Title Block Analysis & Retrieval of Information from Scanned Engineering Drawing Images

A Report for End Semester Evaluation submitted to
Indian Institute of Technology, Kharagpur
in partial fulfilment for the award of the degree of
Master of Technology
in

**Electronics and Electrical Communication Engineering
with specialisation in
Visual Information & Embedded Systems**

by

DEBANJAN NANDI (10EC35026)

Under the supervision of

**Prof. Jayanta Mukhodhyay (Dept. of Computer Science & Engineering)
Prof. Prabir Kr. Biswas (Dept. of Electronics & Electrical Communication Engineering)**

**Department of Electronics and Electrical Communication Engineering,
Indian Institute of Technology Kharagpur,
Autumn Semester 2014-15**

Declaration

I certify that

a) The work contained in this report has been done by me under the guidance of my supervisor.

b) The work has not been submitted to any other Institute for any degree or diploma.

c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d) Whenever I have used materials (data, theoretical analysis, figures and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date:	Debanjan Nandi
Place:	10EC35026

DEPARTMENT OF ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Certificate

This is to certify that the project report entitled "*Title Block Analysis & Retrieval of Information from Scanned Engineering Drawing Images*" submitted by Debanjan Nandi (Roll No. 10EC35026) to Indian Institute of Technology, Kharagpur towards partial fulfilment of requirements for the award of the degree of Master of Technology in the Department of Electronics and Electrical Communication Engineering with specialisation in Visual Information and Embedded Systems, Indian Institute of Technology, Kharagpur, is a record of bona fide work carried by him under my supervision and guidance during academic session 2013 – 14.

Date: Prof. Jayanta Mukhopadhyay
Place:

Date:
Place: Prof. Prabir Kr Biswas

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my supervisors Prof. Jayanta Mukhopadhyay (Dept. of Computer Science and Engineering) and Prof. Prabir Kumar Biswas for their exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am also obliged to the professors of Department of Electronics and Electrical Communications Engineering and the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my project.

Lastly, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this project would not be possible.

CONTENTS

ABSTRACT

In this modern digital age, the increasing use of computer-aided design (CAD) and computer-aided manufacturing (CAM) systems has prompted the move from paper-based documentation towards computerized storage and retrieval systems. Document update and revision is efficiently achieved in this computerised form. However, it still remains a nightmare for enterprises working digital drawings to reap the complete benefits of digitization of documents, especially those related to search and storage using contexts inside these documents, since what the computer sees is only images.

In order to fully utilize the potential of documents digitization, it is extremely necessary to extract the information inside these documents, especially those related to the text. We must be able to identify and segregate these text parts from the actual graphical part in these images, to produce information that can be retrieved and used to create relations and context, enhancing the ability of an organization to store, access, and manage documents, while cutting down costs and stress.

Here in this thesis, I am specifically concerned with title block extraction from architectural design documents. The title block may possess many properties and we are trying to propose a new algorithm for extraction of information from title block. The overall project requires two different segments, namely text-graphics separation as pre-processing operations and development of an Optical Character Recognition (OCR) System which can work at a high accuracy rate aimed for these specific kind of documents.

As part of the project, we are trying to propose a novel pre-processing operations towards extraction of “Sheet No” from the architectural drawing documents. We also wish to modify the existing Open Source OCR engine – Tesseract OCR – which in its available version has very low accuracy. The extracted data will be stored as Meta data for the processed document image to be used for further processes like archiving.

CHAPTER 1: INTRODUCTION

A couple of decade ago, with computers not so common, and being much expensive with lesser and slower computation power, papers were being extensively used everywhere for any document keeping purpose. The advancement of computer technologies and improvement in processing power has led to the world going digital. We have a lot of printed information on papers which is a hassle and requires a lot of man power just for navigating and searching through them. Hence the entire sources of information now needs to be digitized for the ease of information storage and retrieval, thus minimising human efforts. To help with this problem and to improve the speed of work in industries, Document Solutions came into picture.

Beginning in the 1980s, a number of vendors began developing software systems to manage paper-based documents. These systems dealt with not only printed and published documents, but also photographs, prints, and many others. Later, developers began developing a different system which could manage all electronic documents, i.e., all those documents, or files, created on computers, and often stored on users' local file-systems. The earliest electronic document management (EDM) systems managed either proprietary file types, or a limited number of file formats. Many of these systems later became known as document imaging systems, because they focused on the capture, storage, indexing and retrieval of image file formats. EDM systems evolved to a point where systems could manage any type of file format that could be stored on the network. The applications grew to encompass electronic documents, collaboration tools, security, workflow, and auditing capabilities.

Here in this thesis, we are trying to come up with a new way of extracting information from scanned engineering drawing document particularly recognizing the title block. The project consists of two major subsection:

i) Development of image pre-processing techniques to separate text from images.

ii) Development and modification of an open-source OCR system to “read” the text.

Both of these sections are specific in terms of their target document and hence, can be tailored accordingly to achieve better performance with more generic systems. The initial focus is to devise a method for Title Block extraction which is the most important piece of information for any engineering document in order to provide easy archiving and navigation system. In this phase of work, we have worked mainly on extraction of title block, and its recognition. This shall be further extended and improved to extract more useful information from the document and finally converted to an editable format of the original scanned image of the document.

1.1 PROBLEM AND MOTIVATION

The document solution industry, today, is a multi-billion dollar industry with never ending productivity. The advancement in technology has propelled this industry by providing users a really nice, easy and secure way of storing processing searching the documents. So this area also gives the researchers a good platform to work on.

An engineering drawing is a legal document, because it communicates all the needed information about "what is wanted" to the people who will expend resources turning the idea into a reality. It is thus a part of a contract; the purchase order and the drawing together, as well as any ancillary documents, constitute the contract. Thus, if the resulting product is wrong, the worker or manufacturers are protected from liability as long as they have faithfully executed the instructions conveyed by the drawing. Because manufacturing and construction are typically very expensive processes (involving large amounts of capital and payroll), the question of liability for errors has great legal implications as each party tries to blame the other and assign the wasted cost to the other's responsibility. This is the biggest reason why the conventions of engineering drawing have evolved over the decades toward a very precise, unambiguous state.

For centuries, all engineering drawing was done manually by using pencil and pen on paper or other substrate. Producing drawings usually involves creating an original that is then reproduced, generating multiple copies to be distributed to the shop floor, vendors, company archives, and so on. The classic reproduction methods involved blue and white appearances (whether white-on-blue or blue-on-white), which is why engineering drawings were long called, and even today are still often called, "blueprints" or "blue lines". Since the advent of computer-aided design (CAD), engineering drawing has been done more and more in the electronic medium with each passing decade. Today most engineering drawing is done with CAD, but pencil and paper have not disappeared.

