Consistency based Semi-Supervised Learning for Table Detection

Author

Afaq Ahmad

2017-NUST-MS-RIME-204725

Supervisor

Dr. Syed Omer Gilani

Department of Robotics and Artificial Intelligence

School of Mechanical & Manufacturing Engineering (SMME)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

August 2021

Consistency based Semi-Supervised Learning for Table Detection

Author

Afaq Ahmad

2017-NUST-MS-RIME-204725

A thesis submitted in partial fulfillment of the requirements for the degree of

M.S. Mechanical Engineering

Thesis Supervisor:

Dr. Syed Omer Gilani

Supervisor's Signature: _____

Department of Robotics and Artificial Intelligence

School of Mechanical & Manufacturing Engineering (SMME)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

August 2021

Certification

We hereby recommend that the dissertation prepared under our supervision by: **Afaq Ahmad (2017-NUST-MS-RIME-204725)** Titled: **"Consistency based Semi-Supervised Learning for Table Detection"** be accepted in partial fulfillment of the requirements for the award of MS Robotics & Intelligent Machine Engineering degree. (Grade _____)

Examination Committee Members

Member: _____

(Dr. Yasar Ayaz)

Member: _____

(Dr. Jawad Khan)

Member: _____

(Dr. Hasan Sajid)

Supervisor: _____

(Dr. Syed Omer Gilani)

_____ _____

Head of Department Date

COUNTERSINGED

_____ _____

Date Principal

Declaration

I certify that this research work titled " Consistency based Semi-Supervised Learning for Table Detection″ is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

AFAQ AHMAD

2017-NUST-MS-RIME-204725

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

AFAQ AHMAD

Registration Number

2017-NUST-MS-RIME-204725

Supervisor: _____

(Dr. Syed Omer Gilani)

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by Afaq Ahmad (RegistrationNo. 00000204725), of Department of Robotics and Intelligent Machine Engineering (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation

Signature:_____

Date:_____

Dr. Syed Omer Gilani (Supervisor)

Signature HOD:_____

Date:_____

Signature Principal:_____

Date:_____

Copyright Statement

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step, and for every new thought you set in my mind to improve it. Indeed I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents and siblings for their support throughout my master's degree and providing me with financial support to enrol and complete my degree. I want to express special thanks to my supervisor Dr. Syed Omer Gilani for his help throughout my thesis and for everything he taught me.

I would also like to pay special thanks to members of my GEC committee, Head of Department Dr. Hasan Sajid, Dr. Yasar Ayaz, Assistant Professor Dr. Jawad Khan and Assistant Professor Dr. Muhammad Naveed for their support, criticism, cooperation and guidance. I would also like to thank Wajih ul hassnain for her help with different aspects of my thesis.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my Parents

Abstract

A large number of comprehensive variety documents are consist of scanned images, in which tables stored summarized facts and valuable information structurally. Detection of tables is the initial step in extracting valuable information from document images. Recently, many researchers used vision-based deep learning techniques for table localization, which requires many labelled training examples. Due to its expensive labelling cost and time consumption, it is necessitating to develop some level of Semi-Supervised learning (SSL) approach. Semi-supervised learning (SSL) trains a model on many unlabeled data to improve predictive performance.

In this thesis, I have used the SSL method "Consistency-based Self Training" to generate the artificial labels for the semanticity preserved augmented unlabeled data and train the model to predict these artificial labels. Generic SSL models are more prone to generate biased predictions because of foreground-background imbalance in the table detection task. Background overfitting is being handled by parent-child shared learning framework, in which different styles augmented images forward to both parent and child. The parent model predicts pseudo labels, and the child updates the parent model weight via Exponential Moving Average. Faster R-CNN ResNet-50 FPN model first trained on comparatively small labelled tables dataset till model convergence. Trained model separately saved as parent and child duplicates to perform high confidence pseudo labels prediction (ground truth) and loss calculation (backward propagation) on different models. Only those parent predictions considered accurate whose confidence is more than 0.7 after the Non-max suppression stage in Faster R-CNN. Performance analysis carried out on the public benchmark data set TableBank: We received a 0.897 F1-score on the TableBank test-set; it shows that our approach generates comparable results to state-of-the-art techniques even with using only a 10% labeled dataset.

Key Words: *Table Detection, Document Digitalization, Semi-Supervised Learning*

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background, Scope and Motivation

According to the World Bank, estimated 3.4 billion internet users worldwide in 2016 [1]. The extensive Internet adoption created new opportunities; cloud computing evolved, Internet of Things (IoT) systems gained acceptability and began to be employed, and new industrial revolutions began to occur. Industry 3.0 began with the widespread adoption of computers, automation, and information technology (I.T.). It was designed to automate manufacturing processes using rudimentary computers such as Programmable Logic Controllers (PLCs) and human intervention. Industry 4.0 is introduced shortly after that. The new industrial revolution comprises smart gadgets that interact with one another and exchange data, with the ultimate goal of automating all processes entirely without human intervention. The new revolution in IoT devices is becoming more pervasive and ubiquitous, enabling large-scale sensing to improve numerous processes and operations. Because no human intervention is desired, the information generated by these inventions is primarily structured digitally. While specific procedures have been digitized, the complete digitization of all processes is still a long way off. Until the world is entirely digitized, the sector will benefit from traditional forms such as printed papers or visual documents intended to be read, scrutinized, and reviewed by humans.

These data formats, primarily PDF, DOC(x), or scanned documents, can be built for human consumption, as machines cannot easily interpret them. Thus, even when data in an industrial process is automated and benefits from the Internet's ubiquity, A.I., cloud computing, and the IoT, inter-organizational processes continue to rely on old methods and must be changed. For example, corporate and government offices (e.g., hospitals) continue to employ traditional paper documents printed, filled in by hand, and then typically registered by a human operator Figure 1.1. We concentrate on table detection from digital-born documents but scanned documents work just as well because our system accepts image formats as input. This thesis and our earlier works [2, 3, 4] focus on and tackle Table Detection from documents. Our findings demonstrate that the most critical and pertinent information is included in tables inside a document. Commercial technologies that function exclusively on predefined templates and cannot accurately detect tables with unique styles. To address it, we are motivated to tackle the

challenge of table discovery in virtually all documents without prior knowledge of the document while remaining fast and accurate. While operating on non-image files (i.e., PDFs) would enable us to employ metadata, it would be limited to a single document type.

Figure 1.1: Large number of scanned documents digitalization

In earlier works [3, 4], deep learning-based Table detection or instance segmentation model is adequate to provide good performance with little to no pre-or post-processing [5]. Supervised learning is commonly applied to train an object detector, meaning that labelled samples of both the fore and background classes are used to learn a Table detection model. The resulting amount of data required can be substantial to account for the wide variety of possible table types. Typically, labelling is a manual operation that is both time-consuming and costly. Methods that combine labelled and unlabeled data are semi-supervised learning, as they fall midway between supervised and unsupervised learning. This thesis investigates and evaluates semi-supervised learning (SSL) algorithms used to train a Table detection model. The SSL algorithms aim to improve detection accuracy while alleviating the demand for labelled data,

which requires tedious and time-consuming work. The results are compared between the implemented SSL algorithms and current state-of-the-art supervised models taught in identical conditions. The benefits and performance improvements of semi-supervised learning methods are easily demonstrated due to the scarcity of table-labelled datasets.

The table extraction task can potentially save thousands of precious human hours that would otherwise be spent extracting data from tables. Despite the absence of pre-or post-processing, our models outperformed state-of-the-art algorithms on TableBank datasets. We conclude, our higher precision and recall are justified by the increasing diversity and volume of data. While SSL has made tremendous progress in categorization, label-efficient training for tasks requiring a high labelling cost is challenging. By utilizing lessons learned from SSL approaches for classification, we offer a simple (just two easily tuneable hyperparameters) and practical (2 label efficiency in low-label regime) SSL framework for object detection. The simplicity of our solution allows for further research aimed at resolving SSL for object detection. The suggested framework is adaptable to various configurations, including soft labels for classification loss, other detector frameworks to Faster RCNN, and alternative data augmentation methodologies. Successful table detection is useable in Information digitalization, turning old records to digital accessible and Mobile Application usage for real-time document tables data extraction using the camera.

CHAPTER 2: LITERATURE REVIEW AND RELATED WORK

Many quantifiable documents never see the light of the analogue world nowadays. Computers produce, populate, store, and process them. Naturally, we cannot ignore physical data; they must be translated to digital formats, and a study has been conducted specifically on this subject [2, 3, 4, 6, 7]. However, even when historical documents are scanned and transformed into digital ones, the same issue exists. These documents contain crucial information such as product specifications, availability, and hazardous warnings from a supply chain standpoint and safety-critical information on individuals or other organizations from a government standpoint. The most critical data is typically provided in the form of tables, figures, and calculations. Tables contain key-value pairs with typically one key and many values, and the information contained in the rows and columns is critical for comprehending the total content. As a result, locating tables is critical for extracting information from the document.

Table detection approaches range from hand-crafted rule-based, supervised deep learning-based object detection and semi-supervised learning-based object detection. The two most common solutions to this problem are treating the tables as objects within the document image or using alignments, tabs, and white spaces or lines. This section is consists of two parts; in 1, we did brief literature on classical computer vision techniques based on hand-crafted features. In 2, we have explained supervised and semi-supervised deep learning-based techniques.

2.1 Hand-crafted Heuristics

Electronic documents can be classified into two types: those that are image-based and those that are PDF-based. Electronic documents are not always PDF files. Scanned documents (images), online documents (HTML), and documents in other formats, such as .docx or .odt, are all instances. Since these other formats are not as widely used as PDFs for document exchange, the research concentrates on these two formats. These document types can be converted to document images, focusing on tabular data extraction, including all sub-problems. Extracting unstructured information from printed papers is a long-standing need that has been studied extensively [2]. Earlier work has relied heavily on hand-crafted rules retrieved visually by human operators [8, 10]. Before AlexNet [11], object detection was often performed using more

traditional approaches in which features were retrieved, and the k-Means algorithm was applied to structures such as bag-of-features. However, because these methods were not convincing, developing heuristics for structured information detection improved performance.

As in [12, 13, 14, 15], the most often used criteria are identifying lines and white spaces to identify tabular sections. These structures aided in achieving better results, as many tables contain enclosing lines and spaces to denote rows and columns, simplifying table detection. Another often-used technique is splitting words into blocks and utilizing alignment and neighbouring relations to connect them to other words or space [16, 12]. While locating bounding boxes for PDF documents is trivial, another processing layer must extract words from the pixels when the content is in picture format. Mathematical morphology [17] provides two approaches for extracting text chunks as related components. Morphological dilation followed by morphological erosion fills up the small gaps between characters in a picture, and ideally, all the characters in a word are now connected. Following the morphological closure, a linked component discovery approach (Union-Find) is used to locate connected components. The connected components that result are then categorized as words or lines based on their size. The result is employed as word bounding boxes, which serve as the foundation for numerous studies on detecting tabular structures [14, 3].

Researchers created technology mainly geared for word and other structures identification in 1968 [6]. Their hardware can be operated by a human operator who provides input. For more sophisticated texts, such as academic articles, the runtime is longer. Additionally, it can work in an unsupervised mode. With its proprietary hardware design, this effort alone demonstrates why information extraction from documents is critical for many enterprises. Due to the binding nature of the table detection task for information extraction facilities has also been studied extensively. Typically, early works benefit from heuristics discovered through an eye examination and rudimentary feature extraction. Wahl et al. (1982) [8] concentrate on digitized documents and work in the realm of document images. They create a binary operation applied to the image to fill in small gaps with black pixels. This operation is performed horizontally and vertically, followed by a smoothing operation. As a result, the writers categorize the image as meaningful chunks. They use a rule that begins at the left-hand edge of the image and iterates on the right to separate the discovered blocks. Each black pixel close to the previous one is assigned the same label. Then, based on the block segment's height and the

ratio of the block's pixels to the boundary line, these blocks are categorized as various page objects.

Pyreddy and Croft (1997) [19] presented the TINTIN system, a table detection method based on heuristics that work on electronic texts. The phase of table extraction is produced by examining the white spaces in the text. Whitespaces are compared to the text's alignment and the number of whitespaces between the words. The system begins by detecting columns that have whitespace. For instance, there should be at least three whitespaces between the columns. A few factors are set via visual inspection. After extracting table blocks, the second process labels the various components. TINTIN demonstrated that such simple heuristics are capable of spotting tables and are extensible. Jain and Yu (1998) [7] is an early work that addresses transforming paper documents to their electronic counterparts. As with many previous initiatives, this one concentrates on a single category of document: technical journal articles. The extraction procedure is based on documents' geometric layout and the extraction of their related components from their binary pictures. Due to the comparable design of all the papers, the authors incorporated domain knowledge into the solution. The block adjacency method is used to discover connected components while ruling lines detect table, text, and image sections. Apart from that, they also estimate and fix the document's orientation.

Kieninger and Dengel (1999) [20] pioneered the T-Recs table identification approach, which has gained widespread acceptance and success [12, 13]. The study is concerned with document segmentation on arbitrary documents and is organized hierarchically into four components: words, lines, blocks, and the document itself. T-Recs is not a top-down algorithm but rather a bottom-up approach in which words are parsed for blocks. Recognized tables begin with a 'word,' which is a bounding box. This box is then connected to the following boxes if they are near enough to form 'blocks.' If the resulting blocks are adjacent horizontally, they are assumed to be members of the same table. The authors developed rules for detecting internal table structures (rows and columns) within these supposed table regions. Kieninger and Strieder (1999) [16] refined the T-Recs detection model shortly after its initial publication. Tables are recognized in the original T-Recs approach by connecting the blocks. Columns are recognized in [16] by the vertical alignment of the blocks and then aggregated to make tables. T-Recs is a critical step in developing a robust table identification algorithm that is not dependent on non-tabular structures such as lines. That is, lines are not required for the creation of a tabular region.

