptionally well on this task with additions such as Feature Pyramid Networks [15], Group Normalization [17] and Fully Convolutional Networks [16]. On the table detection task, our Mask-RCNN based solution achieves the state-of-the-art with 100% recall and 99% AP on widely known public datasets such as ICDAR 2013 [73], ICDAR 2017 [77] and UNLV [126]. In order to solve the small edges of rows and columns, apart from the architectural decisions, we have applied stretching and various improvements such as line detection, agreement module, judge module and a binary table classifier. The proposed binary table classifier has been shown to be the most effective method when compared to other post-processing steps, and a relative AP increase of 0.3% has been achieved, which is a reasonable increase under the consideration of the nature of the problem and the state of the art. We have finally proposed three End-to-End (E2E) design systems to detect tables rows and columns in arbitrary documents. All three models take images as input to allow them to operate on any format as long as they can be converted into images. The Stitched E2E system bridges the table detection and the table structure detection models with the mentioned post-processing steps applied following the table detection output. The Pure E2E systems are designed to detect the structure in a single pass and observed to have better precision but with a lower recall. The first Pure E2E system is called Pure E2E Table Structure Detection system, which is used to detect only rows and columns in a single pass from the document image. Whereas the second Pure E2E proposal is called Pure E2E Tabular Structure Detection system, and it detects tables alongside rows and columns in a single pass from the document image.

7.1 Future Directions

We achieve near perfect results with our table detection model, but we also acknowledge that there are future improvements to be made in all three aspects, namely table detection, table structure detection and E2E system design.

Coping with the lack of public annotated data

In this study, the lack of public annotated data has been a fundamental challenge. In order to cope with this issue and prevent over-fitting, we hypothesize that hard negative mining and weakly supervised learning techniques can be employed. Hard negative mining, or known as *bootstrapping*, is a method where the misclassified examples are used to re-train the network and learn the edge cases [129]. In our methods, we observe false-positives that have the same characteristics and hard negative mining can help solve this issue alongside with the labelling of common page objects such as figures. Hard negative mining can be used to improve detection performance with the available volume of data.

Increasing the data volume

However, we can also increase the data volume without labelling it by hand. Weakly supervised learning is one method that aims to increase the data volume by using the trained model to label the dataset iteratively [127]. Since the dataset will not be perfectly accurate, as the model gets trained, the dataset is labelled again to obtain a dataset with less label-noise. Another method is called Distant Supervision where instead of the model itself to label the dataset, another method is used to label the dataset with noise and then the machine learning model is trained on top [130]. One natural algorithm to use is one of the rule-based table or table-structure detection methods to label the data with low precision and then use the dataset to train the deep learning models. Other options include hand labelling new data, which is a time-consuming task; or using different methods such as Generative Adversarial Networks (GANs) or augmentations to increase the data volume. Thus, a larger dataset is expected to improve the method's performance.

Investigations on different architectural decisions

Apart from the data volume, different architectural decisions can also be investigated. Instead of using the image as an RGB input, inclusion of table content or text, detected with an OCR engine and fed into a convolutional layer of the feature extractor can be studied as it is already shown to improve the performance [74]. In this thesis, we emphasize the impact of the FCN and FPN architectures. Related work reports that changing the FCN architecture affects the detection performance [10] same can be said for FPN as well. Therefore, further investigation in this area is a potential future direction. In addition to these architectural parameters, hyperparameter search should be conducted. There are many possible parameters to optimize; mask resolution, stride sizes, learning rate and

more. We have conducted a linear search on learning rate as well as mask resolution while applying scheduled decreasing as mentioned before. But a more comprehensive search is bound to increase the performance. As we have analyzed in structure detection models, more complex convolutional neural network based feature extractors can also be studied further with the extended dataset. We have seen promising performance under ResNeXt-101 [85] feature extractor but it is worth noting that the dataset size prevents the model from converging as the more complex the models gets, the higher the data volume is required. The future extension of our study will include enhancements to achieve better detections.

As discussed earlier, the modules that are open to improvement are; table structure detection and table classifier. Table structure detection requires more data to be more robust on false detection cases, and our initial results on table classifier show that searching for a better architecture can be beneficial. The discussed methods to increase the data volume, such as distant supervision, can be employed to increase the table classifier dataset as well.