As the companies are leaving the pen and paper model and shifting towards computerized digital medium, it demands enormous human resources for drafting them to computer understandable and compatible format and to extract the information required from the design. This produced the demand for automation in this industry and to meet this demand many document processing tool are being designed and there are still many unexplored areas which need attention of this areas researchers. We are going to one such unexplored area of document processing which is the extraction of title block of engineering drawing containing the design number/sheet number and our main aim is to extract that number from the scanned copy of the design.

1.2 PROJECT OBJECTIVE AND OVERVIEW

The most important source of information in an Engineering drawing document necessary for its storage, and retrieval is the title block, or rather the sheet number. Our thesis focusses on this aspect of the document processing, where we wish to extract the Title Block & pass it on to the OCR Engine along with certain more information which could be useful in attaining a higher degree of accuracy.

This thesis is primarily divided into 4 sections:

i) Study of text/graphics separation techniques
ii) Analysis of functionality of Tesseract.
iii) Algorithm for Title Block Extraction & Recognition
iv) Results and Conclusions.

CHAPTER 2: TEXT/GRAPHICS SEPARATION

This chapter provides a brief explanation of the existing image processing techniques related to separation of the textual and graphical parts in an image and why text/graphics separation is required for extracting information from scanned Engineering drawing documents.

2.1 TEXT RECOGNITION IN GRAPHICAL DOCUMENTS

Graphical documents contain text labels such as dimensions in mechanical drawings, room names in architectural drawings, in addition to line primitives. Such labels are present mixed to graphics, in different orientation and font sizes. Classical OCR modules, which often assumes a regular layout of organized columns, and paragraphs, are not able to recognize them. It thus leads to refer to the problem of text/graphics separation.

Common problems of text analysis in graphical documents are touching to graphics, multiple orientations or even following curvilinear paths, and reduced lexicons. In this section, we describe a few state-of-the-art algorithms for detecting text strings in graphical documents.

The text-graphics separation process aims at segmenting the document into two different layers; the first layer assumed to contain text characters and annotations and a second layer containing graphical objects. Most approaches performs text separation at early stages in the processing pipeline, usually using image processing techniques and limited knowledge about the configuration of the image in terms of higher-level objects. As Lu suggests in [1], the differentiation between text and graphics in a graphical document is sometimes subjective if only the local distribution of pixels is considered, without taking into account the context. An isolated small line can be part of an "I" character but also part of a dashed line, the end of a dimension component, etc. Although there is some overlapping due to the mentioned higher- level interpretation, a common definition of both categories may be stated as follows:

- **Text:** symbols or strings for the interpretation of the document parts, including letters, words, digits, and/or special symbols.

- **Graphics:** non-textual components having a domain-dependent meaning according to a diagrammatic notation, including all kinds of lines, curves, solid, or textured areas. The use of low-level information for discriminating text in graphical documents requires the consideration of a number of geometric and topological properties at pixel or region level.

These properties will drive the heuristics for the process. According to Lu [1], in technical drawings, some geometric features that differ between textual and graphical components can be observed:

i. The size of text characters is often much smaller than that of graphics components. Changes in text size or shape are within a narrow range.

ii. Text characters usually appear in the form of short strings, having uniform size, inter-character distance and a "smooth path" (usually rectilinear and oriented horizontally, vertically, or slanted at an angle of 45 degree).

iii. The local stroke density of text regions is often much higher than that of graphics.

iv. The length of the linear components included in strokes of text strings is much shorter than that of graphics.

It is quite straightforward to implement algorithms that filter the image parts (connected components or segments) according to the above features. However, the detection of text components in graphical documents has a number of difficulties that require more sophisticated steps. The main difficulties are the following:

i. Graphical components include lines of any length, thickness, orientation, and pattern (continuous, dashed, and dotted). Closed curves of different order (circles, ellipses, etc.) or polylines can be unfilled, filled with solid areas, hatched, tiled.

ii. Text and graphic parts appear mixed, sometimes touching or overlapped. In some cases, there is no clear geometric difference between them (e.g., small components of dotted or dashed lines can be confused with characters or punctuation symbols).

iii. Text can be of any font, size, and orientation. The problem of multi-oriented text is an important difficulty that makes unusable traditional OCR techniques.

Engineering drawings present text strings in vertical, horizontal, or slanted rectilinear orientation. Tombre et al. in [2] proposed three families of text-graphics separation methods. Recently, Hoang and Tabbone [3] slightly reformulated this classification. According to the previous works, the following taxonomy of text/graphics separation approaches is proposed:

- Morphology analysis
- Connected component analysis
- Line (vector)-based segmentation
- Multi-resolution analysis
- Signal processing
- Ad hoc methods (forms, music)

The following subsections describes the different taxonomies of text/graphics separation.

2.1.1 MORPHOLOGICAL ANALYSIS

A number of methods use morphological operators under the assumption that only the text remains after applying iterative opening operations to the original image with structuring elements designed to eliminate rectilinear objects. Wahl et al. [4] was one of the first to propose such methods based on morphological filtering. He uses Run-Length Smoothing Algorithm (RLSA) to detect vertical and horizontal text strings, which can be seen as morphological closing (or opening) operations with vertical and horizontal structuring elements of size according to the text size and graphical lines width. The method of Lu [1] uses RLSA too. Though this method improves the ability to detect slanted lines by performing a stretching operation at different angles, its main drawback is that they tends to wrongly label text as graphics. RLSA has proven to be efficient in separating rule lines in forms, but its use in graphics documents is less frequent.

2.1.2 CONNECTED COMPONENT ANALYSIS

Connected component labelling is one of the most commonly used methods of text graphics separation. Initially proposed by Fletcher and Kasturi [5], the basic idea is to segment text based

on basic perceptual grouping properties. Thus, simple heuristics on font size, inter-character, word and line spacing, and alignment can be easily used. This method requires setting of many thresholds, but the advantage lies in the fact that these values are extracted are extracted from the image itself and are not manually set a priori.

2.1.3 VECTORIAL REPRESENTATION

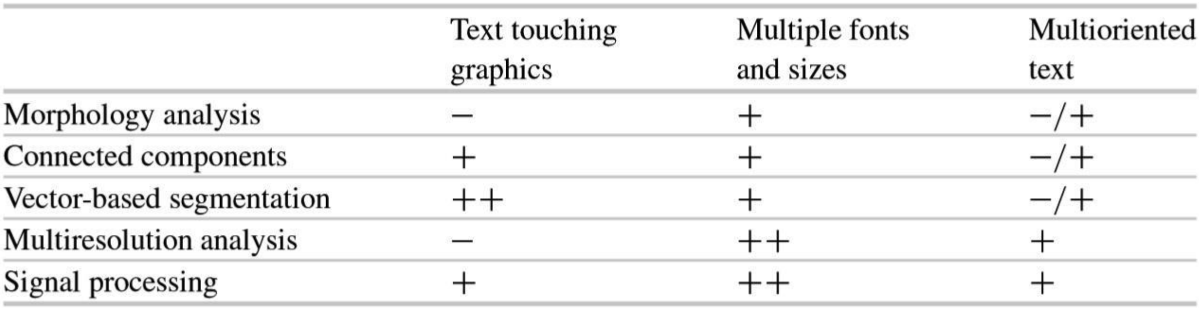
These approaches perform the segmentation of text is performed at vector level, i.e., after the image has been vectorized, instead of segmenting at pixel level [6, 7, 8]. The main idea lies in recursively grouping of short line segments in the vectorial image. Thus, after a contiguous short segment is selected as a seed, touching segments are iteratively merged. This procedure allows to obtain a coarse detection of character bounding boxes, which are then grouped in terms of their alignment. The basic process of this method however does not differ from the previously mentioned methods of connected component grouping.

