High Precision Deep Learning-Based Tabular Data Extraction

by

Ji Chu Jiang

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Electrical and Computer Engineering

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

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 advancements of AI methodologies and computing power enables automation and propels the Industry 4.0 phenomenon. Information and data are digitized more than ever, millions of documents are being processed every day, they are fueled by the growth in institutions, organizations, and their supply chains. Processing documents is a time consuming laborious task. Therefore automating data processing is a highly important task for optimizing supply chains efficiency across all industries. Document analysis for data extraction is an impactful field, this thesis aims to achieve the vital steps in an ideal data extraction pipeline. Data is often stored in tables since it is a structured formats and the user can easily associate values and attributes. Tables can contain vital information from specifications, dimensions, cost etc. Therefore focusing on table analysis and recognition in documents is a cornerstone to data extraction.

This thesis applies deep learning methodologies for automating the two main problems within table analysis for data extraction; table detection and table structure detection. Table detection is identifying and localizing the boundaries of the table. The output of the table detection model will be inputted into the table structure detection model for structure format analysis. Therefore the output of the table detection model must have high localization performance otherwise it would affect the rest of the data extraction pipeline. Our table detection improves bounding box localization performance by incorporating a Kullback–Leibler loss function that calculates the divergence between the probabilistic distribution between ground truth and predicted bounding boxes. As well as adding a voting procedure into the non-maximum suppression step to produce better localized merged bounding box proposals. This model improved precision of tabular detection by 1.2% while achieving the same recall as other state-of-the-art models on the public ICDAR2013 dataset. While also achieving state-of-the-art results of 99.8% precision on the ICDAR2017 dataset. Furthermore, our model showed huge improvements espcially at higher intersection over union (IoU) thresholds; at 95% IoU an improvement of 10.9% can be seen for ICDAR2013 dataset and an improvement of 8.4% can be seen for ICDAR2017 dataset.

Table structure detection is recognizing the internal layout of a table. Often times researchers approach this through detecting the rows and columns. However, in order for correct mapping of each individual cell data location in the semantic extraction step the rows and columns would have to be combined and form a matrix, this introduces additional degrees of error. Alternatively we propose a model that directly detects each individual cell. Our model is an ensemble of state-of-the-art models; Hybird Task Cascade as the detector and dual ResNeXt101 backbones arranged in a CBNet architecture. There is a lack of quality labeled data for table cell structure detection, therefore we hand labeled the

ICDAR2013 dataset, and we wish to establish a strong baseline for this dataset. Our model was compared with other state-of-the-art models that excelled at table or table structure detection. Our model yielded a precision of 89.2% and recall of 98.7% on the ICDAR2013 cell structure dataset.

Acknowledgements

I would like to thank my supervisor Dr. Burak Kantarci for all the support and guidance he has given me. He was the one that encouraged me to pursue thesis-based masters, and gave me the opportunity to work with him. I am ever grateful for all the hard-work he has put in for my sake. All of the endless nights and long discussions were a proof to his dedication for his students. Without his assistance I would not have been able to come so far in my academic career. I couldn't have done it without him.

I would also like to thank Johan Fernandes, Yakup Akkaya and Dr. Murat Simsek for their support and efforts. Their knowledge and feedback were critical to my success. As well as all of the NEXTCON lab members that pursued research alongside me, thank you; Nima Taherifard, Jinxin Liu, Yueqian Zhang, Zhiyan Chen, Ahmed Omara, Nahid Parvaresh, Yu Shen and Yuwei Wang.

I would like to thank Dr. Shahzad Khan and Lytica Inc. for supporting this project by providing their data, resources and input. This work wouldn't have been possible without their support. I want to especially thank the MITACS Accelerate IT14836 internship program for connecting me with our industry partner Lytica. I'd like to thank the efforts of Dr. Burak Kantarci and Shahzad Khan for ensuring that I had the computing resources necessary. As a result we were able to utilize SOSCIP computing resources as a part of SOSCIP Project 3-040. Thank you all at the SOSCIP support team for troubleshooting compatibility errors.

Lastly, I would like to express my gratitude to my parents for their never ending support over all these years. They encouraged and motivated me throughout my life. I wouldn't be here without them. I would also like to express gratitude for Peter Casey for all he has done for me over the years, I wouldn't be who I am without him. Lastly, I'd like to thank my mom, she made the brave decision to move to Canada, by doing so, she gave me a life that I would of never had. I would like to let her know that all of her efforts in raising me would not be wasted.

Table of Contents

List of Tables

List of Figures

Publications of the Candidate During MASc Studies

Direct outcomes of the thesis:

- **JC. Jiang**, M. Simsek, B. Kantarci, and S. Khan, "High Precision Deep Learning-based Tabular Position Detection" in *IEEE Symposium on Computers and Communications (ISCC)*, (Rennes, France), June 2020
  (Accepted)

- **JC. Jiang**, M. Simsek, B. Kantarci, and S. Khan "TabCellNet: Deep Learning-based Tabular Cell Structure Detection," *Elsevier Neurocomputing* (Under Review)

Publication outside of the thesis:

- **JC. Jiang**, B. Kantarci, S. Oktug and T. Soyata, "Federated Learning in Smart City Sensing: Challenges and Opportunities" in *MDPI Sensors*, August 2020
  (Accepted)

Chapter 1

Introduction

The industry 4.0 phenomenon has pushed industries to digitize documentation and enhanced the manufacturing process [2]. Along with it, the use of Internet of Things, smart manufacturing, cloud-based manufacturing and automated processes are becoming more prevalent. Many areas have been enhanced to support the industry 4.0 phenomenon, such as 5G wireless communication, Artificial Intelligence (AI) and the improvements in computing hardware [3]. New products are being rapidly introduced at a unprecedented pace. To support the increased throughput of product creation, supply chains are in a position to benefit from these new automated methodologies. Industries around the world are moving towards digital data, with the benefits of enhanced information organization and querying. Documents can often be found in digital format or born-digital to begin with. Millions of these digital documents are processed throughout supply chains throughout the world. These documents cannot be automatically queried due to the fact that they are created for human consumption without a predetermined format. Efficient processing of these documents through software mediation will allow for complete digital transformation ([4]).

According to the industry 4.0 requirements automated document processing in supply chains is highly desired [5]. This is because supply chains play a integral role in a company's success [6]. Supply chains consists of upstream and downstream firms that connect to the end consumer. The communication and sharing of information between these firms are paramount to the success of an efficient supply chain. This information is often portrayed electronically in formats such as PDF, email and fax. The received data is often unstructured and requires manual processing. Thus, efficient management of supply chains can easily improve the time value spent within the product development cycle [7].

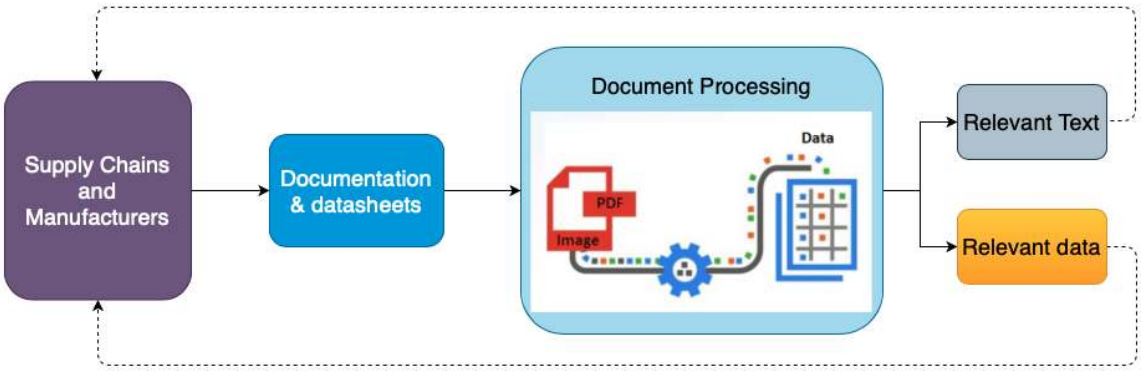


Figure 1.1: Example document flow which exists throughout all industry. [1]

Document analysis consists of processing all parts of a document, this includes figures, graphs, tables and text. Understanding tables and table structures a key aspect to automated data extraction and a crucial part of document analysis. Document extraction can leverage the innovations in AI-based methods, mainly image recognition with deep Convolutional Neural Networks (CNN). There are many public competition datasets that have been created to advance the process of automating document information extraction ([8, 9, 10]).

Document table extraction is difficult due to an absence of absolute global formatting and infinite variations of internal structure layouts. There have been several studies focused on the detection of tables within documents utilizing deep learning-based methodologies ([11, 12, 13, 14, 15, 16, 17, 18]. However, there is much less effort on the detection table structures and often times the table structure is categorized by rows and columns of a table ([19]). A more generalized way of recognizing table structure is by cell recognition.

Most datasheets are found as digital-born documents, however, as their internal structure and meta-data does not follow consistent structures or themes, it is more prudent to approach them as they were scanned images rather than electronic documents. A deep learning approach that specializes in image processing serves as a promising candidate for table extraction. Table extraction has two parts

- First, there is table detection. Which is classification and localization of table objects. Which is the detection of the presence of a table within a document and then generating a bounding box showing the encasing coordinates of the table boundaries.

- Second, there is the detection of the anatomy of a table. This is done through detection of table row and columns. By segmenting each row and column separately and then combining the output results, specific data points can be extracted. This can also be accomplished with cell detection where individual cells of a table is detected. Therefore each cell will be classified and localized within a table to generate an anatomy of the table structure.

Table detection is challenging since it requires classifying tables among surrounding text and other figures that serve as noise. Graphs and figures are the most prominent sources of noise, for tabular cell structure detection on tables separated by white spacing, noise is more prevalent between the lines. Whereas detecting the structure of a table is even more challenging due the presence of split columns or rows as well as nested tables or embedded figures. A study by [20] showed that the deep learning-based detector Mask Region-Based Convolutional Neural Network performed well in table detection and structure detection. For table detection they were able to achieve a precision of 0.974 on the ICDAR2013 dataset, and for structure detection they were able to achieve a precision of 0.95. They achieved state-of-the results and showed great promise for deep learning-based models over heuristic based or mathematical morphology based models. Graphs and figures that present structural features may be falsely classified as tables. Tables have an infinite amount structures especially with heterogeneous tables. Tabular cells can be large in volume and small in size.

The two parts of tabular information extraction are often independent, namely table detection and table structure detection. This thesis tackles the table structure detection in cell recognition. Deep learning approaches have been used for both parts, Convolutional Neural Networks (CNNs) has shown great capabilities in classifying objects from images ([21, 22]). Specifically, Faster-RCNN and Mask-RCNN has shown to be state-of-the-art detectors for many object detection tasks, it has also shown to outperform other models specifically for table detection task ([23]). Table anatomy is often difficult for CNNs to recognize due to the fact that rows and cells will often be only a few pixels in height, this adds complexity for CNNs [24]. The larger filters and pooling layers within the network works better with detailed features and suffers on smaller objects ([25]).

1.1 Motivation

Electronic datasheets within supply chains are published in great quantities and varies greatly in formatting style. The introduction of new products keeps occurring through ingenuity and technological excellence, and thus new products (with new associated datasheets)

are launched, and existing datasheets are frequently revised (and published essentially as new datasheet documents). The sheer volume of these new supply chains product documents is becoming increasing difficult to handle with manual efforts. The difference in formatting styles from different manufactures around the world with the lack of official guidelines and standards is outstripping existing extraction methods that require well structured data with access to sufficient text metadata. Indeed, as many documents are obtained via scanning, or OCR that identified individual characters, they contain no metadata, and cannot be used with most commercial tools. Therefore a robust methodology for table structure detection is both highly desired. For table detection it is vitally important to have high localization performance for the detected bounding boxes. Important information can easily be cropped out due to poor performing models. The evaluation metric is used in object detection tasks rely on an Intersection over Union (IoU) value to determine if an object is deemed detected or not. This is a comparison between the overlap of ground truth bounding boxes and predicted bounding boxes. Researchers often test the models at 50% IoU or an average of IoUs over different thresholds, however, verifying high performance tabular detection results at high IoU thresholds is needed to ensure a proper first step in a highly accurate data extraction pipeline.

For tabular structure detection, there is a lack of quality labeled data present. Recognizing tabular structure is more complex due to the heterogeneous nature that tables can take on; such as split and merged tables, as well as embedded tables. Researchers focus on detecting the row and columns to obtain the tabular structure, however, combining the outputs of the two classes for pin point data extraction is difficult due to the complex structures certain tables make take on. Heterogeneous tables may have many split cells and merged cells that create scenarios where only one element exist in a particular row or column. These scenarios can cause trained models to overestimate or underestimate the outer limits of that particular row and column, thus creating more room for error to occur. The error of both row detection and column outputs would be compounded when joint to determine exact tabular cell coordinates.

1.2 Objectives

The goal is to propel automation in document processing within supply chains. The proposed methods should work on all digital documents whether born digital or scanned without relying on PDF metadata. The focus is extracting content out of tables, since often times the most crucial technical specifications and data are stored within a tabular format. There are two distinct challenges to identifying tables and their location in documents,

specifically detection of tables and detection of the internal structure in tables. The goal of this study is to propose a highly accurate model for table detection that performs well even at high IoU thresholds. As well as to propose an tabular structure detection model that focuses on detecting tables by identifying each individual cell rather than the row and columns for better pin point data extraction on heterogeneous tables.

1.3 Contributions

After examining related works pertaining to the realm of document analysis and table extraction. A document analysis pipeline for data extraction was proposed. Two models were proposed to support the proposed data extraction pipeline; one for table detection and one for table cell structure detection.

Our main contributions can be summarized as follows;

- The first contribution is the proposal of a deep learning based object detection model for table detection that improves bounding box localizing performance at all IoU thresholds and especially at high thresholds of 95%. This is done by improving the proven Faster-RCNN [26] model for table detection by incorporating the Kullback–Leibler loss function [27] that calculates the divergence between the probabilistic distribution between ground truth and predicted bounding boxes. As well as adding a voting procedure into the non-maximum suppression step to produce better localized merged bounding box proposals.

- This thesis generated a benchmark for the ICDAR2013 cell structure dataset and compared state-of-the-art detection models Mask-RCNN, Cascade-RCNN, Cascade-Mask-RCNN and Hybird Task Cascade. Different backbone combinations were presented, namely ResNet101 [28], ResNeXt101 [29], HRNet [30]. ICDAR2013 and ICDAR2017 datasets were hand labeled for cell structure detection as quality labeled data are scarce within this field. The performance metrics across IoUs of 50% up to 95% is shown for each model for better gauge of model performances.

- A deep learning based object model that focuses on tabular cell structure detection. The outputs of the table detection model serves as the inputs of the table structure detection model. We reduce the complexity of structure detection by only training on the detected table region. The proposed tabular cell structure model is an ensemble model that consists of Hybrid Task Cascade [31] and dual ResNeXt101 [29] in the

Combination Backbone Net (CBNet) ([32]) architecture as well as the addition of Soft-Non-Maximum-Suppression (Soft-NMS).

1.4 Thesis Outline

This thesis is comprised of six chapters. In Chapter 2 we present a literature survey on previous works aimed to tackle the same types of problems in the field of document analysis, with specialization in extraction of tables and table structures. In Section 2.1, we present hand-crafted rule based and heuristic based methodologies that have previous been proposed by researchers for document analysis. In section 2.2, we present the machine learning methodologies that have been adapted for document analysis. Often times researchers would use a combination of heuristics to extract features and use a machine learning classifier. In section 2.3 we present the deep learning methodologies that have been used for table detection as well as table structure detection. Most predominantly using CNNs and variations and improvements on CNN architectures.

In Chapter 3 we explain the deep learning methodology that applies to our experiments. Section 3.1 gives a background and explanation of two stage RCNN detectors that we use throughout the paper. Section 3.2 then continues with explanation of backbone networks that serve as a crucial part of a detection pipeline. Section 3.3 explain the architecture of ResNet and how it address key issues of degradation when utilizing deeper networks. Section 3.4 covers the most prominent detector in the field of tabular detection, Faster-RCNN. This model was used in our proposed table detection model as well. Section **??** covered FPNs and how they enable detection performance at different scales, they benefit our table structure detector due to the small nature that table cells can take on. Section 3.6 explains the use of the NMS algorithm and how it integrates with the rest of the detection pipeline.

In Chapter 4, we present our methodology to improving table detection with respect to bounding box localization. The datasets we used are presented in Section 4.2. The training details are then covered alongside the evaluation metric in Section 4.3. Details on the methodology and improvements added are presented in Section 4.1. Section 4.5 concludes the table detection portion of this thesis.

In Chapter 5, we present our methodology to improving table structure detection with focus on detecting cell location instead of row and columns for better pin point extraction of data. The datasets we used are presented in Section 5.2. The training details are then covered alongside the evaluation metric in Section 5.3. Details on the ensemble

model methodology are presented in Section 5.1. Section 5.5 concludes the table structure detection portion of this thesis.

Lastly, Chapter 6 discusses the approaches used within this thesis and summarizes the research findings. The thesis is finalized with future directions in Section 6.1.

Chapter 2

Literature Survey and Background

Documents are more prevalently existing in digital formats versus the traditional physical format. The documents can be created and filled digitally, scanned from physical or filled via converting physical data. These documents can contain valuable data such as product specification, cost, dimensions [1]. Data is often kept in organized formats such as tables where the reader can easily access and understand the content. Tables contain attribute value information for association with a particular key. The rows and columns of a table can be used to group important comparison information [33]. Therefore identifying tables within documents and understanding the table structure is vitally important to the success of data retrieval. Automated data extraction has been extensively studied over the years, methods for extracting tabular information range from hand crafted rule-based methods [34, 35] to PDF-based methods [36] that utilize meta data from PDF document to identify tables and contents. Recently, deep learning-based methods that tackles tabular detection as a computer vision problem has shown great potential [11, 20]. The tables are treated as objects and the the structured nature of the tables are learned, either from the white-spacing , guiding lines or tabs. Table detection is detecting the presence of a table within a document and providing the location of the table itself. Table structure detection is detecting the internal structure of the table, showing the location of each area that contains content, this can be achieved through two methods. Either identifying row and columns and then combining them both to achieve exact locations that contain content or by identifying the individual cells. Documents that contain meta data are much easier to work, information is stored within a structured digital data format that can be parsed and extracted at will, this is however much more time consuming during document generation. It would be ideal for current industrial tools such as MS Azure and AWS to reveal their performance metrics on public competition datasets, it would enable better comparison

between the state-of-the-art methodologies and industrial methods.

2.1 Rule Based Methods

Automating document analysis using computerized tools and techniques have been research for decades. Analyzing documents is a tedious task that is simple for humans to do, therefore it is a prime area for automation. One of the earliest works was presented in 1968 [37], where researchers tried to develop methods to translate physical documents to digital documents. This required a human operator to scan papers with a joystick control, this method proved to work with complex documents such as research journals. They classified text by identifying white spaces above and below the text with a certain density of black spots in the middle. They used a 780 bit shift register for scanning the text. The register will return zero if it hits a white spot and return one, if it hits a black spot, certain combinations will yield different numbers. They also proposed an automated method for which a scanner is used to obtain a negative of the document for unsupervised operation. This work propelled document analysis and established the fact that research in this sector is needed.

The early models and methods usually rely on heuristic approaches; features and rules were selected from an opinion basis through logical deduction. Wahl et al. [38] proposed a methodology to classify and segment regions of text and images for digitized documents. This was achieved through block segmentation that divides the document into subsections based on the type of data, each section would only contain one data type. A constrained run length algorithm was proposed for detecting long lines. A bitmap is used to map black and white pixels in binary and depending, the bit string is applied to each line and when a predefined threshold of adjacent similar binary inputs, the constrained run length algorithm will replace the input with its inverse. The output is formatted in a table describing the positions of the blocks and the respective data types.

Pyreddy et al. proposed TINTIN [39] in 1997 for table retrieval in electronic documents. TINTIN uses a heuristic based model to separate tables from text, this is done by looking for aligned white spaces. A data structure called the Character Alignment Graph is used to check white spacing between contiguous text, this is used to identify gaps within blocks of data. A certain threshold is set for white spacing between characters and if it is shown to be above this threshold, the gap will be considered as a potential separator for a column. The outputted product is a text table that contains the captions and headings of the table. The heuristics used are mainly gap structure heuristics that identify large empty spaces, alignment heuristics to check if characters of two or more lines are aligned, pattern

regularity heuristics to separator lines from captions, differential column count heuristics to identify the beginning of a table and differential gap structure heuristics to identify lines that have an irregular gap alignment. TINTIN proved that simple heuristics can be used to detect tables, it is a pioneering work that can be built upon for further improvements. With the advent of digitization, the formats of tables are ever expanding, complex heuristics may be needed for the vast variety of table structures available in modern times.

Jain et al. [40] propose a high performance representation scheme to convert paper document to a specific electronic version. Those documents contains complex layouts, they utilized skew detection as well as logical and geometrical layout analysis. Through scanning journal papers, representation of document images is achieved by page segmentation. The high level performance of page decomposition system is largely based on the accuracy of the page segmentation as well as several region's label ,e.g., rulers, images,drawings,tables and text. Not only does this novel document model contains top-down generation information but also can represent documents methodically such as transfer, retrieval, storage, editing and layout analysis. Labeling, segmentation and orientation estimation are also involved in this paper, which is different from other papers within the same field.

Kieninger and Dengel [41] presented a well-known and widely used system which is T-Recs table recognition system. The goal of their work is to segment random documents and is based on different level: words, sentences, paragraphs. Instead of focusing on line detection, T-Recs is a bottom-up method that is detected by utilizing words. They start with words and a bounding box. The boxes will be linked together if they are in an adjacent position. If their neighbouring relation is horizontal, they will be recognized as from the same table. A new rule is developed by authors to detect its inside structure such as columns and rows. [41] promote the original T-Recs detection model by linking the vertical blocks together and then generating a new table. Compared to the original method, combine the blocks together to detect tables, their method has a higher accuracy. T-Recs is a significant progress, which can achieve a arbitrary table detection without rely on the lines detection.

Tabfinder, proposed by Cesarini et al. [34], they recognized the line positioning of a table to detect table boundary coordinates within a document image. A hierarchical representation based on the MXY tree is utilized to detect the document. To represent the table, adjacent lines are searched by MXY tree if they are paralleled to each other. By identifying the blank spaces and vertical lines which are the region between the parallel lines to locate table and meet the resemblance criteria. This paper shows the capabilities of hand-crafted rules for table detection.

Gatos et al. [35] 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.(developed a smart table representation for the document environment. The document layout description and the performance of OCR can directly relate to the non-textual content detection such as lines and tables which are continual images. To achieve the effect, the authors pre-processed the images then detect lines and remove it horizontally and vertically, line intersection and table are detected and reconstruction finally. The advantages of this paper is that the author take many formats of documents such as bank cheques, handwritten documents and scientific journals in to consideration.)

Linmen and Xiong [42] introduced a new path representation and analysis of table of contents regions depending on its relevant context. The authors combined layout analysis and natural language processing to promote the tables of contents detection. Different from other works that need machine leaning models for individual documents and only focus on the table of contents without take other contents at different pages into account. The relation between text contents as well as its specific page numbers are integrated together to improve its detection accuracy. Besides, the whole content of document' s information is adequate utilized for the wide variety of documents' detection without relying on the learning models for specific documents. The reason to do so is to leverage the constant association between table of contents and other pages' content from same document. A high performance algorithm based on dynamic tree dictionary ,text chunking and table of content graph description is proposed to detect hidden links which are repetitive in the table or insinuated in the phrases. The primary steps of hidden links detection: set up table of content pages' image description and tree dictionary for every participant; through matching the association with the table of contents and its other pages to divide the candidate page into text chunking; calculate the core of each table of contents page candidate on account of text mining as well as numbers of pages; confirmed candidate table of content page will be associate to the specific articles; greater than a given threshold's title-page-core will be determined as a title page.

TARTAR [43] was able to transform arbitrary tables into logical form and by transforming them into frames. A certain logical step is given: (1)normalize and clean the tables which is to obtain a description and utilized in the subsequent steps. Note that this description is different from the encoding type of input document; (2)accomplish structure detection include token type hierarchy, initial assignment of functional type and proba-

bilities, detecting logical table orientation and discovery and leveling of regions;(3) build functional table model; (4)enriching of functional table model include discovery semantic labels and mapping functional table model into a frame; step2-4 are repeated until the semantic level of conversion process is reached. Each F-logical frame which associated with a specific input table will be system returned as a output.

Liu et al. (2009) [44] proposed two algorithms to solve a common problem which is generated by PDF text extraction tools; the error of text sequencing. Both algorithms are enabling extraction of table content where it is possible to extract sparse lines' sequences. Algorithm 1 has two steps: resorting the cross-column and the within-column. The author simply compared the width of document with the non-spare lines' average length. The algorithm 2 has four steps: fully acquire the sparse lines' non-repeated Y-axis values in a given area; rank these Y-axis values from largest to smallest; sort the sparse lines in the specific area based on the ranked Y-axis values; return all the sparse lines in that area in a sorted sequence.

