Table Detection in Document Images using Deep Learning

Author

Azka Gilani

MSCS5-00000118876

Supervisor

Dr. Faisal Shafait

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science (MS CS)

SCHOOL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE,

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY (NUST),

ISLAMABAD

Approval

It is certified that the contents and form of the thesis entitled “Table Detection in Document Images using Deep Learning" submitted by Azka Gilani have been found satisfactory for the requirement of the degree.

Advisor: Dr. Faisal Shafait
Signature:________________

Date:________________

Committee Member 1: **Dr. Hassan Aqeel**

Signature:________________
Date:________________

Committee Member 2: **Dr. Muhammad Imran Malik**

Signature:________________
Date:________________

Committee Member 3: **Dr. Adnan Ul-Hassan**

Signature:________________
Date:________________

Abstract

Table detection is an essential step in many document analysis applications as tables are used for presenting structural and functional information to the reader. It is a hard problem due to varying table layouts. Researchers have proposed numerous techniques for table detection based on layout analysis of documents. Most of these techniques fail to generalize because they rely on hand engineered features which are not robust to layout variations. The project aims to develop a deep learning based method for table detection. In the proposed method, document images are first pre-processed. These images are then fed to a Region Proposal Network followed by a fully connected neural network for table detection. The proposed method works with high precision on document images with varying layouts that include documents, research papers, and magazines. It has also been evaluated on publicly available UNLV dataset where it beats Tesseract's state-of-the-art table detection system by a significant margin.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Azka Gilani**

Signature: _______________

Acknowledgment

I owe my deepest gratitude to Allah Almighty. I also like to acknowledge the efforts of my parents to make such opportunity available for me. I am very thankful to my Supervisor Dr. Faisal Shafait for guiding me throughout this struggle.

Table of Contents

Chapter 1 8
Introduction and Motivation 8
1.1 Introduction 8
1.1.1 What is a Table? 8
1.1.2 Categories of Tables 10
1.1.2.1 Ruling-lined Table 10
1.1.2.2 Non-ruling lined Table 10
1.2 Problem Statement 11
1.3 Motivation 12
Literature Review 13
2.1 Text Analysis 13
2.2 Non-text Analysis 14
Design and Methodology 16
3.1 Image Transformation 16
3.2 Object Detection Module 18
3.2.1 Typical Neural Networks 18
3.2.2 Faster R-CNN 19
3.2.2.1 Region Proposal Network 19
3.2.2.2 Object Detection Module 21
3.2.2.4 Transfer Learning 23
3.2.2.5 Data Augmentation 23
3.2.2.6 Optimization 24
3.3 Implementation Details 24
3.4 Faster R-CNN Model Training 25
3.4.1 Region Proposal Extraction 25
3.4.2 Object Classification 26
Experiments and Results 27
4.1 Dataset 27
4.2 Performance Measures 28
4.3 Experiments and Results 32
1. Experiment 1 32

2. Experiment 2 33
3. Experiment 3 34
4.4 Comparative Evaluation 36
Conclusion and Outlook 39
Bibliography 40

Table of Figures

Figure 1 Varying table layouts in Documents 9
Figure 2 Classification of tables on the basis of layouts: a) Closed tables b) Non-closed tables c) Parallel tables 11
Figure 3 Algorithm for image transformation 17
Figure 4 Image transformation: (a) Original Image (b) Transformed Image 17
Figure 5 Architecture diagram of a typical neural network............................... 18
Figure 6 Basic Architecture of Faster R-CNN: It takes image as an input and then extracts feature vectors from it after convoluting image through convolutional layers. In the last layer Support Vector Machine is used in order to do object classification 19
Figure 7 Basic architecture of Region Proposal Network 20
Figure 8 Our approach: The document image is first transformed and then fed into a fine-tuned CNN model. It outputs a feature map which are fed into region proposal network for proposing candidate table regions. These regions are finally given as input to fully connected detection network along with the convolutional feature map to classify them into tables or non-tables. 22
Figure 9 Traditional machine learning vs Transfer learning 23
Figure 10 Data Augmentation by image rotation : (a) Original Image, (b)Rotate right by 90 degrees, (c) Rotate by 180 degrees, (d) rotate by 90 degrees, (e) rotate by 270 degrees 25
Figure 11 Sample images from UNLV dataset 28
Figure 12 Some sample images from the UNLV dataset showing detection results of proposed Table Detection approach. Ground-truth is blue while the detected regions are red. 31
Figure 13 Image transformation. Distance transform on red, green and blue channel... 32
Figure 14 Experiment 2: Algorithm for preprocessing 33
Figure 15 Experiment2: Preprocessed Image. (a) Original image (b) Preprocessed Image 34
Figure 16 Experiment 3: Preprocessing of images. (a) Original Image (b) Preprocessed Image 35
Figure 17 Visualization of results of various Engines. Blue color represents ground-truth while blue color represents detected region by proposed methodology. Maroon color represents results of Tesseract while magenta color represents results of Abbyy OCR SDK 38

Chapter 1

Introduction and Motivation

Tables are widely used for presenting structural and functional information present in the documents. They are present in diverse classes of documents including newspapers, research articles and scientific documents, etc. They enable readers to rapidly compare, analyze and understand facts present in documents. Table detection in documents is significant in the field of documents analysis and recognition; hence it has attracted a large number of researchers to make their contributions in this domain.

1.1 Introduction

Since the beginning of structured document analysis, researcher have been trying to find ways of extracting tabular regions from the document. Most of the scientific research has been carried out to extract low level geometric information from the scanned images while recent research work focuses on the analysis of tables in electronic form in order to obtain higher level of understanding.

1.1.1 What is a Table?

In documents, tables are means for representing data in a structured manner. It can either contain graphic, words, numbers and formulas, etc. In other words, it can be regarded as a 2D matrix where data is arranged in the form of rows and columns [1]. Costa e Silva [2] proposed a definition for table as, "graphical grid-like representation of a matrix $M_{i,j}$ where: (1) each element *i*, *j* of the matrix is atomic; (2) there are linear visual clues, i.e. the elements of each row *i* (column *j*) of the matrix tend to be horizontally (vertically) aligned; (3) linear visual clues describe logical connections [. . .]; (4) eventual line art does not add meaning otherwise not present in the relative positioning of the cells in the table." This definition covers a large number of tables with varying layouts but has certain limitations to include tables that can have multiple tables existing inside one table Tables have various layouts. In some of them, rows and columns are separated by ruling lines while in some ruling lines are not present. Apart from the ruling lines, they have different header format. Moreover, complexity in table layouts have greatly been increased with advancement in word processors. Different layouts of the table have been shown in Figure 1.

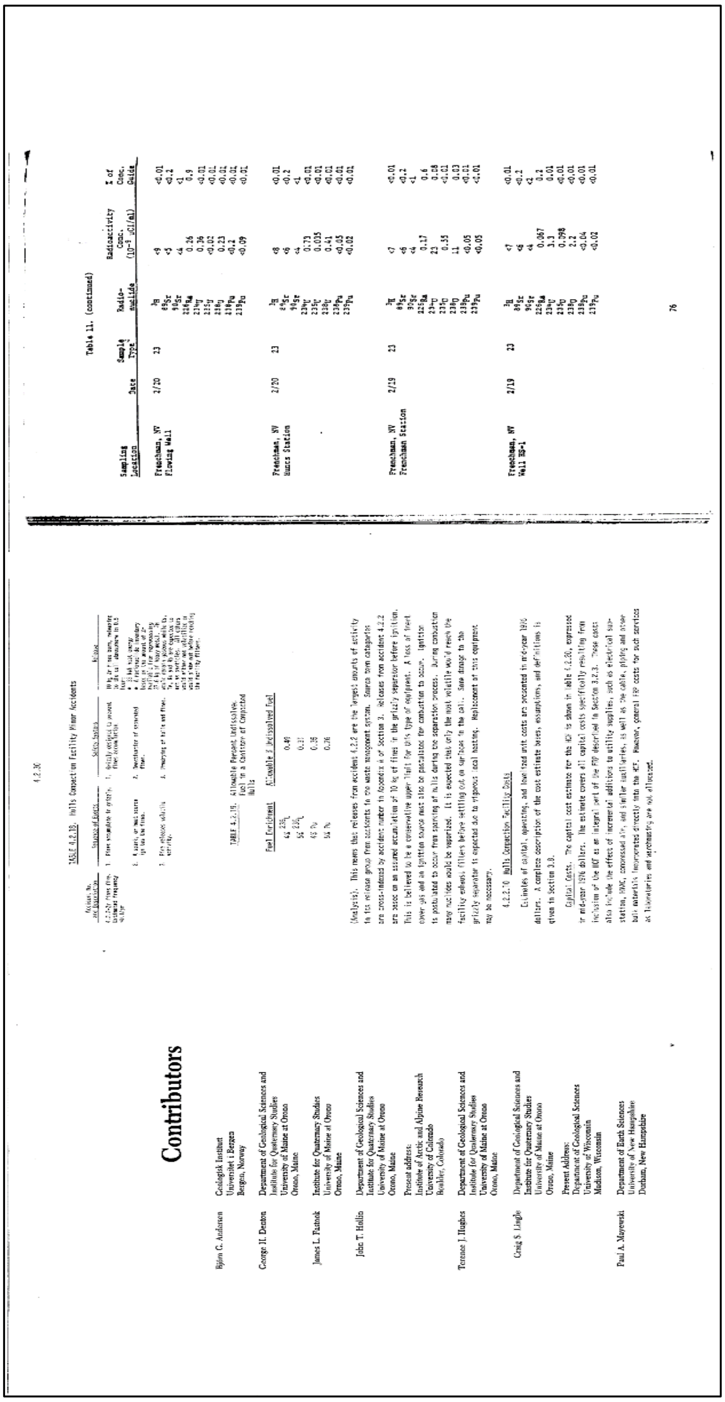


Figure 1 Varying table layouts in Documents

1.1.2 Categories of Tables

Though complexity in table layout has been greatly increased with increase in text processors but generally the tables that are found in document images are divided into two main categories depending upon their structure [3]. Figures of different classes of tables have been shown in Figure2

1.1.2.1 Ruling-lined Table

This category of tables is quite common. In it, the table region is enclosed within a bounding box or it gets discriminated from other regions of the table because of the presence of ruling lines. This category is further subdivided into following categories:

1. **Closed Table**
   A table that is completely enclosed within a bounding box is called closed table. In other words, elements within the table region and outer region are visibly separated by a rectangular boundary.
2. **Non-closed Table**
   A table region whose contents are separated by the presence of horizontal and vertical ruling lines but the presence of outer boundary is not compulsory.
3. **Parallel Table**
   This type of table is bounded by several parallel horizontal line segments. Number of ruling lines usually range from two to three but they can be more in number
4. **Colored Tables**
   These tables contain a layout that is the combination of several colored rectangular blocks.