Cesarini et al. (2003) [21] presented an approach called Tabfinder for utilizing the lines surrounding the tables. Tabfinder distinguishes itself from previous work by utilizing a modified version of XY-trees on the page pictures. Every node in the tree signifies a part of the image containing a table, and it is formed whenever a new line in the sub-image is identified. This tree is constructed iteratively to determine possible table places. As with the previous examples, tables are checked using either lines or suitable spacing between words that create rows or columns. Tabfinder demonstrates that by incorporating additional data structures, it is feasible to improve hand-crafted heuristic-based solutions further. Gatos et al. (2005) [22] suggested a table detector that is not heuristic-dependent and can be used on any document picture. The procedure begins by estimating the character size and line length using black run processing. Connected blobs generated by continuous black pixels are connected (black runs) and labelled. The authors first determine the intersection of all detected lines and then delete them to complete the detection. Following that, the intersection locations are compared to the alignment to complete the table detection. This method is highly dependent on table lines (both horizontal and vertical) and scan quality.

Shafait and Smith (2010) perform the detection in [10] by first determining the document's column layout. The table detection algorithm is based on identifying column divisions identified using connected components and tab-stops between them. The system relies on hand-crafted algorithms to identify these locations and would have difficulty identifying them automatically. Fang et al. (2011) [15] described a technique for detecting tables in PDF documents. Again, detection is contingent upon the presence of white spaces and ruling lines in the manuscript. Preliminary white space processing distinguishes between page columns and table rows. The approach completes the process by performing table detection, mainly relying on hand-crafted rules developed through visual inspection and insights.

Dey et al. (2016) [23] employ a consensus-based approach to extract tables from document images in another work. As with [10, 22, 7], the primary construction block is the connected components, dependent on hand-crafted heuristics. Then, from these components, colour and stroke features are derived. Consensus-based clustering is then performed on each pair of features to produce statistical similarity and anticipate links between related components. Finally, extracted graphs are renamed as tables. Another approach derived from linked component extraction is Tran et al. (2016) [14]. The authors suggest an approach based on

morphology [17] for extracting related components combined to produce Regions of Interest (ROIs) using hand-crafted rules and thresholds. Following the creation of the RoI, the white space between connected components is analyzed to determine if they are distinct or not. The retrieved text chunks are then aligned vertically to create columns. These columns are what make up the tables. Each step of the process is highly engineered and strongly reliant on heuristics, implying that the model may generate incorrect tables when the document changes.

2.2 Supervised and Semi-Supervised

Although hand-crafted heuristics are commonly used, more recent research has examined techniques based on machine learning [2, 4]. Support Vector Machines (SVMs) is a supervised learning technique for classifying data by determining a linear (or non-linear) separator [26]. Machine learning algorithms require identifying line or word bounding boxes, and following the extraction phase, the recovered features are fed into an SVM classifier to detect tables. [27, 28] illustrate that this technique produces high-quality results. Along with SVMs, additional machine learning algorithms have been evaluated, including Decision Trees [30], Hidden Markov Models (HMMs) [31], and Conditional Random Fields (CRFs) [32]. Decision Trees are a machine learning technique that uses the training set to generate a tree-like structure that classifies the input. Because decision trees require organized input, they are typically employed to detect tables within HTML texts. Hidden Markov Models are probabilistic Markov models that contain unknown states. HMMs can locate the tables using information collected from the document metadata, such as the appearance of images and certain words or hyperlinks. CRFs are typically employed for pattern discovery, and in the context of tabular structure detection, they are utilized to model the data's dependencies.

The term "deep learning" belongs to a subgroup of machine learning algorithms. Wherever multilayer perceptrons or more specialized architectures like as LSTMs or CNNs are used in the learning process. This thesis is entirely devoted to CNNs, a commonly used deep learning framework for computer vision applications. Because we focus on detecting tabular structures in document images, CNNs are optimal for this task. CNN's are specialized deep neural networks based on multilayer perceptrons that have been built to mimic how animals perceive visual information. To begin, Kunihiko Fukushima (1980) postulated the recognition [33]. It was inspired by a prior study on monkeys, which shown that specific neurons in these

animals' visual cortexes respond solely to specific portions of the visual field [33]. Back-propagation (the de facto optimization approach used in neural networks) was not as developed during these times today. As a result, these early networks were trained using various optimization techniques and were ultimately unsuccessful. LeCun, Yann (1989) [34] developed a completely autonomous neural network trained using back-propagation and gradient descent. [34] can be considered the forerunner of today's CNNs, and while CNNs have been utilized and refined for various situations since then, the long-awaited breakthrough did not occur until AlexNet [11].

Although SVM performance improved as kernel functions were added [35], AlexNet [11] won the 2012 ImageNet competition [36] and significantly outperformed the previous winner in terms of accuracy. Krizhevsky et al. (2012) [11] demonstrated that CNNs outperform all other image classification algorithms AlexNet's success is attributed to the recent rise in processing capacity, which enables multiple layers and parameters. Although AlexNet appears to have set a new high, approaches based on CNNs consistently outperformed AlexNet on similar tasks [37, 39]. The VGG approach was proposed by Simonyan and Zisserman (2014) [40]. The authors considered the changes that have occurred since the AlexNet was created and adjusted the depth parameter. They investigated various depths and concluded that the best performing model was a 16 layer deep CNN. Szegedy et al. [37] introduced the Inception Module in 2014, consisting of many convolutional neural networks assembled on top of one another and their outputs concatenated. They then built GoogLeNet by stacking numerous inception modules, one of the deepest convolutional neural networks, having 22 layers.

VGG and GoogLeNet attempted to train CNNs with additional layers, but the accuracy degraded. The primary reason for this phenomenon is referred to as the vanishing gradient problem. Because layers contain convolution and max-pooling layers and activation functions such as sigmoid, which reduce the values to the 0 to 1 range, the gradient is lost or not the same as in the final layer where it was calculated. To address this issue, He et al. (2015) [39] presented a unique technique called Residual Learning and Skip Connections (ResNet). They used forward connections between convolutional layers to backpropagate the gradient while preserving as much information as possible. It allows them to train deeper CNNs and attain 1202 layers [39]. These recent enhancements enabled CNNs to perform exceedingly well in various computer vision applications requiring visually prominent objects/environments, such as pedestrian

detection [41] and event detection [42]. They have also been used to time-series tasks with positive results and are sometimes used in place of LSTMs (LSTMs are a form of the neural network developed explicitly for time-series problems) [43].

Apart from their general success, machine learning approaches, specifically CNNs, perform exceptionally well on table detection tasks. Wang et al. (2001) [44] present a portion classification approach, rather than a detection method, utilizing customized decision trees to categorize zones into graphical regions such as text, picture, and table. Classification makes use of vertical and horizontal white spaces, as well as input rows and columns. Although this study is not as significant as others, it is one of the earliest examples of machine learning methods applied to the document analysis problem. Wang and Hu (2004) [28] later concentrated on classifying web tables in a subsequent paper. Decision trees and SVMs are used to classify tables as genuine or non-genuine. Statistical information about rows, columns, and cells is used to train these techniques. Another element that we found intriguing is the average constancy of content type across rows and columns. [28] demonstrates encouraging results demonstrating that machine learning approaches may accurately understand how a table is produced.

Simultaneously, Ng et al. (2007) [45] attempted to classify using the decision tree technique and a simple feed-forward neural network with back-propagation. As with earlier machine learning-based research, Ng et al. utilize these methods as a final step, and domain expertise is still required. Horizontal lines extract features since they may define the borders of tables, rows, and columns. The models are taught to recognize table, row, and column boundaries. As a result, it is a three-step process in which the borders are identified first, and then the vertical and horizontal lines included within the boundary are classified as rows and columns. Fan and Kim (2015) [46] construct their dataset using a technique called Distant Supervision. Distant Supervision is a technique in which unlabelled line data is labelled using a simpler unsupervised classifier based on heuristics. Because the classifier used to label the data is imperfect, there will be errors, but machine learning methods are resistant to these faults. With noisy data [47], the authors' ensemble learning algorithm would learn to detect tables. SVM, Logistic Regression, and Naive Bayes algorithms are included in the ensemble technique. Textual features with unlabelled characteristics are extracted to aid with prediction. The authors also make use of existing knowledge during the feature extraction step. For instance, they extract a feature because the table will contain more nouns than adjectives and adverbs.

Following the success of deep learning in computer vision [11], researchers working on document analysis from document images began incorporating deep learning into their methods. Hao et al. (2016) [48] offer a method for detecting tables based on heuristics and CNNs. Specific regions are identified and labelled as potential tablespaces. The horizontal and vertical lines observed in the document image are used to locate these locations via heuristics-based search. [48] employs CNN and heuristics to validate the located table. As a result, deep learning is used as a judge after a more conventional table detection approach. A fascinating feature of this method is that some information retrieved from the original PDF is included in either the CNN's input or output. Coordinates of lines or words are among the extracted information crops. The authors demonstrate that when the information is introduced to the CNN input, the precision on the ICDAR 2013 [49] dataset may be significantly increased.

Schreiber et al. (2017) [50] offer a method for detecting end-to-end tables and their structures. Previously, these systems featured at least one part of the process that required hand-crafted rules. However, in DeepDeSRT [50], the detection method is end to end, meaning no human participation is required given the document picture. Deep learning model training requires massive volumes of training data are required, which are scarce in the domain of tabular structure identification. As a result, the authors opt for transfer learning, which is the process of taking a previously trained deep neural network on an extensive dataset and fine-tuning it on a smaller dataset to transfer the learned information to the new domain. The suggested table identification approach, which makes use of Faster-RCNN [51], outperforms a large number of others. The ICDAR 2013 test results [49] indicate that

Following the tremendous success of deep learning technologies, many people experimented with novel ways to use them. Gilani et al. (2017) [52] proposed another end-to-end table detection approach using the Faster-RCNN [51] architecture and document pictures. In contrast to DeepDeSRT[50], they alter the input image to appear more natural (i.e., real-life, RGB images). Rather than using RGB, they used three different distance measures for each dimension of the image. The distance metrics indicate the separation of text sections from white spaces. This modification results in the image containing only the architectural remnants and not the figures, lines, or text. [52] demonstrates how picture transformation can increase recall and precision. Li et al. (2018) [53] describe a system for identifying multiple page items, such as tables and formulas. The system is fed an image of a binarized document as input. After the

contour tracing approach identifies related components, the page image is split into columns and lines. At this stage, the machine learning algorithm Conditional Random Field(CRF) is used to classify each line region into four categories. Apart from classifying the lines, the same model is utilized to determine the connection between them. Heuristics are applied as a post-processing step to correct for possible misclassifications. Finally, a verification model is used. The authors assert that certain line regions may be misclassified due to information loss during resizing because the CRF accepts a fixed-size input. As a result, the preliminary classification is verified using a final deep learning approach, CNN. The ICDAR 2017 POD Dataset, [76] achieves an excellent F1 value of 96 percent [54].

Kerwat et al. (2018) [27] opted to examine different recent deep learning-based CNN architectures utilizing the public ICDAR 2013 [49] test set. Faster-RCNN, Single Shot Detector (SSD), and You Only Look Once (YOLO) are the approaches that are compared [51, 55, 56]. SSD is a single-stage object detector based on the anchor concept. Anchors are rectangles with predefined ratios set at each grid location to guide predicting bounding boxes. While performing bottom-up processing, SSD [55] predicts bounding boxes. Results in predictions being made at different scales; thus, several predictions are generated on a single image in a single run. YOLO [56], on the other hand, is primarily concerned with speed and conducts only one detection in a single pass. Comparing these two approaches makes it clear that Faster-RCNN has a significant advantage in terms of the accuracy of table and figure predictions [27]. Siddiqui et al. (2018) [57] propose a model based on Faster-RCNN [51] with a deformable ResNet-101 [39] backbone feature extractor.

In typical CNNs, features are retrieved from a rectangular area that grows larger as the layers progress. While feature extraction is more challenging in deformable CNNs, the receptive fields of the neurons in the network may be modified because they are dynamic and learnable. The suggested technique [57] achieves near-perfect performance on the ICDAR 2013 dataset and excels on the ICDAR 2017 dataset [49, 54]. Although the model performs admirably on ICDAR datasets, it suffers on the UNLV dataset, which contains scanned papers and provides a more significant challenge. The results demonstrate that by adding deformable layers to ResNet-101, the F1 score increases by 4% in ICDAR 2017 but has no effect in ICDAR 2013. It could mean that deformable layers are overfitted to the table layouts, so performance worsens when the style/purpose of the documents changes. Kavasidis et al. (2018) [58] propose using a deep

convolutional neural network to recognize tables and charts inside digital document pictures. Although the detection network is based on VGG-16 [40], filter sizes have been increased to address small/thin objects (lines, white spaces, rows, and columns).

Arif and Shafait (2018) [59] review prior deep learning-based table detection systems and offer enhancements. The enhancements are motivated by the idea that tables include more numerical data than other document sections. Faster-RCNN [51] is the deep learning approach under investigation, and it is evaluated using the publicly available UNLV dataset. The steps involved in pre-processing (improvements) can be classified into two categories: I colourization, in which text is changed to green and numbers to red; (ii) picture transformation depending on the distance between blue pixels, as described in [52, 59]. Faster-RCNN [51] with ResNet-101 [39] backbone feature extractor is used to train the network. They, like us, labelled their own training set and performed tests against the UNLV dataset. The statistics indicate an around 90% F1 score.