Certain methods rely heavily on PDF metadata to achieve tabular detection. Fang et al. [12] uses graphic ruling lines and white spaces as visual separators. They were able to detect tabular regions and table columns. PDF-TREX ([36]) uses a bottom-up heuristics approach to recognize tables within PDF documents. They utilize spatial features to group and align the content elements of a PDF document. The table is then given as a set of cells associated with 2-dimensional coordinates. PDF-based approaches are highly effective for documents such as research papers that have meta data associated with them.

2.2 Machine Learning Methods

In recent years, machine learning and deep learning methodologies are gaining traction propelled by the improvements in processing power.

One of the early works that introduced machine learning for document analysis is Wang et al. [45]. They proposed a decision tree classification method to identify zones where images, text and tables are present. This can be used within a document analysis pipeline to divide the types of data and then process each type with a following methodology. Wang et al. [46] later presented a paper that utilizes machine learning for detection of tables on web pages. They used continuous-value decision trees [47] as well as SVM [48] with layout features, word group features and content type features to classify whether tables are genuine or non-genuine tables. The content type features included the average number of columns, standard deviation of number of columns and geometric data relating to the

table size and cell size. The Content type features contains the different types of data present in the tables. Word group features removes commonly occurring words and maps the remaining into a vector, association with certain words represent whether a table is genuine or non-genuine. The definition of genuine and non-genuine table is a bit ambiguous in this work. It does not focus on locating tables, more so it focuses on analyzing tables to classify whether the contents in the table is useful or not.

Shetty et al. [49] used CRF [50] for labeling extracted content in scanned documents. They mainly focused on identifying printed words, handwritting and noise. Their method performs pseudo-likelihood estimates for the CRF model parameters through the probability of the labels under Gibbs sampling [51]. The model is trained using conjugate gradient descent [52]. Although this work does not directly classify tables, it does show the potential of utilizing CRF models for document analysis and can be adapted to include other content beyond the three that they selected.

Ng et al. [53] classified tables as well as rows and column within free texts such as wall street journal new documents. They utilized back-propagation with C4.5 decision trees for classifier generation. Their focus is first detecting the boundaries of the table and then the row and columns, this is accomplished by detecting horizontal and vertical lines. The leftmost and rightmost vertical lines along with the topmost and bottom-most horizontal line makes the boundaries of the table, and the row and column structure is deducted with lines that are presented within the boundary. This methodology falls in line with our approach of first detecting the presence of a table and then determining the internal structure of the table. However, modern tables with more complex structures or tables that are solely separated by white spaces may prove challenging for this method.

Silva proposed HMMs for detecting tables in text for document analysis [54]. Their methodology is to group potential lines into tables by leveraging probabilistic characteristics of table components. Their work showed that independantly trained document structure detectors can be combined optimally by utilizing HMM to balance them.

Kasar et al. [55] proposed a method that uses a combination of logical heuristic methods with a machine learning classifer to detect tabular regions. Their method solely relies on the separator lines between rows and columns present in the input image. The intersection of the horizontal and vertical separator lines are used to generate a set of 26 low level features to feed into a SVM classifier to identify whether a table is present or not. This study is one of the first to use SVM for tabular detection, however the performance is much more heavily dependant on the heuristic approaches they used to obtain the horizontal and vertical lines.

Distant Supervision was proposed by Fan and Kim [56] for detecting table regions

in PDF documents. Distant Supervision uses heuristic annotation to generate a large database of weakly labeled training data. Three classifiers are then used to jointly vote on whether the proposed region is a table or not; Naives Bayes, Logistic Regression and SVM are used. They solve the problem of labeled data through using weakly labeled data and they compensates the noise in the data by utilizing an ensemble model of three classifiers.

Cermine was proposed by Tkaczyk et al. [57] for extracting structured metadata from scientific documents. Cermine uses PDF metadata to determine sections in the document. The sections are then classified into several classes, namely, content, references, metadata. They use a SVM classifier that has lexical, sequential, geometric , formatting and heuristic feature inputs. Overall the two step process of determining zones and then classifying the zones of a document is a trend that is seen throughout other research papers. This lowers the complexity of the problem and it is also similar to the methodology that our thesis proposes, first identifying the presence and location of a table then specifically focus on that location for understanding the table internal structure.

2.3 Deep Learning Methods

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. Deep learning methodologies for image classification has been propelled through the advent of CNNs. With the support of improved processing power from GPUs [58] showed that capabilities of image classification, their work propelled deep learning methodologies for computer vision problems. Ever since then, researchers have also been utilizing deep learning for tabular detection tasks in document analysis. Deep learning is an image based approach that takes images as an input rather than taking text files as an input. Image-based approaches are more generalized as such it can be used to detect all documents and is not limited to PDF documents or excel documents, specifically scanned documents that do not have proper format.

Hao et al. [59] proposed one of the earlier works on integrating CNNs for tabular detection. Their work focuses on frist proposaling regions that may contain tables, the types of are split into tables with either horizontal rulling lines, vertical ruling lines and no ruling lines. These proposed regions are then fed to the CNN for classification. Their method outperformed other heuristic based approaches on the ICDAR 2013 dataset [8]. However, their method is only utilizing the CNN as a classifer, whereas the central part of their methodology relies on the heuristics methods to obtain the proposed regions for input.

DeepDeSRT was proposed by Schreiber et al. [11], it is an end-to-end model that tackles table detection as well as table structure detection. This is one of the first works that solely depends on deep learning and does not include any heuristic methods. The input is a document image and the output is the output is the detected table regions and tabular structure row and columns. The authors applied transfer learning to deal with the data scarcity. Tabular structure data is vastly limited within the document analysis domain. They implemented Faster-RCNN [26] for table detection and FCN [60] for row and column detection. Their work demonstrated the capabilities of end-to-end use of deep learning methodologies in the document analysis field. Faster-RCNN is a state-of-the-art deep learning architecture architecture that has excelled at many object detection tasks [61, 62, 63]. Variations of Faster-RCNN have been used widely for document analysis ([23, 14, 64, 65, 20, 66, 67, 17]). Faster-RCNN ([26]) was first shown to perform well for table detection by Gilani et al. ([17]). They augmented the input images to look more natural with three distance metrics to show the dimensions in the images. These metrics include the distance between whitespaces and texts. This was on of the earlier works that used image pre-processing for tabular detection performance improvements.

Arif and Shafait [67] adopted Faster-RCNN with their own methodology, they preprocessed different parts of a document by color coding foreground and background features and tackled the table detection as a scene recognition problem. They proposed Faster-RCNN with a combining corner method that groups detected corners through coordinate matching and filters out unreliable corners. The bounding boxes are refined based on the corresponding corner group and this improves pixel-level precision of the table boundary.

Traquair et al. [14] showed that Faster-RCNN performs well on table recognition with only bounding box annotations. Kara et al. [23] utilized Mask-RCNN which adds an additional masking layer to Faster-RCNN to improve instance segmentation to perform table structure detection on rows and columns. In another work, Kara et al.[64] added a judging module to the detection pipeline of Mask-RCNN to show improvements to table detection, structure detection and an end-to-end detection.

DeCNT was proposed by [65] to use a novel combination of Faster-RCNN/FPN with deformable convolution networks. Specifically deformable ResNet 101 with deformable ROI pooling within faster-rcnn. Deformable convolutions allows for extra offsets in the convolutional sliding window appraoch to dynamically adjust its receptive field based on the surrounding location and target object. The offsets are generated by a separate convolutional layer, they are implemented through bi-linear interpolation. DeepTabStR ([68]) also proposed deformable Faster-RCNN with deformable ResNet101.

We [69] introduced a Kullback–Leibler (KL) Loss function version of Mask-RCNN with

a variance voting non-maximum suppression post processing step to improve the precision to 0.987 on the ICDAR2013 dataset. Their work has shown to improve the detection performance at high Intersection over Union(IoU) values of 95%. Therefore generating a model with accurate bounding box predictions of the detected table.

TableNet was proposed by [70] to utilize FCN with VGG19 as a backbone for table detection and row and column structure detection. CascadeTabNet was proposed by [18] to utilize Cascade-Mask-RCNN with High Resolution network (HRNet) to achieve a 1.0 F1 Score on the ICDAR2013 table detection dataset, showing higher results than TableNet and DeepDeSRT. Cascade-Mask-RCNN is the combination of Cascade RCNN and Mask-RCNN that includes the cascading architecture and masking layer to improve detection performance and instance segmentation.

Li et al. [71] proposed a combination of unsupervised clustering with CNNs for POD. They use unsupervised clustering on line regions to classify them into specific objects. First, Binarization is done on the document, then column regions are segmented and line regions are extracted. The regions belonging to the same cluster are merged to obtain objects. CRF model is used to classify the line regions into specific objects such as tables, figures, formulas and text lines. Figure regions that contain more than one figure is split with heuristic based approaches. The classified regions are then fed to a CNN to verify the previous classifications. Their model achieved the highest performance metric results on the ICDAR 2017 POD dataset [72].

Table 2.1: Overview of document analysis and table extraction research covered.

Rule-Based	Machine Learning-Based	Deep-Learning-based
[37, 38, 39, 40, 41, 34, 35, 42] [43, 44, 12, 36]	[45, 46, 49, 53, 54] [55, 56, 57]	[59, 11, 23, 14, 64, 65, 67] [17, 20, 65, 69, 70, 71, 18]

Deep learning shows promise to be more comprehensive and generalize better to the ever varying document formats. Deep learning is superior to PDF-based methodology since it does not rely on metadata, as well as the fact that all document formats are incapsulated due to the input is image based [1]. The augmentation techniques that can be applied within deep learning allows for the model to adapt better to scanned images. Machine learning methods are often combined with heuristic methods for a complete tabular extraction pipeline [55, 56, 57], however the model is always limited by the hand crafted heuristic rules.

Chapter 3

Methodologies

Documents are constantly being processed within supply chains in various industries throughout the globe. Within those documents, often times the most important content is stored in tabular format. Processing documents for transfer of knowledge or for communication is a labour intensive and error prone process. Therefore an automated technique for supply chain document processing is highly desired. Deep learning approaches show promise to deliver an end-to-end extraction model. In this thesis, we want to focus on detection of tables and table structures to automate document processing. Figure 5.1 shows the proposed document analysis pipeline for data extraction. This thesis supports the proposed pipeline by proposing the architecture and deep learning methods for classifying and localizing tables and their respective internal structures. This thesis utilizes state-of-the-art deep learning approaches to improve localization performance for table detection as well as overall performance metrics in recall and precision. For table structure detection, cell structure detection is proposed over the traditional row and column detection methods for better data extraction and performance on heterogeneous table layouts. The tables are first detected with our table detection model, then the detected table cropped based on the bounding box coordinates. The table structure detector then classifies and locates the individual cell locations on the cropped table. Identifying the presence of a table is an easier task and more well studied task when compared to table structure detection. Table detection also has many well-labled public datasets to support research in that field, however, tabular structure detection is far more complex due to the possibilities of different table layouts. By focusing on each task individually, we are reducing the complexity of the problem. Also, since table detection is a simpler task, a lightweight model can be used, whereas a larger complex model is needed for tabular structure detection. Splitting the problem into two distinct steps allows us to narrow the problem down and reduce noise

within the problems. We assume 100% accuracy in table detection stage to ensure that the performance of our structural detection model is strictly based on its capability to capture tabular cells. As with the nature of tabular cell detection, each individual piece of content can be understood as a cell, therefore if the whole document is used as an input then each individual text element can serve as noise to the cell detection model. Deep learning models enables the model to learn the architecture that forms a table, in comparison to heuristic methods that that dependant on hand-crafted rules to pre-determine certain features of a table.

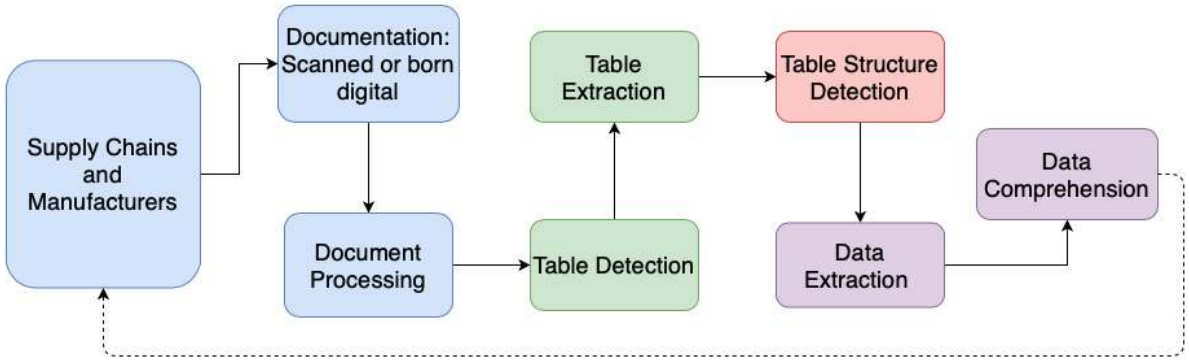


Figure 3.1: Document Analysis pipeline for Data Extraction

This section covers the main parts of the deep learning models that was used to accomplish table and table structure detection. Details on specific methodologies for table detection and table structure detection is covered in Chapter 4 and Chapter 5 respectively.

3.1 Regional Convolutional Neural Networks

Object detection relates to classification and localization of the of the classified object. State of the art detection algorithms are currently all based on two stage detection [73][74]. Two stage-detectors consists of a feature extractor backbone, regional proposal stage and detector, they are categorized as RCNNs. The difference between a two stage versus one stage detector is the additional RPN stage. An example of a two stage RCNN detector can be seen in figure 3.2. The RPN adds an additional stage to the detection pipeline by providing the detector with regional proposals that proposes regions to focus on for the detector, this simplifies the detection task and yields higher accuracies [75]. Whereas

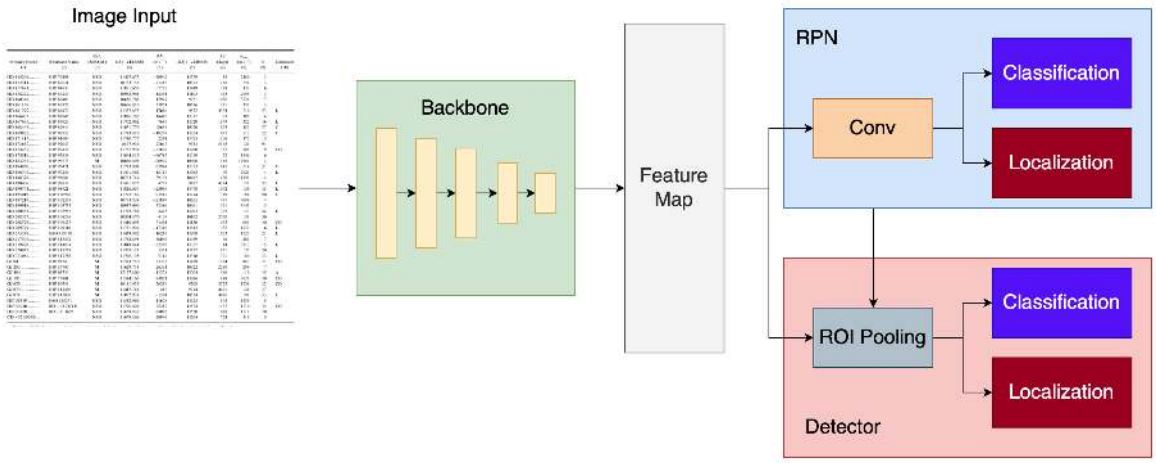


Figure 3.2: Two Stage RCNN detector

one-stage detectors only consists of the detector and backbone, the task is treated as a regression problem that attempts classify and localize the input directly [76, 77, 78].

RCNNs were first proposed by Girshick et al. in 2014 [75] and has found tremendous success in the field of computer vision. The state-of-the-art models at the time were large ensemble models and lacked innovation in detection sturcture, later on the RCNN archetcture dominated the computer vision scene with models such as Fast-RCNN [79], Faster-RCNN [26], Mask-RCNN [22], Cascade-RCNN [80, 81], Grid-RCNN [82], Libra-RCNN [83] , and Dynamic-RCNN [84]. The key contributions from [75] are the addition of the regional proposals within the detection pipeline for a new category of CNN architecture as well as proving the effectiveness of leveraging transfer learning in tasks with scarce labeled data. Transfer learning is a technique that leverages a backbone feature extractor that is pre-trained a large dataset such as the ImageNet Dataset [85]. The pretrained model is inputted and fine-tuned with the limited amount of labeled data to orient the new model to be biased for the chosen application. [86] showed that RCNNs are great for detecting smaller objects in computer vision problems, which is beneficial for our approach to table structure detection, where we are focused on detecting cell regions.

3.2 Backbones

Backbone Networks are an integral part of a detection pipeline, they serve as feature extractors and provides the detection model with a feature map. Fully training an object detection model would require an enormous amount of quality labeled data, which reserchers often do not have. In recent years, the transfer learning approach of utilizing supervised pre-training and then performing application based fine-tuning has proven to be the widely accepted paradigm. Modern state-of-the-art backbones are all pre-trained on versions of the ImageNet dataset, such as VGG [87] , ResNet [28] , ResNeXt [29] and high resolution net [30]. These backbones are initially designed for image classification tasks but have been adopted for object detection to extract basic features. Studies have shown that backbones with more layers provide a richer feature map and improve detection performance [88, 32]. This can be seen through the performance of ResNet50, ResNet101, ResNet152. However, the increased performance comes at the cost of computational complexity. The backbone is one of the areas that was focused on for improving tabular structure detection. This was done by not only utilizing a more complex backbone, but also using a novel backbone integration structure that combining multiple identical backbones.

3.3 ResNet

The trend of larger and deep networks was influenced by [58]. Ever since then, networks are becoming larger to improve performance metrics. Residual Networks [28] was proposed to solve the problem of vanishing gradients and accuracy reduction when more layers are added to a network. This is solved by incorporating shortcut connections. Shortcut connections allow for comparisons of residual networks that have the same characteristics of depth, width and number of parameters while not introducing any additional computational complexity, they are a connection that allows skipping multiple layers. The stacked layers were used to fit a residual mapping instead of expecting the stacked layers to fit according to the original mapping. Suppose the underlying mapping is H(x). By fitting the layers to a residual mapping, it can be denoted as F(x)=H(x)-x. Then the original underlying mapping will be transformed into F(x)+x. It would then be more simple to optimize the residual mapping. F(x)+x is accomplished through shortcut connections of feed-forward neural networks. He et al. [28] showed that residual networks solve the degradation problem, is easy to optimize compared to traditional networks and continuously improve in accuracy even at significantly large depths of over 100 layers.

A 50 layer ResNet model (ResNet50) is used for the table detection task in this thesis.

The archetecture of ResNet50 that was used in this thesis is shown in figure 3.3. There are strides associated with each convolution block, respectivly conv2, conv3, conv4, conv5 has strides 4,8,16,32.

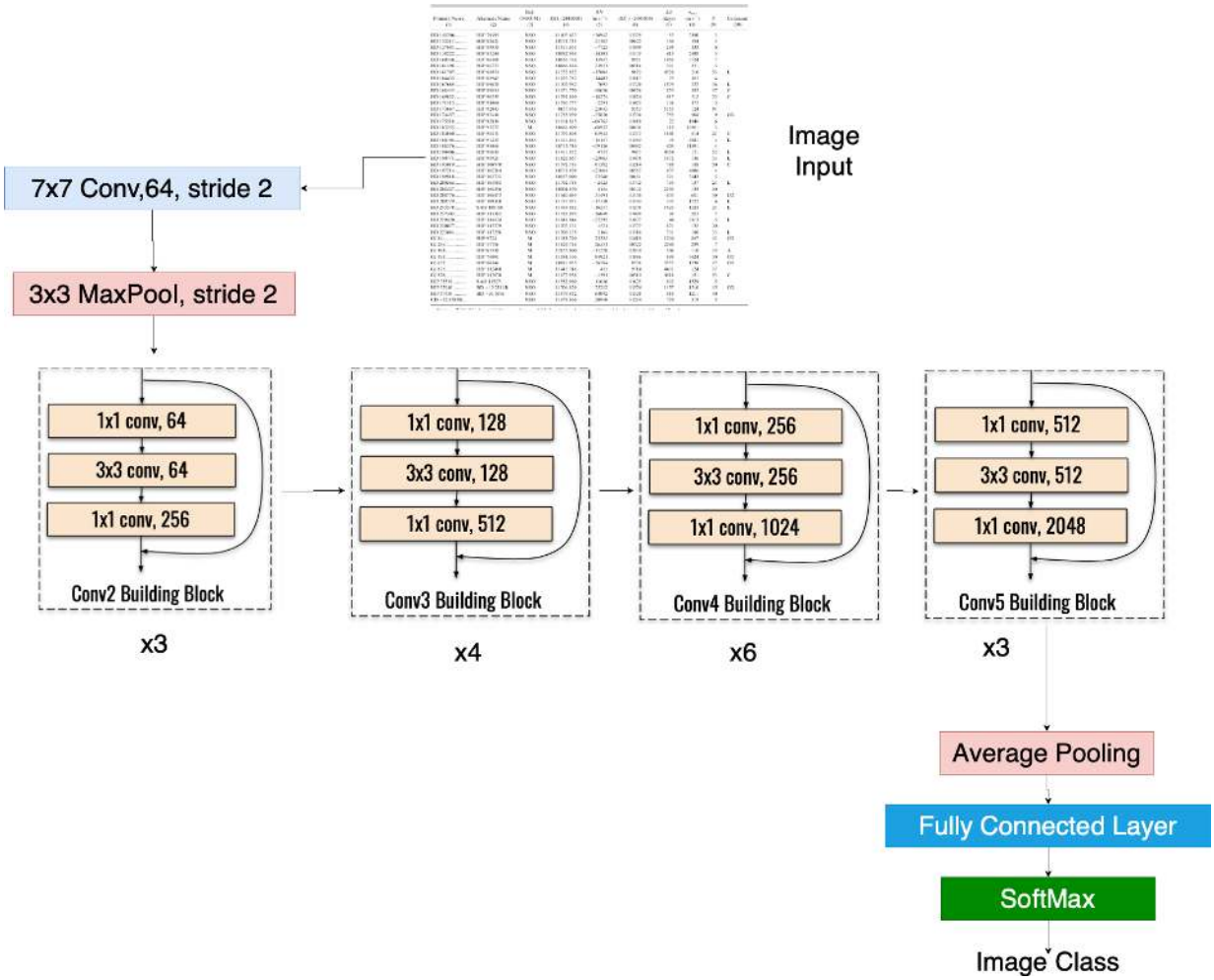


Figure 3.3: ResNet50 Architecture

3.4 Faster-RCNN

Faster-RCNN is an improvement on the RCNN category of detectors by solving key issues in CNN based detection algorithms. A regional proposal method that was commonly

used was selective search [89], it did not have learning capabilities and it had a slow processing speed of two images per second. Faster-RCNN proposed RPN that is a deep neural network that shares layers with the detector. They also proposed the concept of anchor bounding boxes that would enable object detections at varied ratios and scales. Both the RPN and detector takes in the feature map that is generated by the backbone network. The RPN outputs regional proposals and provides confidence scores of the detected regions. During this process, the RPN also performs regression on coordinate localization. The confidence score represents how likely the detected region belongs to the predicted class. Faster-RCNN shares convolutional layers with the detector which is adopted from Fast-RCNN [79], this process enables faster processing times.

The Faster-RCNN pipeline consists of extracting a feature map with the backbone, then scoring/refining the proposals with a regressor in the Regional Proposal Network (RPN), and then finally merging multiple output candidate bounding boxes that belong to the same object with a non-maximum suppression (NMS) algorithm, this is then fed into the detector portion of the Faster-RCNN and undergoes classification and regression again, following with another NMS processing.

The feature map from the backbone is given to the RPN, a small convolutional network that uses the sliding window approach goes over the feature map. We can assume the input of the small network is of size n x n. Each of the sliding windows will then be mapped to a lower-dimensional feature that is sent to the regression and classification fully connected layers. The RPN architecture can be found in figure 3.4.

The anchors within the RPN allows for production of proposals at different ratios and scales. A Tunable parameter k is given to limit the amount of proposals. The regression layer outputs 4k coordinate proposals and the classification layer outputs 2k proposals with confidence scores. The anchors will have predetermined scales and ratios for each location. The default is 3 scales and ratio of 1:1, 1:2 and 2:1, making 9 total anchors at each location. Assume the feature map is W x H, the total anchors to be placed would be W x H x ratios x scales. The scaling capabilities enables the use of feature pyramid networks, which provide higher resolution outputs for improved performance metrics.

The loss function for an image is shown in equation 3.1. The classifcation loss L_cls is the log over the object and non-object classes. The regression loss is used for bounding box localization, it is shown in equation 3.2. The R stands for the smooth L1 loss defined in Fast-RCNN [79].