1.1.2.2 Non-ruling lined Table

This is the type of table in which tabular region is not bounded within a bounding box. In this class, even the cells of tables are not separated by horizontal or vertical lines. These are normally identified as tables on the basis of spacing that is present within the rows and columns of the table.

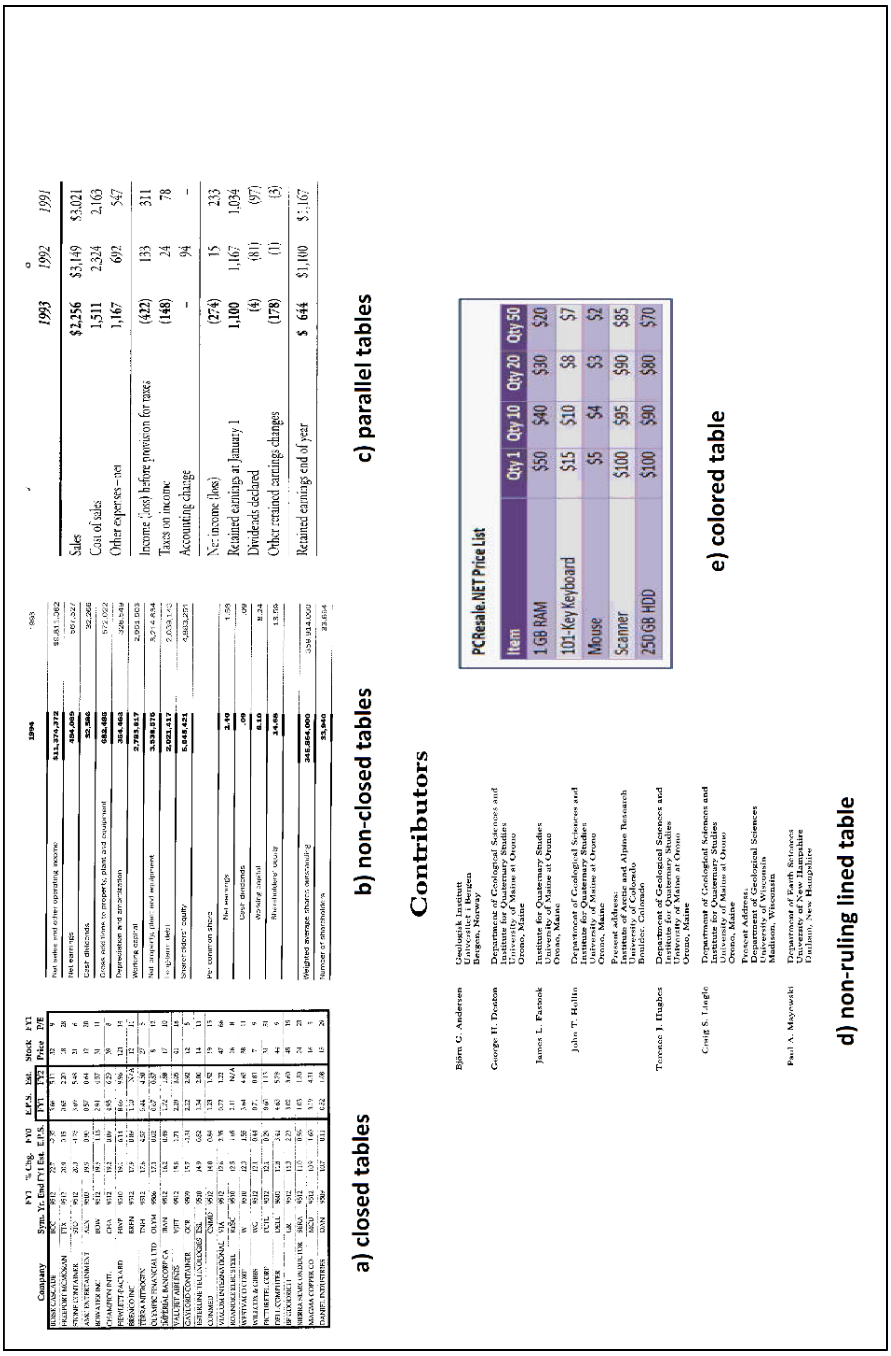


Figure 2 Classification of tables on the basis of layouts: a) Closed tables b) Non-closed tables c) Parallel tables d) Non-ruling lined table e) Colored table

1.2 Problem Statement

Table localization from document images is considered a hard problem in the field of document analysis due to varying table layouts. Most of the techniques that have been proposed earlier focused on the extraction of low level features from scanned images. While recent researches are more focused on the table detection in electronic support in order to obtain high level understanding of tables. Most of the research work that have been carried out earlier have certain limitations due to diversity in table layouts. Therefore, this process of detecting tables from the documents must be robust to diversity in table formatting.

1.3 Motivation

Table detection in documents is of significant importance in the field of document analysis and structure extraction. It is carried out by layout and content analysis of documents. Tables have varying layouts and variety of encodings. Due to this reason, table detection is considered as a hard problem in scientific society. Large number of researches have been carried out in this field but most of them have limitations due to diverse table layouts. Existing commercial and open source techniques for document analysis including OCR Tesseract lack the capability to completely detect table regions from document images.

In the recent years, deep learning techniques have greatly improved the results of various computer vision problems. Recently, Hao et al. [4] presented an approach for table detection from documents using deep learning. Their proposed methodology employs combination of custom algorithms and Convolutional Neural Network (CNN) to detect whether a table exists or not in the proposed region. The major limitation is that they have not discussed the technique for extraction of region proposals from documents in detail. So, it is not possible to reproduce the results. Hence in order to improve the performance of table detection and to make up for the limitations of prior techniques, a new deep learning based methodology is being introduced that can detect tables from documents robustly.

Chapter 2

Literature Review

Table detection and structure extraction has always been an important area of research in the field of document analysis and recognition. It is considered as a hard problem due to varying table layouts. Several researchers have reported their work regarding table detection in document images. Major limitation of these previously proposed techniques is that almost all of them are highly dependent on hand engineered features and thus fail to generalize in the presence of tables of different layouts in documents. Research work in this domain can be divided into two main approaches:

2.1 Text Analysis

Kieninger et al. [5]-[7] proposed an algorithm for table spotting and structure extraction from the documents called as T-Recs. This system takes word bounding boxes as an input. They are clustered to form "segmentation graph" using bottom-up approach. This algorithm with the help of graph detects rows, columns, spanning cells and as well as sparse tables with the high degree of confidence. The key problem with this algorithm is that it depends entirely on word bounding boxes and is thus unable to perform well in the presence of multi-column layouts.

Another approach was proposed by Wang et al. [8]. It detects table lines depending on the distance between consecutive words. This statistical approach assumes that maximum number of columns in document is two and designs three layout templates (single column, double column, mixed column) of document images. Then, column classification algorithm is applied to find out the column layout of the page. Major limitation of this techniques is that it can only work on those templates on which it has been trained. Another limitation is that this algorithm needs large dataset for training.

Hu et al. [9] presented approach for table detection that is based on white space correlation and vertical connected component analysis. It partitions a document into number of tables by using these heuristics. The major limitation is that the input images are single columned. Like previous methods, this technique cannot be applied on multi-column document images. Shafait et al. [10] presented an approach for table detection in heterogeneous documents. This system used layout analysis module of open source Tesseract OCR engine in order to locate table regions in documents of varying layouts. This system works well on large variety of documents but major limitation is that it is a traditional technique rather than data driven approach for table detection.

Tupaj et al. [11] proposed an OCR based table detection technique. The system searches for sequences of table like lines based on the keywords that might be present in the table headers. The line that contains keyword is regarded as the starting line while subsequent lines are then analyzed to match with predefined set of tokens which are then categorized as table structure. The limitation of this technique is that it depends highly on the keywords that might appear in table headers.

Martha O. et al. [12] proposed a system for table organization and to automatically extract and organize information from tables that are present in PDF documents. In order to localize table and to recognize table structure, it processes the document based on k-nearest neighbor and other layout heuristics.

Harit et al. [13] proposed technique for table detection based on the identification of unique table start pattern and table trailer pattern. The proposed technique is the mixture of top-down and bottom-up approach. It forms patches in the document images and then combine those patches of the belong to same logical unit. In case the patches don't match with each other, then the proposed technique automatically draws a line in order to make table segmentation. Table boundary is then decided by the content that is present on both sides of the boundary. It then gives a fair estimate of table boundary. The major limitation of this method is that it will not work properly whenever the table start patterns are not unique in document images.

2.2 Non-text Analysis

Cesarini et al. [14] proposed an approach for locating table regions based on the detection of parallel lines in MXY tree of the image. It is then filtered out by localization of perpendicular lines or white spaces in the region that is present between parallel lines. It uses optimization methods that relies entirely on the location of table index. The major limitation of this approach is that it entirely depends on tabular lines. This limits the scope of this system for tables that do not have any lines in their layout.

Gatos et al. [15] proposed an approach for table detection in document images that have tabular line structure. Their approach is highly dependent on the detection and localization of horizontal and vertical lines from the images. It localizes tables by finding area of intersection between the horizontal and vertical lines. The limitation of this system is that it works only for the documents in which the table rows and columns are separated by ruling lines.

Costa e Silva [16] presented a technique for table detection using Hidden Markov Models (HMMs). The proposed system combines different components of HMM in order to localize table regions in document images. The system extracts text from PDF by using pdftotext Linux utility. Feature vectors are then computed on the basis of the spaces present between the text. The major limitation of this technique is that it will not work for document images as text cannot be extracted from the images using pdftotext Linux Utility.

Kasar T [17] presented a method to locate tables by identifying column and row line separators. This system then employs run length approach in order to detect horizontal and vertical lines from input image. From each group of horizontal and vertical lines, a set of 26 low level features are extracted and passed to Support Vector Machine (SVM) which then detects the table. The major limitation of this techniques is that it is not data driven and it will work only for those images in which table region is properly separated by line separators.

Hao et al. [4] presented deep learning based approach for table detection. This system computes region proposals from document images through some predefined set of rules. These region proposals are then passed to the CNN that detects whether a certain region proposal belongs to table region or not. As already described, the major limitation of their methodology is that they have not shared complete algorithm for extracting region proposal from the image. Thus, this makes it impossible to reproduce the results.