As seen in [2, 58], the general focus of research in recent years, with the advancement and popularity of deep learning, has applied state-of-the-art object detectors to the table detection domain. Semi-Supervised learning (SSL) made remarkable progress; SSL methods have been primarily applied to image classification, whose labelling cost is relatively cheaper than other significant computer vision problems, such as object detection. Object detection requires greater label efficiency due to the high cost of labelling, necessitating the development of dependable SSL technologies.

Semi-supervised learning (SSL) has gained increasing attention in recent years because it enables the use of unlabeled data to enhance model performance in the absence of large-scale annotated data. The term "Consistency-based Self Training" refers to a common class of SSL approaches [61,62,63,64, 68,69,70,71]. The central idea is to construct false labels for unlabeled data and train the model to predict them when unlabeled data is fed with semantically preserved stochastic augmentations. The artificial label might be either a single-prediction (hard) or the predictive distribution of the model (soft). The third foundation of SSL's success is progress in data augmentation. Data augmentations enhance the robustness of deep neural networks [72] and are particularly useful for self-training based on consistency [61, 70, 71, 68]. The augmentation approach can range from a manual mix of fundamental picture changes such as colour jittering, rotation, translation, flipping, or neural image-synthesis [76] and reinforcement learning policies

[77, 78]. Recent work has demonstrated that complicated data augmentation algorithms, such as RandAugment [79] or CTAugment [61], are effective for SSL image classification [61, 68, 70, 71].

"Consistency regularisation" has become a prominent methodology for object detection [63, 70] and inspires [83]. The goal is to impose consistency on the model to create consistent predictions across label-preserving data augmentations. Mean-Teacher [69], UDA [70], and MixMatch [62] are a few examples. Another widely used type of SSL is pseudo labelling [64, 79], which can be considered a more difficult version of consistency regularisation: the model self-trains to generate pseudo labels for unlabeled data trains randomly augmented unlabeled data to match the generated pseudo labels. Understanding how to use pseudo labels is important to SSL's success. For example, Noisy-Student [71] presents an iterative teacher-student architecture to identify assignments using a teacher model and subsequently train a bigger student model. By exploiting additional unlabeled photos in the wild, this technique achieves state-of-the-art performance on ImageNet classification. FixMatch [68] illustrates a straightforward method that beats prior approaches and achieves state-of-the-art performance, particularly on various small labelled data regimes. FixMatch's central concept matches the forecast of strongly augmented unlabeled data to the pseudo label of its weakly augmented counterpart when the model confidence in the weakly augmented counterpart is high. In light of these methods' effectiveness, this thesis effectively uses pseudo labelling, pseudo boxes, and data augmentations to improve table detectors.

CHAPTER 3: METHODOLOGY

Tables appear in various document categories, including technical reports, electronic component datasheets, and medical records. Given that many of these firms are data-driven, extracting structured data from the papers became critical. As seen in Chapter 2, beginning in the 1980s and notably as the years passed, numerous researchers presented solutions to this problem. The most current methods are based on deep learning, and this chapter defines and details the methodologies we studied to offer our table detection system. We present an end-to-end table detection method based on object detection networks trained using deep learning. It involves detecting tables using deep learning networks Faster-RCNN [51].

3.1 Deep Learning-Based Methods

This part begins by describing the backbone feature extractors utilized in the deep learning object detectors. It then thoroughly examines the deep learning methods under consideration, highlighting their differences, benefits, and drawbacks.

3.1.1 Base Networks

In recent years, computer vision has benefited from the benign success of CNNs and deep learning. Initially, conventional CNNs with Dropout [84] and BatchNorm [85] were sufficient to achieve a breakthrough on computer vision competition datasets such as the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [36] and others. As a result, several types of research have been conducted in these areas, with positive outcomes. AlexNet [11], GoogLeNet/Inception Module [37], VGG [40], and ResNet [39] are the first effective deep learning-based based on CNN approaches that had an influence. All of these diverse networks demonstrated the value of exploring alternative topologies. For instance, GoogLeNet demonstrated the benefits of stacking convolutions on top of the other and then concatenating their outputs back together and continuing this process for layers upon layers. VGG demonstrated that even networks with fewer layers and parameters might get outstanding results despite having only 16 or 19 layers, and it is still widely used due to its size and speed. On the other side, ResNet added Skip Connections and went deeper, reaching almost 1200 layers. Their findings demonstrate that delving deeper is not always beneficial and that we must further

explore more efficient designs to improve computer vision systems. The hidden layers of these networks are typically activated by Rectified Linear Units (ReLu) [86]. The ReLu function is a nonlinear function frequently employed in deep learning models [11, 39, 40]. The function is easy; it returns 0 if the input is negative, and the value is unchanged if the input is positive. As a result, the formula is as follows:

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

Deep learning approaches incorporate activation functions since the processes are linear, and the resulting model is similar to a linear regression model. Thus, non-linearity functions are added to networks to represent complicated nonlinear array mappings between input and output. Additionally, the activation functions must be differentiable in order to optimize the model via backpropagation. When we examine Eq. 3.1, we can see that when $x = 0$, the ReLu function is not differentiable. It is still utilized, though, because gradient descent cannot easily reach a local minimum. Implementations of the functions choose a derivative from one of its sides if it occurs. Another aspect to clarify is that activation functions should be nonlinear, but ReLu comprises two connected linear units. Because models consist of numerous connected layers, each layer alters the slope of the linear function at different points, resulting in a function with numerous variable slopes. These back-to-back connections with ReLu allow for the approximation computation of smooth (nonlinear) functions [87].

The S-shaped curves of historical activation functions such as tanh and sigmoid enable them to retain tiny changes in the input and affect the function's output. However, these activation functions are stacked, the derivative is near zero except for the smallest area (the central half of the S-shape). These flat derivatives make it more challenging to update the layer weights, resulting in the vanishing gradient problem. When ReLu is applied to a sufficiently large batch, some nodes will have non-zero activations, increasing the average derivative. Although the usage of ReLu is widespread, activations may appear to be zero layers following layers, resulting in the problem of dead neurons. A dead neuron is never stimulated (activation output > 0), and a modification to the ReLu is proposed to address this issue; the Leaky-ReLu [88]. Leaky-ReLu can be defined as follows:

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

When the "α" value is less than 1.0 or equal to 1.0, a small amount of activity is retained to maintain weight updates and neurons alive when the input is negative.

While pure CNN networks such as Resnet and VGG beat all previous image recognition/classification methods, they are reached on maximum accuracy, and newer research not showing any progress. Researchers have proposed sophisticated architectures as a means of evacuating this plateau. Each layer in a convolutional network encodes different characteristics that describe the input at some abstraction level, and stacked convolutional layers, as demonstrated by [89], can encode hierarchical structures representing the input. For instance, the first layer can encode edges, lines, and corners, and as we progress through the layers, the encoded information transforms into higher-level abstractions such as a car's doors or wheels. Ignoring the results of prior approaches is a bad idea, so they are still utilized as feature extractors and sometimes referred to as backbone networks. Although these CNN models were initially intended to categorize images, they are now exclusively used to extract information from the internal layers by object detectors.

Object detectors of the next generation are classified into two types: (i) single-stage detectors and (ii) two-stage detectors. Single-stage detectors use the backbone network's retrieved features and put anchors to designate possible item positions. Anchors are predefined shapes (often using k-means) that reflect the collection's most frequently occurring shapes of instances. The image is then divided into grids, and anchors are set in the centres of the grids. Following that, these anchors are categorized in order to detect things. On the other side, two-stage detectors include an extra neural network (or, in older approaches, a search algorithm) that proposes possible item positions. Due to the added overhead associated with two-stage detectors, they are often slower but more accurate. This subsection discusses the ResNet-50 model as feature extractors that we utilize in further detail. ResNet-50 used as backbone networks for feature extraction is employed in our table detection technique.

3.1.1.1 ResNet

With the proliferation of models and varied designs presented by academics, numerous previously undiscovered issues appeared. One of these was the vanishing gradient problem, or the loss of accuracy as the network's depth increased. He et al. (2015) [39] discuss the

deterioration problem and offer a method called residual learning. They accomplish this through the use of shortcut connections. Shortcut connections enable the feed-forward and backward paths to bypass a couple of layers, where the outputs of former layers are added to the outputs of subsequent layers.

Consider the following example to gain a better understanding of the problem ResNet is attempting to tackle. Assume we have an external network that produces f (x) when given *x* as an input. We want to improve the accuracy of this neural network by adding additional layers (x). However, as noted in [39], this is not always the case, as it turns out that deeper networks may have lesser accuracy. As a result, ResNet chose to force the network to explicitly learn an identity mapping by learning the residual of input and output. Assume the network's (or a subnetwork's) input is "*x*" and the "true" output is "*H(x)*". This subnetwork's residual is then defined as;

$$f\ (x)\ =\ H(x) - x \tag{3.3}$$

Given our objective of determining the actual output, the equation is reformed as follows:

$$H(x)\ =\ f\ (x)\ +\ x \tag{3.4}$$

This divergence is what distinguishes residual learning from typical neural networks. Earlier networks attempted to learn *H(x)* directly, whereas residual learning is about learning the residual. As an outcome, the network can learn to make *f(x)* equal 0 in Eq. 3.4 and skip some subnetworks. This trait is also visible during the backpropagation phase when the network disregards gradients in some subnetworks and sends them back unchanged. The nature of CNNs with hierarchical feature encoding implied that deeper networks would perform better. Previously, this was not feasible. He et al. [39] demonstrated that deeper networks are achievable using residual learning and shortcut links and established that deeper networks could improve performance. Additionally, when their ResNet-101 feature extractor was utilized in the Faster-RCNN [51] object detector, they saw a relative improvement of 28% over the VGG16 network.

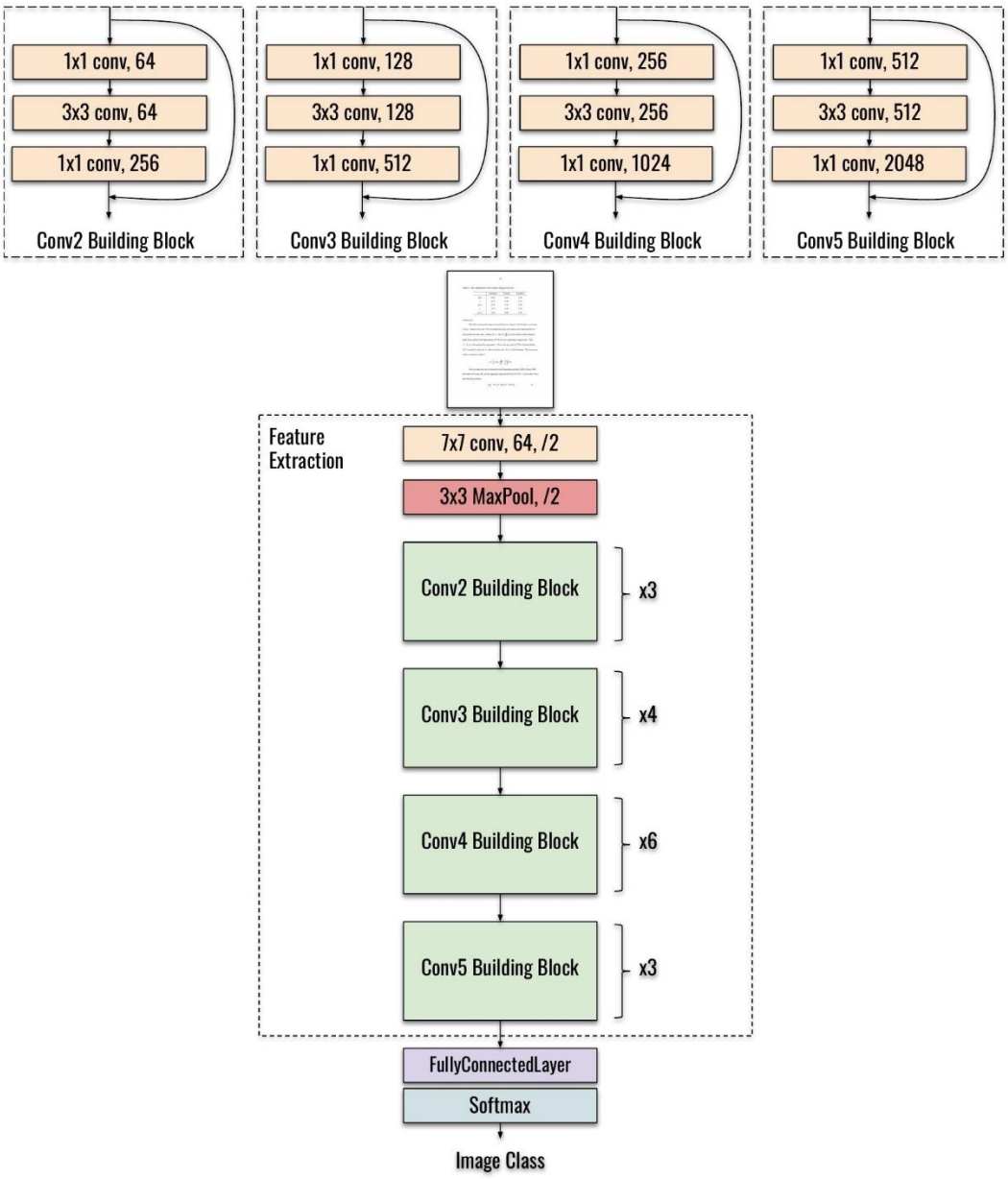


Figure 3.1: ResNet-50 architecture

We conduct our studies using the ResNet-50 architecture, which is a 50-layer variant of the ResNet design. The general architecture is illustrated in Figure 3.1. The variable *f(x)* in Eq. 3.4 denotes the subnetworks outputs or building pieces. Strides connect each convolutional block. Results in the following strides are added to each convolution block's input picture: 4,8,16,32 for conv2, conv3, conv4, and conv5. As with the VGG-16 network, we solely employ the convolutional network for feature extraction, as shown in Figure 3.2. We employ a pre-trained

ResNet-50 on ImageNet [90]. Then we either delete the final connected layers (this model is referred to as ResNet-50) or remove both fully connected layers and the conv5 block, in which case we refer to the model as ResNet-50-C4. By removing the conv5 block, the network's size, making it easier to train on sparse input. Additionally, when the stride of 32 is considered with the input image, conv5 loses the low-level semantics of the input picture. To address these concerns, we suggest designs such as FPN, which we will outline in greater detail in the following subsections.