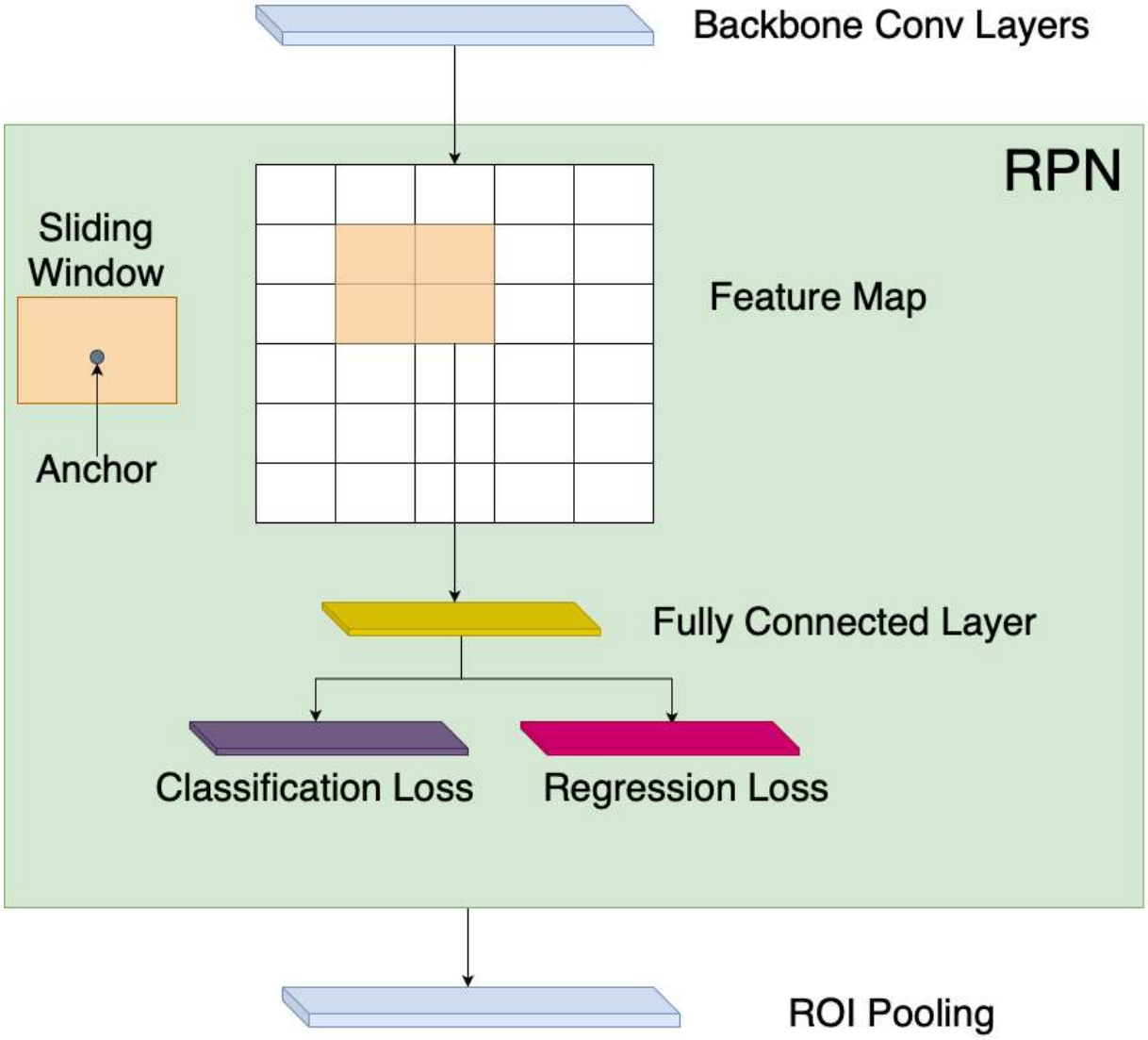


Figure 3.4: RPN Architecture

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*). \tag{3.1}$$

$$L_{reg}((t_i, t_i^*) = R(t_i - t_i^*) \tag{3.2}$$

The parameterization for regression is given by the equation 3.3. Where x and y denote the center location of the bounding box and the w and h denote the width and height. The predicted box is given by x, the anchor box is denoted by x_a the ground truth is denoted by x^*, this format also applies to y, w, h.

$$\begin{aligned} t_x &= (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \\ t_w &= log(w/w_a), \quad t_h = log(h/h_a), \\ t_x^* &= (x^* - x_a)/w_a, \quad t_{y1}^* = (y_1^* - y_1 a)/h_a, \\ t_w^* &= log(w*/w_a), \quad t_h^* = log(h^*/h_a), \end{aligned} \tag{3.3}$$

Faster-RCNN and its adapted modified versions has been a state-of-the-art detector for many object detection applications. This is also the case in the tabular detection domain. In Chapter 4 we improve upon the base Faster-RCNN model to improve bounding box localization for table detection. Table detection is a simpler task than table structure detection therefore a lightweight robust model is better suited. State-of-the-art models have already achieved success with Faster-rcnn in table detection, however, their performance at higher IoU thresholds are not sufficient for industrial deployment. For the purpose of high precision data extraction, the model would require a high degree of IoU overlap to ensure that important information is not cropped out.

3.5 FPN

FPN was proposed by lin et al. [90] as a solution to detecting objects at different scales. This have been a widely studied problem, one solution is manually scaling the

images and running the detection algorithm for each scale [91]. This method can be observed in early works that utilized heuristic image processing methods like in [92, 93]. This allowed the model to be able to perform well irrespective to the scale of the image, however, it increases computational time since the detection is performed many times in succession. Even though CNNs inheretly perform well regarding to detecting objects at different scales. [90] showed that performance gains can be possible by using pyramidal shaped feature maps. For our domain, document analysis, tables are larger objects and can be detected regardless of scale, however, detecting smaller objects such as the cells within a table is more challenging. This is due to the low resolution and noisy representation of smaller objects [94].

FPNs benefit from the pyramidal stucture to combine low-level features and high-level features semantically to generate a top-down architecture for object detection at different scales. There are two major parts to the FPN, a bottom-up and top-down architecture. The feed-forward operation shown in regular convolutional neural networks serves as the bottom-up structure basis. The layers that does not change the input size are grouped into the same level. The last layer in each group is used for the bottom-up architecture. For example, the ResNet50 backbone structure would consist of using the last layer of conv2, conv3, conv4, conv5 to create the bottom-up architecture. Each of these levels are then connected to a top-down architecture of feature maps and merged together. The features are up-scaled before adding them to the upper layers for merging.

Figure 3.5 shows the architecture of the FPN. The left side shows the bottom-up level structure and the right side shows the top-down structure.

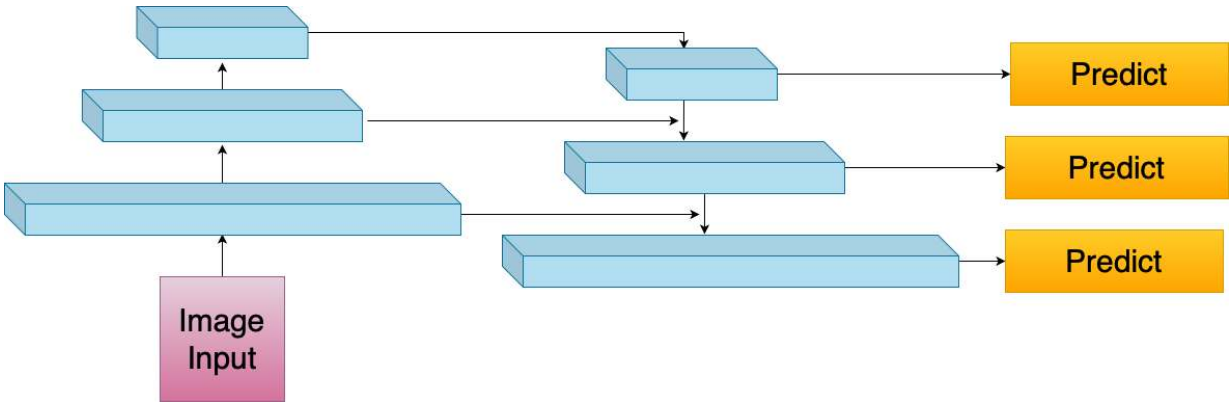


Figure 3.5: FPN Architecture

3.6 Non-Maximum-Suppression

NMS has been an integral part of state of the art computer vision applications [95, 96, 97, 98]. A certain threshold of candidate bounding box proposals are set for both the RPN and the detector. NMS is a post processing algorithm that merges the proposed bounding boxes that belong to a single detection. Soft NMS [96] and learning NMS [99] have been shown to improve over standard NMS results. Standard NMS eliminates candidate boxes with lower classification scores while learning NMS focuses on the bounding box scores to improve localization performance. Instead of eliminating objects, soft-NMS scales the detection scores of an object as a function of overlap with the ground-truth bounding boxes, this has shown to improve mean average precision of Faster-RCNN by 1.7% on PASCAL VOC 2007 [96].

Chapter 4

Table Detection

Table detection has been a well studied problem. There are many methods available, such as rule-based methods, pdf-based methods and deep learning based methods. Recently deep learning based methods have proven to be more prevalent in the field of document table detection [20]. Researchers often focus on classification capabilities in the application of table detection and have reached F1 scores of 97.8 [100]. However, the localization performance of the respective deep learning models have not been studied thoroughly. Often times researchers focus on evaluating their results at 50% IoU or an average value ranging over multiple IoU thresholds, 50% (losely localized object) to 100% (perfectly localized object) [101]. Improving bounding box localization performance is vital for information extraction. The prevention of crucial data being cropped out due to a poor performing detector is paramount to automating data extraction. By improving current detectors in localization performance we can facilitate an end-to-end model that aims to derive semantic meaning from the extracted data.

We propose a deep learning based solution for classifying and localizing tables within document images. The application of this spans from born-digital to scanned images and is generalized for a wide range of documents, ranging from conference papers, journals, newpapers, datasheets and webpages. We transform the documents to an image format and tackle it as a computer vision challenge. Taking images as an input also makes our method more universal, allowing it to work with scanned images or documents without metadata. We achieved state of the art results on public and private datasets as well as greatly improve localization performance at higher IoU thresholds of 95%.

Tabular detection accuracy is not always correlated to the localization performance of the detected bounding boxes. Valuable information may be cropped out due to in accurate

localization accuracy, which is detrimental to information extraction systems. Therefore we propose Faster-RCNN with a KL Loss function. Transfer learning was utilized to train our deep neural network. The ResNet-50 backbone was pre-trained on the ImageNet dataset. The dataset used for training is a combination of many public datasets and a private dataset. We also introduce an additional post processing step that utilizes the bounding box variances acquired from the KL loss function to initiate a voting procedure during NMS.

This chapter first reviews the datasets used in Section 4.2. The methodology used will then be covered in Section 4.1. Afterwards, the training details will be covered in Section 4.3. The results would then be shown in Section 4.4. Finally, the conclusion will be made in Section 4.5.

4.1 Methods

In this section, we present the use of Faster-RCNN, a state-of-the-art two stage object detector and our proposal to improve the tabular localization performance under Faster-RCNN. Two-stage object detection models generate multiple initial candidate bounding box proposals which thus makes it more accurate in comparison to single-stage detectors. Introduction of KL Loss function into Faster-RCNN-based object detection aims to improve its detection capabilities. Then an additional post processing variance voting step is added to improve bounding box localization performance. Fig. 4.1 illustrates our additions to the original Faster-RCNN model [73].

4.1.1 Faster-RCNN with KL Loss Function

Faster-RCNN has proven to be a strong candidate for table detection tasks [14]. Faster-RCNN improves upon Fast-RCNN with the addition of Regional Proposal Networks that can share convolutional layers with the backbone feature extractor. This in turn accelerates the running time. Faster-RCNN has a classification loss function and bounding box regression loss function. The classification loss function is used to classify whether a table is present within the document while the bounding box regression loss function is used to refine bounding box predictions and improve localization performance.

The classification loss is a logarithmic loss over the background and foreground objects, it is adopted from [73] is shown in (4.1).

Table 4.1: Table of Notation

i	Index of an anchor in a mini-batch i
p_i	predicted probability of anchor i being an object
p_i^*	Ground-truth label for p_i
t_i	A vector representing coordinates of the predicted bounding box
t_i^*	A vector representing coordinates of the ground truth box associated with a positive anchor
L_{cls}	Classification loss
L_{reg}	Regression loss
x_1, y_1, x_2, y_2	predicted corner coordinates
x^1*, y^1*, x^2*, y^2*	ground truth corner coordinates

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*). \tag{4.1}$$

The regression loss is originally the smooth-L1 loss function used in [73, 79]. It doesn't consider ambiguous ground truth bounding boxes within a dataset and in turn may result in lower accuracy when the classification score is low [27].

The KL loss function is proposed to accomplish bounding box regression as well as localization uncertainty. This is achieved by modeling the ground-truth bounding boxes as a Gaussian distribution function.

The coordinates are parameterized by a 4-dimensional vector (x_1, y_1, x_2, y_2). Each dimension represents a boundary location of the bounding box shown in (4.2)

$$\begin{aligned} t_{x1} &= (x_1 - x_1 a)/w_a, \quad t_{y1} = (y1 - y_1 a)/h_a, \\ t_{x2} &= (x_2 - x_2 a)/w_a, \quad t_{y2} = (y2 - y_2 a)/h_a, \\ t^*_{x1} &= (x^*_1 - x_1 a)/w_a, \quad t^*_{y1} = (y^*_1 - y_1 a)/h_a, \\ t^*_{x2} &= (x^*_2 - x_2 a)/w_a, \quad t^*_{y2} = (y^*_2 - y_2 a)/h_a, \end{aligned} \tag{4.2}$$

The above equation formulates the offsets of the predicted bounding box coordinates and ground-truth bounding box coordinates. t_{x1}, t_{x2}, t_{y1}, t_{y2} is the predicted bounding box coordinates while t^*_{x1}, t^*_{x2}, t^*_{y1}, t^*_{y2} is the ground-truth bounding box coordinates.

The predicted bounding boxes can be modeled as a single variate Gaussian distribution because we assume the coordinates are independent. Shown in Equation 4.3.

$$P_\Theta(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-x_e)^2}{2\sigma^2}} \tag{4.3}$$

The ground truth bounding boxes can also be modeled as a Gaussian distribution, however since ground truth bounding boxes have a variance of zero, it can be modeled as a Dirac Delta function. shown in Equation 4.4.

$$P_D = \delta(x - x_g) \tag{4.4}$$

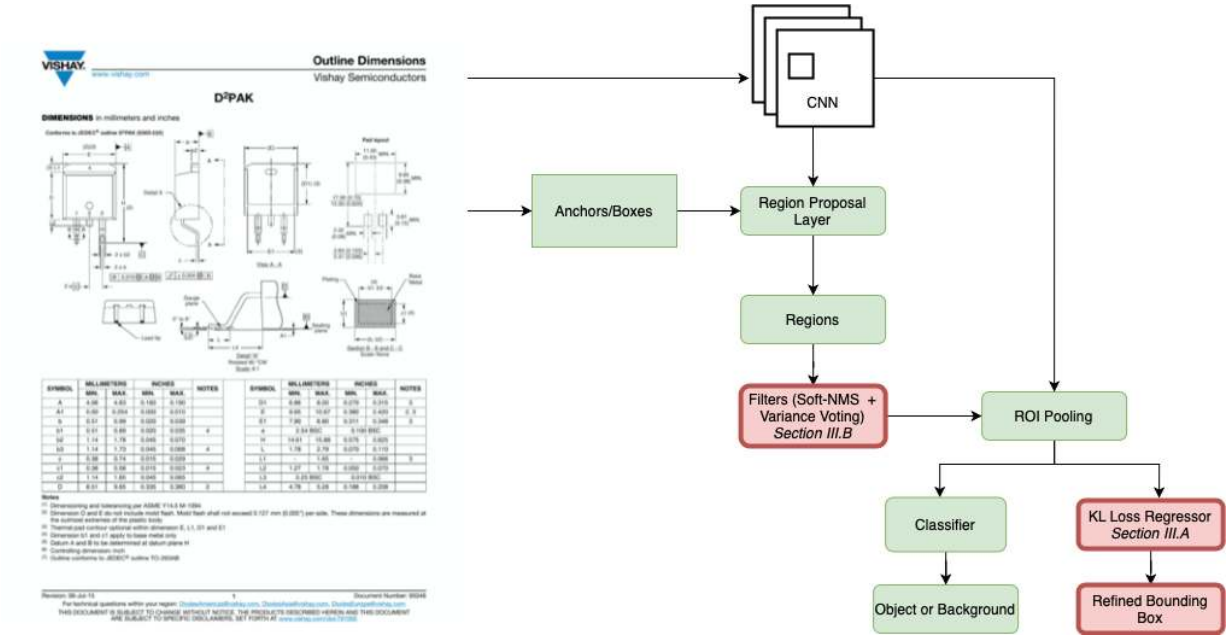


Figure 4.1: Faster-RCNN architecture with the addition of KL Loss and variance voting with Soft-NMS

The KL divergence between these probability distribution functions can serve as the regression loss shown in Equation (4.5). This model has proven to be able to predict larger variances which in turn can produce lower loss values from He et al [27]. If there are ambiguous bounding boxes, a smaller loss will be generated. The variance learned from the KL loss function gives the capability to perform the proposed post processing variance voting step.

$$L_{reg} = D_{KL}(P_D(x)||P_\Sigma(x)) \tag{4.5}$$

We chose Faster-RCNN because it is widely adopted since the first publication of the architecture, and it has been used in many tasks and domains such as face detection [102], medical chart interpretation [103, 104], and many other abstract object detection challenges as presented in [105]. The wide application of Faster-RCNN proves its suitability for transfer learning and domain adaptation. A comparison study by Traquair et al [106] reports that for tabular detection Faster-RCNN achieves high recall and precision values of 0.981 and 0.974, respectively on ICDAR2013 dataset. We use Faster-RCNN with the

ResNet-50 backbone feature extractor. ResNet-50 is trained more than a million images from the ImageNet database. ResNet-50 is 50 layers deep and can classify 1000 object classes [107].

4.1.2 Variance Voting

Variance voting is a method that votes on the location of a candidate bounding box in relation to predicted variances of neighboring bounding boxes. Boxes are given higher weights if it has lower variances and has a high IoU with the ground truth box. Variance voting was used by He et al [27] to improve bounding box localization performance of the COCO dataset. The KL Loss function acquires the variances of the neighboring bounding boxes. This is then followed by voting on the learned variances with perspective to the neighboring bounding boxes. Voting on the perspective bounding boxes with the selection process of soft-NMS can generate new positions. The new positions can be calculated according to the distance and variance to the ground truth bounding box. The formula is presented in (4.6) and (4.7)

$$p_i = e^{(1-IoU(b_i,b))^2/\sigma_t} \tag{4.6}$$

The probability of a given candidate coordinate is calculated with respective of its IoU.

$$x = \frac{\Sigma_i p_i x_i/\sigma_{x,i}^2}{\Sigma_i p_i/\sigma_{x,i}^2} \tag{4.7}$$

The new location of the coordinate is then calculated. It is worth to note that this method focuses on localization confidence therefore it does not consider classification score since the two are not directly related. This method improves the localization accuracy of the final output bounding box, resulting in a higher IoU overlap with the ground truth location.

4.2 Datasets

The datasets used are a mixture of private and public datasets. Deep learning models benefit from large amounts of training data. This ensures the final model is more generalized and works on tabular styles that it has never encountered. The public datasets

are Marmot[1], ICDAR-2017 dataset[2], ICDAR-2013 Table Competition Test Dataset[3]. The private dataset was supplied by Lytica Inc, which we call the Lytica dataset. The private dataset was provided by our industry partners at Lytica Inc[4], a company based in Kanata, Ontario, Canada that specializes in data analytics and supply chain optimization. We can call this private dataset the Lytica Dataset, which was also used in [106, 20].

4.2.1 ICDAR2013

The International Conference on Document Analysis and Recognition (ICDAR) is the flagship conference in the field of document analysis. The 2013 ICDAR table detection dataset has been used widely as a benchmark for table detection. This dataset includes government documents from the websites of the European Union and United States. These are regulated documents therefore the table structure does not differentiate too much between each other. This dataset contains 238 images.

4.2.2 ICDAR2017

The is dataset that was given in the 2017 ICDAR conference, specifically for Page Object Detection. This dataset is much larger than ICDAR2013 dataset, it has 2417 images. It consists of documemnts from academic papers of CiteSeer. A wide spread of page layout styles of tables can be seen within this dataset.

Initially it was created for four detection challenges; formulae, table, figure and page object detection. However, we focus only on the table detection portion of the dataset. The challenges of this dataset is that the other page objects, such as Figures and Graphs can often be classified as tables. A subset of the dataset is used for training and testing, specifically images that include tables; 200 images were used for testing and 617 images were used for training.

4.2.3 Marmot

The Marmot dataset is a popular public dataset for table detection published by Peking University. The dataset contains 2000 images in which 1000 contains tables, of the remain-

[1]http://www.icst.pku.edu.cn/cpdp/data/marmot_data.htm
[2]http://www.icst.pku.edu.cn/cpdp/ICDAR2017_PODCompetition/dataset.html
[3]https://roundtrippdf.com/en/data-extraction/icdar-2013-table-competition/
[4]https://www.lytica.com/

ing 1000 tables, 500 are in English and 500 are in Chinese. The English pages are from CiteSeer spanning from 1500 conference and journal papers from 1970 to 2011. While the Chinese pages were selected from a range of 120 e-books provided by Founder Apabi library in which less than 15 pages are selected from each e-book.

4.2.4 UNLV

The UNLV dataset consists of 2889 documents and was published by the University of Nevada located in Las Vegas. The document sources ranges from magazines, business letters and reports. The subset of the dataset that includes tables consists of 424 images. This dataset contains scanned images, which is a target category that our model is aiming to cover.

4.2.5 Lytica

The Lytica dataset was provided by Lytica Inc, it consists of 2323 tabular images in total. The contents of this dataset was gathered from various supply chain manufactures around the world, it mostly contains electronic component datasheets. This dataset is complex due to the varied styles of tables produced by the diverse manufacturers. This dataset also contains a multitude of heterogenous tables, such as having tables embedded within the row of a larger table. The diverse nature of this dataset allows for the trained model to be more generalized, since many table formats that are present in this dataset are not present in the public datasets. Furthermore, by utilizing industrial documents as a cornerstone, the trained model will be geared more so for commercial use.

4.3 Training Details

For training we use 8 Tesla K80s, images are pre-scaled to 600 by 1000 pixels. The loss is monitored to prevent over-fitting. The learning rate is set to 0.02 based on the scaled value recommended by Detectron [108]. The model is trained to 17500 iterations which translates to 28 epochs. Figure 4.2 shows the learning curve for our trained model, it shows that at 17500 iterations the model shows a clear convergence. Each dataset is split into 80% training and 20% testing partitions. The training partitions of each dataset is combined to form a final training dataset. A singular model is trained with this final training dataset. This model is then tested with the testing partition of each dataset.

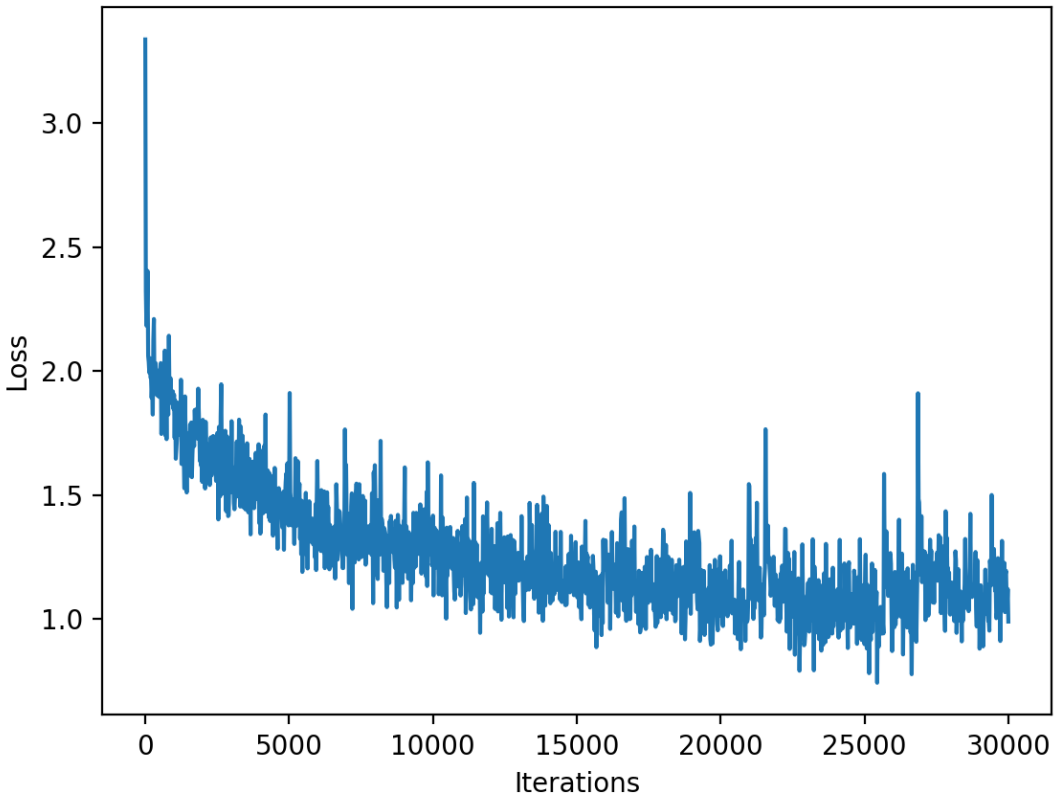


Figure 4.2: Loss function (i.e. learning curve) displaying training loss over iterations

A combination of IoU, precision and recall is used to evaluate the model. The σ_t for Variance voting is set to 0.02 according to [27].