In order to make up for the limitations of prior methodologies, this paper attempts to adopt Faster RCNN, one of the deep learning techniques, for solving table detection problem.

Chapter 3

Design and Methodology

The proposed methodology consists of two major modules: Image transformation and table detection. As we are already aware of the fact that document image consists of textual regions, blank spaces, figures and tabular regions etc. Image is transformed during preprocessing step in order to make clear distinction between white space region and textual region. After transformation, image is passed on to the detection module that uses Faster Recurrent Convolutional Neural Network (R-CNN) as a basic element of deep network. Faster R-CNN is completely dependent on the network that is formed by the combination of Region Proposal Network (RPN) and fully connected Convolutional Neural Network (CNN). Complete description of these modules will be shared further in subsections.

3.1 Image Transformation

It is the basic and most fundamental step for table detection in proposed algorithm. As the base of our object detection module is Faster R-CNN [18] but Faster R-CNN works only for natural images. Hence image transformation plays a vital role in conversion of document image into natural images as close as possible so that it can be used for fine tuning already available Faster R-CNN models.

Distance transform [19]-[21] is the derived representation of digital image. It calculates the distance between white regions and textual regions that gives fare estimate of textual region. As in deep convolutional networks, features of the images are passed from one layer to another so it is important that none of important features get missed or removed during distance transform. In order to make sure that all the features remain in the image, different types of distance transform have been applied on the channels (red, green and blue) channels of the image.

As per the methodology, Euclidian distance transform (distance between two points) on blue channel of the image while linear distance transform has been applied on the green channel. Similarly, distance transform that calculates maximum distance [19]-[21] between two points has been applied on red channel. The algorithm for image transformation has been shown below in Figure 3.

```
procedure IMAGE TRANSFORMATION(I)
    b ← EuclideanDistanceTransform(I)
    g ← LinearDistanceTransform(I)
    r ← MaxDistanceTransform(I)
    P ← ChannelMerge(b,g,r)
    return P
```

Figure 3 Algorithm for image transformation

The image transformation algorithm takes binarized image as an input and performs distance transform on three channels of the image. It computes Euclidian, Linear and Max distance transform on blue, green and red channels of the image. Transformation results has been shown in Figure 4.

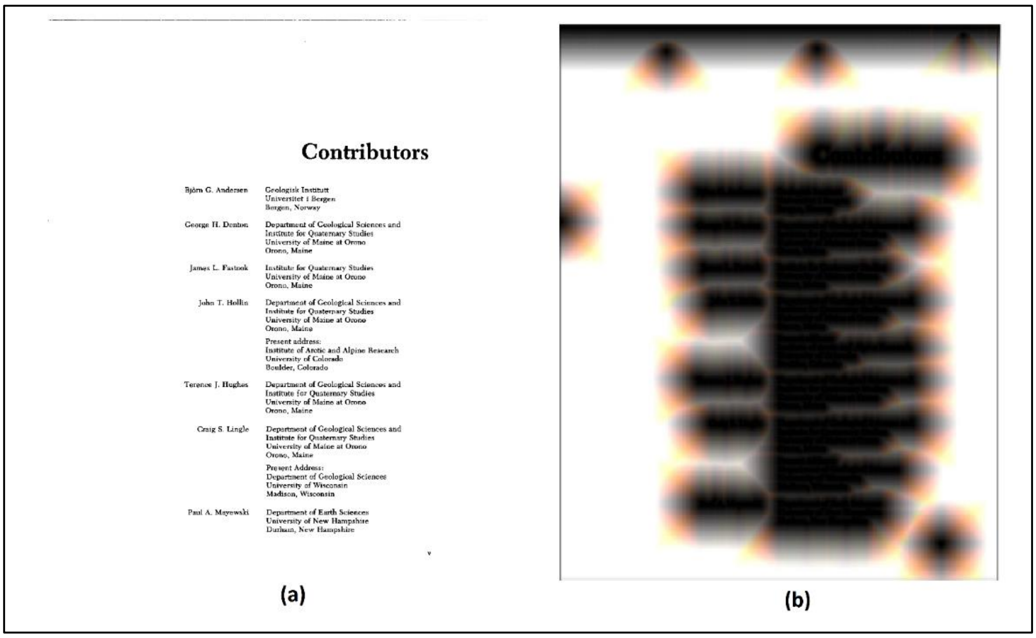


Figure 4 Image transformation: (a) Original Image (b) Transformed Image

3.2 Object Detection Module

Deep learning is an extended branch of machine learning that contains set of multiple powerful techniques that learns from images in multi-level artificial neural networks by extracting features from the image at each stage of neural network. Thus due to this reason of extracting right features from the images, the “Feature Extraction” problem is best addressed by deep neural networks because feature extraction is the core of machine learning. On the basis of these features, these networks learn and trains themselves. As we are also dealing with table detection from images, so the deep neural network that is employed in our pipeline is Faster Recurrent Neural Network (Faster R-CNN). Its architecture is specifically design for object detection in natural images.

3.2.1 Typical Neural Networks

Typical neural networks [22] consist of input layer, a large number of hidden layers and output layers. Output layer is also known as the Classification Layer. Depending upon the architecture, these hidden layers consists of several neurons that are connected to each other. In a fully connected neural network [23], all the neurons are fully connected to each other. The architecture diagram of typical neural net has been shown in Figure 5.

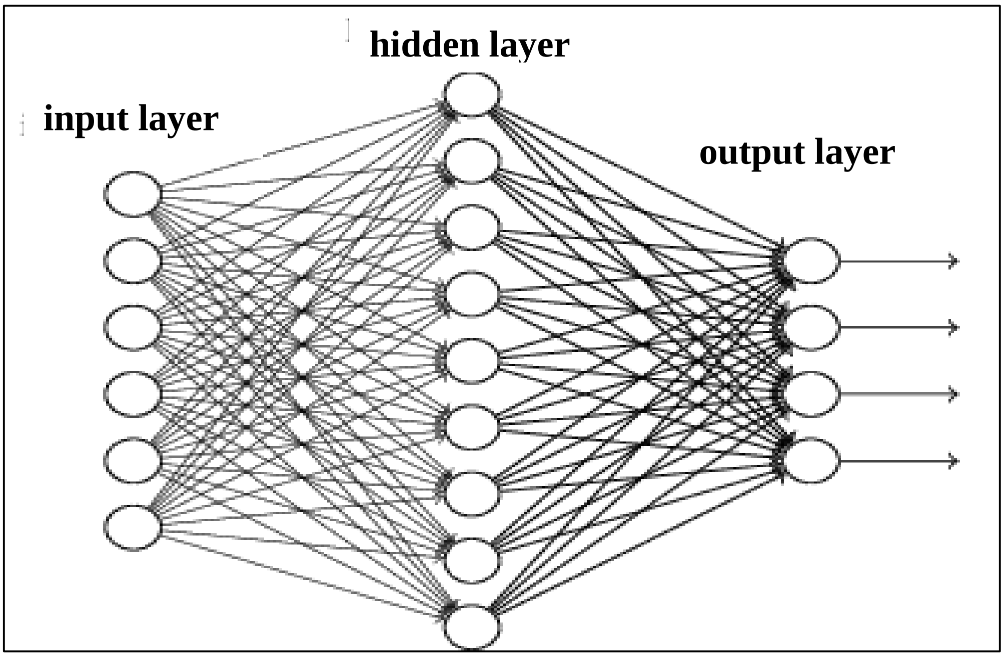


Figure 5 Architecture diagram of a typical neural network

3.2.2 Faster R-CNN

Faster R-CNN [18] is an advanced form of deep learning based fully connected neural network. It consists of three modules i.e. class independent region proposal network, fully connected convolutional network and a layer of linear Support vector machines. Region proposal network generates category independent potential region proposals. These region proposals define the set of candidates' detections available to our second module. Fully connected neural network [24] then extract feature vectors of fixed length from each proposed region. These are then passed to the third module that performs object classification on the basis of these feature vectors using support vector machine. In our problem domain, object will be table region. Basic architecture has been shown in Figure 6.

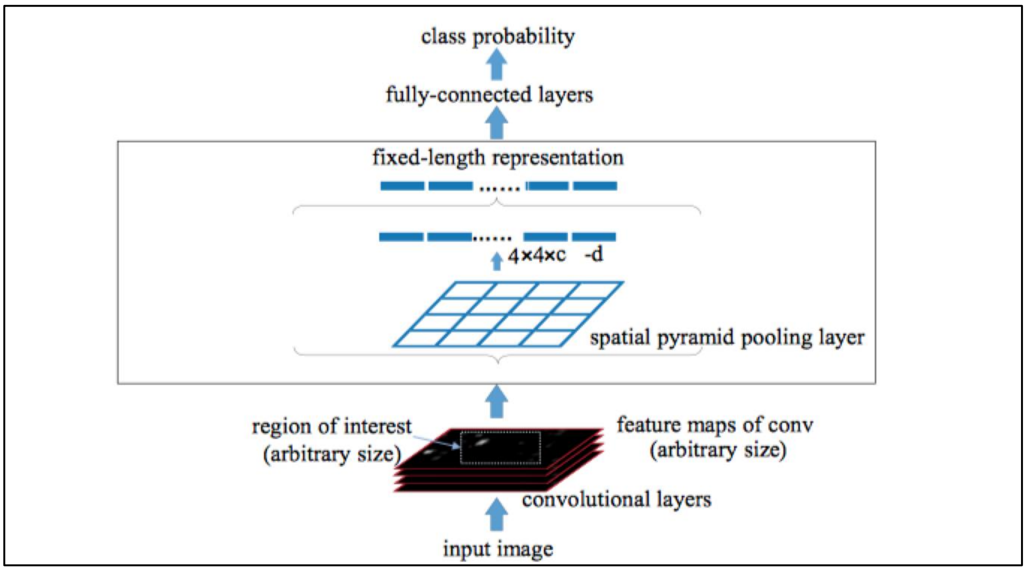


Figure 6 Basic Architecture of Faster R-CNN: It takes image as an input and then extracts feature vectors from it after convoluting image through convolutional layers. In the last layer Support Vector Machine is used in order to do object classification

3.2.2.1 Region Proposal Network

There are several methods that have been proposed earlier for extracting region proposals from the image. These include objectness score, selective search and category independent region extraction, etc. Recurrent Convolutional Neural Network (R-CNN) uses selective search mechanism [18] for proposing regions from the image.