3.1.2 R-CNN Family

Object identification is a significantly more difficult problem to solve than image classification. This complication stems from two issues: the object's location must be determined, and (ii) the discovered region must be categorized. The most straightforward approach to developing the methods was image classification. Each image is assigned to one of the specified classes. However, as we observe our environment, we also notice objects and infer context from them. Computer vision algorithms such as the Histogram of Oriented Gradients (HOG) [91] and the Scale-Invariant Feature Transform (SIFT) [92] were used to perform visual detection tasks. SIFT and HOG are feature histogram-based algorithms similar to the V1 layer (where the input from the human retina is connected in the brain) of the human visual recognition system, the primates' first sensory areas in the optical path [93, 94]. When we observe the human visual recognition system, we see that the visual system is separated into distinct visual areas, each of which is coupled in a feed-forward fashion and depicts a structure hierarchically [94]. PASCAL VOC [95] is a well-known and challenging object detection dataset, and it is frequently used to compare algorithms.

Although SIFT and HOG dominated the PASCAL VOC competition, a plateau was noted in recent years when the precision was measured over time (about 2012). Approaches based on hierarchical and deep learning were also developed. As discussed in Subsection 2.1.2, Fukushima introduced 'neocognitron,' inspired by human biology. Although inspired by hierarchical and shift-invariant processing, this model lacks an appropriate training mechanism. After some time, LeCun et al. [34] use stochastic gradient descent with backpropagation to train CNNs on a comparable network, extending the neocognitron's capabilities. In the years that followed, beginning with AlexNet [11], other highly successful entries to the ILSVRC [36] were

made. However, these approaches (and the ILSVRC) are geared toward image identification, and this knowledge needs to be translated to the object detection task.

We begin this subsection by discussing the R-CNN network [93] in Subsection 3.1.2, which serves as the foundation for a large number of object identification networks currently. Next, the introduction of the R-CNN, the following subsubsections discuss revisions and upgrades and how they addressed significant challenges in object detection.

3.1.2.1 R-CNN

Girshichk et al. (2014) [93] created a CNN-based object detection approach by combining the results of image recognition systems. Girshick et al. demonstrate that CNN-based object detection approaches beat rival systems that use HOG by over 30% compared to VOC 2007. The approach used by CNN to detect objects was the sliding window method. This method convolutionally transforms a specific region of an image and then slides the image until all image regions are searched and categorized. However, big CNNs make it more challenging to determine a specific position given the long strides and filter sizes involved in this operation. Additionally, because the regions are thoroughly examined for items, this method was somewhat slow. RCNN takes a different approach to the sliding-window method in that, rather than endlessly searching, it creates class-agnostic object suggestions from the input image and classifies those proposal areas using SVMs.

A fixed-size input is prepared from the suggested areas, hence Regions with CNN features. Additionally, [93] shown that transfer learning is a viable method for training big CNNs by pre-training them on ImageNet [90] and then fine-tuning them on the smaller PASCAL dataset. The algorithm is structured as follows. At the start, a selective search [96] is utilized to identify potential locations. Then, using the AlexNet network, fixed-size features (4096 features) are extracted from each of those regions. Regions that do not fit the CNN network's input parameters are twisted to the correct size (4096 in this case). Finally, the SVMs are fed fixed-size features to complete the object detection. According to the studies on the selective search algorithm, approximately 2000 regions are selected for an image, and the selected regions have a recall of 91 percent on ILSVRC and 98 percent on PASCAL, indicating that there is considerable room for improvement, as the maximum recall achievable by the entire network cannot exceed that of the region proposal stage. After class-specific SVMs assign scores to areas, bounding-box

regression is used to forecast a new bounding box. Let *x*, *y*, *w*, and *h* denote a proposal's bounding box's coordinates and width and height. Then, define *"x∗"*, *"y∗"*, *"w∗"*, *"h∗"* as the ground-truth bounding box's coordinates, width, and height, respectively. When necessary, the superscript *"i"* is used to express the ith suggestion.

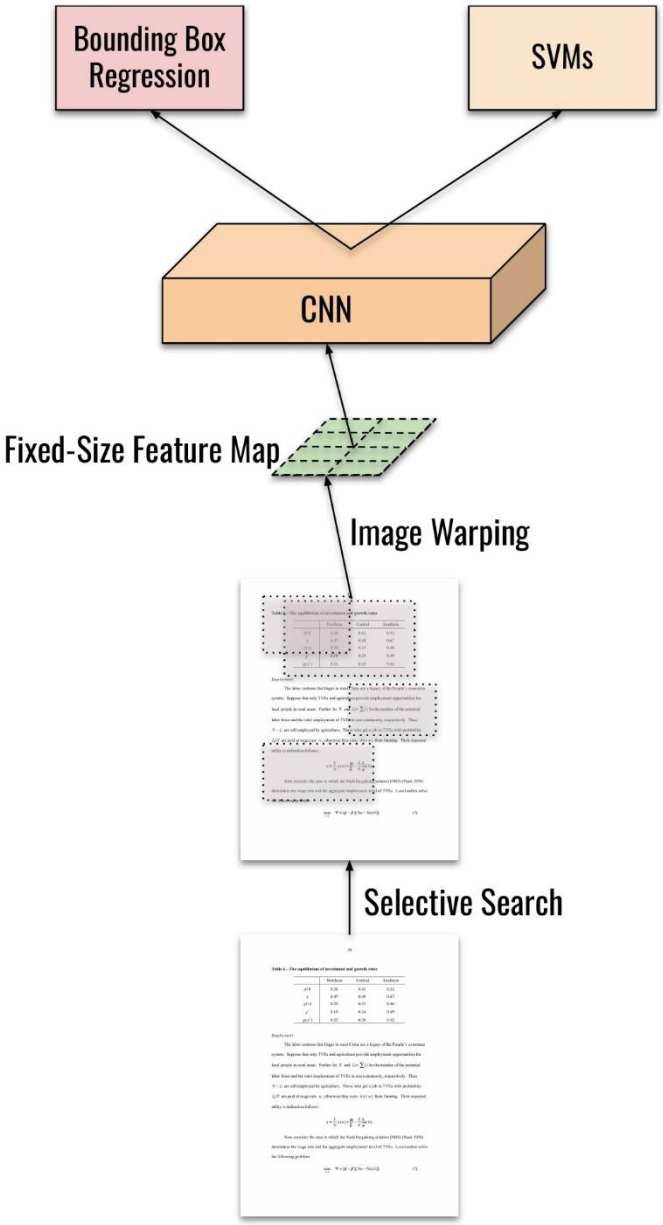


Figure 3.2: R-CNN method architecture, operations are shown for a single bounding box proposal.

The learning process can be summed to learning wθ in:

$$w_\theta = \underset{\hat{w}_\theta}{\operatorname{argmax}} \sum_{i}^{N} \left(t_\theta^i - \hat{w}_\theta^T \phi_5(P^i) \right)^2 + \lambda \|\hat{w}_\theta\|^2 \quad 3.5$$

θ is one of *x,y,w,h,* and $\varphi_5(P)$ is defined as the features recovered from the proposal *P* fifth (and final) convolution layer. To determine w_θ, we require the regression (t_θ) ground truth targets, which are defined as follows (for a single proposal and associated ground truth):

$$tx = (x* - x)/w \quad 3.6$$
$$ty = (y* - y)/h \quad 3.7$$
$$tw = log(w*/w) \quad 3.8$$
$$th = log(h*/h) \quad 3.9$$

Finally, the network is trained as an optimization problem effectively addressed in closed form using the least square loss function. On the PASCAL VOC 2007 test set [97], the R-CNN [93] approach gets 66 percent mAP and 62.4 percent mAP on the PASCAL VOC 2012 test set [98].

3.1.2.2 Fast-RCNN

Ross Girshick (2015) [80] advanced their earlier work, R-CNN, by proposing a set of enhancements. These advancements increased detection accuracy while also increasing detection speed. The author proposed a deep learning method using a single-stage training process, i.e., training the entire system end-to-end in this study. The speed increases indicated above are noteworthy because the R-CNN model with VGG16 takes around 47 seconds to process an image, but Fast-RCNN takes approximately 0.3 seconds without region proposal time [80]. As previously stated, R-CNN has some disadvantages. The training technique is not trivial; after training and fine-tuning a CNN, separate SVMs for each class are trained. Finally, we train bounding box regressors. Additionally, R-CNN is slow because the underlying CNN model is performed for each proposal, approximately 2000 per image.

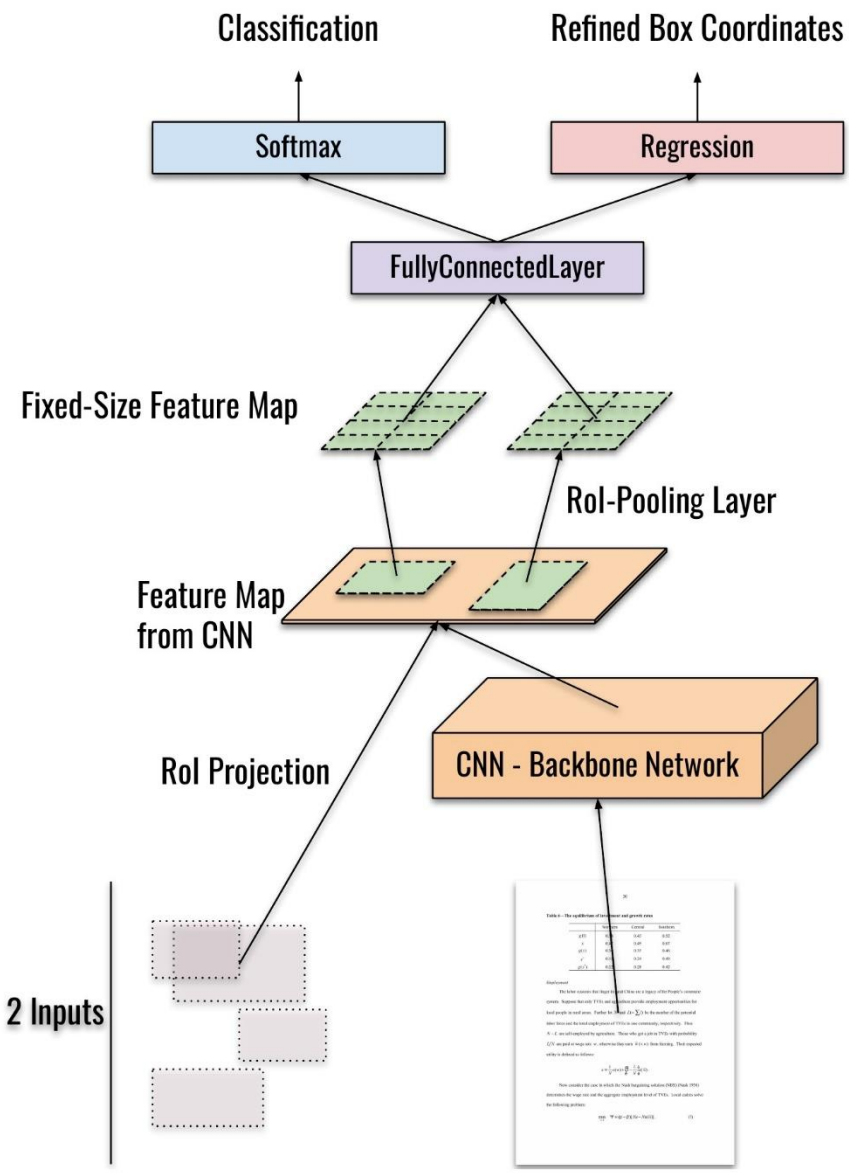


Figure 3.3: Fast-RCNN method architecture

Figure 3.3 illustrates the overall architecture of a Fast-RCNN system. Fast-RCNN requires two inputs: a picture and a set of proposals. As a result, the improvements occur not at the bounding box proposal step but also during the subsequent stages using convolution and SVMs. The image is first processed using the CNN feature extractor, which generates a feature map. Following that, a fixed-length feature vector is derived from the feature map for each proposal using the RoI pooling layer. Instead of running the image through CNN for each proposition, it is processed only once. The fixed-size feature vector is then fed into a fully

connected network divided into two parts: a softmax layer for classification and another layer for proposal box refinement. The RoI-pooling layer stated previously employs maximum pooling to convert features within any regions of interest into fixed-size feature vectors.