2.1.4 MULTI-RESOLUTION REPRESENTATION

Multi-resolution approaches are based on visual perception principles. Each resolution allows to highlight a specific category of information. Tan and Ng proposed a multi-resolution approach in [9] based on the construction of a regular pyramid over the image. A pyramid representation is a well-known multi-scale signal representation in which an image is subject

2.1.5 SIGNAL PROCESSING METHODS

Inspired by techniques used in signal processing, Hoang and Tabbone [11] proposed a novel approach. They saw the image as a bi-dimensional signal composed of two separated bi-dimensional signals of the same size (text and graphics) but containing morphologically different features. They proposed a text-graphics separation method based on sparse representations. The main advantage of this method is that it overcomes the problem of text touching graphics, showing an improved performance regarding the classical state-of-the-art approaches. However, these approaches assumed that documents were structured in a Manhattan layout, where text is organized in columns, paragraphs, and lines, and graphics are contained in figures inserted in the text parts.



	Text touching graphics	Multiple fonts and sizes	Multioriented text
Morphology analysis	−	+	−/+
Connected components	+	+	−/+
Vector-based segmentation	++	+	−/+
Multiresolution analysis	−	++	+
Signal processing	+	++	+

Strong and weak points of text-graphics separation approaches

CHAPTER 3: TESSERACT OCR

Optical Character Recognition helps in the conversion of scanned images of printed text or symbols into text or information which can be understood or edited using a computer program. The most familiar example is the ability to scan a paper document into a computer where it can then be edited in popular word processors such as Microsoft Word.

The Tesseract engine was originally developed as proprietary software at Hewlett Packard labs in Bristol, England and Greeley, Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some migration from C to C++ in 1998. A lot of the code was written in C, and then some more was written in C++. Since then all the code has been converted to at least compile with a C++ compiler. Very little work was done in the following decade. It was then released as open source in 2005 by Hewlett Packard and the University of Nevada, Las Vegas (UNLV). Tesseract development has been sponsored by Google since 2006 (Source: Wikipedia).

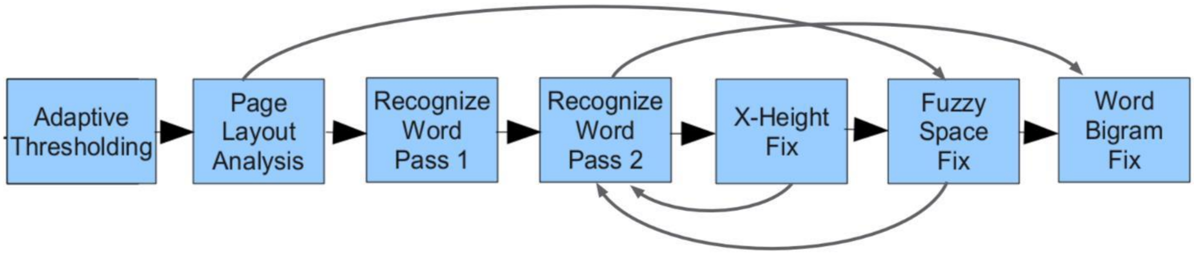
Tesseract is an OCR engine rather than a fully featured program, similar to commercial OCR software such as Nuance's Omnipage, ABBYY FineReader etc. It was originally intended to serve as a component part of other programs or systems. Although Tesseract works from the command line, to be usable by the average user the engine must be integrated into other programs or interfaces, such as FreeOCR.net, WeOCR or OCRpous. Without integration into programs such as these, Tesseract has no page layout analysis, no output formatting and no graphical user interface (GUI).

Tesseract is suitable for use as a backend, and can be used for more complicated OCR tasks including layout analysis by using custom frontend. Tesseract like any other traditional OCR engines assumes Manhattan type layout.

Tesseract's output in its basic mode is however of very poor quality if the input images are not preprocessed to suit it: images must be scaled up such that the text x height is at least 20 pixels, any rotation or skew must be corrected or no text will be recognized, low-frequency changes in brightness must be high-pass filtered, or Tesseract's binarization stage will destroy much of the page. Dark borders must be manually removed, or they would be misinterpreted as characters.

3.1 TESSERACT ARCHITECTURE

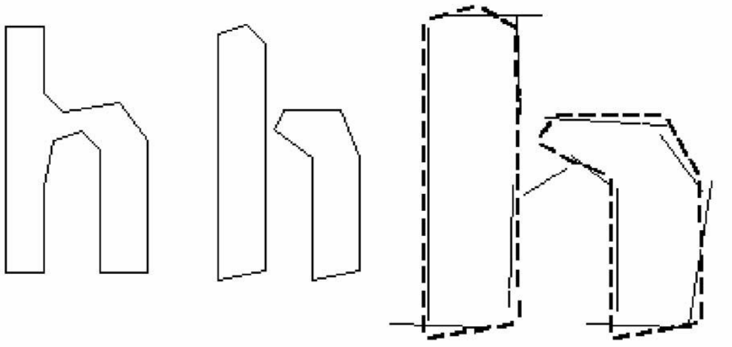
Since HP had independently developed page layout analysis technology that was used in its products, Tesseract never needed its own page layout analysis. Tesseract therefore assumes that its input is a binary image with optional polygonal text regions defined. Processing follows a traditional step-by-step pipeline, but some of the stages were unusual. The first step is a connected component analysis in which outlines of the components are stored. This was a computationally expensive design decision at the time, but had a significant advantage, i.e. by inspection of the nesting of outlines, and the number of child and grandchild outlines, it can easily detect inverse text and recognize it as easily as black-on-white text. Tesseract was probably the first OCR engine able to handle white-on-black text so trivially. After this stage, outlines are grouped together, purely by nesting, into Blobs. Blobs are organized into text lines, and the lines and regions are analysed for fixed pitch or proportional text. Text lines are broken into words differently according to the kind of character spacing. Fixed pitch text is chopped immediately by character cells. Proportional text is broken into words using definite spaces and fuzzy spaces. Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to recognize text lower down the page with increased accuracy. Since the adaptive classifier might have learned something useful too late to make a contribution near the top of the page, a second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces, and checks alternative hypotheses for the x-height to locate small – cap text.