4096 dimensional feature vectors are extracted from each of the proposed region by using Caffe based implementation of convolutional neural net. In order to extract features, RGB image is mean subtracted and then 227×227 dimensional images are then forward propagated through five convolutional and two fully connected layers.

In order to compute regions, it is necessary to convert images into 227 x 227 so that they are compatible with architecture of CNN’s. These images are then convoluted to a lower dimensional feature vector. These are then fed to the regression and classification layer respectively. It is to be noted that mini network operates in a sliding window manner i.e. the fully connected layers are shared across all the spatial locations. It follows the natural architecture that consists of $n \times n$ convolutional layers which are followed by two siblings of 1×1 convolutional layer for regression and classification layer respectively. Basic architecture of region proposal network is shown in Figure 7.

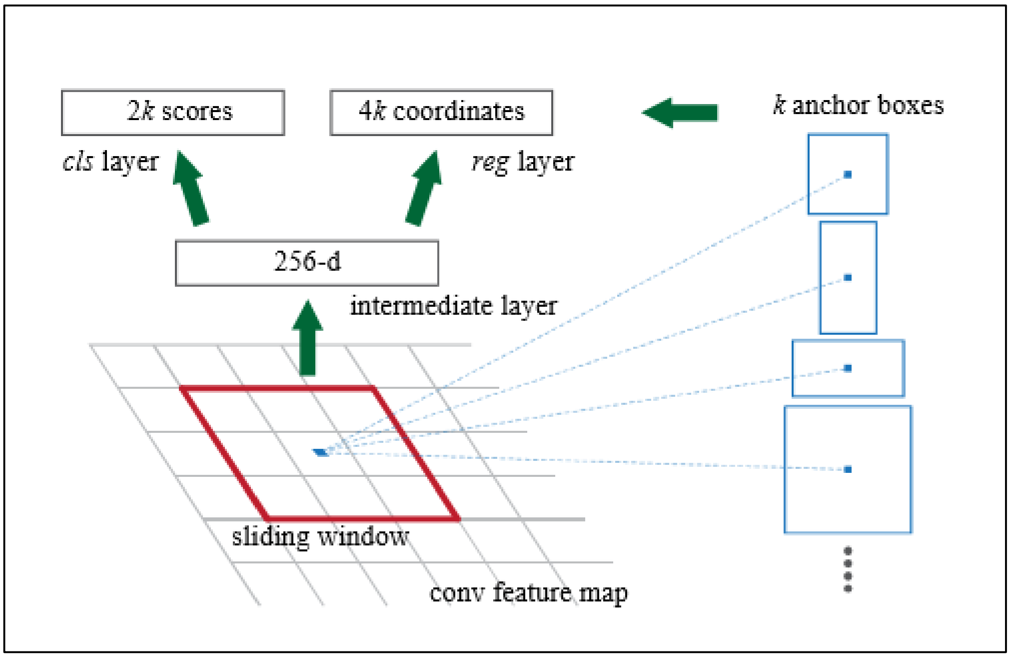


Figure 7 Basic architecture of Region Proposal Network

At each sliding window, *k* region proposals are simultaneously predicted and hence regression layer generates the *4k* outputs while each represents the bounding box of each proposed region. For each proposed layer by regression layer, classification layer then computes objectness score for each region proposal. It is calculated by using following formula [18]:

$$Objectness\ Score = \frac{Probability\ of\ an\ object}{Probablility\ of\ not\ an\ object}$$

These proposals are parametrized in relevance to the *k* reference boxes which are also known as anchors. Each anchor is further associated with aspect ratio of the proposal. Due to this property, all the proposed object proposals are scale and as well as translational invariant.

For training region proposal networks, a binary class label is assigned to each anchor. To assign positive and negative labels, Intersection over Union (IoU) is calculated between proposed anchor and bounding box of ground truth. If the IoU is highest or is greater than 70% then it is assigned a positive label otherwise it is assigned a negative label, if IoU [25] is lower than 0.3. This assignment of negative labels actually helps Faster R-CNN to not aggressively find object in those images in which object is not present. In other words, this helps in reducing number of false detections.
Hence the loss function for the image is defined as follows [18]:

$$L(\{P_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, {p^*}_i) + \lambda \frac{1}{N_{reg}} \sum_i {p^*}_i L_{reg}\ (t_i, {t^*}_i)$$

Here, i represents the index position of an anchor that is present in the mini-batch while p_i represents the probability of anchor to belong to an object. The ground truth label is shown by ${p^*}_i$ is assigned 1 if and only if the anchor is positive while otherwise it is assigned 0. While t_i represents the parametrized coordinates of predicted bounding boxes while ${t^*}_i$ represents the coordinates of ground truth bounding boxes.
As the image contains a large number of negative anchors so if optimization is applied directly, it will cause biasness towards negative samples. So, in order to overcome this 256 anchors are randomly sampled from each image to compute loss function while the ratio of positive and negative labels is 1:1.

3.2.2.2 Object Detection Module

As it has been mentioned already, that Faster R-CNN is the combination of RPN and fully connected CNN's. Both RPN's and CNN's are trained independently but they share same convolutional layers. This fully connected convolutional network is initialized by the Zeiler and Fergus model [26]. After this, detector module is used to initialize RPN's after fixing shared convolutional layer. Hence in this fashion, both networks now share the same convolutional layers. After that, the layers are fine-tuned and thus it forms the basis of shared network. Resultantly, it detects table regions from the test set and returns the bounding boxes for each detection table region.
Complete architectural diagram that includes preprocessing and object detection module has been shown in Figure 8.

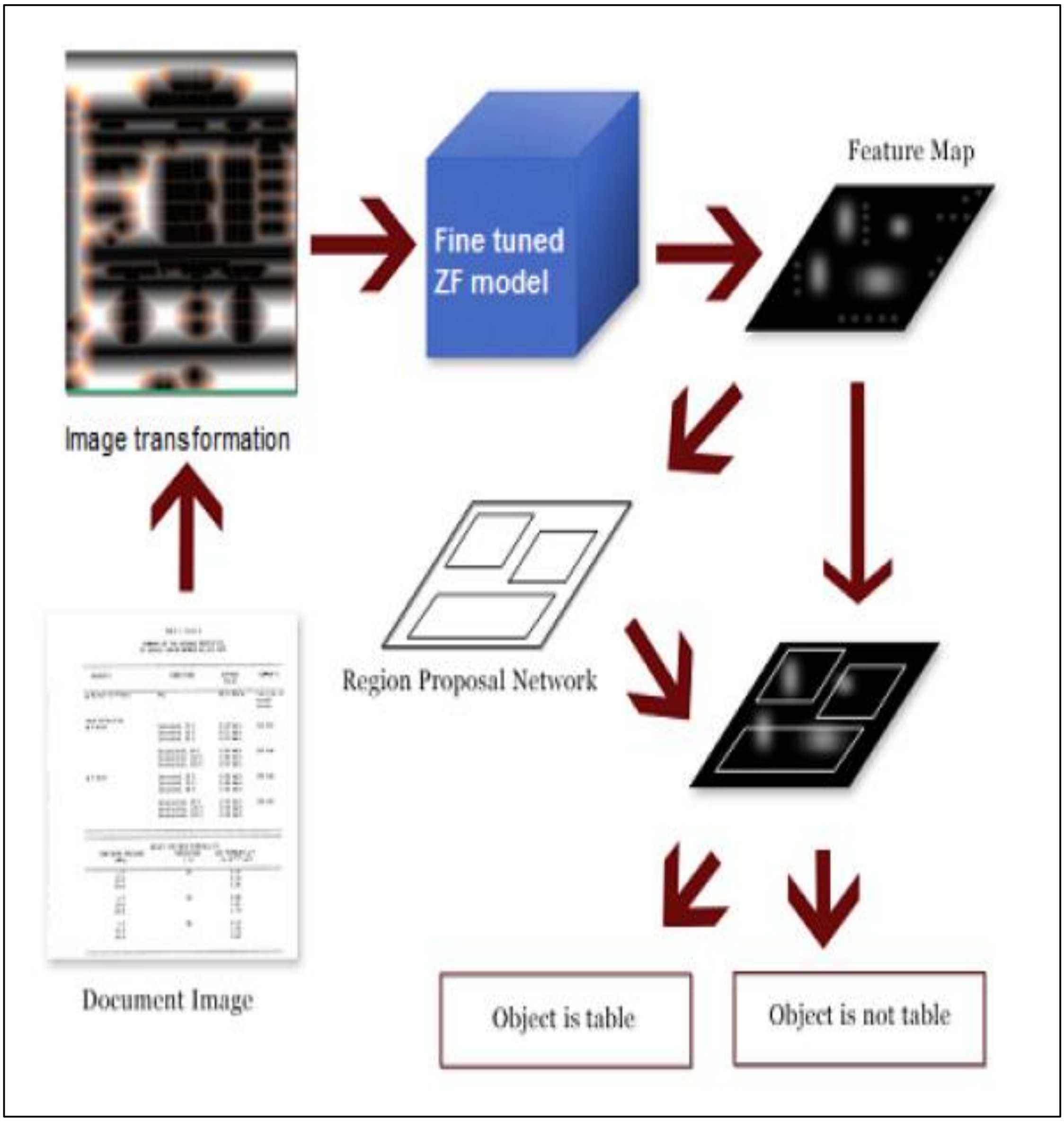


Figure 8 Our approach: The document image is first transformed and then fed into a fine-tuned CNN model. It outputs a feature map which are fed into region proposal network for proposing candidate table regions. These regions are finally given as input to fully connected detection network along with the convolutional feature map to classify them into tables or non-tables.

3.2.2.4 Transfer Learning

In the early days of machine learning and neural networks, it was assumed that in order to train the model, data on which the model has to be trained and tested should belong to same feature space and should have same distribution. As we know that neural nets need a huge amount of data in order to be trained so it is not possible to have data of such magnitude for each problem domain. After research and advancements, it has been observer that it is not essential to training and test data to share same set of features and distribution. Hence the concept of transfer learning came into existence. It was observed that in order to get more optimized results on test data, model gives more optimized results on data with different attributes and feature space.
It is very rare to train the fully connected convolutional network from scratch because it is not possible practically to have a huge dataset in order to train the model. Fully connected convolutional neural nets are mostly trained by using transfer learning [27] of already present models e.g. Zeiler and Fergus model [26] and then retain it on our new dataset as per our requirements. Comparison of traditional machine learning model and transfer learning has been shown in Figure 9.