This fixed size is given to the network as hyper-parameters, assuming *Y* and *Y'*. Then the fixed-size small feature map would have the size YxY'. Each RoI is defined with four coordinates *(r, c, w, h)* that denote the top-left corner (r, c) and its width and height *(w, h)*. RoI max-pooling layer divides this RoI into equal length spatial sub-windows of the approximate size of *h/Y* and *w/Y'*. Following that, the standard max-pooling process is performed. Fast-RCNN also takes advantage of pre-trained ImageNet [90] convolutional networks in order to leverage successful CNNs. The aforementioned RoI-pooling layer takes the role of the feature extractor CNN's final pooling layer.

Fast-RCNN capitalizes on architectural advantages. They share the features retrieved for each RoI during training to avoid recalculating the fixed-size feature maps, which resulted in a training speedup of approximately 64 times [80]. Additionally, while R-CNN trained SVMs independently, Fast-RCNN trains the system as a whole by combining the box regression and classification branches with the CNN and RoI-pooling. As a result, the Fast-RCNN network outputs two layers. One softmax layer outputs a probability distribution per RoI, $p = (p_0, ..., p_K)$ over K + 1 classes where K is the count of categories in the dataset and 1 comes from the background represents the negative class. The

second layer outputs offset for bounding boxes, $t^u = (t^u_x , t^u_y , t^u_w, t^u_h)$ for $u \in K$. A ground Struth class k and a ground truth bounding box coordinate target v is associated with each training RoI. A multi-task loss L is then jointly trained for bounding box regression and RoI classification:

$$L(p, k, t^k, v) = L_{cls}(p, k) + \lambda[k \geq 1]L_{reg}(t^k, v) \quad 3.10$$

in which Lcls and Lreg are defined as the following:

$$L_{cls}(p, k) = -\log (p_k) \quad 3.11$$

$$L_{reg}(t^k, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}\left(t_i^k - v_i\right) \quad 3.12$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad 3.13$$

Girshick [80] chose smooth *L1* loss over the *L2* loss used in R-CNN because it is less susceptible to changes and outliers. Additionally to this advantage, the classical *L1* loss is not differentiable at the zero points. Circumvent this difficulty, and a smooth *L1* loss is converted to *L2* loss around zero. Because the regression targets are unbounded, hyperparameters such as the learning rate must be carefully controlled to avoid the exploding gradients problem. The Fast-RCNN [80] approach gets an mAP of 70% on the PASCAL VOC 2007 test set [97] and 68.4% on the PASCAL VOC 2012 test set [98].

3.1.2.3 Faster-RCNN

Ren et al. (2015) [51] significantly improved Fast-RCNN by creating Faster-RCNN. This novel solution tackles some significant concerns associated with deep learning-based object detection techniques. The research community employed selection search [96] as one of several strategies for area proposal. However, it was slow, at two photos per second, and because it is not learnable, we were unable to learn domain-specific features. Using the CPU-intensive selective search algorithm, Faster-RCNN proposes a deep convolutional neural network with shared convolutional layers. The new network, dubbed Region Proposal Network, was established (RPN). Since RPNs are composed of a few convolutional layers, they can be learned in their entirety. [51] also coined the term 'anchor' in the context of object detection, which enables it to recognize objects of varying scales and ratios.

The RPN accepts a feature map as input and produces area proposals with given objectness ratings. The RPN can simultaneously regress the coordinates of all potential regions and assign the regions an objectness score. The objectness score indicates whether the region is considered part of the background class or user classes. To accelerate the process, RPN shares layers with the convolutional network of Fast-RCNN (backbone network). For instance, if the VGG-16 [40] backbone is employed, a feature map from the 13th shared convolutional layer is taken, and a tiny network is slid over it. The input to this little network is $n * n$. The RPN output is subsequently transformed into a 512-dimensional vector. Ren et al. propose that we use n = 3, which we do. The recovered feature vector (512-d) is processed by two fully connected layers: the objectness and regression layers. Figure 3.4 depicts the whole RPN architecture. As discussed previously, the anchors enable the RPN to generate proposals at various scales and ratios. The system requires an input of *k*, the maximum number of recommendations for each

site. As a result, the regression layer produces 4k outputs with the coordinates, whereas the classification layer produces 2k outputs with the objectness score. These k proposals are constructed using anchors. Anchors are scaled differently (3 by default) and have varied ratios (2:1, 1:2, and 1:1 by default), resulting in $k = 9$ anchors placed at each position.

All of these anchors are set in each sliding window slice, and objectness scores are assigned. Thus, if the feature map is $W * H$ in size, the total number of anchors is $W * H * k$. Scaling is included to take advantage of the feature pyramid structure, comparable to FPNs [81]. When scaling the image and repeating the detection procedure, resizing the anchors adds less computational effort. However, considerable increases can be found when utilizing anchors with varied scales, which results in an mAP gain of roughly 2% on the PASCAL VOC 2007. Anchors are given a score of 1 during network training if they have an IoU of 0.7 or greater with any ground truth box. Otherwise, o The loss function is then simply a modification of the Fast-RCNN loss function. Assume that i is an index for an anchor in a mini-batch with a probability of being an object equal to p_i.

Also, the ground truth label for the anchors is denoted as p^*i. Lastly, t_i represents the four coordinates that define a predicted bounding box, and t^*_i is the coordinates for the corresponding ground truth bounding box. Then the multi-task loss L is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad 3.14$$

The loss L_{cls} is the same as Eq. 3.11 as defined in the Fast-RCNN chapter. The regression loss is smooth *L1* loss (Eq. 3.13), adapted from the Fast-RCNN [80] and defined as:

$$L_{\text{reg}}(t_i, t_i^*) = \text{smooth}_{L_1}(t_i - t_i^*) \quad 3.15$$

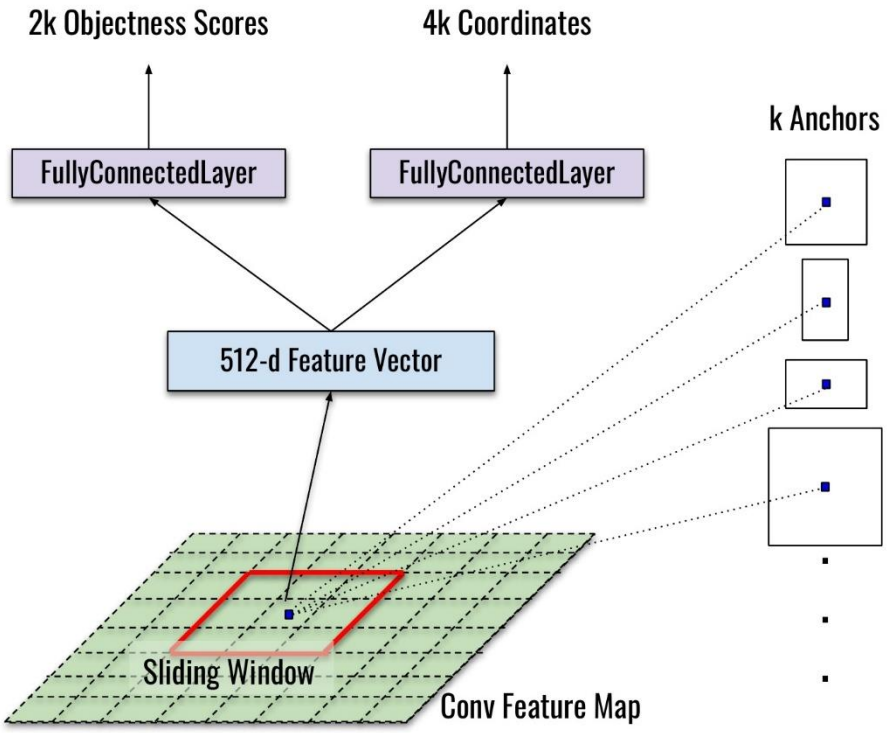


Figure 3.4: Region Proposal Network

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

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

$$t_x = \frac{(x - x_a)}{w_a}; t_y = \frac{(y - y_a)}{h_a}$$
$$t_w = \log\left(\frac{w}{w_a}\right), t_h = \log\left(\frac{h}{h_a}\right)$$
$$t_x^* = \frac{(x^* - x_a)}{w_a}, t_y^* = \frac{(y^* - y_a)}{h_a} \qquad 3.16$$
$$t_w^* = \log\left(\frac{w^*}{w_a}\right), t_h^* = \log\left(\frac{h^*}{h_a}\right)$$

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

The Faster-RCNN method's overall architecture is depicted in Figure 3.5. We used the Faster-RCNN approach with a pre-trained VGG-16 and ResNet-50 backbone on the MS COCO dataset [82]. This option is because the Faster-RCNN approach is widely used and has state-of-the-art performance and speed. Additionally, it is applied to a variety of domains and has aided in the recognition of tables and table structures [2, 3], pedestrian detection [75], and instance retrieval [74]. The model achieves 78.8 percent mAP on the PASCAL VOC 2007 test set [97], and 75.9 percent mAP on the PASCAL VOC 2012 test set [98], outperforming previous methods and claiming the throne in object detection research while improving speed by 0.2 seconds per image, including proposals using the VGG-16 backbone network. Previous researchers have applied the Faster-RCNN architecture to the domain of document analysis [27, 50, 52, 57]. The data indicate that this network architecture is well-suited for this area, and hence we include it in our comparisons of various backbone networks.

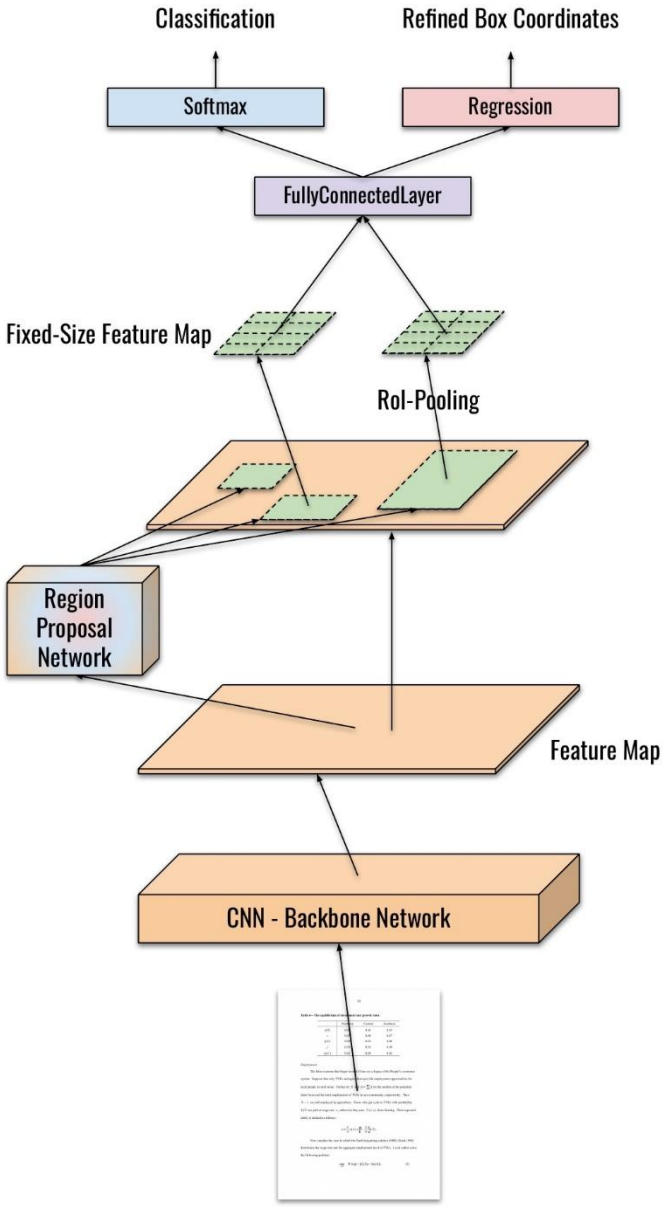


Figure 3.5: Faster-RCNN architecture

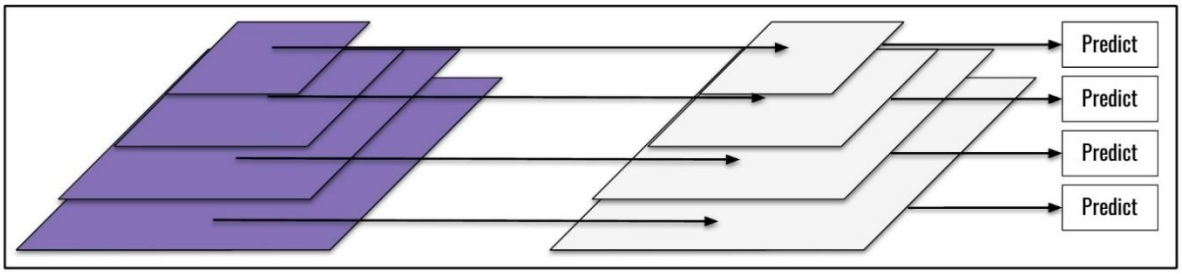
3.1.3 Feature Pyramid Networks

Detecting objects at various scales has always been a challenge in computer vision, and numerous techniques have been presented. One of them is manually scaling images and conducting detection at various scales, which is based on [72](Figure 3.6a) and is a typical technique in hand-crafted rule-based image processing methods [67]. This method makes the procedures scale-invariant at the performance cost, as the detection is conducted multiple times.