Tesseract System Architecture

3.2 FEATURE EXTRATION

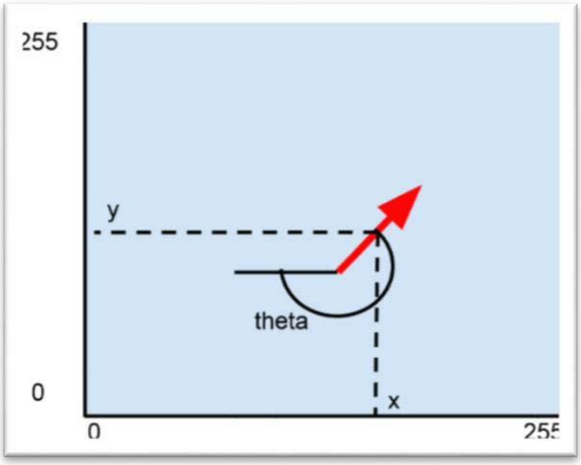
The early versions of Tesseract used topological features developed from the work of Shillmanet.al. [17] .Though nicely independent of font and size, these features were not robust enough for the problems found in real- life images. An intermediate idea involved the use of segments of the polygonal approximation as features, but this approach was also not robust to damaged characters.



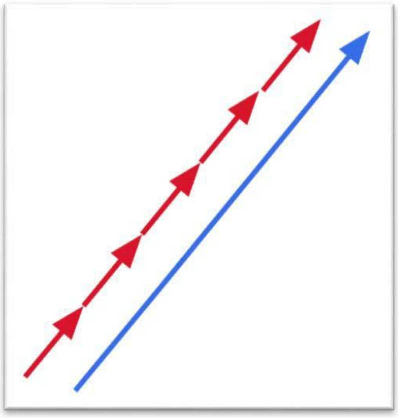
(a) Pristine 'h' (b) Broken 'h' (c) features matched to prototypes

The breakthrough achieved was that the features in the unknown need not be the same as the features in the training data. During training, the segments of a **polygonal approximation** are used for features, but in recognition, features of a small, fixed length (in normalized units) are extracted from the outline and matched many-to-one against the clustered prototype features of the training data. In Figure (c), the short, thick lines are the features extracted from the unknown, and the thin, longer lines are the clustered segments of the polygonal approximation that are used as prototypes. One prototype bridging the two pieces is completely unmatched. Three features on one side and two on the other are unmatched, but, apart from those, every other prototype and every other feature vectors are well matched. This example shows that this process of small features matching large prototypes is easily able to cope with recognition of damaged images. However the main problem lies in the computational cost of computing the distance between an unknown and a prototype which is usually very high.

The features which are extracted from the unknown are thus 3- dimensional, (x, y position, angle), with typically 50- 100 features in a character, and the prototype features are 4-dimensional (x, y, position, angle, length), with typically 10-20 features in a prototype configuration.



Features extracted from the unknown; Each feature is a short, fixed length, directed, line segment, with (x,y) position and theta direction making a 3-D feature vector (x, y, theta) from an integer space [0, 255]



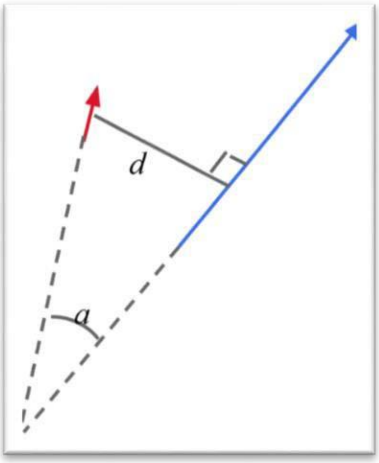
Features in the training data; x,y position, direction, and length (as a multiple of feature length)

3.3 FEATURE CLASSIFICATION

Feature classification is a two-step process. In the first step, a class pruner creates a shortlist of character classes that the unknown might match. Each feature fetches, from a coarsely quantized 3-dimensional look- up table, a bit-vector of classes that it might match, and the bit-vectors are summed over all the features. The classes with the highest counts (after correcting for expected number of features) are short-listed for the next step. Each feature of the unknown looks up a bit vector of prototypes of the given class that it might match, and then the actual similarity between them is computed. Each prototype character class is represented by a logical sum-of-product expression with each term called a configuration, so the distance calculation process keeps a record of the total similarity evidence of each feature in each configuration, as well as of each prototype. The best combined distance, which is calculated from the summed feature and prototype evidences, is the best over all the stored configurations of the class.

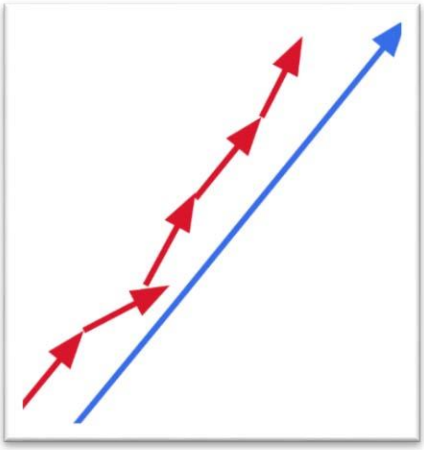
The Distance function: Single Feature to Single Proto

- $d = perpendicular\ distance\ of\ feature\ f\ from\ proto\ p$
- $a = angle\ between\ feature\ f\ and\ proto\ p$
- $Feature\ distance\ dfp{=}\ d2{+}a2\ (in\ appropriate\ units)$
- F$eature\ evidence\ efp{=}1/(1{+}kdfp2)$



Feature Evidence and Proto Evidence

- $Feature\ evidence\ ef{=}max$p $in\ config\ (efp)$
- $Proto\ evidence\ e$p$=\ \sum efptop\ l$p $(Proto\ p\ is\ of\ length\ lp)$



CHAPTER 4: TITLE BLOCK EXTRACTION ALGORITHM

The title block of a drawing, usually located on the bottom or lower right hand corner, contains all the information necessary to identify the drawing and to verify its validity. This is the area of the drawing that conveys header-type information about the drawing. However, if the sheet is oriented differently during scanning, the title block might be present at any other location. Nevertheless, a title block is divided into several areas such as:

i. Drawing title (hence the name "title block").

ii. Drawing number.

iii. Sheet Number.

iv. Part number(s).

v. Name of the design activity.

vi. Identifying code of the design activity.

vii. Address of the design activity (such as city, state/province, and country).