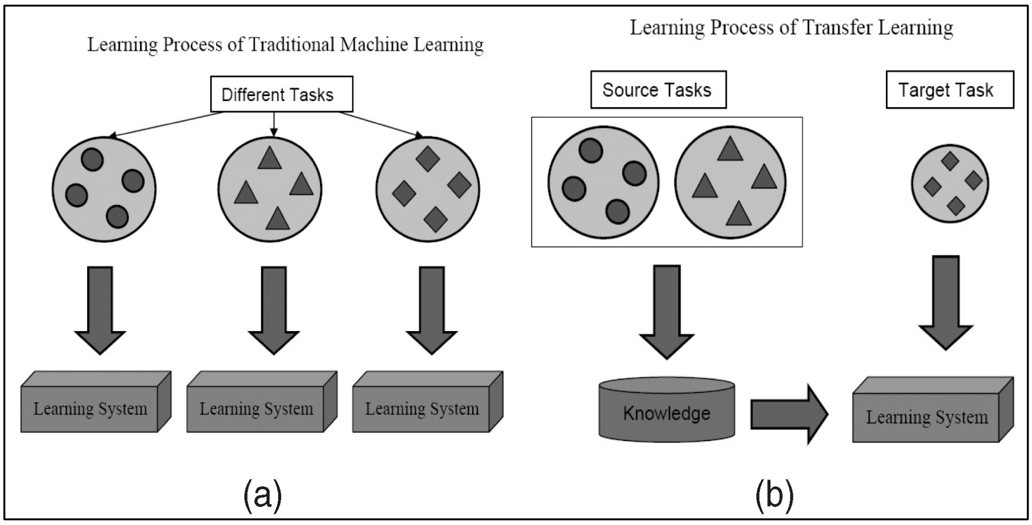


Figure 9 Traditional machine learning vs Transfer learning

In our table detection problem, we have used Zeiler and Fergus model [26] for transfer learning purposes. We have used that model as a feature extractor and then in the last layer, support vector machine is used in order to do binary classification.

3.2.2.5 Data Augmentation

Deep neural nets need a huge amount of data in order to give satisfactory results on test data. If the original dataset contains limited number of samples then it is

better to do data augmentation [28],[29] in order to boost the performance of trained model. It is the essential step while training on deep network. There are many techniques through which data augmentation can be implied. Some of them includes random crops, horizontally flipping and color jittering etc. In case of colored images, data can be augmented by changing values of HSV color space.
In case of Faster R-CNN [18] data has been augmented by rotating, cropping and flipping of the images. This not only boosts the performance of the model but also makes the model translation and scale invariant. It has been explained in Figure 10.

3.2.2.6 Optimization

The region proposal network that has been implemented as a fully connected convolutional network. It can be trained via end-to-end back propagation. Back propagation is a technique that can be used to calculate the error contribution of each neuron layer in a batch of processed images. It is mostly used in gradient descent optimization algorithms. Similarly, RPN can be trained with the help of stochastic gradient. It is also known as the incremental gradient descent [18]. To train the model as per our requirement, we have used image centric methodology for training the network on our training data.
Each mini batch contains a large number of positive and negative anchors. As the image contains a large number of negative anchors so it is not possible to optimize it for the loss function of all anchors. In order to make up for this problem, randomly 250 anchors from each image have been sampled where the ratio of positive and negative anchors in 1:1. If in an image, there are less than 128 anchors then it is padded with negative anchors in order to maintain the size of the samples. Model has been fine-tuned by using Zelier and Fergus model [26].

3.3 Implementation Details

We have used Caffe based implementation of Faster R-CNN. In deep neural networks, a large number of dataset consisting of thousands of images is needed in order to train the model. In order to make up for this limitation, we have fine-tuned our images by using Zeiler and Fergus model [26]. Momentum optimizer has been set to 0.001 and a momentum of 0.9 has been used in order to train the model on new dataset.

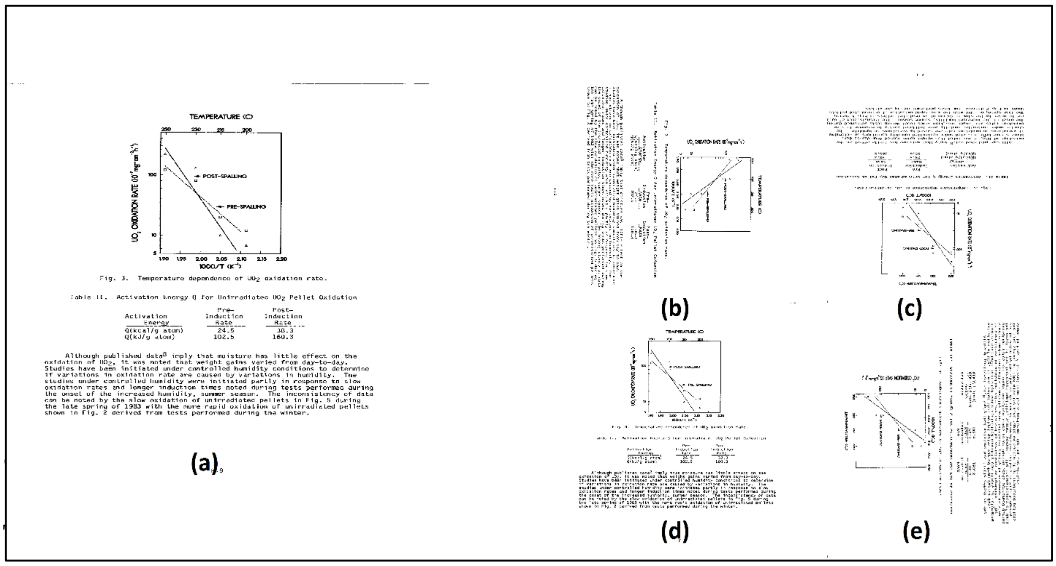


Figure 10 Data Augmentation by image rotation : (a) Original Image, (b)Rotate right by 90 degrees, (c) Rotate by 180 degrees, (d) rotate by 90 degrees, (e) rotate by 270 degrees

The feature vectors or proposed regions from each image have been extracted through Region Proposal Network. These proposed regions are then passed as feature vectors to fully connected convolutional network. After that, Classification at last layer of convolutional network is done by using support vector machine library.

3.4.1 Region Proposal Extraction

In this part of the training, first of all it loads the pre-trained model (ZF model in our case). It then loads the preprocessed images from training set and extracts features from it and then normalizes it. After that, it resizes the input images as per requirement of Faster R-CNN model. Ground truth is given in Pascal VOC format. The pseudo code for extracting region proposal by attention based mechanism is as follows:

1. Loads Zeiler and Fergus pre-trained model
2. Reads input image
3. Resizes it in order to make it compatible with Faster R-CNN library
4. Apply model on input images
5. Data augmentation
6. Get proposed regions from RPN
7. Save proposed regions in the form of MAT

3.4.2 Object Classification

The extracted regions proposals are passed as feature vectors to the Fully connected convolutional neural network. RPN and CNN then share the same convolutional network. Fully connected convolutional network then goes through each region proposal and then return detected proposals as bounding boxes

The pseudo code for classification is as follows:

1. Region proposals are passed as feature vectors to CNN
2. RPN and CNN share same convolutional network
3. CNN convolutes through each region proposal as proposed by RPN
4. Optimization
5. Support vector machine then classifies each region proposal as object or not
6. It returns coordinates of detected bounding boxes

Chapter 4

Experiments and Results

Efficiency of algorithms based on deep neural networks is highly dependent on the fact that on how good the network has been trained. Training of deep neural networks require a large amount of data in order to be trained. In order to compensate for this, we have used Zeiler and Fergus model for fine-tuning purposes.

4.1 Dataset

To evaluate the performance of proposed algorithm, we have used publicly available UNLV dataset [30]. This dataset was collected by DFKI university from Germany for research purposes related to document analysis. This dataset consists of wide range of documents with varied layouts. It includes research papers, scanned documents and magazines, etc. with varying and complex table layouts and structures. UNLV dataset [30] consists of 10,000 images with various resolutions. For each image, manually keyed ground truth text is available. Ground truth also contains zone information (table, text and figure, etc.) about the images. Each zone has been further categorized into subclasses depending upon the content. Amongst 10,000 images, only 427 images contain tabular structure. We have used all of these 427 images for evaluating the performance of proposed algorithm. Dataset has been divided into training and test set. Before dividing the data into training and test set, it was completely shuffled in order to avoid under fitting or over fitting. Training, validation and test data has been divided in ratio of 60:20:20. Cross validation [31] has also been applied in order to avoid over fitting.

As in deep neural networks, there is high chance of over or under fitting. So in in order to cater that techniques of fine tuning and transfer learning have been applied. As the dataset is small so we have use transfer learning approach for

training our model. We have used ZF model [26] for fine tuning our model. The major benefit of using fine tuning techniques is that optimizations of models are robust. In addition to that, neural nets need less amount of data for training purposes in case of fine-tuning. Due to fine tuning, learning rate of neural net on unknown dataset in high. Apart from that, we have also used data augmentation techniques that includes rotation, scaling, flipping of the image. Due to this data augmentation, our proposed model in translation and scale variant.
All the tasks related to preprocessing and model training has been performed using python, Caffe and CUDA environment.
Sample images from publicly available UNLV dataset has been shown in Figure 11.