For predicted bounding box (P) and ground truth bounding box (Gt), IoU can be described as;

$$IoU(P, Gt) = \frac{P \cap Gt}{P \cup Gt} \tag{4.8}$$

IoU is the intersection area of the candidate bounding box with the ground truth bounding box over the total spanned area of the ground truth and candidate bounding boxes.

The model is tested at different IoU thresholds of 0.5 to 0.95 in increments of 0.05 . If the detection results are higher than the set threshold the table is deemed detected.

Therefore recall and precision values are often lower as IoU thresholds improves.

4.4 Results

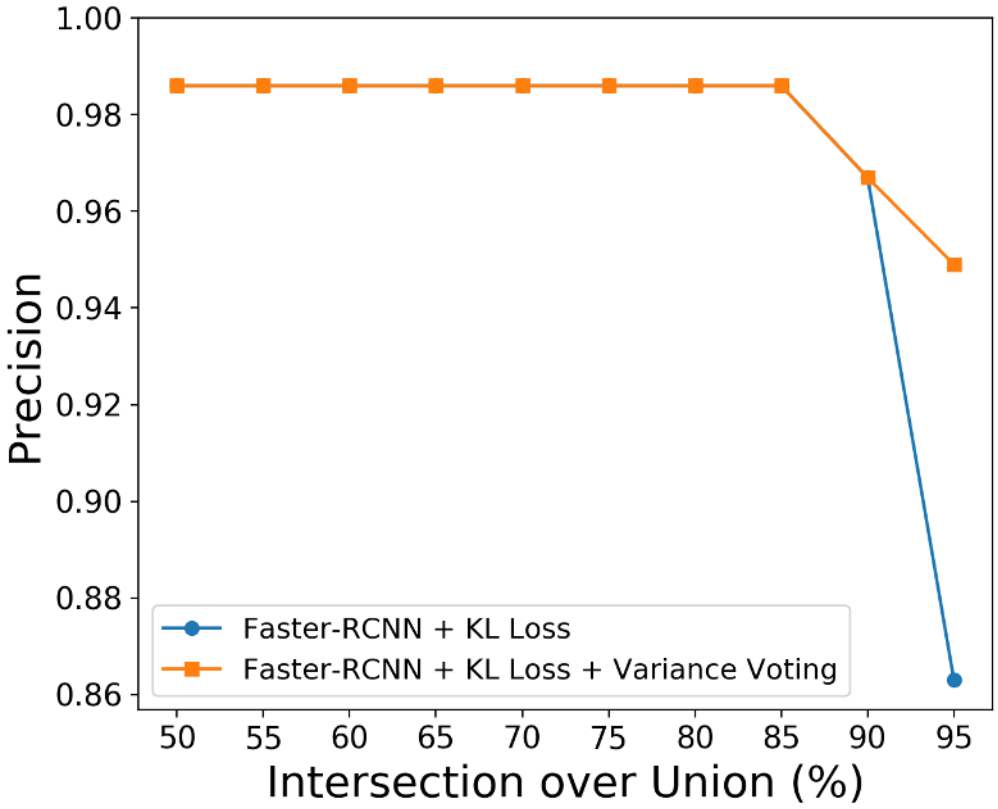


Figure 4.3: Precision values for ICDAR2017 dataset from IoU 50% to 95%

The Lytica dataset contains a wide array of tabular images for electronic components. It is diverse in it's tabular structure since it contains different electronic components from multiple manufactures. See Appendix, Figures A.1 A.2 A.3 A.4 show the example output of our Faster-RCNN model with KL Loss and variance voting on the Lytica dataset. It can be seen that the Dataset includes various electronic component figures and graphs that

Table 4.2: Table detection performance comparison on ICDAR2013 test set.

Models	**ICDAR2013 Test Set**		
	Recall	**Precision**	**F1**
Kavaisidis et al.[100]	0.981	0.975	0.978
DeepDeSRT[11]	0.962	0.974	0.968
Tran et al.[13]	0.964	0.952	0.958
Faster-RCNN [106]	0.981	0.974	0.977
Faster-RCNN + KL Loss	0.965	0.983	0.974
Faster-RCNN + KL Loss + Variance Voting	0.981	0.987	0.984

Table 4.3: Table detection comparison for models on Lytica test set.

Models	Lytica Test set			ICDAR2017 Test set		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**
RetinaNet w/ResNeXt-101 [106]	0.910	0.777	0.838	0.975	0.924	0.949
RetinaNet w/ResNeXt-50 [106]	0.944	0.765	0.845	0.994	0.903	0.946
Faster-RCNN [106]	0.953	0.911	0.932	0.969	0.939	0.953
Faster-RCNN + KL Loss	0.990	0.998	0.994	0.992	0.986	0.989
Faster-RCNN + KL Loss + Variance Voting	0.997	0.998	0.998	0.996	0.986	0.991

share similarities with tables. The tabular data obtained from public datasets are often of scientific origin. Therefore they generally stem from a similar structure.

The Lytica dataset contains 2323 hand-labeled documents. Along with the ICDAR-2017, ICDAR-2013 and Marmot datasets, we have a total of 5569 images to apply transfer learning. Testing is done individually on each dataset to evaluate and observe the performance.

For pre-processing we stretched the images by 250% vertically to increase the row size, according to Kara et al [20] this method can improve recall and precision by up to 20% . For post-processing we implement the aforementioned variance voting step.

In Table 4.2, we present the comparison of the recall, precision, values of various detection models in comparison to the introduced Faster-RCNN + KL Loss, and Faster-RCNN + KL Loss + Variance voting. They are all tested on the ICDAR 2013 dataset. It can

be seen that with the addition of the KL Loss function alone, the precision value rises to 0.983 which is 1.3% higher than the the results of Kavaisidis et al [100]. After adding variance voting to the model the precision improves further to 0.987. Overall the F1 score of the Faster-RCNN + KL Loss + variance voting is highest at 0.984. Table 4.3 shows the performance results on the Lytica and ICDAR2017 dataset. With the addition of the KL Loss function the F1 score improves by 6.2% from to 0.994 from 0.932. This is due to the 8.7% improvement in precision. After adding the variance voting step the F1 score improves to 0.998. For the ICDAR2017 test set the KL Loss alone improves by 4.6% from 0.953 to 0.989. The variance voting step improved the F1 score even further to 0.991. Figs 4.6 4.7 4.8 show the comparison of KL loss Faster-RCNN versus the model with the addition of variance voting on the ICDAR 2017, ICDAR 2013 and Lytica dataset. We can observe that at IoU of 95% the addition of variance voting improves the precision by 10.9% on ICDAR2013, 8.6% on ICDAR2017 and 6.7% on Lytica dataset.

4.5 Conclusion

In this chapter, we have proposed the integration of KL loss function and variance voting into deep learning based table detection i.e., (Faster-RCNN) to improve bounding box localization performance at high IoU thresholds of 95% as well as improve overall performance metrics. This can help optimize supply chain efficiency by automating document processing and removing additional sources of human error. By improving the bounding box regression with KL Loss we can see an improvement in precision of 5.3% on the ICDAR2017 dataset. The variance voting step has shown to improve the precision at 95% IoU by a drastic 10.9% on the ICDAR2013 dataset. Both the addition of KL Loss and variance voting has shown to improve the standard state-of-the-art object detector Faster-RCNN on ICDAR2013, ICDAR2017 and Lytica dataset.

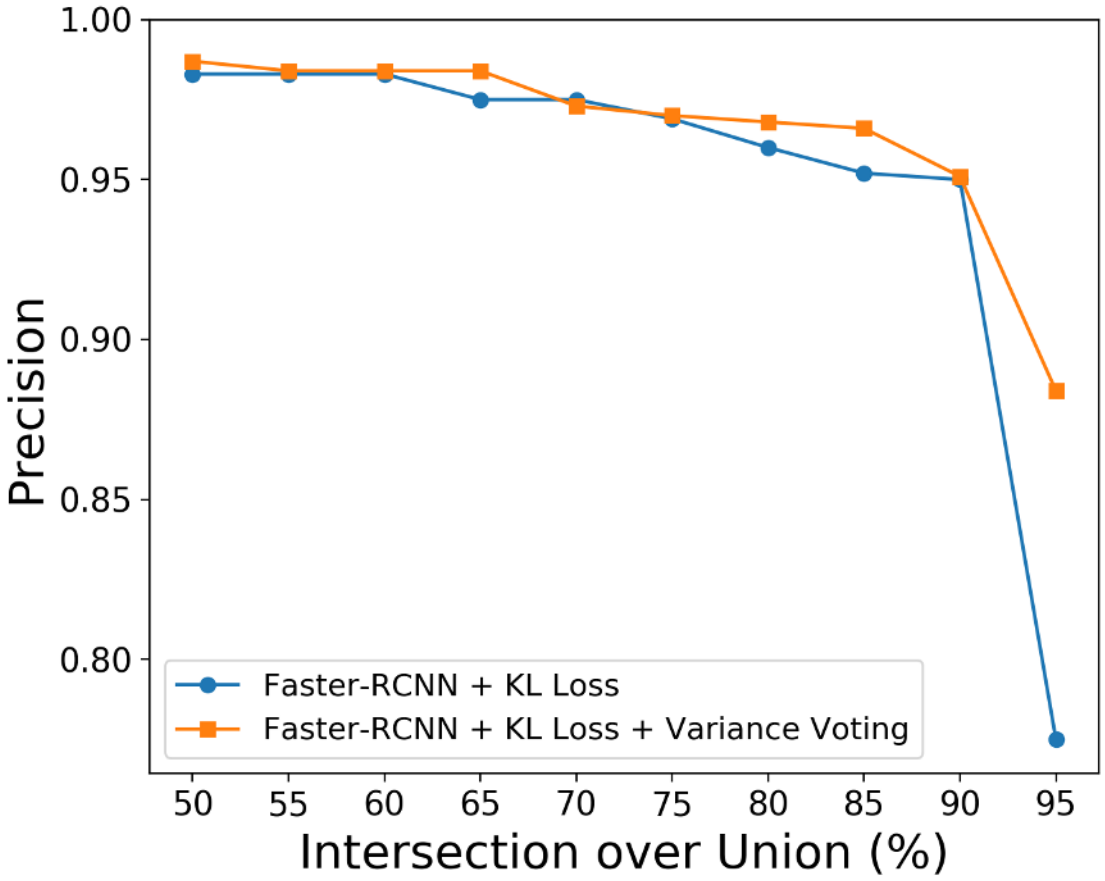


Figure 4.4: Precision values for ICDAR 2013 dataset from IoU 50% to 95%.

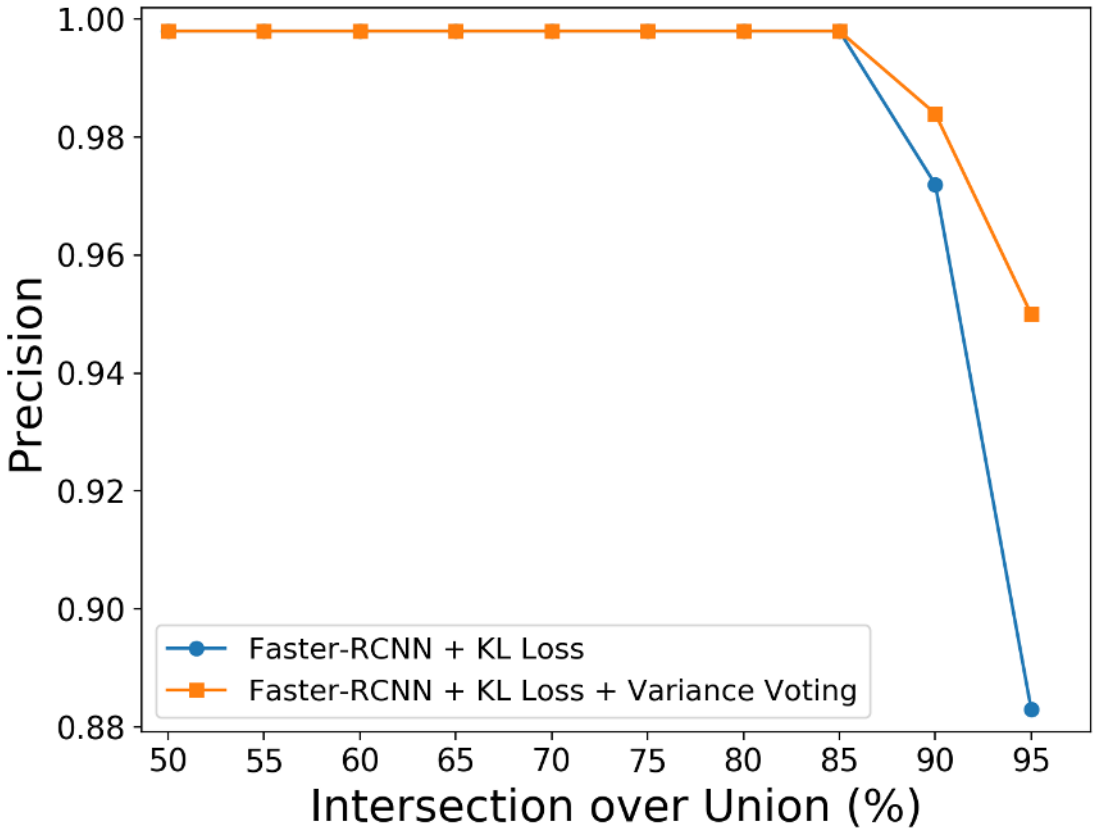


Figure 4.5: Precision values for Lytica dataset from IoU 50% to 95%.

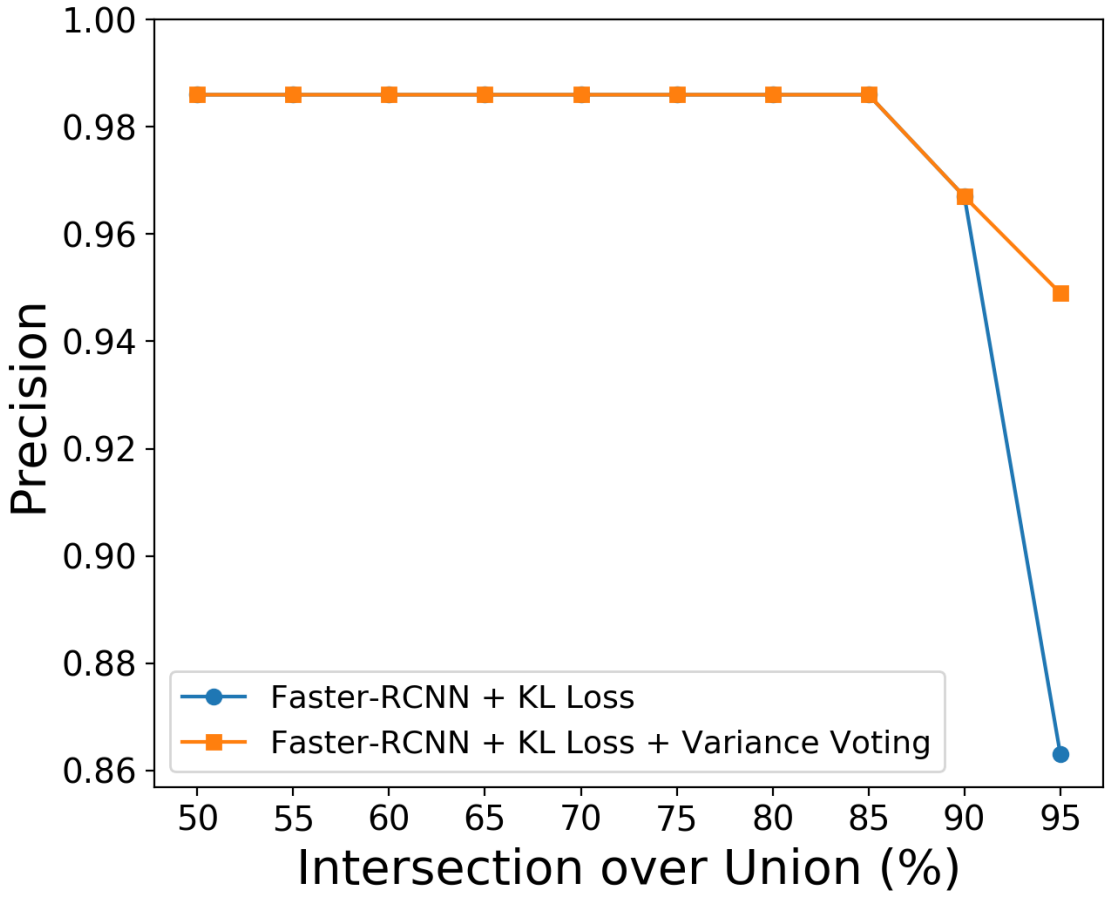


Figure 4.6: Precision values for ICDAR2017 dataset from IoU 50% to 95%

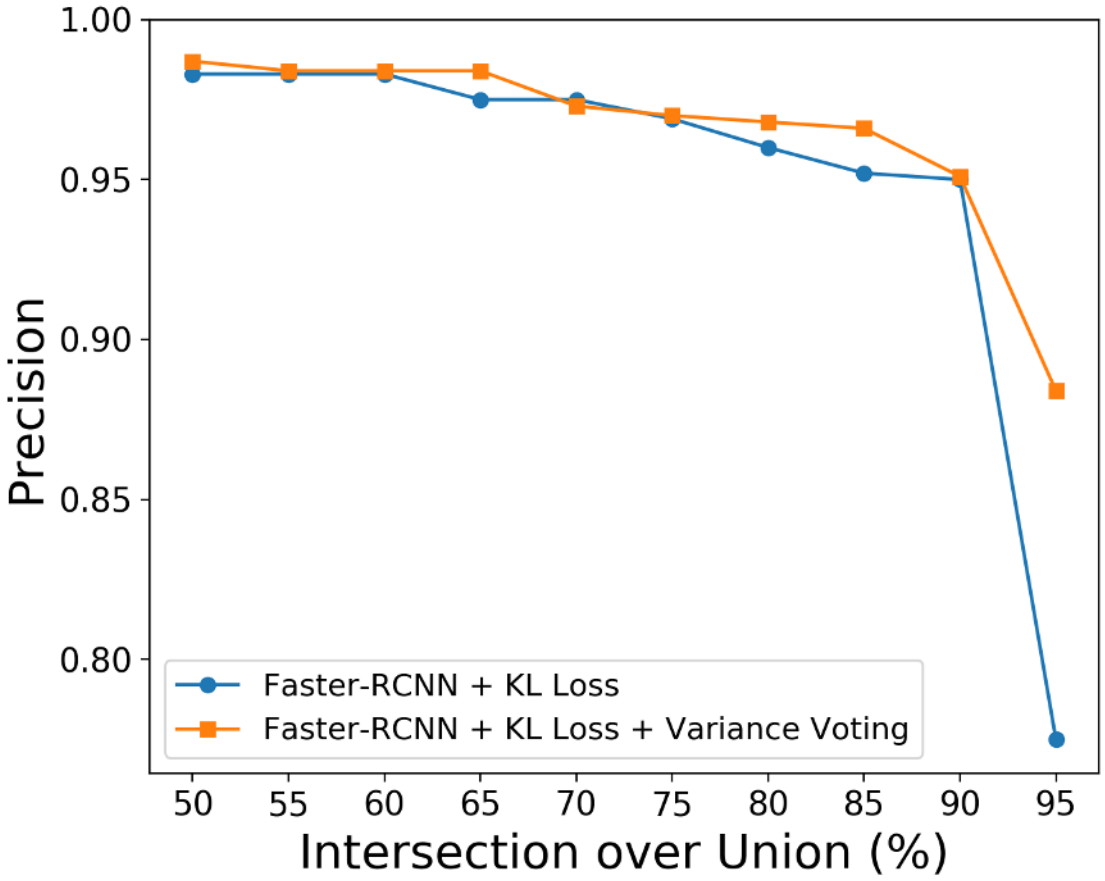


Figure 4.7: Precision values for ICDAR 2013 dataset from IoU 50% to 95%.

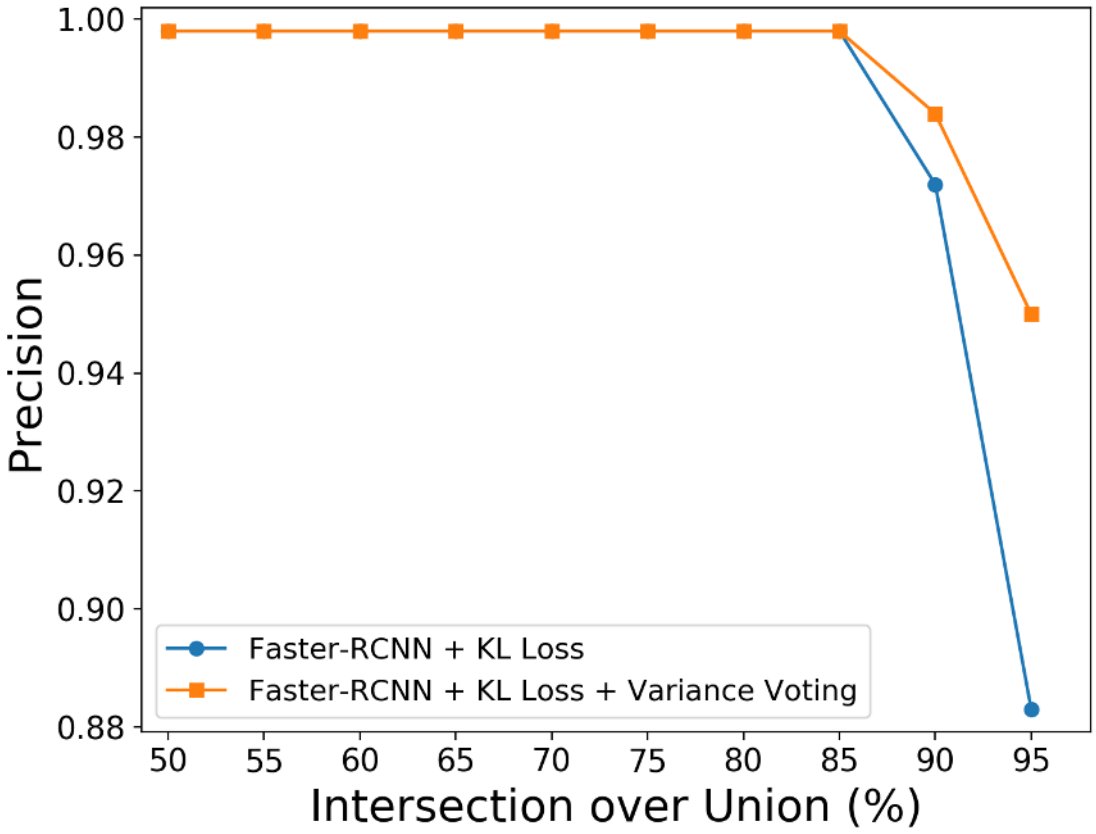


Figure 4.8: Precision values for Lytica dataset from IoU 50% to 95%.

Chapter 5

Table Structure Detection

There is an increasing demand for automated document processing techniques as the volume of electronic component documents increase. This is most prevalent in the supply chain optimization sector where vast amount of documents need to be processed and is time consuming and prone to error. Detection of tables and table structures serves as a crucial step to automating document processing. While table detection is a well investigated problem, tabular structure detection is more complex, and requires further improvement. To address this, this study proposes a deep learning model that focuses on high precision tabular cell structure detection. The proposed model creates a benchmark for the ICDAR2013 dataset cell structure with comparison to the previously state of the art table detection models and proposing alternative models. Our methodology approaches improving table structure detection through the detection of cells instead of row and columns for better generalization capabilities for hetergenous table structures. Our proposed model advances prior models by improving major parts of the detection pipeline compared to previous state of the art table structure detection models, mainly the two-stage detector, backbone, backbone architecture, and non-maximum-suppression (NMS). TabCellNet consists of Hybrid Task Cascade (HTC) with Combinational Backbone Network (CBNet), dual ResNeXt101 and Soft-NMS to achieve a precision of 89.2% and recall of 98.7% on the hand annotated ICDAR2013 cell structure dataset.

5.1 Methods

This section describes the methodology used for tabular cell detection. Figure 5.1 describes the proposed document flow pipeline for document processing. This paper is

focused on the table structure detection portion of the proposed pipeline.