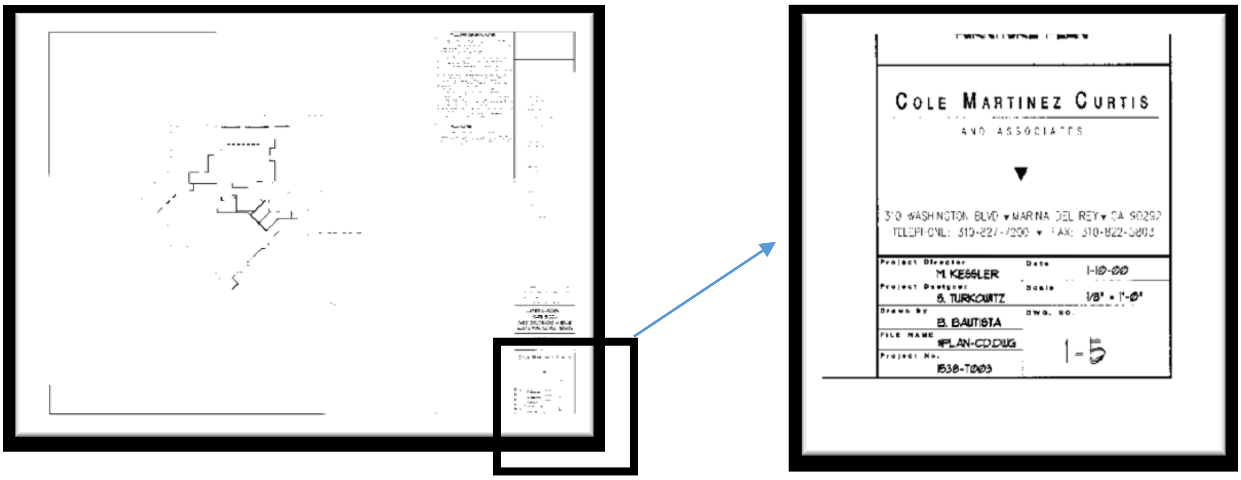
viii. Measurement units of the drawing.

ix. Default tolerances for dimension callouts where no tolerance is specified.

x. Boilerplate callouts of general specs.

xi. Intellectual property rights warning.

In this thesis our aim is to detect the Sheet Number part using document image analysis.

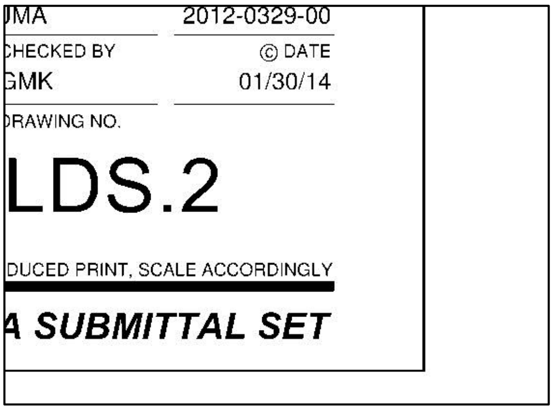


Position of Title block in an Engineering Drawing (Image Credit: ARC Document Solution)

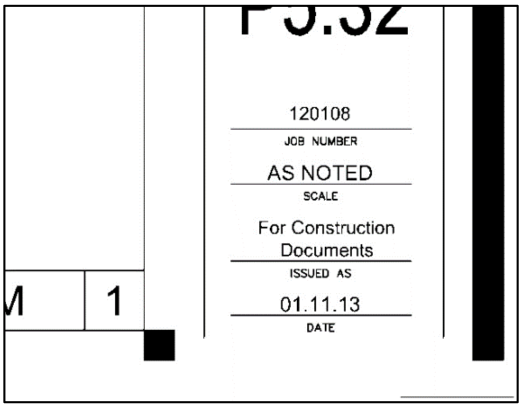
4.1 PROCEDURE

For our system in general, we would be given a scanned architectural design document for extraction of the sheet number. The title block is in most cases present at the right bottom corner of the document, so for now, we work under the assumption that the true title block region is cropped from the image More sophisticated algorithms to detect the region of the text block is not a top priority right now. Considering the huge size of Engineering Drawing images (in order of 10000 x 10000 pixels), this helps in faster work during development & testing

So in our test case we were given some cropped images which were cropped from the bottom right corner of the document. So the inputs provided have some images, where there is no design or sheet number to be retrieved and but most of the cropped images have the required data available as shown below 2 samples show a correct input and a false positive input.



Correct Input



Incorrect Input

We are going to start working considering the observations we are making. Some of the most important observations are as follows:

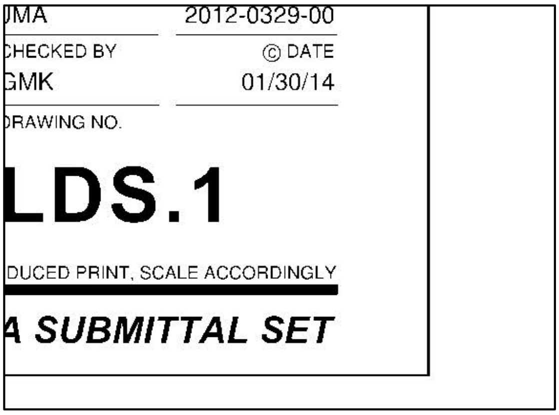
1. The sheet number had the largest font. In most of the documents it holds true but some parts of the sheet number might be of smaller font or even in some cases some other part of the document may be of the same size.
2. Some characters of the sheet number fields were not separated.
3. Some characters from the sheet number were overlapping with the table or some lines or some other object of the image.
4. There are stray lines and random small font texts are available which are of no relevance.
5. The sheet numbers are not skewed or tilted so no need of tilt correction is required.
6. The Cropped input image is significantly larger than the text of concern.
7. We don’t need any color information of the document as the visual properties don’t matter only the structure of the objects is what we are worried about.
8. Some characters were of weird style fonts which are not used generally which needed to be extracted.
9. Due to cropping a cropped table or cropped graphics content might be present inside the image.

Based on the above made observations, we applied some algorithms for the removal of the unwanted objects they are as follows:

4.1.1 BINARIZATION OF TITLE BLOCK IMAGE AND NOISE REMOVAL

Since, we have no use of the colour information, binarization helps in reducing memory usage. Additionally, it was observed that characters which were very close were often connected by few very light pixels. Setting a threshold value biased towards black pixels helped separating those characters without any effective loss of information about the necessary structural details.

Salt-and-pepper noise which appeared here was removed by the application of a median filter.



JMA 2012-0329-00

CHECKED BY © DATE

GMK 01/30/14

DRAWING NO.

LDS.1

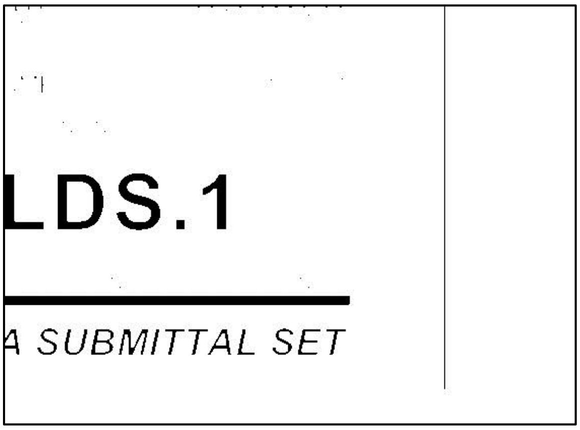
DUCED PRINT, SCALE ACCORDINGLY

A SUBMITTAL SET

Binarised Input after Noise Removal

4.1.2 REMOVAL OF THIN LINES/ CHARACTERS

The thin lines and the considerably smaller characters in the image by morphological operations of the erosion using a structural mask size of 3x3 where all the elements were 1. This stage not only eliminated the thin lines which are graphical elements, but it also helped in removing the small, low-width characters, which could never possibly be title blocks.