A different architectural decision would be extracting text from PDF data (or through an OCR tool) and using this data to annotate additional data in distant supervision fashion or feeding them to a chosen convolutional layer is promising value. We focus on table detection and table structure detection from images to encapsulate a broader range of documents. However, converting to PDFs or using an OCR tool to extract text from the document could be useful. Some cases make the table detection problem trivial such as 'Table' captions. Incorporating text from the document into the model would turn these cases from object detection to a more straightforward process.

Semantic modelling

The ongoing extensions of the work presented in this thesis include the extraction of semantic meaning and introducing further improvements to the models presented here. These include extraction of the detected rows and columns, and detection of the intersections (cells). Each cell represents a piece of information that we aim to extract as our end goal. A distinctive property of table cells is, as we show in Figure 4.5, the possibility of including other tables or even figures. These cases and semantic connections of these cells are in our current agenda, which is an immediate extension of the work presented in this thesis.

References

- [1] W. B. I. E. D. D. Group, *World development indicators*. World Bank, 1978.
- [2] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, “Defining supply chain management,” *Journal of Business Logistics*, vol. 22, no. 2, pp. 1–25.
- [3] S. Vaidya, P. Ambad, and S. Bhosle, “Industry 4.0 a glimpse,” *Procedia Manufacturing*, vol. 20, pp. 233 – 238, 2018. 2nd International Conference on Materials, Manufacturing and Design Engineering (iCMMD2017), 11-12 December 2017, MIT Aurangabad, Maharashtra, INDIA.
- [4] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, “Historical document layout analysis competition,” in *2011 International Conference on Document Analysis and Recognition*, pp. 1516–1520, IEEE, 2011.
- [5] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, “Page segmentation of historical document images with convolutional autoencoders,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1011–1015, IEEE, 2015.
- [6] M. Bulacu, R. van Koert, L. Schomaker, and T. van der Zant, “Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen,” in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 1, pp. 357–361, IEEE, 2007.
- [7] M. Traquair, E. Kara, B. Kantarci, and S. Khan, “Deep learning for the detection of tabular information from electronic component datasheets,” in *IEEE Symposium on Computers and Communications (ISCC)*, (Barcelona, Spain), June 2019 (accepted) Available for reviewer: <http://nextconlab.academy/iscc19/1570516006.pdf>.

- [8] E. Kara, M. Traquair, B. Kantarci, and S. Khan, “Deep learning for recognizing the anatomy of tables on datasheets,” in *IEEE Symposium on Computers and Communications (ISCC)*, (Barcelona, Spain), June 2019 (accepted) Available for reviewer: <http://nextconlab.academy/iscc19/1570532191.pdf>.
- [9] E. Kara, M. Traquair, M. Simsek, B. Kantarci, and S. Khan, “End-to-end system design for deep learning-based tabular information discovery in datasheets,” *IEEE Access Special Section: 'AI-Driven Big Data Processing: Theory, Methodology, and Applications'*, 2019 (submitted, under review) Available for reviewer: <http://nextconlab.academy/access19/2019-12062.pdf>.
- [10] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, “DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2018.
- [11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [12] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, pp. 91–99, 2015.
- [13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.
- [16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
- [17] Y. Wu and K. He, “Group normalization,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

- [18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pp. 448–456, JMLR.org, 2015.
- [19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [21] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- [22] R. M. Haralick, S. R. Sternberg, and X. Zhuang, “Image Analysis Using Mathematical Morphology,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 532–550, jul 1987.
- [23] G. Nagy, “Preliminary investigation of techniques for automated reading of unformatted text,” *Communications of the ACM*, vol. 11, pp. 480–487, jul 1968.
- [24] A. K. Jain and B. Yu, “Document representation and its application to page decomposition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.
- [25] F. M. Wahl, K. Y. Wong, and R. G. Casey, “Block segmentation and text extraction in mixed text/image documents,” *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 375–390, 1982.
- [26] A. Pivk, P. Cimiano, Y. Sure, M. Gams, V. Rajković, and R. Studer, “Transforming arbitrary tables into logical form with TARTAR,” *Data and Knowledge Engineering*, vol. 60, no. 3, pp. 567–595, 2007.
- [27] F. Shafait and R. Smith, “Table detection in heterogeneous documents,” pp. 65–72, 2010.
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