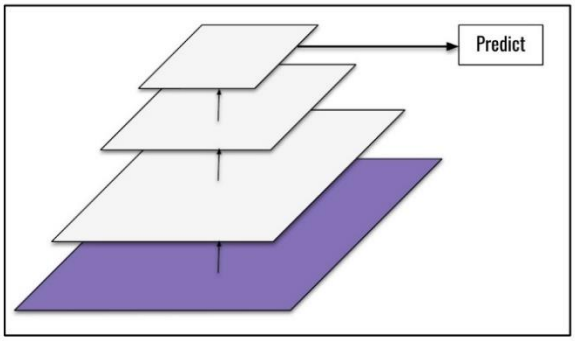
While convolutional networks have been proven to be more resistant to object scale changes [80, 51] (Figure 3.6b), a closer examination reveals that pyramid-like architectures can achieve more significant performance benefits [81]. Detection at various scales is advantageous for detecting objects at various scales. Tables, for example, can be spotted at various scales, whereas smaller items such as rows can only be detected at a larger scale, as [3] demonstrates. While image pyramids are the most natural technique to establish scale invariance for models, they are not the only way. Because CNN's already calculate feature pyramids in their forward pass, which is essentially a pyramid-like shape with hierarchical features at each level. SSD [55] is a method that utilized CNN's hierarchical characteristics and detected at various levels (Figure 3.6c), but they added more layers to complete the pyramid and avoided using lower layers. Lin et al. (2017) [81] proposed a new method called FPN that takes advantage of the CNNs' inherent feature pyramids by combining semantically low-level and semantically high-level features to mimic the image pyramids scheme and form a top-down architecture capable of performing detections at all levels, as illustrated in Figure 3.6d.

Lin et al. [81] demonstrate that FPNs significantly improve recall on an object detection task. In terms of numerical performance, FPN design increased Average Recall (AR) by eight and Average Precision (AP) by 2.3 or 3.8 for COCO-style [82] or PASCAL-style [95] precision calculation, respectively. The FPN is separated into two sections: bottom-up and top-down. The bottom-up pathway is represented by the convolutional networks' frequent feed-forward operations. Because many convolutional layers do not alter the input size (e.g., the convolution layers in the conv4 block of the ResNet [39] model), they are regarded to be at the same level in the bottom-up route. Then, a bottom-up feature pyramid is formed using the final convolutional layers of each of these tiers. For example, feature maps are extracted from the last layers of the conv2, conv3, conv4, and conv5 blocks in the ResNet models. The top-down route is formed by extracting feature maps at specified layers starting from the last layers of the convolutional network and progressing downward in the network. Laterally coupled to the top-down path, these feature maps are then blended with the other feature maps. Each layer along the path is upscaled using the closest neighbour technique before being transferred to the bottom layer. The lower layer is then blended with the upper layer simply by adding the features. The FPN architectural method is depicted in Figure 3.6d. The bottom-up path is depicted on the left, while the top-down path is depicted on the right.

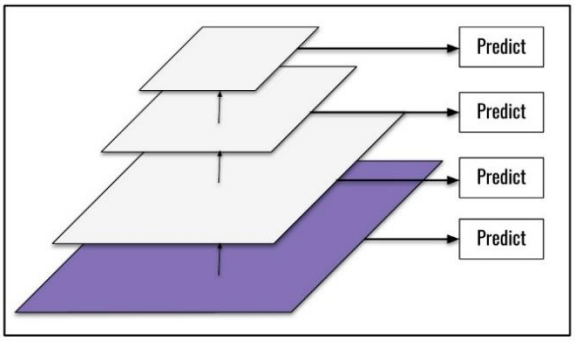
Since our challenge contains both wide and short items (rows and columns), detecting various scales might aid us. According to the studies in [81], the Faster-RCNN model with FPN outperforms the COCO test set somewhat better [82]. We also tested with FPN, and our prior research in [3] indicated that the model with FPN (and FCN) increased the AP and AR by 4-6 per cent in the publicly available dataset.



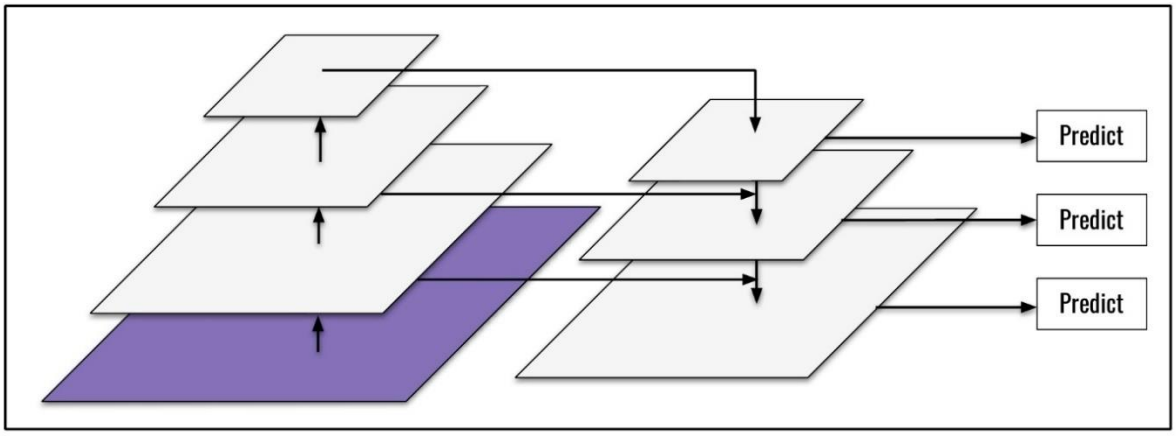
(a) Image scaling



(b) Single feature map



(c) Hierarchical features



(d) Feature pyamid network

Figure 3.6: Feature pyramid networks, Colored boxes represent input images and white boxes for feature maps.

3.2 Semi-Supervised Learning for Object Detection

We present a basic SSL framework based on Self-Training (through pseudo label) and Augmentation-driven Consistency regularisation. First, we use a stage-based training approach with Noisy Child [71] due to its scalability and flexibility. Following that, at least two training phases are required: first, we tr ain a parent model using all available labelled data; second, we train both the child and parent models using labelled and unlabeled data. Second, we employ a high confidence threshold for FixMatch-inspired confidence-based thresholding to manage the quality of pseudo labels composed of bounding boxes and their class labels in object detection. The summarization of the steps involved in training is explained in further sub-topics.

3.2.1 Parent Model Training

Parent model trained on all available labelled datasets. We base our formulation on the Faster RCNN [66], which has been demonstrated to be a highly representative detection framework. On top of the shared backbone network, the faster RCNN has a classifier (CLS) and a region proposal network (RPN) heads. Each head contains two modules: area classifiers (for the CLS head, this is a (K+1)-way classifier; for the RPN head, this is a binary classifier) and bounding box regressors (REG). To keep things simple, we present the supervised and unsupervised losses of the Faster RCNN for the RPN head. The following is a description of the supervised loss:

$$\ell_s(x, \mathbf{p}^*, \mathbf{t}^*) = \sum_i \ell_s(x, p_i^*, t_i^*) \\ = \sum_i \left[\frac{1}{N_{\text{cls}}} \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{reg}}} \mathcal{L}_{\text{reg}}(t_i, t_i^*) \right] \quad 3.17$$

Where *i* is the index of a mini-batch anchor. "p_i" is the predictive likelihood of an anchor being positive, and t_i denotes the anchor's four-dimensional coordinates. "p^*_i" denotes the binary label of an anchor to ground-truth boxes, while t^*_i denotes the box "*i*" ground-truth coordinates for all "$p^*_i = 1$".

3.2.2 Generating Pseudo Labels

Utilize the trained parent model to generate pseudo labels for unlabeled images (i.e., bounding boxes and associated class labels). To produce pseudo labels, we make a test-time inference of the object detector from the parent model. The production of pseudo labels entails the forward pass of the backbone, RPN, and CLS networks and post-processing such as non-maximum suppression (NMS). This contrasts with standard classification methods, which generate the confidence score from the raw prediction probability. We use the score of each returning bounding box following NMS, which averages the anchor box prediction probabilities—using box predictions following NMS benefits over using raw predictions (before to NMS), as it eliminates redundant detection. However, as illustrated in Figure 3.7, this does not filter out boxes in incorrect locations. To further exclude potentially incorrect pseudo boxes, we employ confidence-based thresholding [68,71].

3.2.3 Unsupervised Loss

Calculate both unsupervised and supervised loss in order to train a detector. We determine *"$q*i$"* a binary label for an anchor I about pseudo boxes, for all anchors, given an unlabeled image *"x"* a collection of anticipated bounding boxes, and respective area proposal confidence ratings. Notably, the primary threshold mechanism *"w"* is applied to *"$q*i$"* via the CLS head, resulting in a value of "1" if the anchor is connected with any pseudo boxes whose CLS prediction confidence scores of the parent model are above the threshold "τ" and 0 otherwise (i.e., treated as background). Assume that "s∗" is the box coordinates of pseudo boxes. Then, STAC's unsupervised RPN loss is expressed as $\ell_u(\mathcal{A}(x_u, \mathbf{s}^*), \mathbf{q}^*) = \ell_s(x_{\mathcal{A}}, \mathbf{q}^*, \mathbf{s}^*_{\mathcal{A}})$. Significant data augmentation is applied to an unlabeled image x, resulting in the string "X_A". Since some transformation operations (e.g., global geometric transformation [78]) are not invariant to the box coordinates, the operations "A" are also applied to the pseudo box coordinates, generating "s^*_A". Finally, the RPN is trained by simultaneously minimizing two losses:

$$\ell = \ell_s(x_s, \mathbf{p}^*, \mathbf{t}^*) + \lambda_u \ell_u(\mathcal{A}(x_u, \mathbf{s}^*), \mathbf{q}^*) \qquad 3.18$$

The overall model adds two new hyperparameters τ and λ_u. We found that "τ = 0.9" and "$\lambda_u \in [1, 2]$" work well in experiments. Notably, the consistency-based SSL object recognition approach described in [83] necessitates a complex weighting schedule for "λ_u" including temporal ramp-up and ramp-down. Rather than that, our system displays effectiveness with a simple constant schedule by utilizing a strong data augmentation technique and confidence-based thresholding.

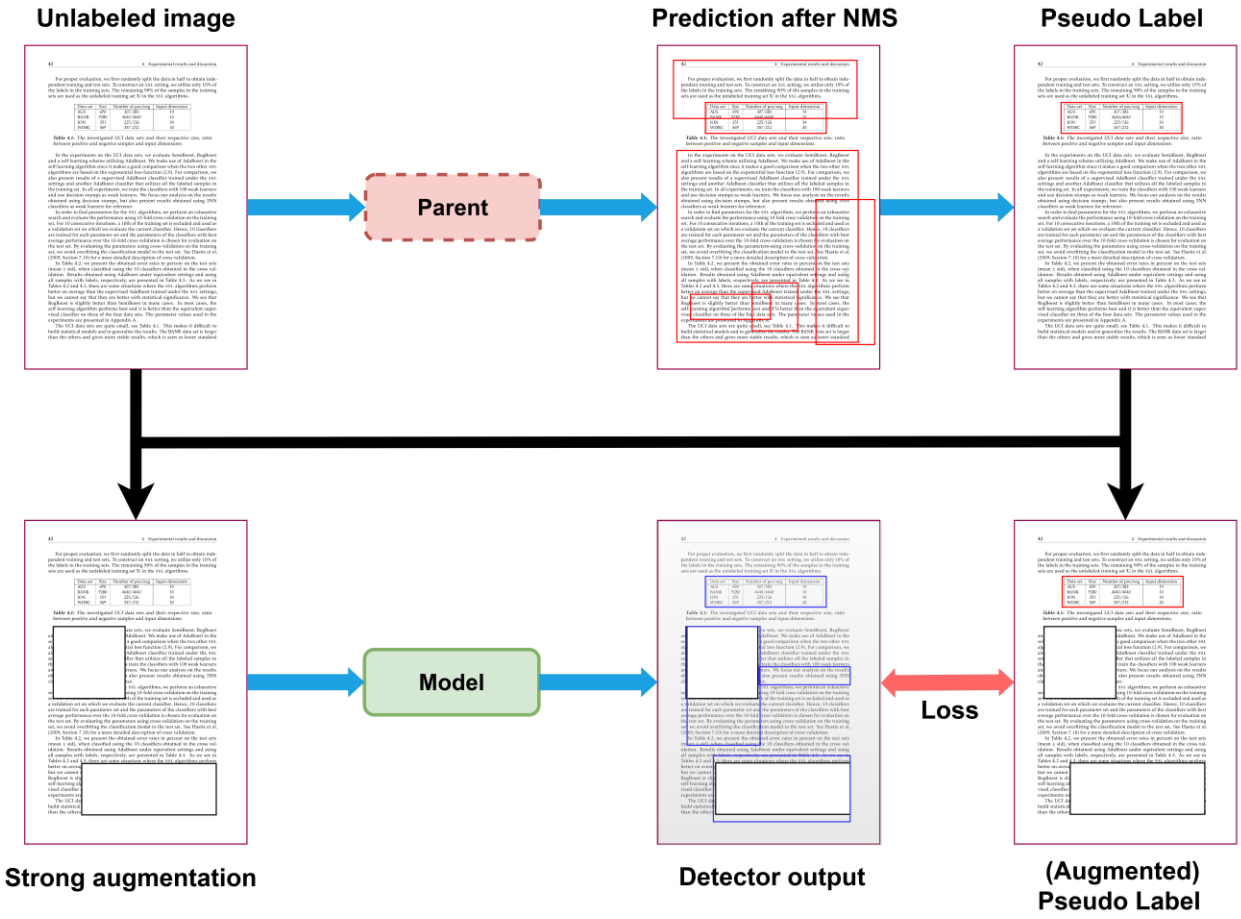


Figure 3.7: SSL framework for object detection

We produce pseudo labels (i.e., bounding boxes and associated class labels) for unlabeled data using test-time inference techniques, including NMS, with the instructor model trained on labelled data. We compute unsupervised loss in the presence of pseudo labels with confidence scores greater than a predefined threshold. The strong augmentations are used to ensure that the augmentations are consistent throughout the model training process. When global geometric transformations are performed, target boxes are augmented.