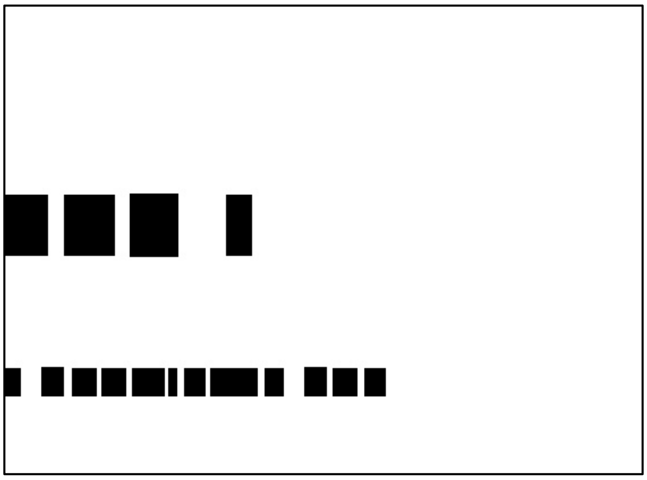


LDS.1

A SUBMITTAL SET

Post morphological operations

4.1.3 CONNECTED COMPONENT LABELLING AND BOUNDING BOXES



The text components were now extracted using connected component analysis on the segmented images. Connected component analysis not only locates the black connected component in the image, but also computes their sizes/areas, densities, bounding boxes.

4.1.4 REMOVAL OF NON TEXT IMAGES BASED ON CERTAIN PARAMETERS

In this procedure, the connected component inside each bounding box is checked with respect to the original image. The following important parameters are then calculated.

1) **MaxBox**, which represents the maximum and minimum co-ordinates of the circumscribing rectangle.
2) **Density,** which represents the stroke (black) pixel density in the circumscribing rectangle MaxBox.
3) **HWRatio,** which represents the dimensional ratio of the circumscribing rectangle.
4) **Average Width, Average Height** of the Connected Components.

We outright rejected all connected components which had a density in excess of threshold T1 and thresholds T2, because all textual components are expected to lie within these two parameters.

4.1.5 DETECTION OF TITLE ELEMENTS/SHEET NUMBER

Having obtained a near all text image, from the previous procedure, connected component analysis was again performed on the remaining elements in the image. We calculated the above mentioned parameters again for the remaining elements in the image.We extracted the text parameters based on the following properties.

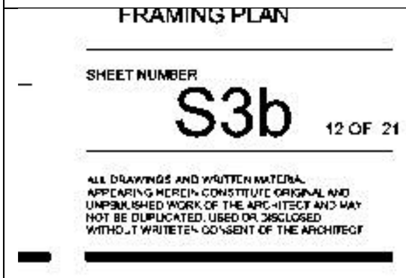
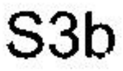
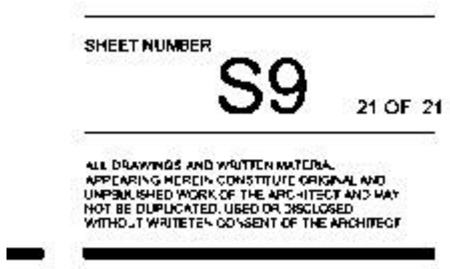
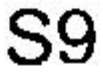
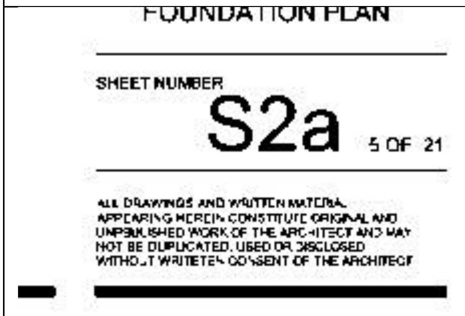
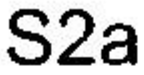
i) HWRatio > T3
ii) Height < T4 (= N1 * Average height)
iii) Width < T5 (= N2 * Average width.

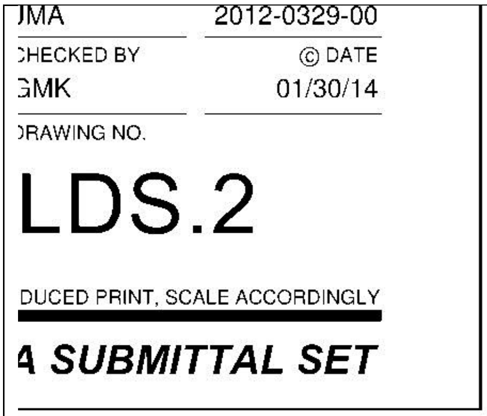
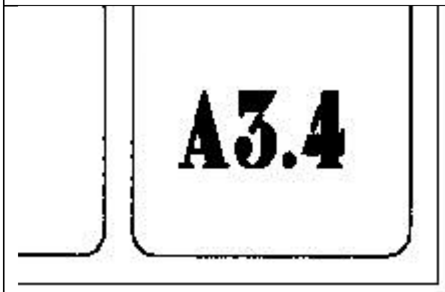
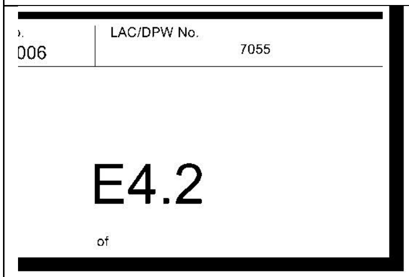
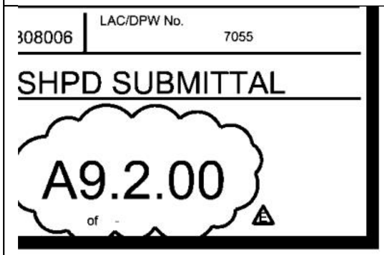
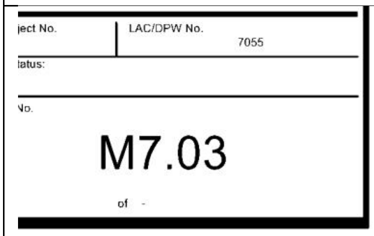
CHAPTER 5: RESULTS

The above procedure generated some satisfactory results but still in some cases it could not perform well and hence some extra measures have to be taken. In this section we are going to present a set of results generated by the process described above. The results below include the correctly detected title block results and also the incorrect interpretation of the test cases, which we have to fix in the next segment of our work.