- [29] T. Kieninger and A. Dengel, “Applying the T-recs table recognition system to the business letter domain,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2001-Janua, pp. 518–522, 2001.
- [30] E. Oro and M. Ruffolo, “PDF-TREX: An approach for recognizing and extracting tables from PDF documents,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 906–910, IEEE, 2009.
- [31] D. N. Tran, T. A. Tran, A. Oh, S. H. Kim, and I. S. Na, “Table Detection from Document Image using Vertical Arrangement of Text Blocks,” *International Journal of Contents*, vol. 11, no. 4, pp. 77–85, 2016.
- [32] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, “A table detection method for multipage PDF documents via visual separators and tabular structures,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 779–783, IEEE, 2011.
- [33] T. G. Kieninger and B. Strieder, “T-recs table recognition and validation approach,” in *AAAI Fall Symposium on Using Layout for the Generation, Understanding and Retrieval of Documents*, 1999.
- [34] N. Amamoto, S. Torigoe, and Y. Hirogaki, “Block segmentation and text area extraction of vertically/horizontally written document,” in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR’93)*, pp. 739–742, IEEE, 1993.
- [35] M. A. Hearst, “Support vector machines,” *IEEE Intelligent Systems*, vol. 13, pp. 18–28, July 1998.
- [36] M. Kerwat, R. George, and K. Shujaee, “Detecting Knowledge Artifacts in Scientific Document Images - Comparing Deep Learning Architectures,” in *2018 5th International Conference on Social Networks Analysis, Management and Security, SNAMS 2018*, pp. 147–152, IEEE, 2018.
- [37] Y. Wang and J. Hu, “A machine learning based approach for table detection on the web,” p. 242, 2004.
- [38] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, “Learning to detect tables in scanned document images using line information,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 1185–1189, IEEE, 2013.

- [39] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, pp. 81–106, Mar. 1986.
- [40] L. R. Rabiner and B. H. Juang, “An Introduction to Hidden Markov Models,” *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.
- [41] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML ’01, (San Francisco, CA, USA), pp. 282–289, Morgan Kaufmann Publishers Inc., 2001.
- [42] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, pp. 193–202, Apr 1980.
- [43] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, pp. 541–551, Dec. 1989.
- [44] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- [45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F. F. Li, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, 09 2014.
- [46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [47] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2874–2883, 2016.
- [48] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, “Scale-aware fast r-cnn for pedestrian detection,” *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2018.
- [49] Z. Xu, Y. Yang, and A. G. Hauptmann, “A discriminative cnn video representation for event detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1798–1807, 2015.

- [50] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio.,” *SSW*, vol. 125, 2016.
- [51] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.
- [52] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.
- [53] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, “Deep learning for content-based image retrieval: A comprehensive study,” in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 157–166, ACM, 2014.
- [54] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359–2367, 2017.
- [55] P. Pyreddy and W. B. Croft, “TINTIN : A System for Retrieval in Text Tables,” in *Proceedings of the second ACM international conference on Digital libraries*, pp. 193–200, 1997.
- [56] T. Kieninger and A. Dengel, “The T-Recs Table Recognition and Analysis System,” pp. 255–270, Springer, Berlin, Heidelberg, 1999.
- [57] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, “Trainable table location in document images,” pp. 236–240, 2003.
- [58] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, “Automatic Table Detection in Document Images,” 2005.
- [59] X. Lin and Y. Xiong, “Detection and analysis of table of contents based on content association,” *International Journal on Document Analysis and Recognition*, 2006.
- [60] J. W. Son, H. J. Song, J. A. Lee, S. J. Lee, S. B. Park, and S. Y. Park, “Discriminating meaningful web tables from decorative tables using a composite kernel,” in *Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2008*, pp. 368–371, 2008.