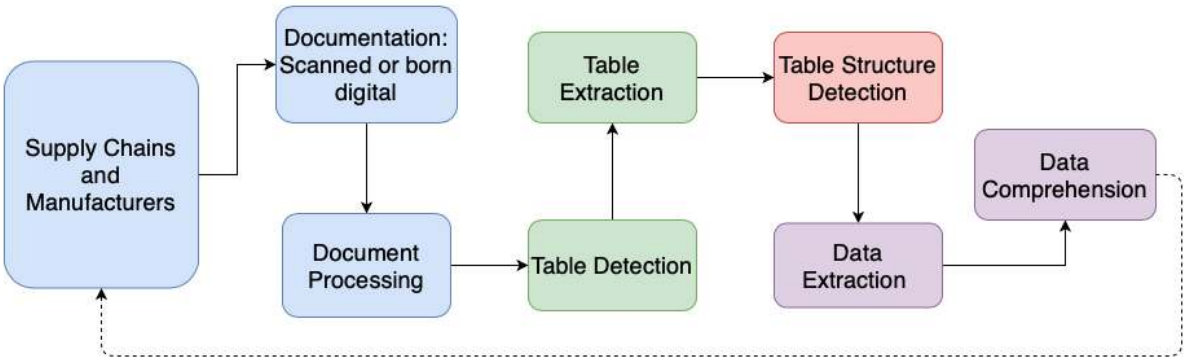


Figure 5.1: Document flow pipeline for document processing

5.1.1 Hybrid Task Cascade

Hybrid Task Cascade (HTC) was proposed by [31] to improve upon the classic and powerful Cascade-Mask-RCNN. Cascade-RCNN has a cascading structure that focuses on iterative refinement on predictions and adaptive training distributions. The cascading architecture would would well for object detection but not fully adaptive for instance segmentation. HTC integrates the cascading structure for instance segmentation by interlocking the detection and segmentation features to create a collaborative multi-stage process. The refinement process is benefited by the shared contextual information among detection, mask prediction and semantic segmentation tasks. There is also a direct path that incorporates information flow between mask branches.

Figure 5.2 depicts the architecture of the HTC model. Where the model takes in input from the Feature Map given by the CBNet backbone and RPN to predict the bounding box and mask regions. The bounding box proposals are receiving information from the previous layers, adopting from Cascade-Mask-RCNN, while the Masking proposals are improved with the benefit of using the updated bounding box proposals as well as communicating between each mask layer. The connected masking layer is where HTC improves upon Cascade-Mask-RCNN, the details on the information transfer between masking layers can be seen in Figure 5.3.

HTC is formulated by Equation 5.1, where x_t^{box} and x_t^{mask} respectively denote the the bounding box and mask features. Where x denotes the features obtained from the

backbone network. $P()$ represents the pooling operator. While the predicted mask and bounding boxes are denoted as m_t and r_t. The mask and bounding box heads are denoted by M_t and B_t. m_{t-1}^{-} is the features of the previous mask layer, while G_t represents the function that transforms the previous mask features into a 1x1 convolutional layer. The mask layers are connected by the 1x1 convolutional layer, this is shown in Figure 5.3 for further clarification. Each Masking layer consist of four 3x3 convolutional layers and this information is shared to the next masking layers.This shows that the predicted mask m_t is a composition of the mask features from a previous layer $G_t(m_{t-1}^{-})$ and mask features of the current layer x_t^{mask}. Whereas the current mask layer takes the updated bounding box proposals r_t and feature map x as the inputs.

$$\begin{aligned} x_t^{box} &= P(x, r_{t-1}), \quad r_t = B_t(x_t^{box}), \\ x_t^{mask} &= P(x, r_t), \quad m_t = M_t(x_t^{mask} + G_t(m_{t-1}^{-})), \end{aligned} \tag{5.1}$$

Feature Pyramid Networks

The Feature Pyramid Network (FPN) proposed by [109] has been widely used in various CNNs to improve feature map quality. It consists of a bottom up, top-down and lateral connections. This allows for building high level semantic feature maps at all scales. It accomplishes this by extracting features from high-resolution to low resolution then combines them from low-resolution to high resolution. The tabular structures consists of row, columns and cell that can be small in nature, after layers of convolution and pooling operations there is a reduced number of meaningful features. Therefore FPN can aid in tabular structure detection applications to produce more favourable results.

Region Proposal Network

The backbone network extracts features that creates the FPN, which are then fed into the Region Proposal Network (RPN). Backbone networks such as ResNet-50 extract features and then these features are fed into the RPN which then proposes candidate bounding boxes that may include an object of interest. Mask-RCNN ([22]) is almost identical to the Faster-RCNN but it employs additional methods to further improve the results. As presented in ([26]), Faster-RCNN improved vastly upon previous object detection architectures and gained wide adoption ([14, 11, **?**]).

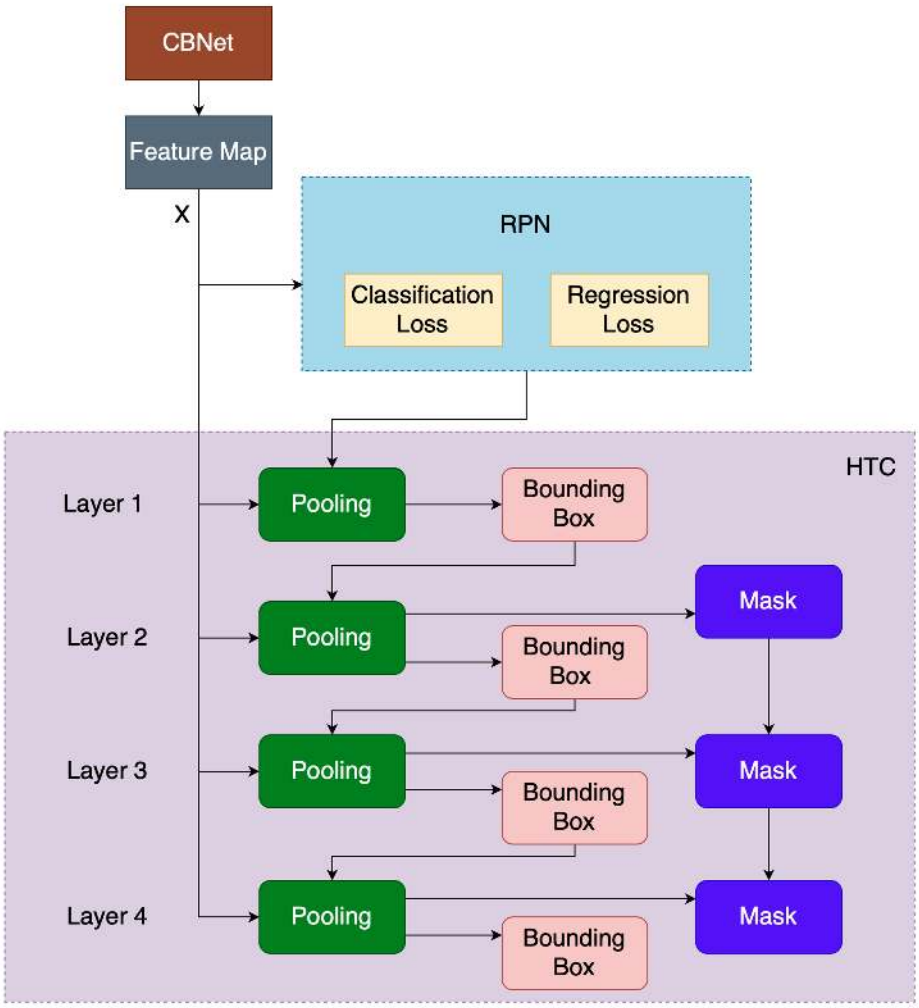


Figure 5.2: Hybrid Task Cascade Architecture

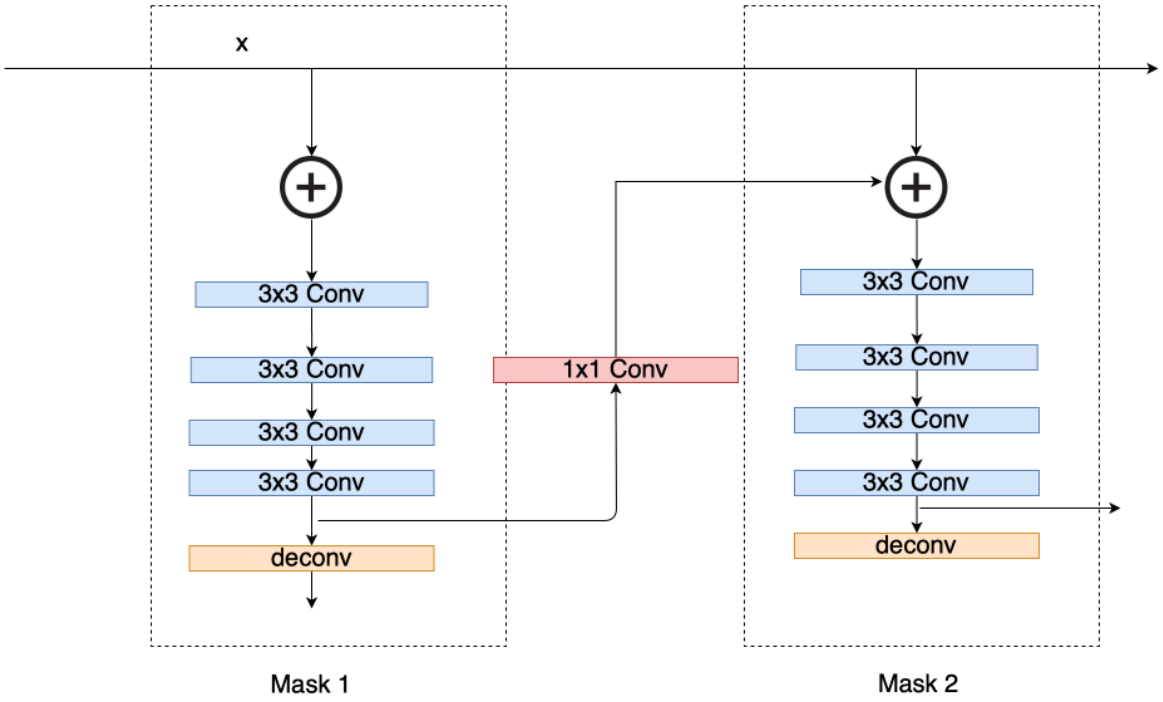


Figure 5.3: Masking Layer Connection

Batch Normalization

Batch Normalization (BN) was proposed by [110] is a training technique that uses a regularizer between the layers to solve the internal covariate shift issue. During training, the distribution between inputs of each layer changes and thus requires lower learning rates. BN normalizes the input of each mini-batch,

5.1.2 ResNeXt101

ResNeXt101 was proposed by [111] to improve on the robust VGG and ResNet architectures in a simplistic yet effective way. ResNext architecture performs a set of transformations of the same topology on a low dimensional setting and then aggregates each transformation by summation. This methodology is more effective and has less computational complexity; a 101 layer ResNeXt is able to outperform a 200 layer ResNet while only having half of its complexity ([112]). The archetecture of ResNeXt101 can be seen in figure 5.4.

5.1.3 Combinational Backbone Network

The backbone is of paramount importance within a detection pipeline, it is tasked with extracting features out of input images. Studies have proven that the larger and more complex a backbone is, the better the performance gains to a degree, like in the case of ResNet 50, ResNet 101, ResNet152 and ResNet200. The Combinational Backbone Network (CBNet) was proposed by [32] and in combination with Cascade-Mask-RCNN it was able to achieve an astonishing mAP of 53.3 on the COCO dataset, which was the state-of-the-art results at the time.

CBNet consists of a lead backbone and assistant backbones. The lead backbone provides the feature map for the detector, while the assistant backbones supports the lead backbone by feeding its output features to the input features of the next backbone.

The amount of identical backbones is denoted by K where $K \geq 2$. The lead backbone can be denoted as B_K, While the assistant backbones can be denoted as $B_1, B_2, ...B_{K-1}$. A backbone consists of several stages, at each stage there are several convolutional layers. A stage can be denoted as S and often times there are S=5 stages. The output of a stage is denoted as x^{s-1}. In a traditional single backbone architecture, the input of a stage is the output of the stage before it. However, in the CBNet structure, the input of a stage is the combination of the output of the stage before it (x^{s-1}) as well as the output of the previous

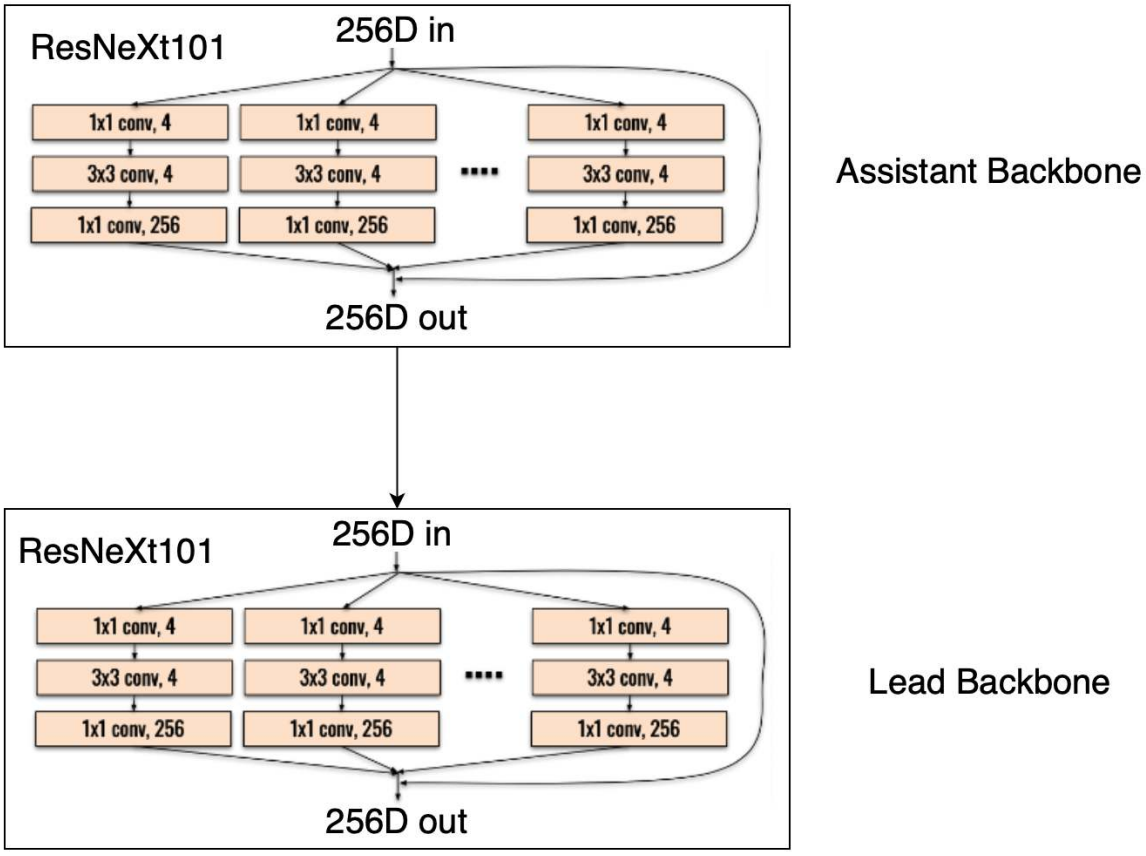


Figure 5.4: ResNeXt101 Architecture

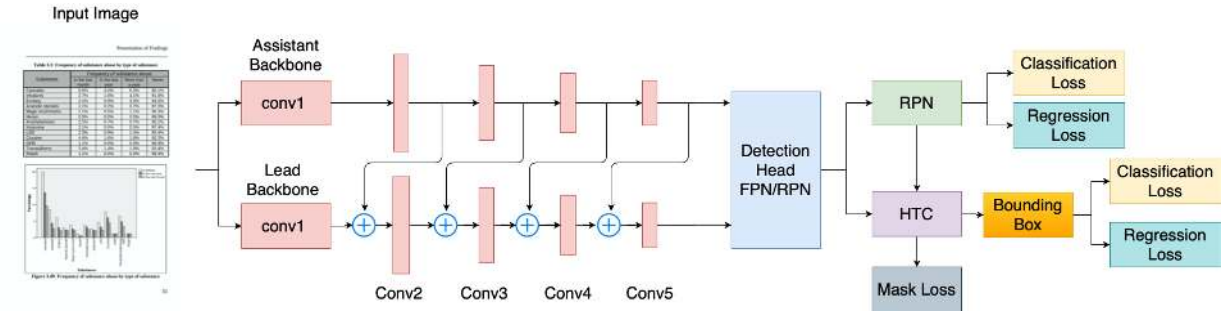


Figure 5.5: Dual Combinational Backbone Network

backbone at the same stage (x^l_{k-1}). This is shown in equation 5.2. We have visualized a dual combinational backbone network in Figure 5.5. The document image is taken as an input and two backbone networks are used in parallel to generate a feature map, the convolutional stages correspond to stages within ResNeXt101. This information is then fed to the RPN and used alongside to help with the bounding box and mask predictions in HTC.

$$x^s_k = F^s_k(x^{s-1}_k + g(x^s_{k-1})), s \geq 2 \tag{5.2}$$

Dual Backbone

[32] showed that as the number of backbones increase, the performance also increases, however, computational complexity and inference time also increases. This is mainly due to the composite architecture instead of due to the increase in network parameters. For the COCO dataset, it was shown that the performance converges at around triple backbone architectures, however, tabular structure detection is a less complex detection task compared to the 80 classes in the COCO dataset, therefore our results has shown that the dual backbone architecture performed best.

5.1.4 Soft Non Maximum Suppression

Non-maximum suppression (NMS) is a vital part of a modern object detection pipeline [96]. The proposed bounding boxes are given by the RPN proposals, the NMS step merges the proposed bounding boxes. Standard NMS eliminates bounding boxes with lower classification scores. While Soft-NMS improves over standard NMS by eliminating bounding boxes by not eliminating objects with lower classification scores, but instead scale the detection score as a function of IoU witht he ground truth, this method has shown to improve many baseline predictions, for example there is a gain of 1.7% mean average precision on Faster-RCNN with PASCALVOC 2007 dataset.

5.2 Dataset

The dataset used in this study is the cell-annotated version of the ICDAR 2013 ([8]) dataset. Currently, there is a shortage of annotated data for table structure analysis.

ICDAR 2013 has been a benchmark for table detection analysis. Therefore preprocessing stage of this research involves hand annotation of the ICDAR2013 dataset for tabular cell recognition. The goal of structure detection is to be able to pin point extract cell information for querying purposes. Often times rows and columns are used to obtain tabular structure and the combination of row and column coordinates can generate a matrix of cell coordinates. This method works well for homogeneous tables where there the table consists of x rows and y columns with no split or merged cells. Heterogeneous tables pose an increased complexity for the detection model and makes the structure detection model less generalized to fit the wide array of tabular structures within document processing. Thus, our solution is to mainly focus on cell detection instead of row and column detection. The dataset is cropped to only include the tables in the images, it is assumed that table detection performance is 100%. Table classification and position detection has achieved results as high as 100% F1 score on the ICDAR2013 dataset, therefore the proposed methodology focuses solely on structure detection. The document processing pipeline starts with the detection and extraction of the table, then the detection of the tabular structure. When a table is fully extracted and the position of each cell is known then creating a semantic understanding of the tabular contents can be done effectively using various OCR and NLP techniques.

The ICDAR2017 cell structure dataset consists of in total 1081 cropped table images, of which 865 are used for training and 216 for testing. The dataset is labeled oriented towards content focused annotations. This means that for tables without separator lines between the rows and columns we do not infer a dimension for the cell width and height, instead we only annotate the area around the content. We removed the EmptyCell class, therefore the dataset only have the Cell class. Through experimentation the Empty Cell class overall did not have an apparent positive impact to the model, removing it does not reduce the performance of our model, however it does reduce confusion on inference results. This is most likely due to the indeterminate nature of empty cells in tables separated by white spacing.

5.3 Training Details

To formulate a benchmark for ICDAR2013 cell structure dataset we experimented with deep learning models that performed well for the ICDAR2013 table dataset. We also experimented with model combinations that havent been used before but have shown promising results on other public datasets such as the COCO dataset.

All experiments were done on the Mist GPU cluster utilizing 4 Tesla V100 GPUs with

32GB of VRAM each. The mmdetection toolbox that was based on pytorch was used to implement the models. Table 5.1 contains more detailed attributes for our network. To identify which models would perform well on cell structure detection, potential models were trained using the same backbone network, ResNeXt101. Faster-RCNN has been a benchmark detector for several related works [14, 66, 65], Mask-RCNN was shown to be a better candidate than Faster-RCNN based on [20, 64, 23]. The addition of a masking layer that uses Fully Convolutional Network (FCN) on top of Faster-RCNN presents more spatial features. Furthermore, Cascade-Mask-RCNN was proposed by [18] to achieve even higher results on the ICDAR2013 table detection dataset. These models all showed promising performance on the ICDAR2013 table dataset therefore we wanted to use them to create a benchmark for the ICDAR2013 cell structure dataset. Deformable convolutions networks (DCN) was used by [65] for table detection, this was tested but it was not able to outperform our final selected model. For each model the learning rate was tuned for optimal performance, the learning curve for each model was also observed to ensure that the model is fully trained. An average training accuracy of 99.5% was observed on each model. A learning curve for a HTC CBNet Dual ResNeXt101 model can be seen in figure 5.6. State-of-the-art backbones were then studied as well as different backbone structures. ResNet101 and HRNet have both previously been used for tabular detection, therefore we compared them with ResNeXt101, as the proposed CBNet double ResNeXt101 and triple ResNeXt101. For bounding box regression, we tried Balanced L1 loss ([113]), IoU Loss ([114]), Bounded IoU Loss ([115])and General IoU Loss ([116]), however, none showed improvements over the smooth L1 loss for this task. Soft-NMS has shown improvements on many detectors and over many applications therefore it is implemented on the best performing model seeking to improve the performance results further.

Table 5.1: Model Settings

GPU	Learning Rate	Optimizer	momentum	weight decay	Image Scale	batch size
Tesla V100 32GB RAM x4	0.01	SGD	0.9	0.0001	1333x2000	50

5.3.1 Evaluation Metrics

The models are tested in regards to precision, recall and F1 score under the IoU range of 50% to 95% as well as Mean average precision (mAP), mean average recall (mAR) and mean average F1 Score (mAF1). F1 score is a combination of Recall and Precision values, it gives us a gauge of the models performance. We favour recall over precision due to the fact that for data extraction we want to make sure that the detected cells are actual valid

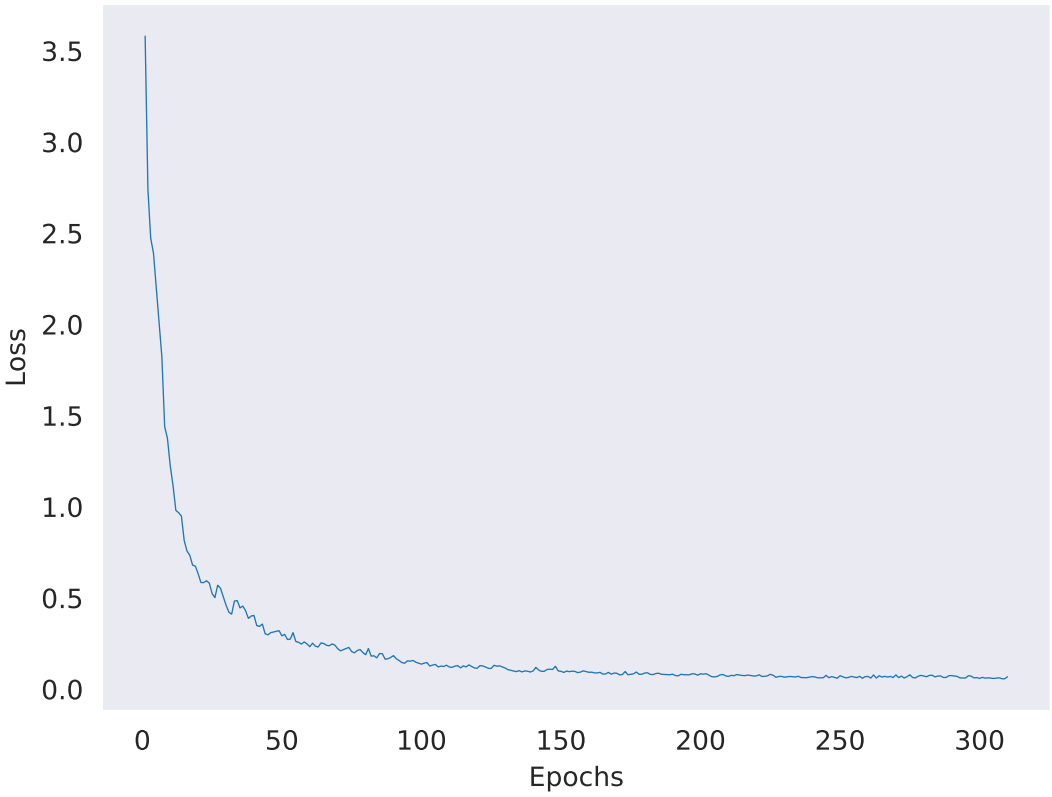


Figure 5.6: Learning Curve for HTC CBNet Dual ResNeXt101

detection, this shows how complete the tabular cells within a table are detected. Whereas precision is a metric to evaluate the validity of the detected results.

$$F1(Precision, Recall) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.3}$$

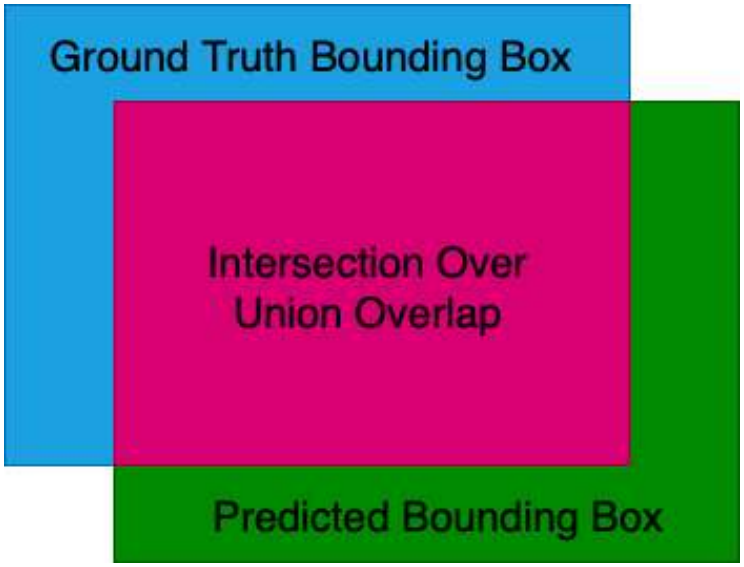


Figure 5.7: IoU Illustration

For predicted bounding box (P) and ground truth bounding box (Gt), IoU can be described as;

$$IoU(P, Gt) = \frac{P \cap Gt}{P \cup Gt} \tag{5.4}$$

IoU is the area of intersection between the predicted bounding box and ground truth bounding box over the total spanned area of the two bounding boxes. An illustration of IoU for predicted and ground truth bounding boxes can be seen in Figure 5.7. Utilizing different degrees of IoU will allow us to gauge the localization performance of our detector. This is important because often times the focus is the classification capabilities of the model rather than the localization performance. In order for deep learning-based document analysis to reach a industrial level, the localization performance of the detector must be studied to ensure that crucial information is not left out during the data extraction phase.