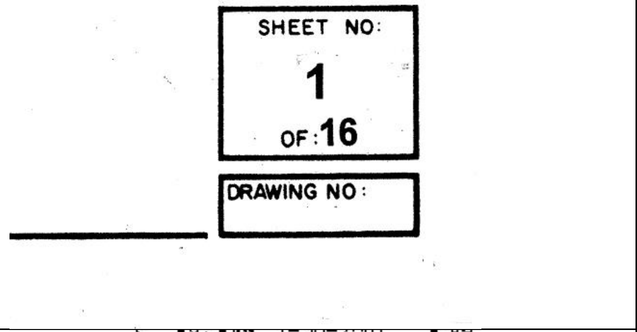
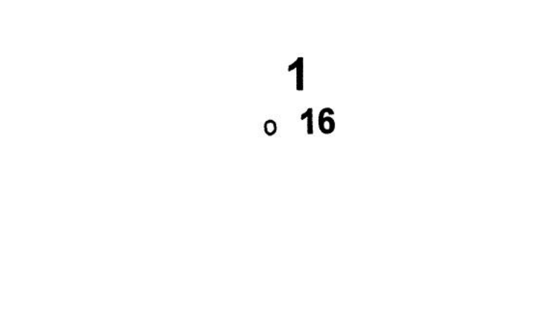
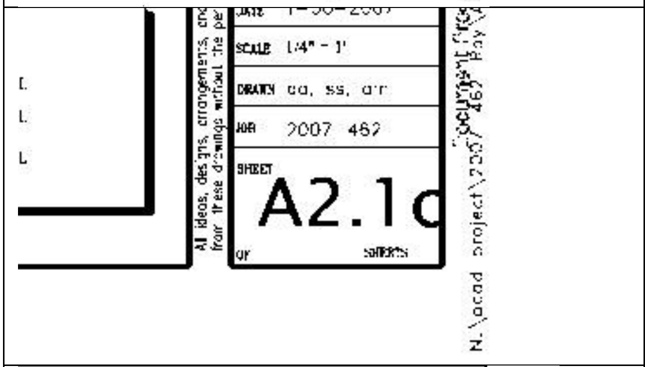
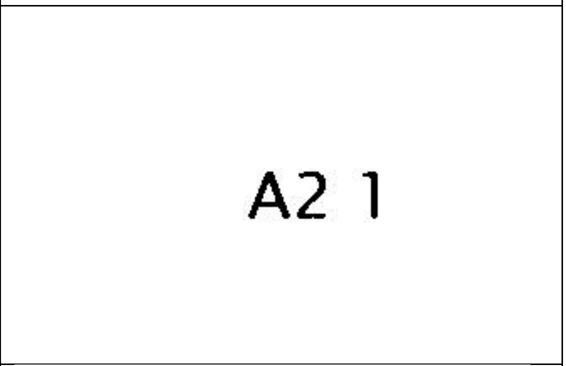
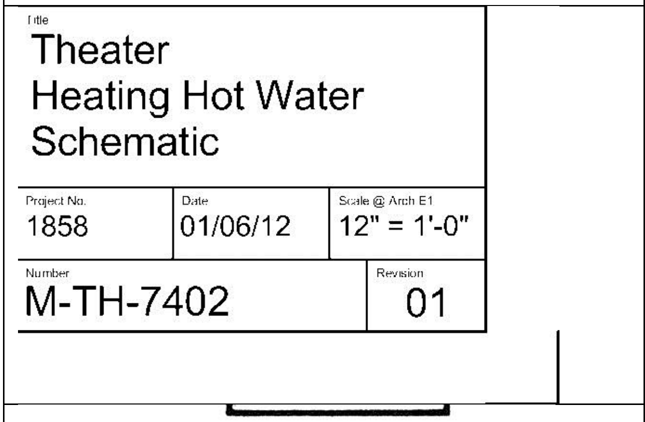
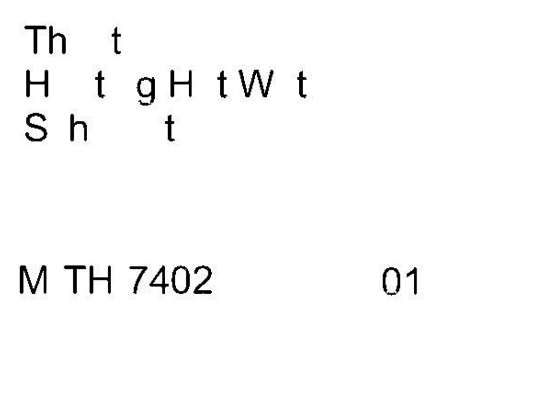
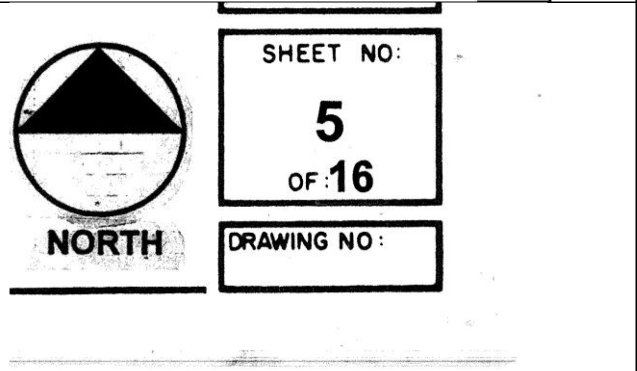
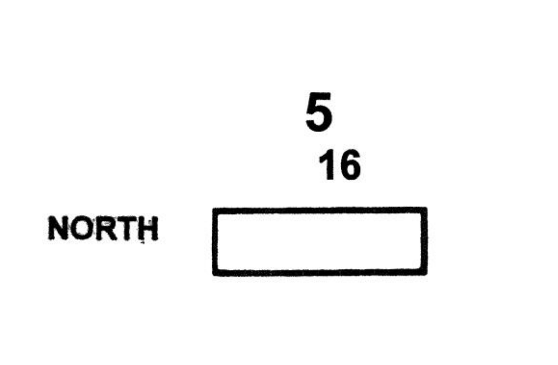
The image set that we are using for testing our system is the one provided by the company ARC Document Solution.