- [61] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak, “Towards domain-independent information extraction from web tables,” p. 71, 2007.
- [62] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, “Improving the table boundary detection in PDFs by fixing the sequence error of the sparse lines,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009.
- [63] S. Dey, J. Mukherjee, and S. Sural, “Consensus-based clustering for document image segmentation,” *International Journal on Document Analysis and Recognition*, vol. 19, no. 4, pp. 351–368, 2016.
- [64] K. Zuyev, “Table image segmentation,” pp. 705–708, 2002.
- [65] Y. Wang, R. Haralick, and I. T. Phillips, “Zone content classification and its performance evaluation,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2001-Janua, pp. 540–544, IEEE Comput. Soc, 2001.
- [66] S. Shetty, H. Srinivasan, M. Beal, and S. Srihari, “Segmentation and labeling of documents using conditional random fields,” in *Document Recognition and Retrieval XIV*, vol. 6500, p. 65000U, 2007.
- [67] H. T. Ng, C. Y. Lim, and J. L. T. Koo, “Learning to recognize tables in free text,” in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics -*, (Morristown, NJ, USA), pp. 443–450, Association for Computational Linguistics, 2007.
- [68] A. C. Silva, “Learning rich hidden Markov models in document analysis: Table location,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 843–847, IEEE, 2009.
- [69] M. Fan and D. S. Kim, “Detecting Table Region in PDF Documents Using Distant Supervision,” 2015.
- [70] D. P. Lewis, T. Jebara, and W. S. Noble, “Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure,” *Bioinformatics*, vol. 22, pp. 2753–2760, nov 2006.
- [71] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and Ł. Bolikowski, “CER-MINE: Automatic extraction of structured metadata from scientific literature,” *International Journal on Document Analysis and Recognition*, vol. 18, no. 4, pp. 317–335, 2015.

- [72] L. Hao, L. Gao, X. Yi, and Z. Tang, “A Table Detection Method for PDF Documents Based on Convolutional Neural Networks,” in *Proceedings - 12th IAPR International Workshop on Document Analysis Systems, DAS 2016*, pp. 287–292, IEEE, 2016.
- [73] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras, “Icdar 2013 robust reading competition,” in *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, ICDAR '13*, (Washington, DC, USA), pp. 1484–1493, IEEE Computer Society, 2013.
- [74] X. Yang, E. Yumer, P. Asente, M. Kralej, D. Kifer, and C. L. Giles, “Learning to extract semantic structure from documents using multimodal fully convolutional neural networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [75] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, “Table Detection Using Deep Learning,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 771–776, 2017.
- [76] X. H. Li, F. Yin, and C. L. Liu, “Page Object Detection from PDF Document Images by Deep Structured Prediction and Supervised Clustering,” in *Proceedings - International Conference on Pattern Recognition*, vol. 2018-Augus, pp. 3627–3632, IEEE, 2018.
- [77] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, “Icdar2017 competition on page object detection,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 1417–1422, IEEE, 2017.
- [78] S. Ares Oliveira, B. Seguin, and F. Kaplan, “DhSegment: A generic deep-learning approach for document segmentation,” in *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*, pp. 7–12, 2018.
- [79] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, pp. 21–37, Springer, 2016.
- [80] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.
- [81] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, “DeCNT: Deep deformable CNN for table detection,” *IEEE Access*, vol. 6, pp. 74151–74161, 2018.

- [82] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, “A Saliency-based Convolutional Neural Network for Table and Chart Detection in Digitized Documents,” pp. 1–13, 2018.
- [83] S. Arif and F. Shafait, “Table Detection in Document Images using Foreground and Background Features,” in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, dec 2018.
- [84] N. Milosevic, C. Gregson, R. Hernandez, and G. Nenadic, “A framework for information extraction from tables in biomedical literature,” *International Journal on Document Analysis and Recognition*, vol. 22, no. 1, pp. 55–78, 2019.
- [85] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [86] J. Fang, P. Mitra, Z. Tang, and C. L. Giles, “Table Header Detection and Classification,” in *In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [87] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [88] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.
- [89] P. Petersen and F. Voigtlaender, “Optimal approximation of piecewise smooth functions using deep relu neural networks,” *Neural Networks*, vol. 108, pp. 296–330, 2018.
- [90] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, vol. 30, p. 3, 2013.
- [91] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*, pp. 818–833, Springer, 2014.
- [92] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