Table 5.2: Structure Detection with different models

Methods with ResNeXt101 Backbone	IoU	50%:95%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
Mask-RCNN	Precision	0.672	0.866	0.859	0.857	0.845	0.826	0.780	0.714	0.605	0.309	0.059
	Recall	0.827	0.938	0.926	0.924	920	0.916	0.901	0.878	0.842	0.705	0.320
	F1-Score	0.741	0.901	0.891	0.889	0.881	0.869	0.836	0.788	0.704	0.430	0.100
Cascade-RCNN	Precision	0.702	0.875	0.859	0.853	0.844	0.826	0.793	0.740	0.641	0.473	0.116
	Recall	0.831	0.916	0.898	0.896	0.893	0.885	0.877	0.858	0.834	0.778	0.588
	F1-Score	0.761	0.895	0.878	0.874	0.868	0.855	0.833	0.795	0.725	**0.588**	0.186
Cascade-Mask-RCNN	Precision	0.697	0.861	0.855	0.855	0.849	0.833	0.809	0.729	0.623	0.435	0.121
	Recall	0.875	0.963	0.952	0.946	0.944	0.937	0.930	0.913	0.888	0.800	0.477
	F1-Score	0.776	0.909	0.901	0.898	**0.894**	**0.882**	**0.865**	0.811	0.732	0.564	**0.193**
Hybrid Task Cascade	Precision	0.705	0.891	0.878	0.865	0.846	0.823	0.793	0.757	0.674	0.425	0.098
	Recall	0.876	0.978	0.963	0.954	0.946	0.934	0.922	0.909	0.876	0.805	0.473
	F1-Score	**0.781**	**0.932**	**0.919**	**0.907**	0.893	0.875	0.852	**0.826**	**0.762**	0.556	0.162

Mask-RCNN has been previously proven to be a promising candidate for tabular detection [23, 14, 64]. Cascade RCNN and Cascade-Mask-RCNN has shown to improve upon Mask-RCNN. Hybrid Task Cascade was proposed by [31] to further improve Cascade-Mask-RCNN.

Table 5.3: Cascade-Mask-RCNN with different Backbones

Backbones with Cascade-Mask-RCNN	IoU	50%:95%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
HRNet	Precision	0.655	0.821	0.802	0.797	0.787	0.777	0.755	0.707	0.593	0.371	0.140
	Recall	0.802	0.886	0.866	0.862	0.860	0.855	0.843	0.825	0.0.799	0.731	0.493
	F1-Score	0.721	0.852	0.833	0.828	0.822	0.814	0.797	0.761	0.681	0.492	**0.218**
Resnet101	Precision	0.705	0.875	0.862	0.856	0.849	0.832	0.803	0.754	0.663	0.469	0.087
	Recall	0.828	0.920	0.899	0.894	892	0.888	0.882	0.863	0.838	0.739	0.465
	F1-Score	0.762	0.897	0.880	0.875	0.870	0.859	0.841	0.800	0.740	0.574	0.147
ResNeXt101	Precision	0.697	0.861	0.855	0.855	0.849	0.833	0.809	0.729	0.623	0.435	0.121
	Recall	0.875	0.963	0.952	0.946	0.944	0.937	0.930	0.913	0.888	0.800	0.477
	F1-Score	0.776	0.909	**0.901**	**0.898**	**0.894**	0.882	**0.865**	0.811	0.732	0.564	0.193
CBNet-Double ResNeXt101	Precision	0.720	0.887	0.872	0.865	0.859	0.848	0.824	0.775	0.695	0.464	0.111
	Recall	0.859	0.955	0.930	0.927	0.925	0.921	0.912	0.892	0.866	0.782	0.480
	F1-Score	**0.783**	**0.920**	0.900	0.895	0.891	**0.883**	**0.865**	**0.829**	**0.771**	0.582	0.180
CBNet-Triple ResNeXt101	Precision	0.708	0.876	0.859	0.852	0.846	0.834	0.799	0.738	0.678	0.484	0.114
	Recall	0.861	0.953	0.932	0.931	0.928	0.923	0.914	0.896	0.869	0.816	0.448
	F1-Score	0.777	0.913	0.894	0.890	0.885	0.876	0.853	0.810	0.762	**0.608**	0.182

Cascade-Mask-RCNN has been shown to result in the highest F1 score at certain IoUs based on Table 5.2. Cascade-Mask-RCNN was tested with different backbones. Cascade-Mask-RCNN with HRNet backbone was proposed by [18], and achieved 1.0 F1 score under the ICDAR2013 for table detection. Therefore, this study aims to test Cascade-Mask-RCNN on the ICDAR2013 structure dataset.

Table 5.4: HTC with different backbones

Backbones with Hybrid Task Cascade	IoU	50%:95%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
ResNeXt101	Precision	0.697	0.861	0.855	0.855	0.849	0.833	0.809	0.729	0.623	0.435	0.121
	Recall	0.875	0.963	0.952	0.946	0.944	0.937	0.930	0.913	0.888	0.800	0.477
	F1-Score	0.776	0.909	0.901	0.898	0.894	0.882	0.865	0.811	0.732	0.564	0.193
CBNet-Double ResNeXt101	Precision	0.742	0.893	0.881	0.879	0.869	0.853	0.820	0.772	0.710	0.563	0.180
	Recall	0.887	0.971	0.953	0.948	0.946	0.942	0.939	0.924	0.897	0.826	0.524
	F1-Score	0.808	0.930	0.916	**0.912**	0.906	0.895	0.875	0.841	0.793	0.670	0.268
CBNet-Triple ResNeXt101	Precision	0.726	0.896	0.876	0.872	0.862	0.844	0.816	0.749	0.695	0.551	0.099
	Recall	0.872	0.982	0.957	0.953	0.950	0.946	0.932	0.911	0.875	0.802	0.412
	F1-Score	0.792	**0.937**	0.915	0.911	0.904	0.892	0.870	0.822	0.775	0.653	0.160
ResNeXt101 - Soft-NMS	Precision	0.734	0.899	0.884	0.878	0.872	0.856	0.826	0.763	0.700	0.529	0.133
	Recall	0.897	0.975	0.950	0.947	0.946	0.943	0.934	0.917	0.890	0.833	0.455
	F1-Score	0.800	0.935	0.916	0.911	0.907	0.897	0.877	0.833	0.784	0.647	0.206
CBNet-Double ResNeXt101 - Soft-NMS	Precision	0.750	0.892	0.883	0.877	0.874	0.859	0.830	0.787	0.728	0.581	0.189
	Recall	0.899	0.975	0.955	0.949	0.948	0.946	0.945	0.933	0.912	0.853	0.574
	F1-Score	**0.818**	0.932	**0.918**	**0.912**	**0.909**	**0.900**	**0.884**	**0.854**	**0.810**	**0.691**	**0.284**
CBNet-Triple ResNeXt101- Soft NMS	Precision	0.734	0.892	0.876	0.872	0.867	0.855	0.829	0.770	0.710	0.561	0.108
	Recall	0.891	0.987	0.958	0.954	0.952	0.950	0.946	0.930	0.897	0.829	0.507
	F1-Score	0.805	**0.937**	0.915	0.911	0.908	**0.900**	**0.884**	0.842	0.793	0.669	0.178

HTC with CBNet Double ResNeXt101 and soft-NMS performed the best for almost every category compared to all models tested. The dual ResNeXt101 performs the triple ResNeXt101 due to the model reaching a point of diminishing return on features obtained. The additions of Soft-NMS outperforms standard NMS due to its ability to scale confidence scores instead of eliminating bounding boxes.

5.4 Results

In Table 5.2, The structure detection results are presented and compared with various state-of-the-art detectors with ResNeXt101 Backbone. The state of the art does not contain research that have trained and tested the ICDAR2013 and ICDAR2017 structure datasets, hence this work will serve as a this research will serve as a baseline for research in this field. Mask-RCNN shows promise for table detection as well as tabular structure detection based on [23, 14, 64, 20].

Table 5.2 shows that Hybrid Task Cascade achieves the highest mAF1 score of 0.781 and continues to achieve the highest F1 score at various IoUs. Cascade-Mask-RCNN also shows promise in terms of performance at lower IoUs of 65%, 70%, 75% and 95%. Often times benchmarks are evaluated at 50% IoU or the mean average of 50% to 95% IoU. This is due to most deep learning-based models are more concerned with classification of objects in the image than the localization performance, therefore 50% IoU is enough to suffice that an object is positively detected in the image. However, in order to ensure data is extracted properly without error the localization performance needs to be high enough so that data is not cropped out. The mean average value that is often used by COCO detection [117] challenges go up till 95% IoU as the highest threshold. Therefore we evaluate the model individually at each of the threshold values in the mean average calculation to gauge an overall performance of the system. At 50% IoU the highest precision and recall are both taken by HTC at 0.891 and 0.978 respectively.

Both Cascade-Mask-RCNN and HTC are experimented with further due to their performance capabilities at different IoUs. They perform better due to them both incorporating benefits from the cascading architecture and masking layers of Cascade-RCNN and Mask-RCNN.

Cascade-Mask-RCNN is tested with different backbones to evaluate the best combination between backbone and detector. This includes the use of ResNet50 [28], ResNet101 [28], ResNeXt101 [29] and HRNet [118]. ResNet50 have been used by us and other researchers for table detection [69, 14], ResNet101 was used by Siddiqui et al. [**?**] and ResNeXt101 was used by Kara et al. [20]. The combination of Cascade-Mask-RCNN and High Resolution Net (HRNet) was proposed by [18] to achieve 1.0 F1 score on the ICDAR2013 table detection dataset, which is currently the state-of-the-art results for that dataset. Therefore its performance for the ICDAR2013 structure dataset should also be included for the baseline formulation. The results are shown in Table 5.3. ResNeXt101 led to an improvement over Resnet101, with a mAF1 of 0.776 compared to 0.762. Deploying double and triple ResNeXt101 in a CBNet showed even further improvements, with double

ResNeXt101 performing the best at a mAF1 of 0.783, and a $F1_{50}$ of 0.920. This is due to architectural improvements in major aspects of the model developed on top of proven state-of-the-art detectors. Faster-RCNN has been proven many times to be a strong detector for document analysis, Kara et al. [1, 20] showed that Mask-RCNN that is built upon Faster-RCNN is a promising candidate for tabular structure detection. Therefore HTC improves upon the detector by sharing information through masking layers as well as utilizing the Cascading architecture from Cascade-RCNN [81]. While CBNet improves the detection by combining adjacent backbones for a more detailed representational feature map. Soft-NMS preserves and scales detection scores to retain all elements so non is eliminated.

Table 5.4 shows the performance of HTC detector with various backbone structures with combinations of standard NMS and Soft-NMS. Once again, the addition of arranging double or triple backbones in a CBNet structure improves the the model's performance in precision, recall and F1 score due to the improved feature map that is fed to the RPN and detector. The additional backbones allows for the extraction of more representational basic features compared to a singular backbone. Single backbones are specifically designed for image classification therefore they are not optimized for object detection [32]. Aligning with the results of Cascade-Mask-RCNN, double ResNeXt101 outperforms triple ResNeXt101. For HTC an increase of 1.6% mAF1 score for double ResNeXt101, and for Cascade-Mask-RCNN a 0.6% incresae. When compared to the single ResNeXt101, Double ResNeXt101 shows a mAP increase of 3.5%, mAR increase of 1.2% and mAF1 increase of 3.2%. The Soft-NMS post processing operation further improves the performance of the HTC model with double ResNeXt101, with Soft-NMS added, it has the highest F1 score in every IoU category except IoU of 50%, this is because at IoU of 50% the triple ResNeXt101 backbone provides more basic features to the feature map, however, these additional features do not contribute more at a higher IoU threshold, on the contrary it over saturates the feature map due to tables having simple background and foreground features, and this is probably why dual ResNeXt101 over performs triple ResNeXt101. While also acheiving the highest mAP of 75% and mAR of 89.9%. Therefore our proposed model consists of the HTC model with CBNet dual ResNeXt101 and soft-nms. The inference results of this model on the ICDAR2013 cell structure dataset can be observed in the Appendix.

All models perform relatively well up until an IoU of 85%. At higher IoUs of 90% and 95% a decrease in precision of 14.7% and 39.2% is observed respectively for the best performing HTC model. As IoU thresholds go up, the performance metrics in recall and precision go down due to increased localization performance threshold. The cells are not deemed detected if they cannot satisfy this threshold. The cells within a table can be very small in size, with a few pixels in height and width, therefore a slight shift in bounding

box proposals would result in an invalid detection at high thresholds. Also, there can be hundreds or thousands of cells within a single table, the sheer quantity as well as the small size nature of the cells adds to the complexity of the challenge. By having a recall of 91.2% at 85% IoU represents that 91.2% of all cells were detected with at least 85% area overlap, with the nature of the IoU calculation an 85% IoU is actually higher than 85% total area of the ground truth bounding box detected. This shows that the best performing HTC model performs well at even high IoUs of 85%. The recall is generally much higher than the precision, we favour recall over precision due to it being a value that allows us to gauge the percentage of cells detected in the table. Recall is often higher in tabular cell detection due to the fact that the model outputs more bounding box predictions that are necessary. Therefore most of the correct cells will be detected, however, out of the bounding box proposals a higher quantity would be incorrect cells, leading to a lower precision value. Hetergenous tables with split cells, merged cells as well as tables separated by white spacing without seperator lines complicates the problem, for example, the detector might detect more than one cell for merged cells since there is a change in format at that location, leading to a lower precision value. The recall at 50% IoU is 8.3% higher and the recall at 95% IoU it is 38.5% higher, meaning that there are often much fewer false negatives than false positives. Which means their are less missed cells in the detection model. We want to ensure that as many cells are detected to not lose any information, this is more important than if the cells are classified wrong, since compensation in the semantic modeling part of the data extraction pipeline can phase out the wrongly detected cells. If a detection identifies that a cell is present, then there is high probability that the cell is actually present.

Furthermore, augmentation is applied to the image inputs, specifically resizing, padding and applying horizontal and vertical flipping. The threshold for bounding box proposals were increased to 10000 for the RPN and detector. We found that certain images may contain over 500 cells, therefore more candidate boxes are required, this effectivly improved the performance of the model during testing, without needing to retrain the model. NMS for the RPN was reduced from 0.7 to 0.5. This objects are small objects, therefore more tolerance during the NMS operation during RPN allows for more bounding box proposals for the detector. This operation also allowed for increases in performance without needing to retrain the model.

The proposed model that consists of HTC and dual ResNeXt101 backbone in CBNet architecture performed the best on the ICDAR2013 dataset therefore we utilize it again for performance evaluation on the ICDAR2017 dataset.

The results for our proposed model on the ICDAR2017 dataset can be seen in Table 5.5. A comparison has been done for a model with image augmentations and without.

Overall augmentations improved the average F1 score by 1% and improved the F1 score at 50% IoU by 2.3%. The model with image augmentations shows consistent improved performance up until the high IoU thresholds of 85%, 90% and 95%. The results follow the same trend of achieving a higher Recall value than precision. The results for ICDAR2017 infernce outputs can be found in the Appendix.

Table 5.5: Proposed Model on ICDAR2017 dataset

Backbones with Hybrid Task Cascade		**IoU** 50%:95%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
No Augmentation	Precision	0.590	0.894	0.870	0.843	0.809	0.753	0.681	0.555	0.358	0.127	0.010
	Recall	0.713	0.954	0.893	0.865	0.838	0.789	0.754	0.703	0.627	0.501	0.206
	F1-Score	0.646	0.923	0.881	0.854	0.823	0.771	0.716	0.620	**0.456**	**0.203**	**0.019**
Augmentation	Precision	0.600	0.915	0.895	0.870	0.829	0.772	0.687	0.559	0.348	0.117	0.008
	Recall	0.747	0.980	0.942	0.906	0.876	0.840	0.772	0.726	0.650	0.514	0.264
	F1-Score	**0.665**	**0.946**	**0.918**	**0.888**	**0.852**	**0.805**	**0.727**	**0.632**	0.453	0.191	0.016

The proposed HTC with dual ResNeXt101 in CBNet archetecture on the ICDAR2017 cell structure dataset, a comparison of before and after image augmentations.

5.5 Conclusion

The flow of information through supply chains are increasing more than ever, in which automated processing of valuable data is crucial in order to further optimize supply chains. Important data within documents is often stored in a tabular structure. Detection of tables is the first step while detection of table structures is the second step to effective information retrieval. Locating data points in heterogeneous tables is more effective if table cells are extracted versus compounding the intersections of row and columns.

This study proposed a deep learning-based model that consists of a novel combination of HTC and CBNet double ResNeXt101 with Soft-NMS for cell structure detection within tables. Models that performed well for table detection were tested at each IoU from 50% to 95%. Various combinations were proposed, and the best performing by far was selected. cell structure detection is significantly more complex than table detection. The proposed model builds upon the models that performed well on table detection. The proposed model includes the use of HTC that improves upon Cascade-Mask-RCNN as well as Mask-RCNN and Faster-RCNN, the use of ResNeXt101 that significantly improves over ResNet101, the use of CBNet architecture that improves upon a singular backbone and the use of Soft-NMS that improves over standard NMS. The proposed model achieves a precision of 89.2%

and a recall of 97.5%.

Chapter 6

Conclusion

Advancements in deep learning Convolutional Neural Networks for object detection methodologies can be leveraged for improving document analysis models for data processing, especially within the sector of optimizing supply chain efficiency. Extracting data from data-sheets is a laborious task that is prone to error. Millions of documents are processed each day, an automated method to extract content is highly desired. These documents can be from industry supply chains, government, research papers etc. The source of these documents range from different industries and countries without a unified format. The vast variation in format makes it difficult for pre-defined heuristic methods to be generalized enough for industrial use. Data is often stored in structured tabular format. Focusing on understanding tables is a key area in document analysis that will enable efficient data extraction. This thesis is dedicated to analyzing tables within documents. There are two crucial steps for analyzing tables within documents, first is determining the presence of table and its coordinates, second is determining the anatomy of the classified table. Table detection serves as a key point in a document analysis pipeline, the detected results will be given to the next part of the pipeline which would be information retrieval and semantic understanding of tabular contents.

Regarding table detection, various rule-based methodologies have been developed over the years, as well as methods that rely on document metadata. However, in recent years, deep learning based CNNs prove to be the most efficient and robust methodology. Table detection requires classification and localization of the table boundary. While table structure detection requires anatomical detection of tables. This can be achieved through various methods, the most studied is detection of row and columns, however, this thesis focus on detection of individual cells, which enables pin point data extraction for the semantic extraction portion of the document analysis pipeline. This thesis tackled each

detection task individually in a methodical way. The proposed pipeline is first detecting the boundary of tables within a document, then solely focus on detected table region. We simplify the tabular structure detection task by eliminating the noise that the other elements in a document presents. The tabular structure model input would only be confirmed detected tables. In order for this method to be successful, the table detector must be highly capable at localization the table boundaries. Therefore we improved the tested state-of-the-art Faster-RCNN detector for better localization performance. This was achieved by utilizing a KL Loss function as the regression loss and adding a voting methodology during the NMS step. Our tabular detection model achieved state-of-the-art results in regards to precision and recall and we showed the vast improvements at extremely high IoUs of 95%. IoU thresholds are a reflection of localization performance needed before the detection is deemed positive. Therefore at 95% IoU, the ground truth bounding box and predicted bounding box has an overlap area of 95%. Previous table detection research only only evaluate their models at 50% IoU or an average across multiple IoUs, localization is highly important for table detection due to the possibility of data loss and its affects on the latter stages of data extraction. The outputs of the table detection model is then fed into our table structure detection model. Table structure detection is much more complex due to the heterogeneous nature of table formats. Tables without guiding lines are especially difficult to analyze due to there being no guidelines on exactly how width and high each tabular cell should be. Tables can also contain thousands of cells with each cell being small in size, small objects are difficult to detect in the field of computer vision due to their lack of resolution. We propose an ensemble model that leverages multiple state-of-the-art networks. HTC was used in conjunction to dual ResNeXt101 backbones in CBNet architecture. this model performed the best on the ICDAR2013 cell structure dataset that we hand-labeled. It was compared extensively to other state-of-the-art detection models in the domain of table detection as well as table structure detection. Our proposed method provides a baseline performance metric results on the ICDAR2013 cell structure dataset. Overall, this thesis presents a high precision table and table structure detection model that serves as an intermediate process within document analysis for data extraction.

6.1 Future Directions

Table detection is well studied and consists of a wide array of well annotated and diverse datasets. Currently detection performance is reaching high degrees of accuracy, such as the 99.8% precision that our proposed model achieves on the ICDAR2017 dataset.

However, the performance of table structure detection still has room for improvement,

in particular, the lack of quality datasets for cell structure detection is still a major issue for research within tabular document extraction. Our contributions of ICDAR2013 and ICDAR2017 will contribute to the advancement the table structure detection domain, however, it is still not enough to achieve the same amount of success as table detection since table structure detection is inherently more difficult. The field of cell structure detection is still in it's early stages, it is even more difficult than row and column detection due to many aspects, they can be abundant in volume and they can be extremely small objects which are challenging for CNNs. For example, our private dataset contains electronic component datasheets from a diverse spread of manufacturers, with each using vastly varied formats for their tabular content. Heterogeneous tables that has split cells and merged cells are abundant, the cells are also often separated by white spacing with no clear dictation on the dimensions for each cell. Better comparisons can be made between proposed research methodologies if the performance metrics of a model over different IoU thresholds is presented. The application of document analysis requires high localization accuracy to ensure that data is not lost during the early stages of the extraction pipeline.

We are still exploring other state-of-the-art deep learning strategies to further enhance document structure detection performance, such as training models that do not require annotations as well as attention based models. If we can train models that do not require pre-annotated data then it would be a great breakthrough in table cell structure detection. Annotating table cells is far more time consuming that annotating rows and columns, this is also one of the reasons why annotations in this field are scarce.

Future directions are given as below;

- Tabular cell structure detection poses two difficulties, first there is the challenge of large quantity of detections within a single image; there can be over 1000 cells within a table. The study of Deep learning object detectors for large quantity of object detections within a singular image have not been studied and optimized. Second there is the challenge of detecting tiny objects, depending on the resolution, cells can be a few pixels in height and width. The combinations of large quantity of detections and small objects in tabular cell detection adds complexity to the problem. Object detectors that focus specifically on this area should be researched to advance tabular cell structure detection.

- Even with our annotated ICDAR2013 and ICDAR2017 cell structure datasets, quality labeled data within this field is still scarce. Further annotation on the Marmot dataset, UNLV dataset [10] and private datasets can be used to train an even more generalized detector.

- Extensions to thesis also include next part of the data extraction pipeline is semantic understanding of the extracted table contents. The model should built upon this study and utilize NLP techniques to relate and understand the nomenclature and attributes of the data contained within the tables. This is currently being investigated by other researchers.

- The integration between the table extraction part that was presented in this thesis and the semantic modeling portion is an integral part to a complete end-to-end data extraction pipeline that needs to researched. For homogeneous tables it would be easy to understand the key values of the table and relate to its attributes, however, for heterogeneous tables that have inconsistent formatting, the problem becomes vastly complex.

References

[1] E. Kara, M. Traquair, M. Simsek, B. Kantarci, and S. Khan, "Holistic design for deep learning-based discovery of tabular structures in datasheet images,," *Elsevier Engineering Applications of Artificial Intelligence*, 2020 (Accepted).