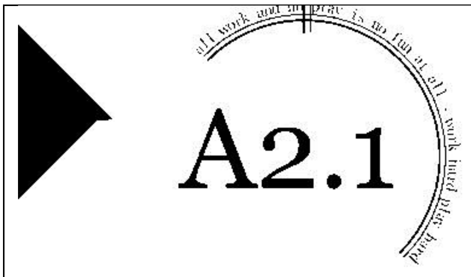
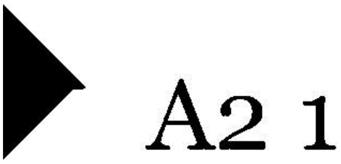
5.1 CORRECT OUTPUT

INPUT IMAGE	OUTPUT IMAGE
 FRAMING PLAN SHEET NUMBER S3b 12 OF 21 ALL DRAWINGS AND WRITTEN MATERIAL APPEARING HEREIN CONSTITUTE ORIGINAL AND UNPUBLISHED WORK OF THE ARCHITECT AND MAY NOT BE DUPLICATED, USED OR DISCLOSED WITHOUT WRITTEN CONSENT OF THE ARCHITECT	 S3b
 SHEET NUMBER S9 21 OF 21 ALL DRAWINGS AND WRITTEN MATERIAL APPEARING HEREIN CONSTITUTE ORIGINAL AND UNPUBLISHED WORK OF THE ARCHITECT AND MAY NOT BE DUPLICATED, USED OR DISCLOSED WITHOUT WRITTEN CONSENT OF THE ARCHITECT	 S9
 FOUNDATION PLAN SHEET NUMBER S2a 5 OF 21 ALL DRAWINGS AND WRITTEN MATERIAL APPEARING HEREIN CONSTITUTE ORIGINAL AND UNPUBLISHED WORK OF THE ARCHITECT AND MAY NOT BE DUPLICATED, USED OR DISCLOSED WITHOUT WRITTEN CONSENT OF THE ARCHITECT	 S2a

 JMA 2012-0329-00 HECKED BY © DATE GMK 01/30/14 RAWING NO. LDS.2 DUCED PRINT, SCALE ACCORDINGLY A SUBMITTAL SET	LDS 2
 A3.4	A3 4
 006 LAC/DPW No. 7055 E4.2 of	E4 2
 308006 LAC/DPW No. 7055 SHPD SUBMITTAL A9.2.00 of -	A9 2 00
 ject No. LAC/DPW No. 7055 tatus: No. M7.03 of -	M7 03

5.2 INCORRECT OUTPUT

INPUT IMAGE	OUTPUT IMAGE
 SHEET NO: 1 OF :16 DRAWING NO :	 1 o 16
 SCALE 1/4" - 1' DRAWN cd, ss, drr JOB 2007 462 SHEET A2.1c OF SHEETS All ideas, designs, arrangements, and from these drawings without the per N:\acad project\2007\462 Roy	 A2 1
 Title Theater Heating Hot Water Schematic Project No. 1858 Date 01/06/12 Scale @ Arch E1 12" = 1'-0" Number M-TH-7402 Revision 01	 Th t H t gH tW t S h t M TH 7402 01
 SHEET NO: 5 OF :16 NORTH DRAWING NO :	 5 16 NORTH

A2.1 all work and no play is no fun at all - work hard play hard	A2 1

CHAPTER 6: CONCLUSION AND FUTURE WORK

The main objective of the project is to automate the detection and conversion of the title block of the scanned Engineering Drawing documents. In this part we have got a good rate of success in detecting the title block design number. The results need a little more refinement both in detection & recognition segments and some more cases has to be considered to get measure of accuracy.

Image pre-processing is at present a little sensitive to height of characters, this dependence needs to be weakened using more statistical measures to set the height parameter. Several improvements can also be made while re-growing boxes. Certain heuristics can be used to get back text parts close-by which were deleted in earlier processing steps. Currently, the speed of processing is quite fast, so there is no issue in real time usage of this system. This is important taking into consideration the fact that we are processing only a small section of the image and further extension of this project will perform pre-processing operation on entire image which is in order of 10000x10000 pixels. Hence, maintaining this speed & increasing accuracy is way forward for this part.

As far as the OCR module is concerned, we have barely started scratching its surface, though the results are pretty good. The Tesseract library is too much flexible & huge. Customization is of utmost importance to achieve higher levels of accuracy. While we have tailored it at present to restrict recognition to certain characters & area in the image, this is not enough. Since we are providing complete image pre-processing, tweaking is required with Tesseract's internal pre-processing operations – making it use only the bare essential. Another most import work would be to train the Tesseract with wide variety of custom fonts that are found in the Engineering Drawing industry. This would ensure that we don't miss any information which is properly processed and given to OCR.

REFERENCES

1. ***"Detection of text regions from digital engineering drawings"***, Lu Z (1998), IEEE Trans PAMI 20(4):431–439

2. ***"Text/graphics separation revisited"***, Tombre K, Tabbone S, Pelissier L, Lamiroy B, Dosch P (2002), In: Document analysis systems V. Lecture notes in computer science, vol 2423. Springer, Berlin/New York, pp615–620

3. ***"Text extraction from graphical document images using sparse representation"***, Hoang TV, Tabbone S (2010), In: Proceedings of the 9th IAPR international workshop on document analysis systems, Boston, pp143–150

4. ***"Block segmentation and text extraction in mixed text/image documents"***, Wahl F, Wong K, Casey R (1982), Computer Graphics & Image Processing 20(4):375–390

5. ***"A robust algorithm for text string separation from mixed text/graphics images"***, Fletcher LA, Kasturi R (1988), IEEE Trans PAMI 10(6):910–918

6. ***"Line structure extraction from line-drawing images"***, Kaneko T (1992), Pattern Recognition Society 25(9):963–973

7. ***"Vector-based segmentation of text connected to graphics in engineering drawings"***, Dori D, Liu W (1996), In: Advances in structural and syntactical pattern recognition. Lecture notes in computer science, vol 1121. Springer, Berlin/New York, pp322–331

8. ***"Automated CAD conversion with the machine drawing understanding system: concepts, algorithms, and performance"***, Dori D, Liu W (1999), IEEE Trans System, Man, and Cybernetics-part A: System and Humans 29(4):411–416

9. ***"Text extraction using pyramid"***, Tan CL, Ng PO (1998), Pattern Recognition Society 31(1):63–72

10. ***"Detection of word groups based on irregular pyramid"***, Loo PK, Tan CL (2001), In: Proceedings of the 6th international conference on document analysis and recognition, Seattle, pp200–204.

11. ***"Text extraction from graphical document images using sparse representation"***, Hoang TV, Tabbone S (2010), In: Proceedings of the 9th IAPR international workshop on document analysis systems, Boston, pp143–150

12. ***"Text segmentation using gabor filters for automatic document processing"***, Jain A, Bhattacharjee S (1992), Machine Vision Applications 5(3):169–184

13. ***"Document image segmentation using wavelet scale-space features"***, Acharyya M, Kundu MK (2002), IEEE Trans Circuits & Systems for Video Technology 12(12):1117–1127

14. ***"An introduction to vectorization and segmentation"***, Doermann D (1998), In: Graphics recognition algorithms and systems. Lecture notes in computer science, vol 1389. Springer, Berlin/New York, pp 1–8

15. ***"Page rule line removal using linear subspaces in monochromatic handwritten arabic documents"***, Abd-Almageed W,Kumar J,Doermann D (2009), In: Proceedings of the 12th

international conference on document analysis and recognition, Barcelona, pp768–772

16. ***"Text line extraction in graphical documents using background and foreground information"***, Roy PP, Pal U, Llad´os J (2012), IJDAR 15(3):227–241

17. ***"Empirical Tests for Feature Selection Based on a Pscychological Theory of Character Recognition "***,B.A. Blesser, T.T. Kuklinski, R.J. Shillman (1976), Pattern Recognition 8(2), Elsevier, NewYork