3.2.4 Data Augmentation Strategy

When global geometric alterations are implemented, apply strong data augmentations to unlabeled images and proportionally augment pseudo labels (i.e., bounding boxes). Robust data augmentation is critical for the success of consistency-based SSL approaches such as UDA [70] and FixMatch [68]. While the augmentation technique has been extensively investigated for supervised and semi-supervised image classification [61, 68, 70, 77, 79], little effort has been made to date for object identification. We extend [77]'s RandAugment for object detection by

using the recently proposed augmentation search space (e.g., box-level alteration) and the Cutout [65]. We investigate several transformation options and identify a set of effective combinations. Each operation has a magnitude that determines the degree of strength augmentation.

- Global colour transformation (C): The methods described in [79] are employed, as are the specified magnitude ranges for each operation.
- Global geometric transformation (G): [79] uses geometric transformation operations such as x-y translation, rotation, and x-y shear.
- Box-level transformation [78] (B): Three transformation procedures similar to those employed in global geometric transformations but with lower magnitude ranges.

We execute the following transformation procedures sequentially to each image. First, we apply one of the procedures from C. Second, we perform one of the operations on the data sampled from G or B. Finally, we apply Cutout to numerous random spots across a picture to avoid a trivial solution when used exclusively within the bounding box.

CHAPTER 4: RESULTS AND DISCUSSION

We propose semi-supervised based solutions to find tables in arbitrarily styled documents, be it research papers, magazines, journals, newspapers, web pages, and more. The document format is in digital image format because every other document format can be converted into an image, allowing our system to take any document as input. The proposed table detection model also works on scanned and possible tilted images that result from scanning progress, and we achieve almost state-of-the-art results on a publicly available dataset TableBank. The detections by this model are accurate, and speed-wise it competes with other methods we present. Since the annotated dataset for the table detection task is lacking, the research community use existing public datasets and benefit from transfer learning capabilities of the deep learning methods to adapt the models to table detection domain. We propose that unlike how public datasets and their training sets are annotated, annotating all table-like areas, which we call tabular areas, increases the overall recall and precision with a trade-off of an increased number of false positives. Tabular regions include the table of content pages, some form of lists that has several columns and rows, actual tables or sometimes figures as well, or in other words, everything that looks like tables. In training the models, even though we use transfer learning to adapt pre-trained backbone networks in our models to the document domain, we still require more data.

4.1 Dataset and Experimentation

Minghao et al. [60] recognised the table community's demand for huge datasets in early 2019 and produced TableBank, a dataset including 278,582 tagged photos with tabular information. This dataset was produced by crawling through online documents in the.docx format. Another source of data for this dataset is LaTeX documents obtained from arXiv's database. The 78,399 photos were taken from Microsoft Word documents, while the 200,183 images were extracted from latex documents. According to the dataset's publishers, this addition will enable academics to harness the potential of deep learning and finetuning approaches. The size of training, testing, and validation are 260582, 10000, and 8000, respectively. TableBank consists of multiple types of tables, without border, with border and landscape type. Figure 4.1.

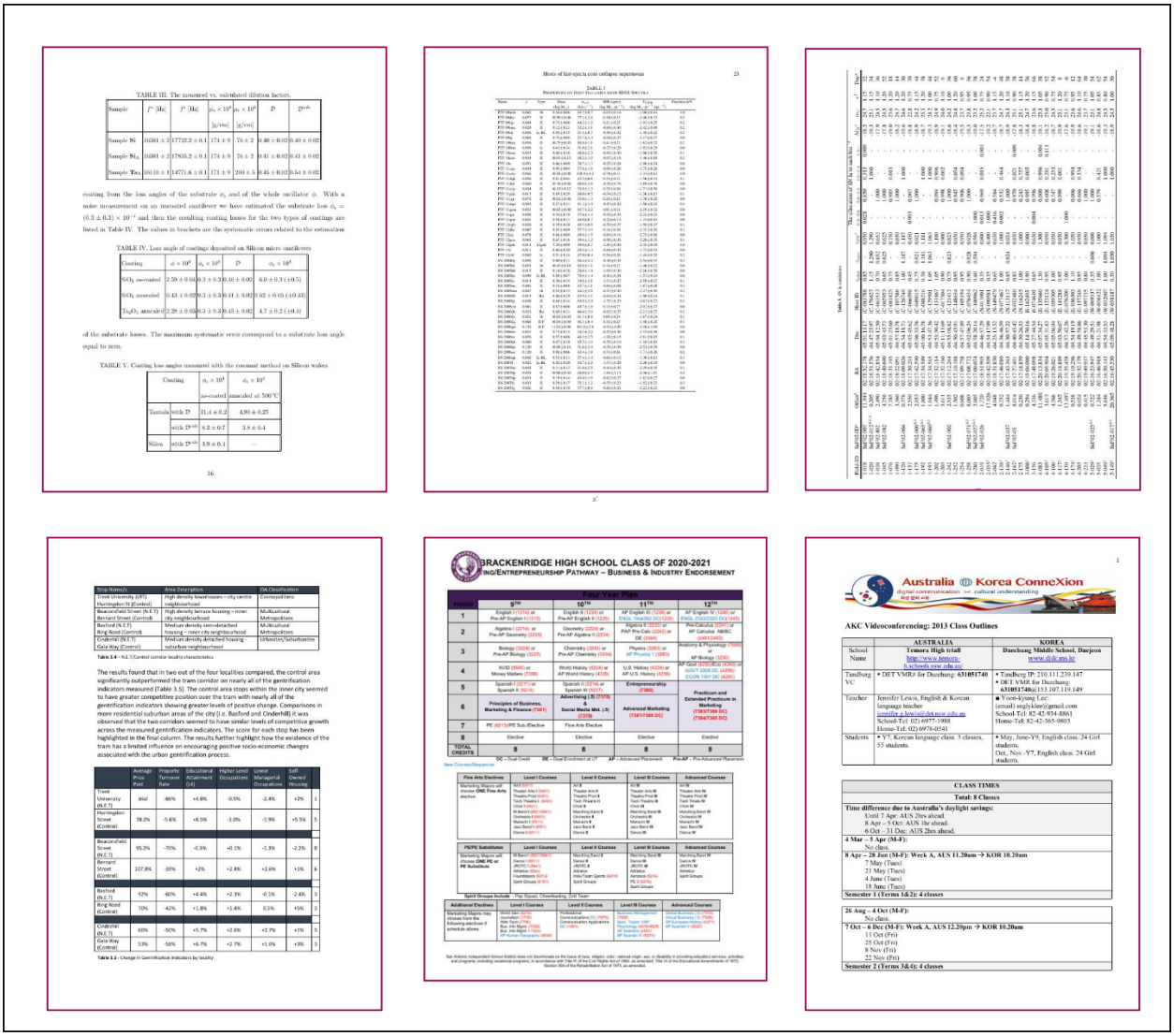


Figure 4.1: TableBank tables types

We experiment with two different SSL configurations. First, we randomly choose 1, 2, 5, and 10% of labelled training data and utilise the remainder as an unlabeled set. We generate five data folds for these tests. The 1% protocol contains roughly 2.6k labelled images chosen at random from TableBank's labelled set. 2% protocol contains additional ∼2.6k images, and 5, 10% protocol datasets are constructed similarly. Second, following [38], We define a labelled set as an entire set of labelled training data and an unlabeled set as extra unlabeled data. Notably, the first experiment assesses the model's efficacy when just a few labelled examples are provided.

However, the second protocol assesses the possibility of improving the state-of-the-art object detector using unlabeled data in addition to the large-scale labelled data currently available. We report on the mAP distribution over table classes. On a test set of 10,000 images, the detection performance is evaluated in precision, recall, and F1 score at an IoU of 0.5.

4.2 Implementation and Results

Our version is based on Tensorpack's Faster RCNN and FPN libraries [29]. Our object detection models are built on the ResNet-50 [24] backbone. Unless otherwise specified, the ImageNet pre-trained model initializes the network weights at all phases of training. Due to the complexity of training the object detector, we use the default learning settings for all our trials except the learning schedule. The majority of our experiments employ a rapid learning schedule. We find that when more labelled training data and more elaborate data augmentation procedures are applied, longer training considerably improves the model's performance. Two new hyperparameters have been introduced: τ for the confidence threshold and λu for the unsupervised loss. For all tests, we use $\tau = 0.9$ and $\lambda u = 2$. Due to the lack of research on deep semi-supervised learning of visual object detectors for table recognition, we compare our technique to supervised models (i.e., models trained on labelled data from TableBank) for various experimental procedures utilizing various data augmentation strategies. The results are summarised in Table 1. For protocols of 1, 2, 5, and 10%, we train models using a fast learning schedule and report the F1-score over five data folds, as well as the standard deviation.

The complete training and testing data set have been made public. We calculate precision, recall, and F1 for table detection in the same method as in [52], where the metrics for all documents are derived by adding the area of overlap, prediction, and ground truth. The definition is defined as follows:

$$\text{Precision} = \frac{\text{Area of Ground truth regions in Detected regions}}{\text{Area of all Detected table regions}}$$

$$\text{Recall} = \frac{\text{Area of Ground truth regions in Detected regions}}{\text{Area of all Ground truth table regions}}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

To begin, we confirm [79]'s findings that the RandAugment significantly improves the supervised learning performance of a detector on a 10000 image test dataset, 0.21 f1-score at 5% protocol and 0.30 f1-score at 10% protocol, when compared to the supervised baselines with default data augmentation of resizing and horizontal flipping.

Table 4-1: Comparison in scores for different methods on TableBank.

Models	**Training Dataset Count**	**Method**	**Precision**	**Recall**	**F1**
Faster R-CNN ResNet-50 FPN	2605 (1 %)	Supervised	0.186	0.283	0.224
	5210 (2 %)	Supervised	0.268	0.382	0.315
	13025 (5 %)	Supervised	0.439	0.479	0.458
	26050(10 %)	Supervised	0.550	0.640	0.591
	260582(100 %)	Supervised	**0.964**	**0.904**	**0.933**
Faster R-CNN ResNet-50 FPN	2605 (1 %)	Semi-Supervised	0.353	0.342	0.348
	5210 (2 %)	Semi-Supervised	0.483	0.469	0.476
	13025 (5 %)	Semi-Supervised	0.686	0.667	0.677
	26050(10 %)	Semi-Supervised	0.908	0.886	0.897

The results in Table 4-1 indicate that ResNet-50 is an appropriate backbone for this task in terms of speed and f-score. Because the primary objective is supply chain optimization, this procedure requires millions of documents, making speed a critical aspect of the decision-making process. The term 'Scratch' refers to when the network is trained from scratch (with randomly initialised weights), and the results indicate that employing pre-trained feature extractors improves detection performance. However, the results are close, and with additional enhancements to the training dataset (such as increasing the number of data points), pre-trained weights may be unnecessary. The trials are carried out using the ResNet50 backbone feature extractor in conjunction with the Faster-RCNN model. Figure 4.2 illustrates table detections from a TableBank dataset. Our results demonstrate that even with only 10% data, it outperforms supervised models trained on 10% data and is comparable to supervised models trained on 100% data. It is useful for industrial applications since it eliminates the need for labelled data, an expensive, time-consuming operation.