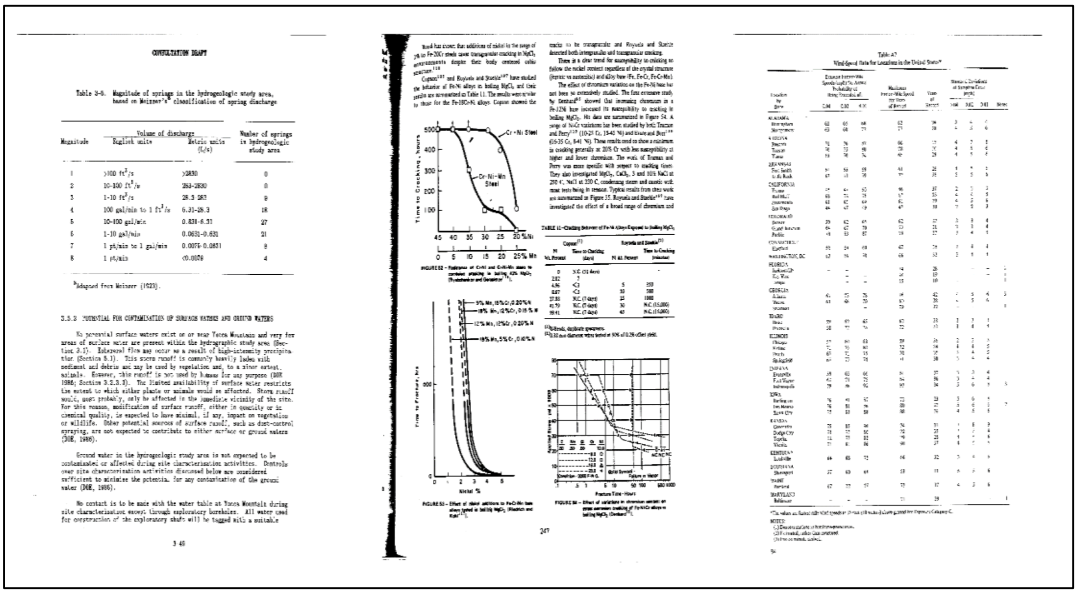


Figure 11 Sample images from UNLV dataset

4.2 Performance Measures

We performed 3 experiments in order to evaluate the performance of the proposed algorithm. We benchmarked the experiment with highest accuracy against commercial OCR Engines that includes Abby Fine Reader [34]. We also evaluated our experiment with state of the Art Tesseract's table detection system. Different techniques and measures have been mentioned in the literature for evaluation of the performance of table detection system. These measures include Precision and Recall [32],[33] for evaluating the performance of any system.
We have compared our proposed approach with Shafait et al. [10] So in order to make a fair comparison of both techniques, we used the same dataset. Due to this reason, we didn't compare our approach with techniques that were mentioned in ICDAR 2013, Table Competition. Most of the techniques that were proposed in

ICDAR 2013 [17],[35] were either layout specific or data driven and hence this makes these techniques non robust to layout analysis.
Consider ground truth bounding box is represented by G_i while the bounding box that has been detected by the proposed system is represented by D_j. The formula for finding the overlapped region between ground truth and detected bounding box is as follows [10]:

$$A(G_i, D_j) = \frac{2 \times |G_i \cup D_j|}{|G_j| + |D_j|}, A \in [0,1] \quad (4.1)$$

$A(G_i, D_j)$ represents the overlapped region between detected bounding box and ground truth bounding box while G_i represents the bounding box of ground truth. Similarly, D_j represents the bounding box of detected region by our proposed algorithm. Depending on the overlapped region, values of area lies between 0 and 1. If the two table overlap each other completely then the value of area is 1 while if two bounding boxes don't interest or overlap each other at all then value of area is 0. It is to be noted that we are using same threshold values as mentioned in Shafait et al. to make a fair comparison of both techniques. Complete definition of performance evaluation parameters is described as follows.

1. **Complete Detection**
   These are the number of detections in which ground truth bounding boxes have a major overlap with detected bounding box. Area of overlap must be greater than or equal to 0.9 ($A \geq 0.9$) in order to be considered as a complete detection. Formula for calculating overlapped region has been shown in Eq. 4.1.

2. **Partial Detections**
   These are the number of detections that have a partial overlap ($0.1 < A < 0.9$) between ground-truth and detected bounding box. Formula for calculating Area has been explained in Eq.4.1.

3. **Over-segmented Detections**
   These are the number of detections in which multiple detected bounding boxes overlaps ($0.1 < A < 0.9$) with one ground-truth bounding box. In other words, it means that different sections of ground truth bounding box were detected as different tables during detection.

4. **Under-Segmented Tables**
   In this category, different tables are merged during detection and are reported as single detection.

In other words, these are the number of detections in which detected bounding box overlaps ($0.1 < A < 0.9$) with more than one ground-truth bounding boxes.

5. **False Positive Tables**
These are the number of tables in which detected bounding box doesn't overlap ($A \leq 0.1$) with any of the ground truth bounding boxes. It means that tables are falsely classified as table by detection algorithm.

6. **Missed Tables**
These are the number of tables in which detection algorithm fails to detect table region. In other words, ground-truth bounding boxes don't overlap ($A \leq 0.1$) with any of the detected bounding boxes.

7. **Precision**
This performance measure evaluates the overall performance of algorithm. It calculates the percentages of detected bounding boxes that actually overlap with ground-truth bounding boxes.
The formula that has been used for calculating precision is as follows:

$$Precision = \frac{Area\ of\ ground - truth\ regions\ in\ Detected\ regions}{Area\ of\ all\ Detected\ table\ regions} \quad (4.2)$$

8. **Recall**
It is evaluated by calculating the percentage of ground-truth table regions that were correctly detected by proposed algorithm. The formula for calculating Recall is as follows:

$$Recall = \frac{Area\ of\ ground - truth\ regions\ in\ Detected\ regions}{Area\ of\ all\ ground - truth\ table\ regions} \quad (4.3)$$

9. **F1-Score**
This measure considers both precision and recall in order to evaluate the accuracy of methodology. Formula for calculating F1-Score is as follows:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4.4)$$

Results of these performance measures have been shown in the Figure 12. Ground truth bounding boxes have been represented by blue color while detected table region has been represented by red color.

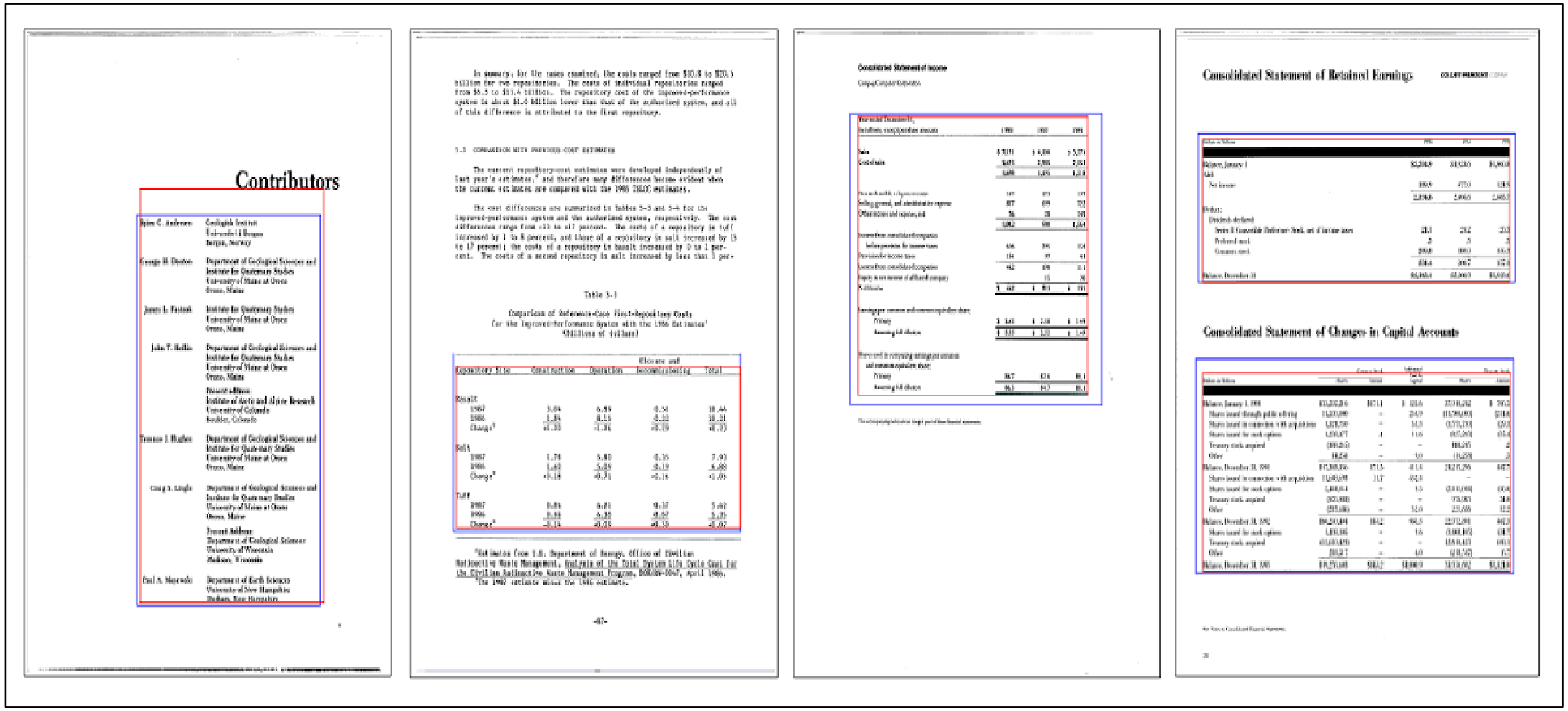


Figure 12 Some sample images from the UNLV dataset showing detection results of proposed Table Detection approach. Ground-truth is blue while the detected regions are red.

4.3 Experiments and Results

We performed 3 different experiments in order to evaluate the performance of all the techniques. In all these experiments, we used different preprocessing pipeline but used the same detection module. Performance of each experiment was evaluated using different performance measures. These measures have been described in detail in the section above.

1. Experiment 1

In order to perform this experiment, we used 3 different types of distance transform (Euclidian distance, Linear Distance and Maximum distance transform) on red, green and blue channels of the image separately. These preprocessed images are then passed to the detection module which then detects table regions in the form of bounding box from the images. It has been described in detail in Chapter 3. This is the proposed methodology as it gives more accurate results as compared to other experiments that are described below. Preprocessed images have been shown in Figure 13.

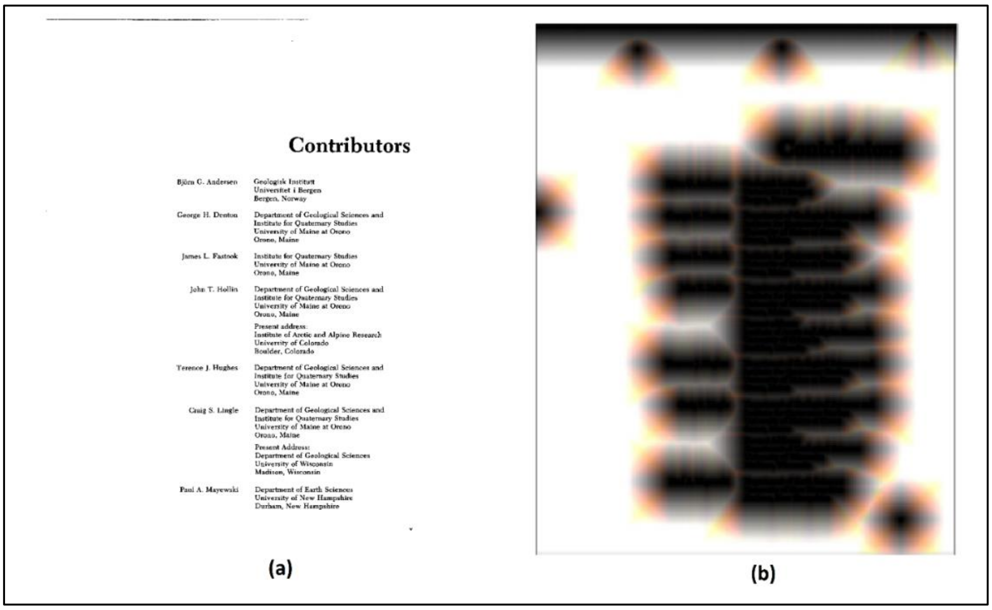


Figure 13 Image transformation. Distance transform on red, green and blue channel

The results of proposed methodology on various samples of UNLV dataset has been shown in the Figure 13. Ground-truth has been represented by blue color while detected table region has been represented by red color.