[2] 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.

[3] S. K. Rao and R. Prasad, "Impact of 5g technologies on industry 4.0," *Wireless personal communications*, vol. 100, no. 1, pp. 145–159, 2018.

[4] B. Tjahjono, C. Esplugues, E. Ares, and G. Pelaez, "What does industry 4.0 mean to supply chain?," *Procedia Manufacturing*, vol. 13, pp. 1175 – 1182, 2017. Manufacturing Engineering Society International Conference 2017, MESIC 2017, 28-30 June 2017, Vigo (Pontevedra), Spain.

[5] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[6] M. J. Meixell and V. B. Gargeya, "Global supply chain design: A literature review and critique," *Transportation Research Part E: Logistics and Transportation Review*, vol. 41, no. 6, pp. 531–550, 2005.

[7] L. Liang, F. Yang, W. D. Cook, and J. Zhu, "Dea models for supply chain efficiency evaluation," *Annals of Operations Research*, vol. 145, no. 1, pp. 35–49, 2006.

[8] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *Intl. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 1484–1493, IEEE, 2013.

[9] 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. 01, pp. 1417–1422, 2017.

[10] S. V. Rice, F. R. Jenkins, and T. Nartker, "The fourth annual test of ocr accuracy," 03 2012.

[11] 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 *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 1162–1167, Nov 2017.

[12] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual seperators and tabular structures," in *2011 International Conference on Document Analysis and Recognition*, pp. 779–783, IEEE, 2011.

[13] 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, 2015.

[14] 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.

[15] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *Object recognition supported by user interaction for service robots*, vol. 3, pp. 236–240, IEEE, 2002.

[16] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for pdf documents based on convolutional neural networks," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pp. 287–292, IEEE, 2016.

[17] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," 09 2017.

[18] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure, "Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 572–573, 2020.

[19] S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: a literature survey," in *Document Recognition and Retrieval X*, vol. 5010, pp. 197–208, International Society for Optics and Photonics, 2003.

[20] E. Kara, "End-to-end tabular information extraction in datasheets with deep learning," 2019.

[21] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[23] 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.

[24] C. Eggert, S. Brehm, A. Winschel, D. Zecha, and R. Lienhart, "A closer look: Small object detection in faster r-cnn," in *2017 IEEE international conference on multimedia and expo (ICME)*, pp. 421–426, IEEE, 2017.

[25] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 951–959, 2017.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, (Cambridge, MA, USA), pp. 91–99, MIT Press, 2015.

[27] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2888–2897, 2019.

[28] 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.

[29] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5987–5995, 2017.

[30] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," 2019.

[31] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Hybrid task cascade for instance segmentation," 2019.

[32] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling, "Cbnet: A novel composite backbone network architecture for object detection," 2019.

[33] S. S. Paliwal, D. Vishwanath, R. Rahul, M. Sharma, and L. Vig, "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 128–133, IEEE, 2019.

[34] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," pp. 236–240, 2003.

[35] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, "Automatic Table Detection in Document Images," 2005.

[36] E. Oro and M. Ruffolo, "Trex: An approach for recognizing and extracting tables from pdf documents," in *2009 10th International Conference on Document Analysis and Recognition*, pp. 906–910, IEEE, 2009.

[37] G. Nagy, "Preliminary investigation of techniques for automated reading of unformatted text," *Communications of the ACM*, vol. 11, pp. 480–487, jul 1968.

[38] 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.

[39] 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.

[40] A. K. Jain and B. Yu, "Document representation and its application to page decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[41] 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.

[42] X. Lin and Y. Xiong, "Detection and analysis of table of contents based on content association," *International Journal on Document Analysis and Recognition*, 2006.

[43] A. Pivk, P. Cimiano, Y. Sure, M. Gams, V. Rajkovič, and R. Studer, "Transforming arbitrary tables into logical form with TARTAR," *Data and Knowledge Engineering*, vol. 60, no. 3, pp. 567–595, 2007.

[44] 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.

[45] 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, '.

[46] Y. Wang and J. Hu, "A machine learning based approach for table detection on the web," p. 242, 2004.

[47] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, pp. 81–106, Mar. 1986.

[48] M. A. Hearst, "Support vector machines," *IEEE Intelligent Systems*, vol. 13, pp. 18–28, July 1998.

[49] 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.

[50] 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.

[51] E. I. George and R. E. McCulloch, "Variable selection via gibbs sampling," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 881–889, 1993.

[52] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on optimization*, vol. 16, no. 1, pp. 170–192, 2005.

[53] 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*, pp. 443–450, 1999.

[54] 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.

[55] 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.

[56] M. Fan and D. S. Kim, "Detecting Table Region in PDF Documents Using Distant Supervision," 2015.

[57] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and Ł. Bolikowski, "CERMINE: Automatic extraction of structured metadata from scientific literature," *International Journal on Document Analysis and Recognition*, vol. 18, no. 4, pp. 317–335, 2015.

[58] 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.

[59] 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.

[60] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 640–651, Apr 2017.

[61] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3339–3348, 2018.

[62] Z. He and L. Zhang, "Multi-adversarial faster-rcnn for unrestricted object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6668–6677, 2019.

[63] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified faster r-cnn," *Applied Sciences*, vol. 8, no. 5, p. 813, 2018.

[64] 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*, vol. 90, p. 103551, 2020.

[65] 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.

[66] N. Sun, Y. Zhu, and X. Hu, "Faster r-cnn based table detection combining corner locating," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1314–1319, 2019.

[67] 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, 2018.

[68] S. A. Siddiqui, I. A. Fateh, S. T. R. Rizvi, A. Dengel, and S. Ahmed, "Deeptabstr: Deep learning based table structure recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1403–1409, 2019.

[69] J. Jiang, M. Simsek, B. Kantarci, and S. Khan, "High precision deep learning based tabular detection," in *IEEE Symposium on Computers and Communications (ISCC)*, (Rennes, France), 2020.

[70] S. S. Paliwal, V. D, R. Rahul, M. Sharma, and L. Vig, "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 128–133, 2019.

[71] 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.

[72] 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.

[73] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, p. 1137–1149, Jun 2017.

[74] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016.

[75] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[76] 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.

[77] 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.

[78] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[79] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[80] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," 2017.

[81] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," 2019.

[82] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7363–7372, 2019.

[83] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 821–830, 2019.

[84] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic r-cnn: Towards high quality object detection via dynamic training," *arXiv preprint arXiv:2004.06002*, 2020.

[85] 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.

[86] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-cnn for small object detection," in *Asian conference on computer vision*, pp. 214–230, Springer, 2016.

[87] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[89] 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.

[90] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection.," in *CVPR*, vol. 1, p. 4, 2017.

[91] 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.

[92] 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.

[93] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[94] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1222–1230, 2017.

[95] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 850–855, IEEE, 2006.

[96] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms – improving object detection with one line of code," 2017.

[97] C. Sun and P. Vallotton, "Fast linear feature detection using multiple directional non-maximum suppression," *Journal of Microscopy*, vol. 234, no. 2, pp. 147–157, 2009.

[98] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Asian conference on computer vision*, pp. 290–306, Springer, 2014.

[99] J. H. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," *CoRR*, vol. abs/1705.02950, 2017.

[100] 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," *arXiv preprint arXiv:1804.06236*, 2018.

[101] S. Gidaris and N. Komodakis, "Locnet: Improving localization accuracy for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[102] 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.

[103] 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.

[104] 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.

[105] 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.

[106] 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)*, pp. 1–6, June 2019.

[107] T. Akiba, S. Suzuki, and K. Fukuda, "Extremely large minibatch SGD: training resnet-50 on imagenet in 15 minutes," *CoRR*, vol. abs/1711.04325, 2017.

[108] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron." https://github.com/facebookresearch/detectron, 2018.

[109] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016.

[110] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[111] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2016.

[112] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, pp. 630–645, Springer, 2016.

[113] S. Wu and X. Li, "Iou-balanced loss functions for single-stage object detection," 2019.

[114] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, "Iou loss for 2d/3d object detection," 2019.

[115] L. Tychsen-Smith and L. Petersson, "Improving object localization with fitness nms and bounded iou loss," 2017.

[116] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," 2019.

[117] 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," *Lecture Notes in Computer Science*, p. 740–755, 2014.

[118] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," 2020.

Appendix A

Appendix

A.1 Table Detection Outputs

The table detection outputs for our proposed Faster-RCNN with KL loss and Variance voting are shown. The four examples show that our model has near perfect localization capabilities for the tables of our complex Lytica dataset. Figure A.1 shows that only the table in the middle is detected, whereas the other items within the image can easily be misclassifed as tables due to them having a structured layout. Figure A.2 and A.4 shows that all three tables are detected with high localization accuracy, the detector understood that three tables are present, instead of grouping the three tables into one singluar table detection. Figure A.3 shows the correct detection of tables among graphical figures that have grid lines.

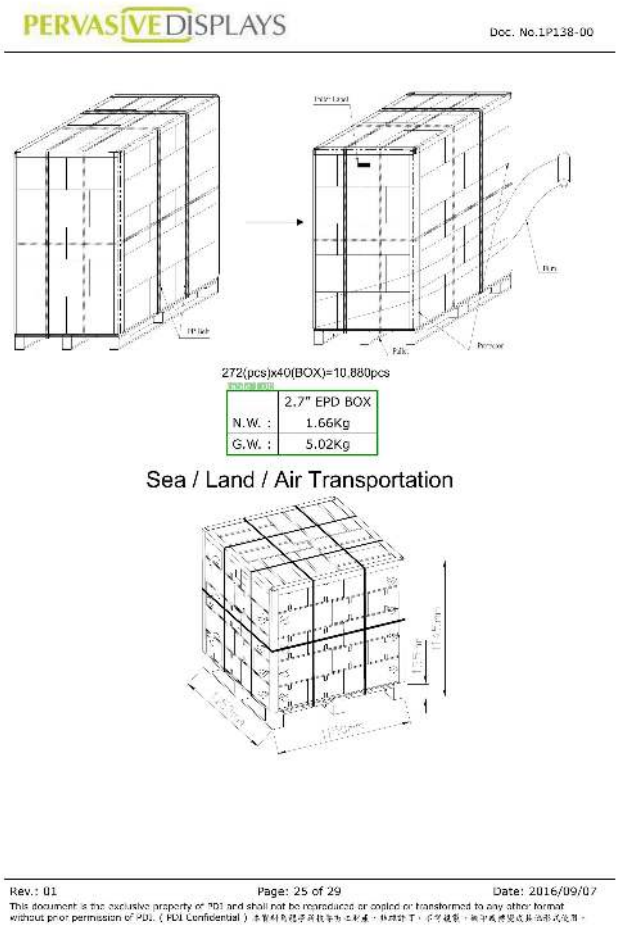
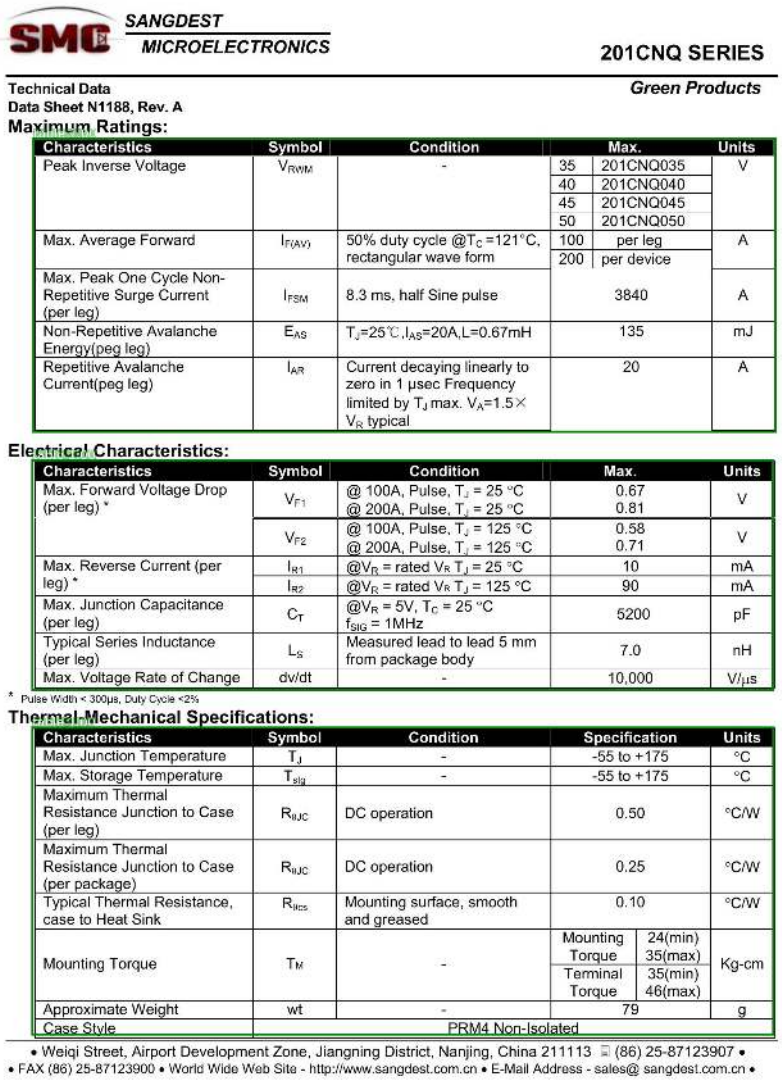


Figure A.1: Example output for KL Loss + Variance voting on Lytica Dataset



SANGDEST MICROELECTRONICS

201CNQ SERIES

Technical Data
Data Sheet N1188, Rev. A

Green Products

Maximum Ratings:

Characteristics	Symbol	Condition	Max.		Units
Peak Inverse Voltage	V_{RWM}	-	35	201CNQ035	V
			40	201CNQ040	
			45	201CNQ045	
			50	201CNQ050	
Max. Average Forward	$I_{F(AV)}$	50% duty cycle @T_C =121°C, rectangular wave form	100	per leg	A
			200	per device	
Max. Peak One Cycle Non-Repetitive Surge Current (per leg)	I_{FSM}	8.3 ms, half Sine pulse	3840		A
Non-Repetitive Avalanche Energy(peg leg)	E_{AS}	T_J=25℃,I_{AS}=20A,L=0.67mH	135		mJ
Repetitive Avalanche Current(peg leg)	I_{AR}	Current decaying linearly to zero in 1 μsec Frequency limited by T_J max. V_A=1.5× V_R typical	20		A

Electrical Characteristics:

Characteristics	Symbol	Condition	Max.	Units
Max. Forward Voltage Drop (per leg) *	V_{F1}	@ 100A, Pulse, T_J = 25 °C @ 200A, Pulse, T_J = 25 °C	0.67 0.81	V
	V_{F2}	@ 100A, Pulse, T_J = 125 °C @ 200A, Pulse, T_J = 125 °C	0.58 0.71	V
Max. Reverse Current (per leg) *	I_{R1}	@V_R = rated V_R T_J = 25 °C	10	mA
	I_{R2}	@V_R = rated V_R T_J = 125 °C	90	mA
Max. Junction Capacitance (per leg)	C_T	@V_R = 5V, T_C = 25 °C f_{SIG} = 1MHz	5200	pF
Typical Series Inductance (per leg)	L_S	Measured lead to lead 5 mm from package body	7.0	nH
Max. Voltage Rate of Change	dv/dt	-	10,000	V/μs

* Pulse Width < 300μs, Duty Cycle <2%

Thermal-Mechanical Specifications:

Characteristics	Symbol	Condition	Specification		Units
Max. Junction Temperature	T_J	-	-55 to +175		°C
Max. Storage Temperature	T_{stg}	-	-55 to +175		°C
Maximum Thermal Resistance Junction to Case (per leg)	$R_{\theta JC}$	DC operation	0.50		°C/W
Maximum Thermal Resistance Junction to Case (per package)	$R_{\theta JC}$	DC operation	0.25		°C/W
Typical Thermal Resistance, case to Heat Sink	$R_{\theta cs}$	Mounting surface, smooth and greased	0.10		°C/W
Mounting Torque	T_M	-	Mounting Torque	24(min) 35(max)	Kg-cm
			Terminal Torque	35(min) 46(max)	
Approximate Weight	wt	-	79		g
Case Style	PRM4 Non-Isolated				

• Weiqi Street, Airport Development Zone, Jiangning District, Nanjing, China 211113 ☎ (86) 25-87123907 •
• FAX (86) 25-87123900 • World Wide Web Site - http://www.sangdest.com.cn • E-Mail Address - sales@ sangdest.com.cn •

Figure A.2: Example output for KL Loss + Variance voting on Lytica Dataset

Table 1.00

THERMAL CHARACTERISTICS (T_A = 25 °C unless otherwise noted)									
PARAMETER	SYMBOL	KBU8A	KBU8B	KBU8D	KBU8G	KBU8J	KBU8K	KBU8M	UNIT
Typical thermal resistance	$R_{\theta JA}$ (1)	18							°C/W
	$R_{\theta JC}$ (2)	3.0							

Notes

(1) Units mounted in free air, no heatsink, PCB at 0.375" (9.5 mm) lead length with 0.5" x 0.5" (12 mm x 12 mm) copper pads

(2) Units mounted on a 3.0" x 3.0" x 0.11" thick (7.5 cm x 7.5 cm x 0.3 cm) aluminum plate heatsink

Table 1.01

ORDERING INFORMATION (Example)				
PREFERRED P/N	UNIT WEIGHT (g)	PREFERRED PACKAGE CODE	BASE QUANTITY	DELIVERY MODE
KBU8J-E4/51	8.0	51	250	Anti-static PVC tray

RATINGS AND CHARACTERISTICS CURVES (T_A = 25 °C unless otherwise noted)

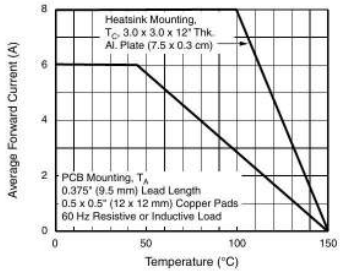


Fig. 1 - Derating Curve Output Rectified Current

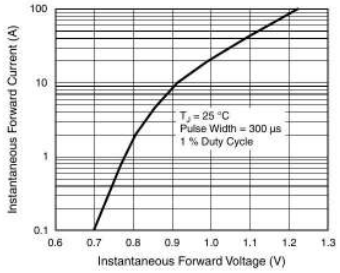


Fig. 3 - Typical Instantaneous Forward Characteristics Per Diode

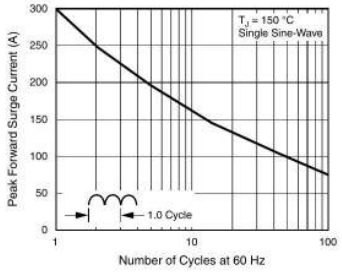


Fig. 2 - Maximum Non-Repetitive Peak Forward Surge Current

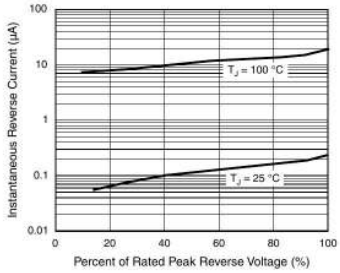


Fig. 4 - Typical Reverse Leakage Characteristics Per Diode

Figure A.3: Example output for KL Loss + Variance voting on Lytica Dataset

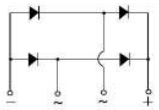
VISHAY

KBU8A, KBU8B, KBU8D, KBU8G, KBU8J, KBU8K, KBU8M

www.vishay.com

Vishay General Semiconductor

Single-Phase Bridge Rectifier



Case Style KBU

FEATURES

- UL recognition, file number E54214
- Ideal for printed circuit boards
- High surge current capability
- High case dielectric strength of 1500 V_{RMS}
- Solder dip 275 °C max. 10 s, per JESD 22-B106
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

RoHS COMPLIANT

PRIMARY CHARACTERISTICS	
Package	KBU
$I_{F(AV)}$	8 A
V_{RRM}	50 V, 100 V, 200 V, 400 V, 600 V, 800 V, 1000 V
I_{FSM}	300 A
I_R	10 µA
V_F at I_F = 8 A	1.0 V
T_J max.	150 °C
Diode variations	In-Line

TYPICAL APPLICATIONS

General purpose use in AC/DC bridge full wave rectification for monitor, TV, printer, SMPS, adapter, audio equipment, and home appliances applications.

MECHANICAL DATA

Case: KBU

Molding compound meets UL 94 V-0 flammability rating Base P/N-E4 - RoHS-compliant, commercial grade

Terminals: Silver plated leads, solderable per J-STD-002 and JESD22-B102

Polarity: As marked on body

Mounting Torque: 10 cm-kg (8.8 inches-lbs) max.

Recommended Torque: 5.7 cm-kg (5 inches-lbs)

MAXIMUM RATINGS (T_A = 25 °C unless otherwise noted)

PARAMETER		SYMBOL	KBU8A	KBU8B	KBU8D	KBU8G	KBU8J	KBU8K	KBU8M	UNIT
Maximum repetitive peak reverse voltage		V_{RRM}	50	100	200	400	600	800	1000	V
Maximum RMS voltage		V_{RMS}	35	70	140	280	420	560	700	V
Maximum DC blocking voltage		V_{DC}	50	100	200	400	600	800	1000	V
Maximum average forward rectified output current at	T_C = 100 °C (1)(3)	$I_{F(AV)}$	8.0							A
	T_A = 40 °C (2)		6.0							
Peak forward surge current single sine-wave superimposed on rated load		I_{FSM}	300							A
Operating junction and storage temperature range		T_J, T_{STG}	- 50 to + 150							°C

Notes

(1) Recommended mounting position is to bolt down on heatsink with silicone thermal compound for maximum heat transfer with #6 screw

(2) Units mounted in free air, no heatsink, PCB at 0.375" (9.5 mm) lead length with 0.5" x 0.5" (12 mm x 12 mm) copper pads

(3) Units mounted on a 3.0" x 3.0" x 0.11" thick (7.5 cm x 7.5 cm x 0.3 cm) aluminum plate heatsink

ELECTRICAL CHARACTERISTICS (T_A = 25 °C unless otherwise noted)

PARAMETER	TEST CONDITIONS	SYMBOL	KBU8A	KBU8B	KBU8D	KBU8G	KBU8J	KBU8K	KBU8M	UNIT
Maximum instantaneous forward drop per diode	I_F = 8.0 A	V_F	1.0							V
Maximum DC reverse current at rated DC blocking voltage per diode	T_A = 25 °C	I_R	10							µA
	T_A = 125 °C		1.0							mA

Figure A.4: Example output for KL Loss + Variance voting on Lytica Dataset

A.2 Table Structure Detection Outputs

A.2.1 ICDAR2013 Results

This section presents a selection of inference results on ICDAR2013 dataset to further support the numerical results in 5.

	2000	2002	2004
General upper secondary education	15 455	13 951	12 068
Competence-based qualifications	35 190	44 307	60 152
Polytechnics	20 527	20 922	22 083

Figure A.5: A homogeneous table that contains guiding lines.

Fig. A.5 presents a homogeneous table that contains guiding lines. TabCellNet performs well at classifying and localizing cells on homogeneous tables that contains guiding lines.

	1997	1995*	1992*	1990**	1980**	By leading retailers (1993/4)
UK	42.3	29	25	31	22	Sainsbury 55; Tesco 46; Safeway 38; Asda 32.
Belgium/Lux	24.9	22	16			
Netherlands	19.1	16	16			
France	18.2	16	16	20	11	Monoprix 28; Casino 25; Intermarche 23; Carrefour 22; Auchan 19; Leclerc 10
Denmark		13				
Germany	12.6	11	6	24	15	Aldi 90; Metro 33; Tengelmann 18.
Spain		10	8	9	2	Eroski 24; Pryca 20; Alcampo 15
Portugal		9				
Austria		9				
Finland		8	8			
Sweden		8	8			
Italy		6	4			
Greece		3				

Figure A.6: A homogeneous table with guiding lines and contains empty cells.

Fig. A.6 illustrates a homogeneous table with guiding lines and contains empty cells. There are two classes listed within the ICDAR2013 cell structure dataset, cells and empty cells. The detection of cells is often much better than the detection of empty cells, this is due to the small quantity of empty cells within the dataset as well as the sparseness of empty cell distribution.

Fig. A.7 illustrates a heterogeneous table without guiding lines. TabCellNet is able to consistently detect and localize all of the cells within this image. This type of table is occurs often within the ICDAR2013 dataset. Once again the detection of empty cells is poor this is because in a scenario that does not contain guiding lines the empty spaces can be easily mistaken for empty cells.