- [93] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01*, CVPR ’05, (Washington, DC, USA), pp. 886–893, IEEE Computer Society, 2005.
- [94] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.
- [95] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, “How does the brain solve visual object recognition?,” *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.
- [96] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [97] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.
- [98] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.” <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [99] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.” <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [100] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, pp. 740–755, Springer, 2014.
- [101] L. Zhang, L. Lin, X. Liang, and K. He, “Is faster r-cnn doing well for pedestrian detection?,” in *European conference on computer vision*, pp. 443–457, Springer, 2016.
- [102] A. Salvador, X. Giró-i Nieto, F. Marqués, and S. Satoh, “Faster r-cnn features for instance search,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 9–16, 2016.
- [103] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, “Spatial transformer networks,” in *Advances in neural information processing systems*, pp. 2017–2025, 2015.

- [104] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, “Pyramid methods in image processing,” *RCA engineer*, vol. 29, no. 6, pp. 33–41, 1984.
- [105] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *international Conference on computer vision & Pattern Recognition (CVPR’05)*, vol. 1, pp. 886–893, IEEE Computer Society, 2005.
- [106] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [107] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results.” <http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html>.
- [108] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359–2367, 2017.
- [109] S.-W. Lee, Y. Nakano, A. Schreyer, P. Suda, and G. Maderlechner, *Document Analysis Systems: Theory and Practice*, vol. 1655. 1999.
- [110] S. Mao, A. Rosenfeld, and T. Kanungo, “Document structure analysis algorithms: a literature survey,” *Document Recognition and Retrieval X*, vol. 5010, no. January 2003, pp. 197–207, 2003.
- [111] D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy, “Table-processing paradigms: A research survey,” 2006.
- [112] X. Sun, P. Wu, and S. C. Hoi, “Face detection using deep learning: An improved faster rcnn approach,” *Neurocomputing*, vol. 299, pp. 42 – 50, 2018.
- [113] X. Mo, K. Tao, Q. Wang, and G. Wang, “An efficient approach for polyps detection in endoscopic videos based on faster r-cnn,” *arXiv preprint arXiv:1809.01263*, 2018.
- [114] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hashoul, R. Ben-Ari, and E. Barkan, “A cnn based method for automatic mass detection and classification in mammograms,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–8, 2017.
- [115] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *IEEE CVPR*, vol. 4, 2017.

- [116] L. Ale, N. Zhang, and L. Li, “Road damage detection using retinanet,” in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5197–5200, IEEE, 2018.
- [117] J. W. Johnson, “Adapting mask-rcnn for automatic nucleus segmentation,” *arXiv preprint arXiv:1805.00500*, 2018.
- [118] R. Alp Güler, N. Neverova, and I. Kokkinos, “Densepose: Dense human pose estimation in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7297–7306, 2018.
- [119] X. Zhao, S. Liang, and Y. Wei, “Pseudo mask augmented object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4061–4070, 2018.
- [120] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI’16, (Berkeley, CA, USA), pp. 265–283, USENIX Association, 2016.
- [121] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS-W*, 2017.
- [122] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017.
- [123] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- [124] J. Canny, “A computational approach to edge detection,” in *Readings in computer vision*, pp. 184–203, Elsevier, 1987.
- [125] N. Kiryati, Y. Eldar, and A. M. Bruckstein, “A probabilistic hough transform,” *Pattern recognition*, vol. 24, no. 4, pp. 303–316, 1991.
- [126] S. V. Rice, F. R. Jenkins, and T. Nartker, “The fourth annual test of ocr accuracy,” 03 2012.

- [127] Z.-H. Zhou, “A brief introduction to weakly supervised learning,” *National Science Review*, vol. 5, no. 1, pp. 44–53, 2017.
- [128] Z. Sun, F. Li, and H. Huang, “Large scale image classification based on cnn and parallel svm,” in *International conference on neural information processing*, pp. 545–555, Springer, 2017.
- [129] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.
- [130] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pp. 1003–1011, Association for Computational Linguistics, 2009.