This methodology has been evaluated against the performance measures. Results have been shown in Table 1.

Performance Measures	Evaluation
Correct Detections	60.5
Partial Detections	30.2
Missed Tables	9.17
Over Segmented Tables	24.7
Under Segmented Table	30.27
False Positive Detections	10.17
Area Precision	82.3
Area Recall	90.67
F1-Score	86.29

Table 1 Experiment 1: Evaluation of proposed technique

2. Experiment 2

In this experiment, we applied distance transform of Red and green channel of the images and text in blue channel. This experiment was performed in order to evaluate if features of text information contribute to the table detection. Algorithm for preprocessing has been shown in Figure 15.

```
procedure IMAGE TRANSFORMATION(I)
   b ← EuclideanDistanceTransform(I)
   g ← LinearDistanceTransform(I)
   r ← TEXT INFORMATION(I)
   P ← ChannelMerge(b,g,r)
   return P
```

Figure 14 Experiment 2: Algorithm for preprocessing

Apart from the preprocessing pipeline, we kept the same module for object detection. Preprocessed images have been shown in Figure 16.

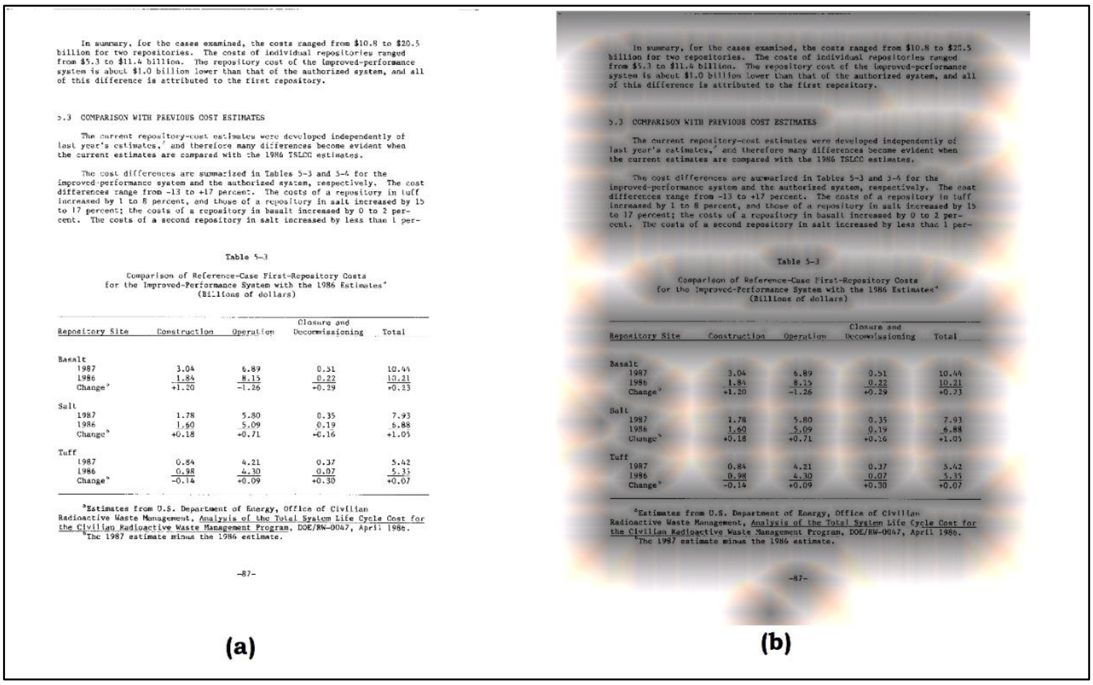


Figure 15 Experiment2: Preprocessed Image. (a) Original image (b) Preprocessed Image

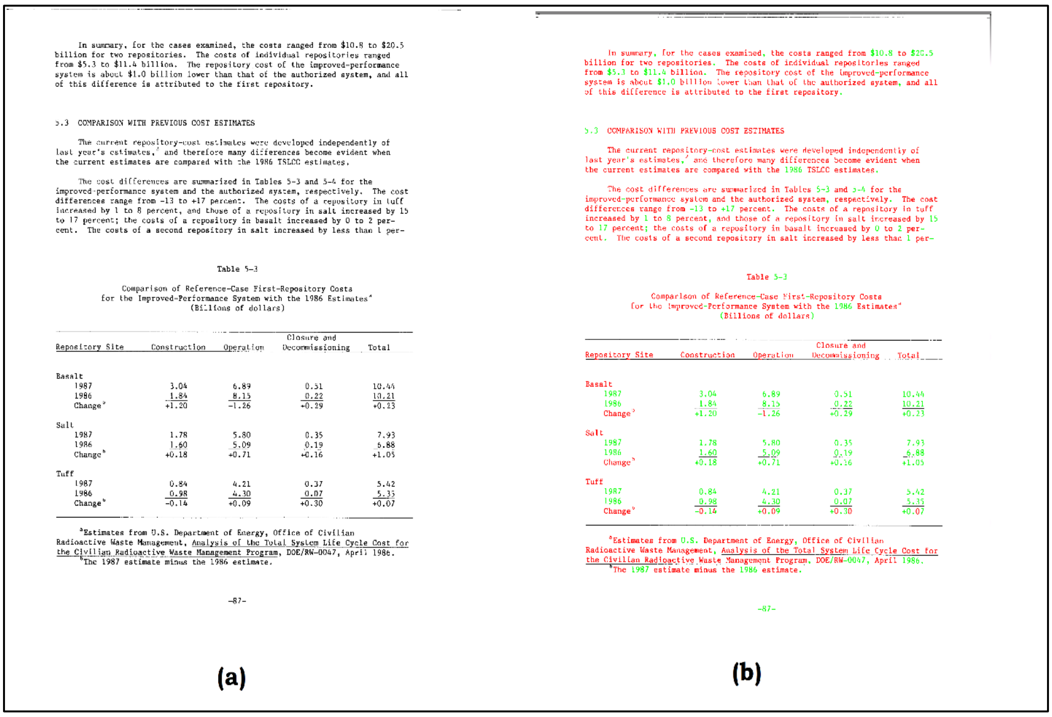
Results of object detection module on such preprocessed images have been represented in Table 2.

Performance Measures	Evaluation
Correct Detections	55.9633
Partial Detections	36.6972
Missed Tables	6.42202
Over Segmented Tables	37.6147
Under Segmented Table	36.6972
False Positive Detections	8.83721
Area Precision	83.1421
Area Recall	89.9455
F1-Score	86.40

Table 2 Experiment 2: Evaluation Results

3. Experiment 3

While conducting this experiment, it was assumed that numeric digits contribute more towards table detection. For this purpose, all the numeric information was colored as green while text was color coded as red. This color coding helped in differentiating numeric numbers from the text. Apart from preprocessing, object detection module was kept same during experiment. Result of preprocessed image have been shown in Figure 17.



In summary, for the cases examined, the costs ranged from $10.8 to $20.5 billion for two repositories. The costs of individual repositories ranged from $5.3 to $11.4 billion. The repository cost of the improved-performance system is about $1.0 billion lower than that of the authorized system, and all of this difference is attributed to the first repository.

5.3 COMPARISON WITH PREVIOUS COST ESTIMATES

The current repository-cost estimates were developed independently of last year's estimates,[a] and therefore many differences become evident when the current estimates are compared with the 1986 TSLCC estimates.

The cost differences are summarized in Tables 5-3 and 5-4 for the improved-performance system and the authorized system, respectively. The cost differences range from -13 to +17 percent. The costs of a repository in tuff increased by 1 to 8 percent, and those of a repository in salt increased by 15 to 17 percent; the costs of a repository in basalt increased by 0 to 2 percent. The costs of a second repository in salt increased by less than 1 per-

Table 5-3

Comparison of Reference-Case First-Repository Costs for the Improved-Performance System with the 1986 Estimates[a] (Billions of dollars)

Repository Site	Construction	Operation	Closure and Decommissioning	Total
Basalt				
1987	3.04	6.89	0.51	10.44
1986	1.84	8.15	0.22	10.21
Change[b]	+1.20	-1.26	+0.29	+0.23
Salt				
1987	1.78	5.80	0.35	7.93
1986	1.60	5.09	0.19	6.88
Change[b]	+0.18	+0.71	+0.16	+1.05
Tuff				
1987	0.84	4.21	0.37	5.42
1986	0.98	4.30	0.07	5.35
Change[b]	-0.14	+0.09	+0.30	+0.07

[a]Estimates from U.S. Department of Energy, Office of Civilian Radioactive Waste Management, Analysis of the Total System Life Cycle Cost for the Civilian Radioactive Waste Management Program, DOE/RW-0047, April 1986.
[b]The 1987 estimate minus the 1986 estimate.

-87-

Figure 16 Experiment 3: Preprocessing of images. (a) Original Image (b) Preprocessed Image

These preprocessed images were then fed to the table detection module. It returned bounding boxes of detected table regions for each image. Results of object detection module on these images have been represented in Table 3.