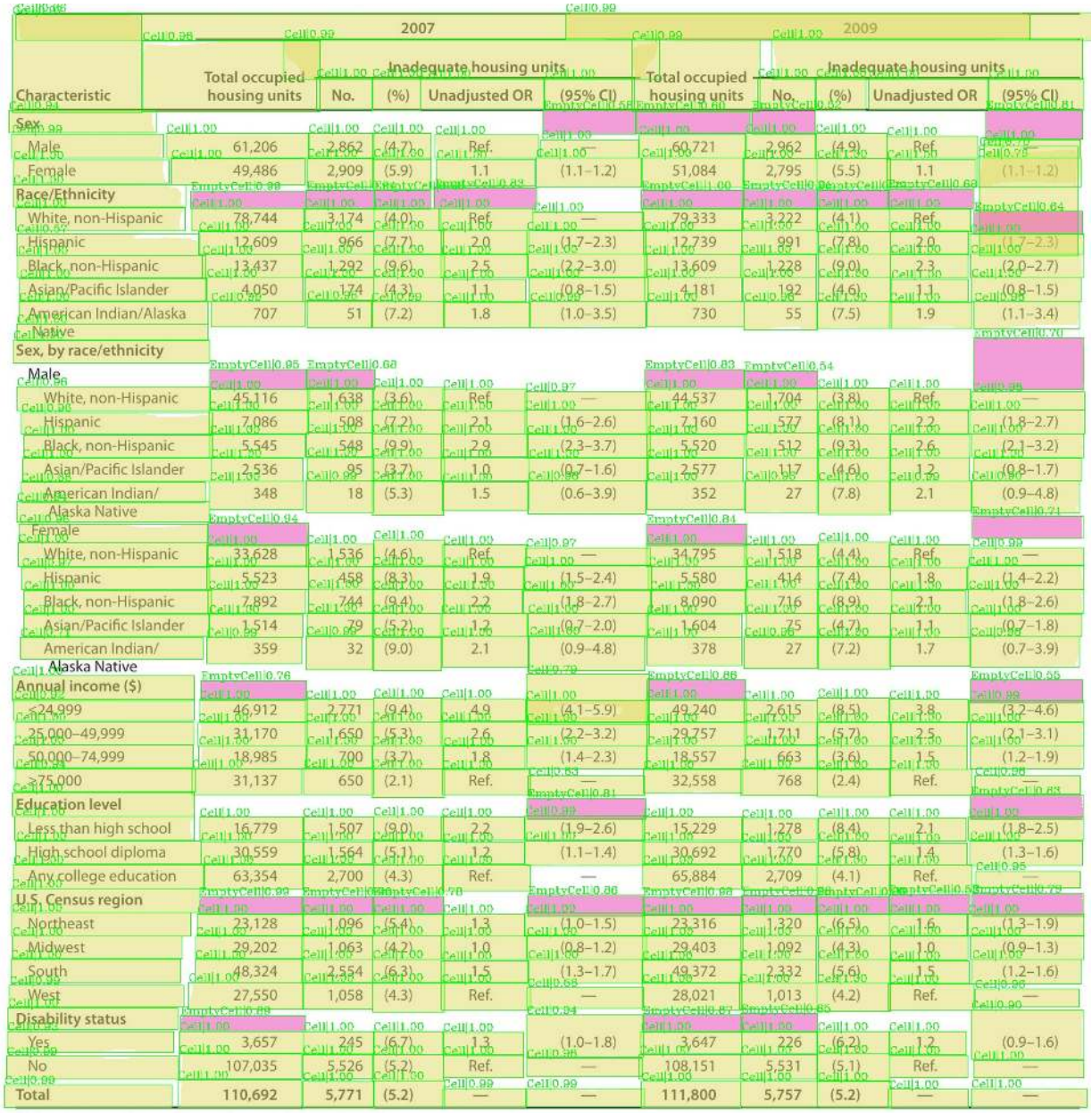
The table in Fig. A.8 illustrates an example of false positives. This table has guiding lines for columns but no guiding lines for rows, there are many rows that are separated by different amounts of spacing. The detector has the most difficulty when classifying cells that are separated by inconsistent spacing. The spacing of most rows and small, however the spacing increases when the category changes. The Row that the detector struggles the most with is the row with category "Aged 6 and older", this is most likely due to the rapid changing of row spacing; a large spaced row following by a small spaced row and then ending with a large spaced row. With the absence of guiding lines the cells are is mostly determined through the white spacing. Another observation is that the detector struggles with symbols, such as "****" and "(X)", these symbols distort the trend of white spaces compared to the letters or numbers and adds complexity to the image.

A.2.2 ICDAR2017 Results

This section shows the inference results of our proposed model of HTC with dual ResNeXt101 backbones arranged in CBNet architecture on the ICDAR2017 cell structure dataset.

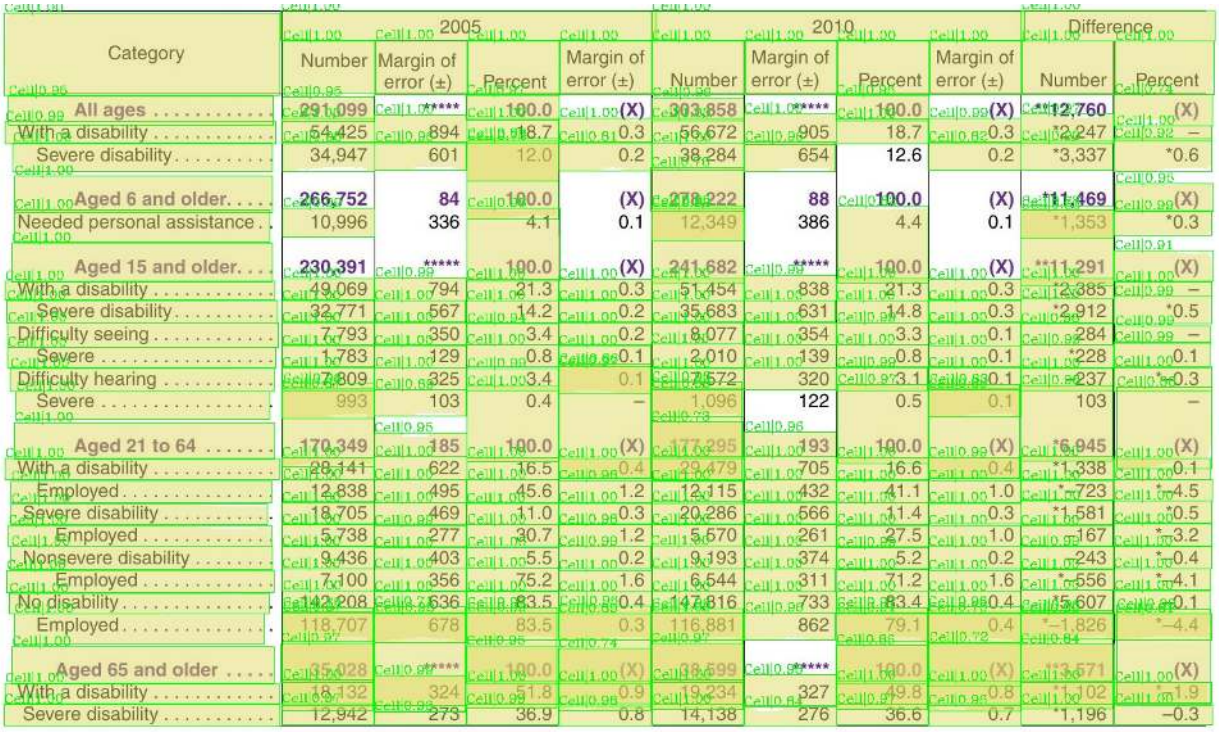
Figure A.9 shows the inference output of a homogeneous table that is separated by white spacing. Homogeneous tables are easier to detect while tables with white spacing presents a challenge as the width or height of the cells is indeterminate.

Figure A.10 shows the inference results on a heterogeneous table separated with white spacing. The hetergenous nature of the table layout is challenging, however our detector is able to correctly classify all cells as well as localize them to a degree in which no content would be cropped out.



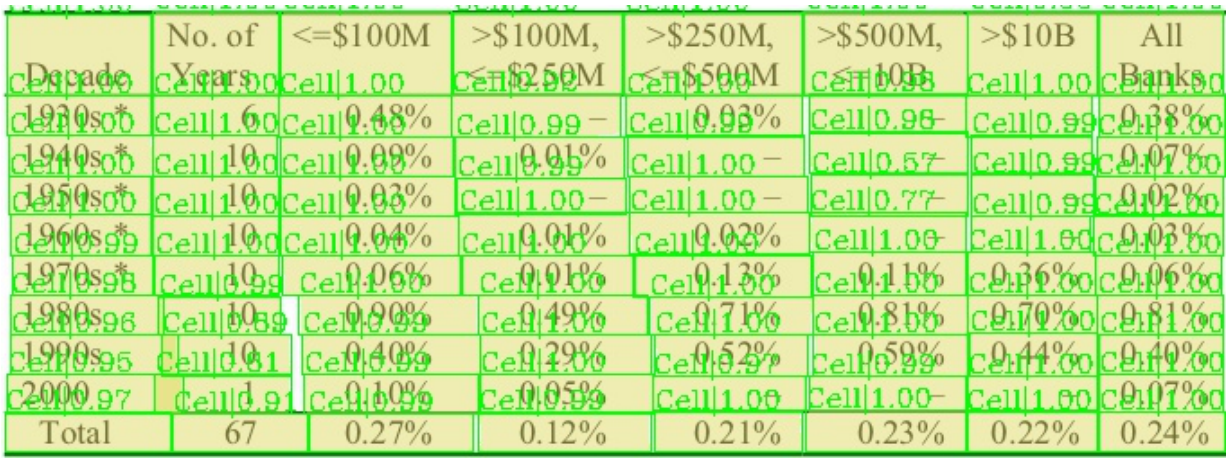
Characteristic	2007					2009				
	Total occupied housing units	Inadequate housing units				Total occupied housing units	Inadequate housing units			
		No.	(%)	Unadjusted OR	(95% CI)		No.	(%)	Unadjusted OR	(95% CI)
Sex										
Male	61,206	2,862	(4.7)	Ref.	—	60,721	2,962	(4.9)	Ref.	—
Female	49,486	2,909	(5.9)	1.1	(1.1–1.2)	51,084	2,795	(5.5)	1.1	(1.1–1.2)
Race/Ethnicity										
White, non-Hispanic	78,744	3,174	(4.0)	Ref.	—	79,333	3,222	(4.1)	Ref.	—
Hispanic	12,609	966	(7.7)	2.0	(1.7–2.3)	12,739	991	(7.8)	2.0	(1.7–2.3)
Black, non-Hispanic	13,437	1,292	(9.6)	2.5	(2.2–3.0)	13,609	1,228	(9.0)	2.3	(2.0–2.7)
Asian/Pacific Islander	4,050	174	(4.3)	1.1	(0.8–1.5)	4,181	192	(4.6)	1.1	(0.8–1.5)
American Indian/Alaska Native	707	51	(7.2)	1.8	(1.0–3.5)	730	55	(7.5)	1.9	(1.1–3.4)
Sex, by race/ethnicity										
Male										
White, non-Hispanic	45,116	1,638	(3.6)	Ref.	—	44,537	1,704	(3.8)	Ref.	—
Hispanic	7,086	508	(7.2)	2.1	(1.6–2.6)	7,160	577	(8.1)	2.2	(1.8–2.7)
Black, non-Hispanic	5,545	548	(9.9)	2.9	(2.3–3.7)	5,520	512	(9.3)	2.6	(2.1–3.2)
Asian/Pacific Islander	2,536	95	(3.7)	1.0	(0.7–1.6)	2,577	117	(4.6)	1.2	(0.8–1.7)
American Indian/ Alaska Native	348	18	(5.3)	1.5	(0.6–3.9)	352	27	(7.8)	2.1	(0.9–4.8)
Female										
White, non-Hispanic	33,628	1,536	(4.6)	Ref.	—	34,795	1,518	(4.4)	Ref.	—
Hispanic	5,523	458	(8.3)	1.9	(1.5–2.4)	5,580	414	(7.4)	1.8	(1.4–2.2)
Black, non-Hispanic	7,892	744	(9.4)	2.2	(1.8–2.7)	8,090	716	(8.9)	2.1	(1.8–2.6)
Asian/Pacific Islander	1,514	79	(5.2)	1.2	(0.7–2.0)	1,604	75	(4.7)	1.1	(0.7–1.8)
American Indian/ Alaska Native	359	32	(9.0)	2.1	(0.9–4.8)	378	27	(7.2)	1.7	(0.7–3.9)
Annual income ($)										
<24,999	46,912	2,771	(9.4)	4.9	(4.1–5.9)	49,240	2,615	(8.5)	3.8	(3.2–4.6)
25,000–49,999	31,170	1,650	(5.3)	2.6	(2.2–3.2)	29,757	1,711	(5.7)	2.5	(2.1–3.1)
50,000–74,999	18,985	700	(3.7)	1.8	(1.4–2.3)	18,557	663	(3.6)	1.5	(1.2–1.9)
≥75,000	31,137	650	(2.1)	Ref.	—	32,558	768	(2.4)	Ref.	—
Education level										
Less than high school	16,779	1,507	(9.0)	2.2	(1.9–2.6)	15,229	1,278	(8.4)	2.1	(1.8–2.5)
High school diploma	30,559	1,564	(5.1)	1.2	(1.1–1.4)	30,692	1,770	(5.8)	1.4	(1.3–1.6)
Any college education	63,354	2,700	(4.3)	Ref.	—	65,884	2,709	(4.1)	Ref.	—
U.S. Census region										
Northeast	23,128	1,096	(5.4)	1.3	(1.0–1.5)	23,316	1,320	(6.5)	1.6	(1.3–1.9)
Midwest	29,202	1,063	(4.2)	1.0	(0.8–1.2)	29,403	1,092	(4.3)	1.0	(0.9–1.3)
South	48,324	2,554	(6.3)	1.5	(1.3–1.7)	49,372	2,332	(5.6)	1.5	(1.2–1.6)
West	27,550	1,058	(4.3)	Ref.	—	28,021	1,013	(4.2)	Ref.	—
Disability status										
Yes	3,657	245	(6.7)	1.3	(1.0–1.8)	3,647	226	(6.2)	1.2	(0.9–1.6)
No	107,035	5,526	(5.2)	Ref.	—	108,151	5,531	(5.1)	Ref.	—
Total	110,692	5,771	(5.2)	—	—	111,800	5,757	(5.2)	—	—

Figure A.7: A heterogeneous table without guiding lines.



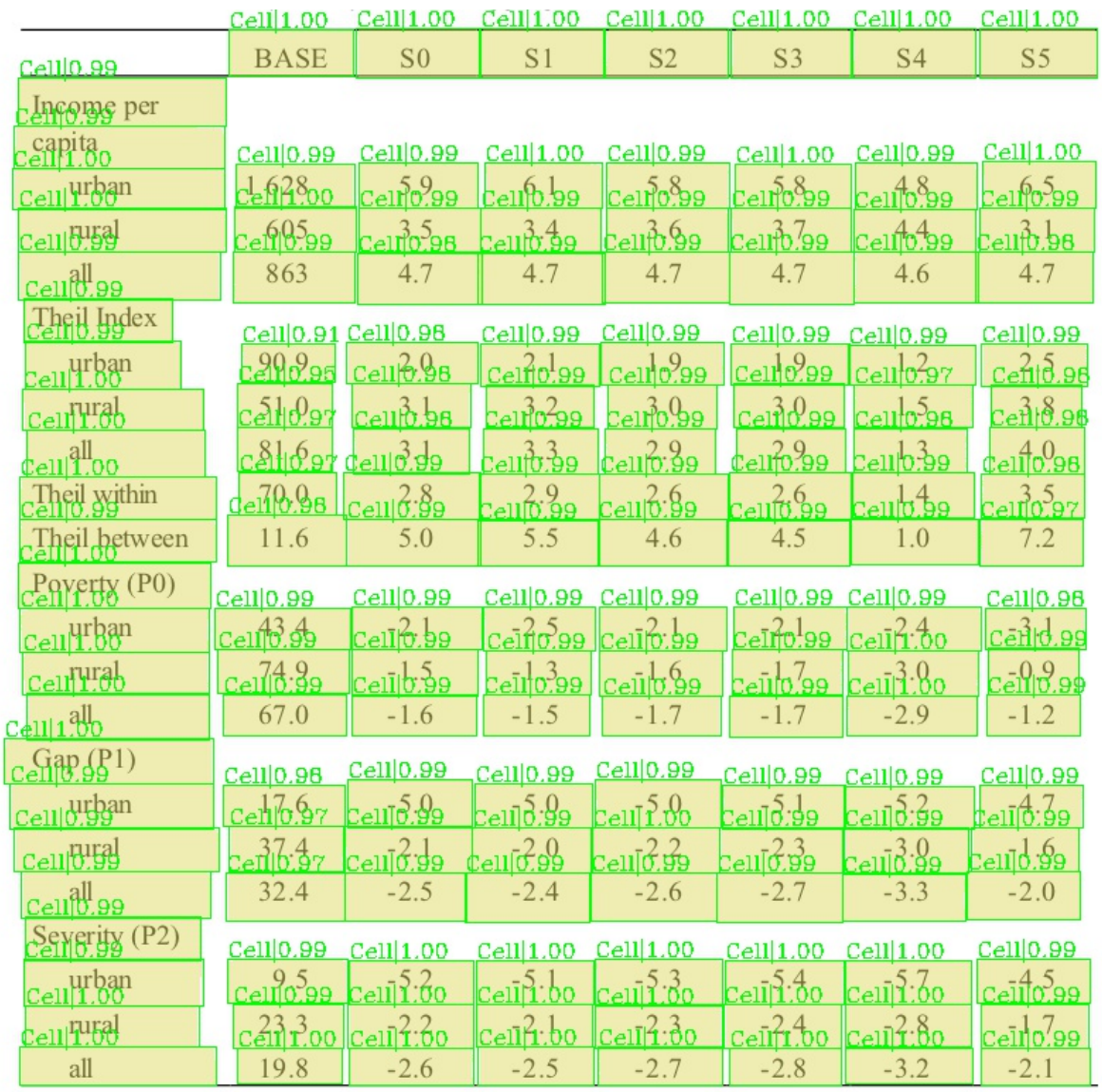
Category	2005				2010				Difference	
	Number	Margin of error (±)	Percent	Margin of error (±)	Number	Margin of error (±)	Percent	Margin of error (±)	Number	Percent
All ages	291,099	*****	100.0	(X)	303,858	*****	100.0	(X)	*12,760	(X)
With a disability	54,425	894	18.7	0.3	56,672	905	18.7	0.3	*2,247	–
Severe disability	34,947	601	12.0	0.2	38,284	654	12.6	0.2	*3,337	*0.6
Aged 6 and older	266,752	84	100.0	(X)	278,222	88	100.0	(X)	*11,469	(X)
Needed personal assistance	10,996	336	4.1	0.1	12,349	386	4.4	0.1	*1,353	*0.3
Aged 15 and older	230,391	*****	100.0	(X)	241,682	*****	100.0	(X)	*11,291	(X)
With a disability	49,069	794	21.3	0.3	51,454	838	21.3	0.3	*2,385	–
Severe disability	32,771	567	14.2	0.2	35,683	631	14.8	0.3	*2,912	*0.5
Difficulty seeing	7,793	350	3.4	0.2	8,077	354	3.3	0.1	*284	–
Severe	1,783	129	0.8	0.1	2,010	139	0.8	0.1	*228	0.1
Difficulty hearing	7,809	325	3.4	0.1	7,572	320	3.1	0.1	*−237	*−0.3
Severe	993	103	0.4	–	1,096	122	0.5	0.1	103	–
Aged 21 to 64	170,349	185	100.0	(X)	177,295	193	100.0	(X)	*6,945	(X)
With a disability	28,141	622	16.5	0.4	29,479	705	16.6	0.4	*1,338	0.1
Employed	12,838	495	45.6	1.2	12,115	432	41.1	1.0	*−723	*−4.5
Severe disability	18,705	469	11.0	0.3	20,286	566	11.4	0.3	*1,581	*0.5
Employed	5,738	277	30.7	1.2	5,570	261	27.5	1.0	*−167	*−3.2
Nonsevere disability	9,436	403	5.5	0.2	9,193	374	5.2	0.2	*−243	*−0.4
Employed	7,100	356	75.2	1.6	6,544	311	71.2	1.6	*−556	*−4.1
No disability	142,208	636	83.5	0.4	147,816	733	83.4	0.4	*5,607	0.1
Employed	118,707	678	83.5	0.3	116,881	862	79.1	0.4	*−1,826	*−4.4
Aged 65 and older	35,028	*****	100.0	(X)	38,599	*****	100.0	(X)	*3,571	(X)
With a disability	18,132	324	51.8	0.9	19,234	327	49.8	0.8	*1,102	*−1.9
Severe disability	12,942	273	36.9	0.8	14,138	276	36.6	0.7	*1,196	−0.3

Figure A.8: An example table to illustrate false positives.



Decade	No. of Years	<=$100M	>$100M, <=$250M	>$250M, <=$500M	>$500M, <=$10B	>$10B	All Banks
1930s*	6	0.48%	–	0.03%	–	–	0.38%
1940s*	10	0.09%	0.01%	–	–	–	0.07%
1950s*	10	0.03%	–	–	–	–	0.02%
1960s*	10	0.04%	0.01%	0.02%	–	–	0.03%
1970s*	10	0.06%	0.01%	0.13%	0.11%	0.36%	0.06%
1980s	10	0.90%	0.49%	0.71%	0.81%	0.70%	0.81%
1990s	10	0.40%	0.29%	0.52%	0.59%	0.44%	0.40%
2000	1	0.10%	0.05%	–	–	–	0.07%
Total	67	0.27%	0.12%	0.21%	0.23%	0.22%	0.24%

Figure A.9: Homogenous table separated by white spaces



	BASE	S0	S1	S2	S3	S4	S5
Income per capita							
urban	1,628	5.9	6.1	5.8	5.8	4.8	6.5
rural	605	3.5	3.4	3.6	3.7	4.4	3.1
all	863	4.7	4.7	4.7	4.7	4.6	4.7
Theil Index							
urban	90.9	2.0	2.1	1.9	1.9	1.2	2.5
rural	51.0	3.1	3.2	3.0	3.0	1.5	3.8
all	81.6	3.1	3.3	2.9	2.9	1.3	4.0
Theil within	70.0	2.8	2.9	2.6	2.6	1.4	3.5
Theil between	11.6	5.0	5.5	4.6	4.5	1.0	7.2
Poverty (P0)							
urban	43.4	-2.1	-2.5	-2.1	-2.1	-2.4	-3.1
rural	74.9	-1.5	-1.3	-1.6	-1.7	-3.0	-0.9
all	67.0	-1.6	-1.5	-1.7	-1.7	-2.9	-1.2
Gap (P1)							
urban	17.6	-5.0	-5.0	-5.0	-5.1	-5.2	-4.7
rural	37.4	-2.1	-2.0	-2.2	-2.3	-3.0	-1.6
all	32.4	-2.5	-2.4	-2.6	-2.7	-3.3	-2.0
Severity (P2)							
urban	9.5	-5.2	-5.1	-5.3	-5.4	-5.7	-4.5
rural	23.3	-2.2	-2.1	-2.3	-2.4	-2.8	-1.7
all	19.8	-2.6	-2.5	-2.7	-2.8	-3.2	-2.1

Figure A.10: Heterogenous Table separted by white spacing

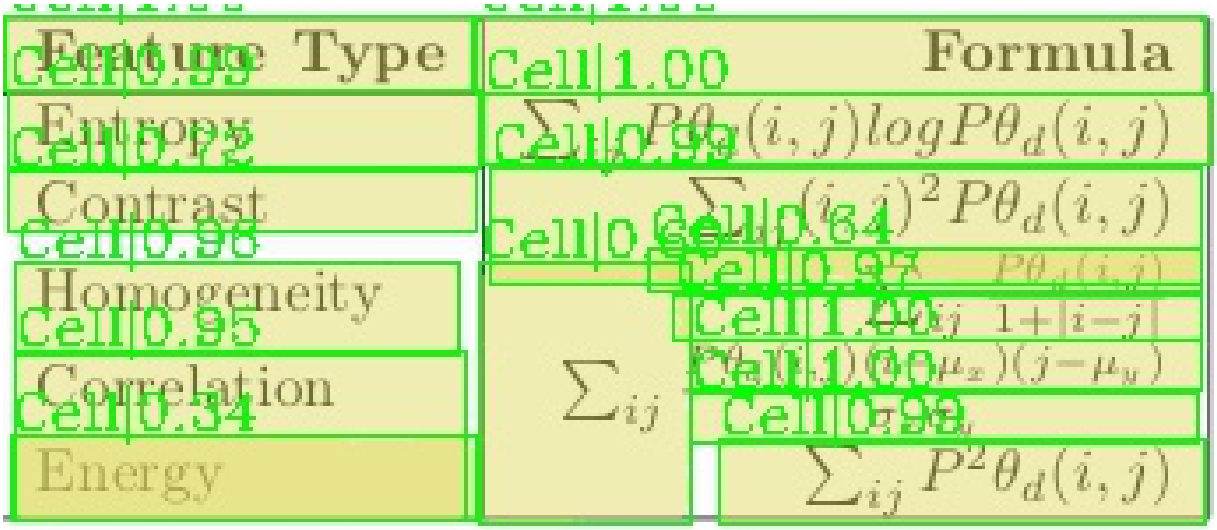


Figure A.11: Table with diverse row spacing

Figure A.11 shows a table that is significantly diverse in row spacing as well as content. However, our cell structure detector was able to identify each line of the equations individually to a high localization accuracy.