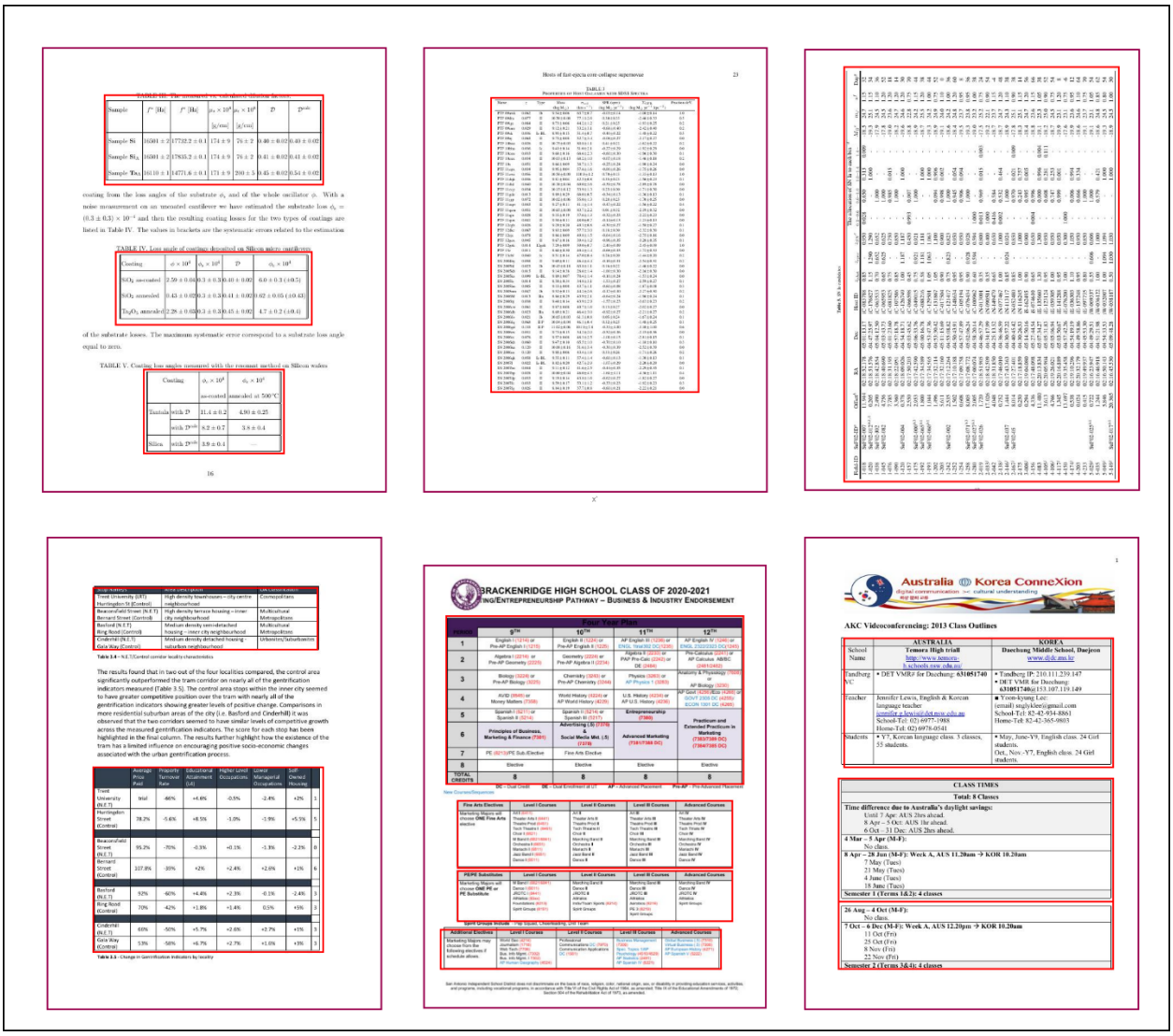


Figure 4.2: Examples of TableBank detection results

CHAPTER 5: CONCLUSION

The table extraction task can potentially save thousands of precious human hours that would otherwise be spent extracting data from tables. Despite the absence of pre-or post-processing, our models outperformed state-of-the-art models on TableBank datasets. We conclude that our higher precision and recall are justified by the increasing diversity and volume of data. While SSL has made tremendous progress in categorization, label-efficient training for tasks requiring a high labelling cost is challenging. By utilizing lessons learned from SSL approaches for classification, we offer a simple (just two easily tuneable hyperparameters) and practical (2 label efficiency in low-label regime) SSL framework for object detection. The simplicity of our solution allows for further research aimed at resolving SSL for object detection. The suggested framework is adaptable to various configurations, including soft labels for classification loss, other detector frameworks to Faster RCNN, and alternative data augmentation methodologies. While our approach achieves an impressive performance improvement without considering confirmation bias [18], it may become troublesome when used with a detection system that employs a more robust kind of hard negative mining [9] as noisy pseudo labels can be overused. Further research into learning with noisy labels, confidence calibration, and uncertainty estimation in the context of object detection are just a few critical areas to investigate in order to improve SSL's performance for table detection further.

CHAPTER 6: REFERENCES

[1] W. B. I. E. D. D. D. Group, World development indicators. World Bank, 1978.

[2] 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.

[3] 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.

[4] E. Kara, M. Traquair, M. Simsek, B. Kantarci, and S. Khan, "End-to-end system design for deep learning-based tabular information discovery in datasheets," IEEE Access Special Section: 'AI-Driven Big Data Processing: Theory, Methodology, and Applications', 2019.

[5] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, pp. 2961–2969, 2017.

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

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

[8] 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.

[9] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard ex ample mining. In CVPR, 2016.

[10] F. Shafait and R. Smith, "Table detection in heterogeneous documents," pp. 65–72, 2010.

[11] 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.

[12] T. Kieninger and A. Dengel, "Applying the T-recs table recognition system to the business letter domain," in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, vol. 2001-Janua, pp. 518–522, 2001.

[13] E. Oro and M. Ruffolo, “PDF-TREX: An approach for recognizing and extracting tables from PDF documents,” in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, pp. 906–910, IEEE, 2009.

[14] D. N. Tran, T. A. Tran, A. Oh, S. H. Kim, and I. S. Na, “Table Detection from Document Image using Vertical Arrangement of Text Blocks,” International Journal of Contents, vol. 11, no. 4, pp. 77–85, 2016.

[15] 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 Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, pp. 779–783, IEEE, 2011.

[16] 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.

[17] R. M. Haralick, S. R. Sternberg, and X. Zhuang, “Image Analysis Using Mathematical Morphology,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, pp. 532–550, jul 1987.

[18] Zixing Zhang, Fabien Ringeval, Bin Dong, Eduardo Coutinho, Erik Marchi, and Bjorn Sch ¨ uller. Enhanced semi¨ supervised learning for multimodal emotion recognition. In ICASSP, 2016.

[19] 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.

[20] T. Kieninger and A. Dengel, “The T-Recs Table Recognition and Analysis System,” pp. 255–270, Springer, Berlin, Heidelberg, 1999.

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

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

[23] S. Dey, J. Mukherjee, and S. Sural, “Consensus-based clustering for document image segmentation,” International Journal on Document Analysis and Recognition, vol. 19, no. 4, pp. 351–368, 2016.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[25] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. arXiv preprint. arXiv:2005.04757, 2020.

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

[27] M. Kerwat, R. George, and K. Shujaee, "Detecting Knowledge Artifacts in Scientific Document Images Comparing Deep Learning Architectures," in 2018 5th International Conference on Social Networks Analysis, Management and Security, SNAMS 2018, pp. 147–152, IEEE, 2018.

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

[29] Yuxin Wu et al. Tensorpack. https://github.com/ tensorpack/, 2016.

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

[31] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," IEEE ASSP Magazine, vol. 3, no. 1, pp. 4–16, 1986.

[32] 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.

[33] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, pp. 193–202, Apr 1980.

[34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Comput., vol. 1, pp. 541–551, Dec. 1989.

[35] B. Scholkopf and A. J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2001.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F. F. Li, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, 09 2014.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9, 2015.

[38] Peng Tang, Chetan Ramaiah, Ran Xu, and Caiming Xiong. Proposal learning for semi-supervised object detection. arXiv preprint arXiv:2001.05086, 2020.

[39] 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.

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

[41] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," IEEE Transactions on Multimedia, vol. 20, no. 4, pp. 985– 996, 2018.

[42] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative cnn video representation for event detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1798–1807, 2015.

[43] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio.," SSW, vol. 125, 2016.

[44] 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, '.

[45] H. T. Ng, C. Y. Lim, and J. L. T. Koo, "Learning to recognize tables in free text," in Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics -, (Morristown, NJ, USA), pp. 443–450, Association for Computational Linguistics, 2007.

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

[47] D. P. Lewis, T. Jebara, and W. S. Noble, “Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure,” Bioinformatics, vol. 22, pp. 2753–2760, nov 2006.

[48] 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.

[49] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almaz`an, and L. P. de las Heras, “Icdar 2013 robust reading competition,” in Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, ICDAR ’13, (Washington, DC, USA), pp. 1484– 1493, IEEE Computer Society, 2013.

[50] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, “DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images,” in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2018.

[51] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, pp. 91–99, 2015.

[52] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, “Table Detection Using Deep Learning,” in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 771–776, 2017.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A Saliency-based Convolutional Neural Network for Table and Chart Detection in Digitized Documents," pp. 1–13, 2018.

[59] S. Arif and F. Shafait, "Table Detection in Document Images using Foreground and Background Features," in 2018 Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8, IEEE, dec 2018.

[60] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: Table benchmark for image-based table detection and recognition. arXiv preprint arXiv:1903.01949, 2019.

[61] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Ku rakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remix match: Semi-supervised learning with distribution matching and augmentation anchoring. In ICLR, 2020.

[62] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In NeurIPS, 2019.

[63] Samuli Laine and Timo Aila. Temporal ensembling for semi supervised learning. In ICLR, 2017.

[64] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In ICML Workshops, 2013.

[65] Terrance DeVries and Graham W Taylor. Improved regular ization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

[66] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NeurIPS, 2015.

[67] 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.

[68] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685, 2020.

[69] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In NeurIPS, 2017.

[70] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for con sistency training. arXiv preprint arXiv:1904.12848, 2019.

[71] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Self-training with noisy student improves imagenet clas sification. arXiv preprint arXiv:1911.04252, 2019.

[72] 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.

[73] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In ICDAR, 2003.

[74] A. Salvador, X. Gir´o-i Nieto, F. Marqu´es, and S. Satoh, "Faster r-cnn features for instance search," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–16, 2016.

[75] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?," in European conference on computer vision, pp. 443–457, Springer, 2016.

[76] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle consistent adversarial networks. In ICCV, 2017.

[77] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasude van, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In CVPR, 2019.

[78] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data aug mentation strategies for object detection. arXiv preprint arXiv:1906.11172, 2019.

[79] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no sep arate search. arXiv preprint arXiv:1909.13719, 2019.

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

[81] T.-Y. Lin, P. Doll´ar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125, 2017.

[82] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in European conference on computer vision, pp. 740–755, Springer, 2014.

[83] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object de tection. In NeurIPS, 2019.

[84] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[85] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proceedings of the 32Nd International Conference on International Conference on Machine Learning Volume 37, ICML’15, pp. 448–456, JMLR.org, 2015.

[86] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814, 2010.

[87] P. Petersen and F. Voigtlaender, “Optimal approximation of piecewise smooth functions using deep relu neural networks,” Neural Networks, vol. 108, pp. 296–330, 2018.

[88] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. icml, vol. 30, p. 3, 2013.

[89] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in European conference on computer vision, pp. 818–833, Springer, 2014.

[90] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A largescale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, Ieee, 2009.

[91] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) Volume 1 Volume 01, CVPR ’05, (Washington, DC, USA), pp. 886–893, IEEE Computer Society, 2005.

[92] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vision, vol. 60, pp. 91–110, Nov. 2004.

[93] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587, 2014.

[94] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, “How does the brain solve visual object recognition?,” Neuron, vol. 73, no. 3, pp. 415–434, 2012.

[95] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International journal of computer vision, vol. 88, no. 2, pp. 303–338, 2010.

[96] 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.

[97] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge, 2007.

[98] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge, 2012.

thesis_v3_plagiarism

ORIGINALITY REPORT

4%	2%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	1%
2	Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017 Publication	<1%
3	Submitted to Monash University Student Paper	<1%
4	"Computer Vision", Springer Science and Business Media LLC, 2017 Publication	<1%
5	Lecture Notes in Computer Science, 2014. Publication	<1%
6	ndltd.ncl.edu.tw Internet Source	<1%
7	Baoxiang Jiang, Jingbo Xia, Shiyan Li. "Few training data for Objection Detection", Proceedings of the 2020 4th International	<1%

	Conference on Electronic Information Technology and Computer Engineering, 2020 Publication	
8	Submitted to University of St. Gallen Student Paper	<1 %
9	towardsdatascience.com Internet Source	<1 %
10	Diping Song, Yu Qiao, Alessandro Corbetta. "Depth driven people counting using deep region proposal network", 2017 IEEE International Conference on Information and Automation (ICIA), 2017 Publication	<1 %
11	"Computer Vision – ACCV 2018", Springer Science and Business Media LLC, 2019 Publication	<1 %
12	Submitted to Coventry University Student Paper	<1 %
13	"Computer Vision – ECCV 2016", Springer Nature, 2016 Publication	<1 %
14	Submitted to Cummins College of Engineering for Women, Pune Student Paper	<1 %
15	Submitted to Leeds Metropolitan University Student Paper	<1 %

16	Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, Xiangyang Xue. "Object Detection from Scratch with Deep Supervision", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020 Publication	<1 %
17	opensiuc.lib.siu.edu Internet Source	<1 %
18	Submitted to City University Student Paper	<1 %
19	epdf.pub Internet Source	<1 %
20	pic.org.kh Internet Source	<1 %
21	Wan Ding, Xinguo Yu, Nan Ye. "Goal Detection for Broadcast Basketball Video Using Superimposed Texts", Proceedings of International Conference on Internet Multimedia Computing and Service - ICIMCS '14, 2014 Publication	<1 %
22	"Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020 Publication	<1 %
23	"Deep Learning and Data Labeling for Medical Applications", Springer Science and Business	<1 %

Media LLC, 2016
Publication

24	Libin Lan, Chunxiao Ye, Chengliang Wang, Shangbo Zhou. "Deep Convolutional Neural Networks for WCE Abnormality Detection: CNN Architecture, Region Proposal and Transfer Learning", IEEE Access, 2019 Publication	<1 %
25	Weidong Min, Hao Cui, Qing Han, Fangyuan Zou. "A Scene Recognition and Semantic Analysis Approach to Unhealthy Sitting Posture Detection during Screen-Reading", Sensors, 2018 Publication	<1 %
26	Jack C.P. Cheng, Mingzhu Wang. "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques", Automation in Construction, 2018 Publication	<1 %
27	Keren Gu, Sam Greydanus, Luke Metz, Niru Maheswaranathan, Jascha Sohl-Dickstein. "Meta-Learning Biologically Plausible Semi-Supervised Update Rules", Cold Spring Harbor Laboratory, 2019 Publication	<1 %
28	Éric Keiji Tokuda. "Computer vision analysis of unconstrained urban ground-level images",	<1 %

Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2019

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		