Performance Measures	Evaluation
Correct Detections	47.7064
Partial Detections	40.367
Missed Tables	11.0092
Over Segmented Tables	21.1009
Under Segmented Table	40.367

False Positive Detections	7.79221
Area Precision	86.3722
Area Recall	88.2595
F1-Score	87.30

Table 3 Experiment 3: Evaluation Results

4.4 Comparative Evaluation

After performing various experiments, we finalized experiment 1 as it is more robust to layout analysis and give better results as compared to other approaches. The results of the proposed experiment have been benchmarked against state-of-the-art tesseract technique [10] and commercial methodologies e.g. Abbyy Fine Reader [34] for table detection. The results have been shown in the Table 4 as follows:

	Accuracy(%)			
Performance Measures	**Tesseract**	**Abbyy**	**Without preprocessing**	**Proposed Methodology**
Correct Detections	44.9	41.28	51.37	60.5
Partial Detections	28.4	32.1	42.2	30.2
Missed Tables	25.68	25.68	6.42	9.17
Over Segmented Tables	3.66 7	7.33	29.35	24.7
Under Segmented Table	3.66	7.33	42.20	30.27
False Positive Detections	22.72	7.21	5	10.17
Area Precision	93.2	95.0	84.5	82.3
Area Recall	64.29	64.3	89.17	90.67
F1-Score	76.09	76.69	86.77	86.29

Table 4 Performance Comparison of different Table Detections

Visualization results of various engines with proposed methodology have been shown in Figure 18. Ground-truth has been represented by blue color while the detected table region by proposed methodology has been represented by green color. Similarly, results of ABBY OCR SDK have been shown by Magenta color while maroon color represents results of Tesseract.

Performance comparison between our proposed methodology, open sourced Tesseract and commercial Abbyy Cloud OCR SDK have been shown in Figure 15. While parsing a table, rows and column headers are often used as keys. If the header information is missing, then it is meaningless to parse remaining table information and the whole detected table becomes useless. Thus it is safe to say that the number of Correct detections is the most expressive and fundamental measure for evaluating any table detection methodology. Open sourced Tesseract and Commercial Abbyy fails to detect tables from document images in the presence of complex layouts in which huge white spaces are present. Thus it depicts that proposed approach has better performance as the number of correct detection greatly improved from 44% to 60%.

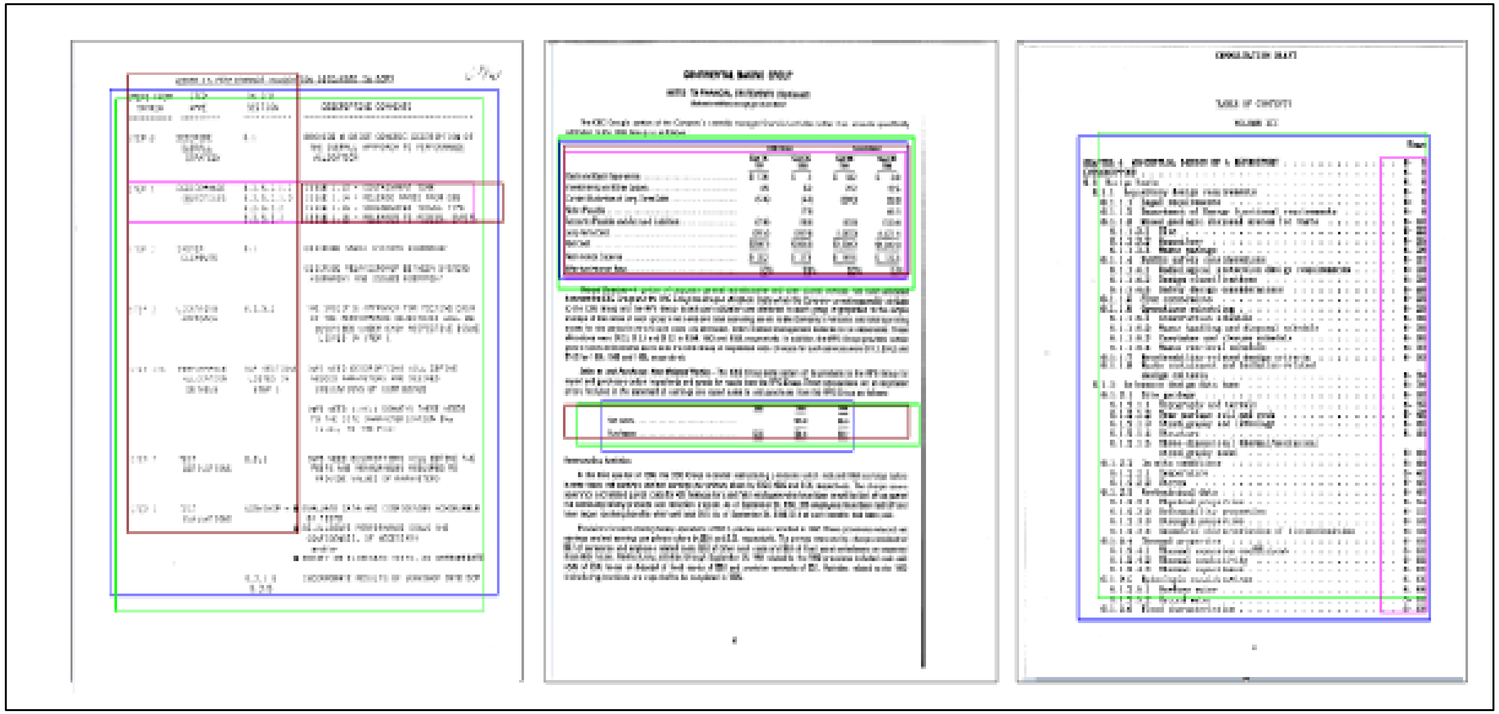


Figure 17 Visualization of results of various Engines. Blue color represents ground-truth while blue color represents detected region by proposed methodology. Maroon color represents results of Tesseract while magenta color represents results of Abbyy OCR SDK

Chapter 5

Conclusion and Outlook

The proposed algorithms used image transformation for separating text regions from non-text regions. These transformed images are then passed to Region proposal network followed by fully connected convolutional network of table detection module. The proposed algorithm has been evaluated using publicly available UNLV dataset. Experimental results have shown that proposed methodology is robust to layout analysis as it is not dependent on hand engineered features. As it is shown in Figure 13 that our proposed algorithm performs better in terms of accuracy as compared to open sourced OCR engines e.g. Tesseract [10] and commercial systems like Abbyy Cloud OCR SDK [34]. As in case of parsing tables, row and column headers are often used as keys. If they are missed by the detection module, then parsing of remaining table is meaningless. So, it can be assumed that correct detection is the most expressive measure for evaluation any table detection system. Our proposed methodology performs better as compared to other system as number of correct detections have improved greatly from 44% to 60%. In other words, it gives better results as compared to the state-of-the-art table detection system. We plan to extend this work in the direction of table structure and content extraction in the future.

Bibliography

1. Embley, David W., et al. "Table-processing paradigms: a research survey." *International Journal on Document Analysis and Recognition* 8.2 (2006): 66-86.
2. e Silva, Ana Costa, Alípio M. Jorge, and Luís Torgo. "Design of an end-to-end method to extract information from tables." *International Journal on Document Analysis and Recognition* 8.2 (2006): 144-171.
3. Tran, Tuan Anh, et al. "A mixture model using Random Rotation Bounding Box to detect table region in document image." *Journal of Visual Communication and Image Representation* 39 (2016): 196-208.
4. Hao, Leipeng, et al. "A Table Detection Method for PDF Documents Based on Convolutional Neural Networks." *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 2016.
5. Kieninger, Thomas, and Andreas Dengel. "A paper-to-HTML table converting system." *Proceedings of document analysis systems (DAS)*. Vol. 98. 1998.
6. Kieninger, Thomas, and Andreas Dengel. "Table recognition and labeling using intrinsic layout features." *International Conference on Advances in Pattern Recognition*. Springer, London, 1999.
7. Kieninger, Thomas, and Andreas Dengel. "Applying the T-RECS table recognition system to the business letter domain." *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001.
8. Wangt, Yalin, I. T. Phillipst, and Robert Haralick. "Automatic table ground truth generation and a background-analysis-based table structure extraction method." *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001.
9. Hu, Jianying, et al. "Medium-independent table detection." *Document Recognition and Retrieval*. Vol. 7. 2000.
10. Shafait, Faisal, and Ray Smith. "Table detection in heterogeneous documents." *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010.
11. Tupaj, Scott, et al. "Extracting tabular information from text files." *EECS Department, Tufts University, Medford, USA* (1996).

12. Perez-Arriaga, Martha O., Trilce Estrada, and Soraya Abad-Mota. "TAO: System for Table Detection and Extraction from PDF Documents." *FLAIRS Conference*. 2016.
13. Harit, Gaurav, and Anukriti Bansal. "Table detection in document images using header and trailer patterns." *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2012.
14. Cesarini, Francesca, et al. "Trainable table location in document images." *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 3. IEEE, 2002.
15. Gatos, Basilios, et al. "Automatic table detection in document images." *International Conference on Pattern Recognition and Image Analysis*. Springer, Berlin, Heidelberg, 2005.
16. e Silva, Ana Costa. "Learning rich hidden markov models in document analysis: Table location." *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009.
17. Kasar, Thotreingam, et al. "Learning to detect tables in scanned document images using line information." *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013.
18. Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.
19. Breu, Heinz, et al. "Linear time Euclidean distance transform algorithms." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.5 (1995): 529-533.
20. Fabbri, Ricardo, et al. "2D Euclidean distance transform algorithms: A comparative survey." *ACM Computing Surveys (CSUR)* 40.1 (2008): 2.
21. Ragnemalm, Ingemar. "The Euclidean distance transform in arbitrary dimensions." *Pattern Recognition Letters* 14.11 (1993): 883-888.
22. Specht, Donald F. "A general regression neural network." *IEEE transactions on neural networks* 2.6 (1991): 568-576.
23. Hecht-Nielsen, Robert. "Theory of the backpropagation neural network." *Neural Networks* 1.Supplement-1 (1988): 445-448.
24. Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2015.
25. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
26. Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.
27. Oquab, Maxime, et al. "Learning and transferring mid-level image representations using convolutional neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

28. Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.
29. Cui, Xiaodong, Vaibhava Goel, and Brian Kingsbury. "Data augmentation for deep neural network acoustic modeling." *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23.9 (2015): 1469-1477.
30. Shahab, Asif, et al. "An open approach towards the benchmarking of table structure recognition systems." *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010.
31. Kohavi, Ron. "A study of cross-validation and bootstrap for accuracy estimation and model selection." *Ijcai*. Vol. 14. No. 2. 1995.
32. Hu, Jianying, et al. "Medium-independent table detection." *Document Recognition and Retrieval*. Vol. 7. 2000.
33. Kieninger, Thomas, and Andreas Dengel. "An approach towards benchmarking of table structure recognition results." *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE, 2005.
34. Abbyy. (2017) OCR SDK engine. [Online]. Available: https://www.abbyy.com/en-eu/ocr-sdk/
35. Fang, Jing, et al. "A table detection method for multipage pdf documents via visual seperators and tabular structures." *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011