Optical Character Recognition - A Combined ANN/HMM Approach

Dissertation

submitted to the

Department of Computer Science

Technical University of Kaiserslautern

for the fulfillment of the requirements for the doctoral degree

Doctor of Engineering

(Dr.-Ing.)

by

Sheikh Faisal Rashid

Dean:
Prof. Dr. Klaus Schneider

Thesis supervisors:
Prof. Dr. Thomas Breuel, TU Kaiserslautern
Prof. Dr. Andreas Dengel, TU Kaiserslautern

Chair of supervisory committee:
Prof. Dr. Karsten Berns, TU Kaiserslautern

Kaiserslautern, 11 July, 2014

D 386

Abstract

Optical character recognition (OCR) of machine printed text is ubiquitously considered as a solved problem. However, error free OCR of degraded (broken and merged) and noisy text is still challenging for modern OCR systems. OCR of degraded text with high accuracy is very important due to many applications in business, industry and large scale document digitization projects. This thesis presents a new OCR method for degraded text recognition by introducing a combined ANN/HMM OCR approach. The approach provides significantly better performance in comparison with state-of-the-art HMM based OCR methods and existing open source OCR systems. In addition, the thesis introduces novel applications of ANNs and HMMs for document image preprocessing and recognition of low resolution text. Furthermore, the thesis provides psychophysical experiments to determine the effect of letter permutation in visual word recognition of Latin and Cursive script languages.

HMMs and ANNs are widely employed pattern recognition paradigms and have been used in numerous pattern classification problems. This work presents a simple and novel method for combining the HMMs and ANNs in application to segmentation free OCR of degraded text. HMMs and ANNs are powerful pattern recognition strategies and their combination is interesting to improve current state-of-the-art research in OCR. Mostly, previous attempts in combining the HMMs and ANNs were focused on applying ANNs as approximation of the probability density function or as a neural vector quantizer for HMMs. These methods either require combined NN/HMM training criteria [ECBG-MZM11] or they use complex neural network architecture like time delay or space displacement neural networks [BLNB95]. However, in this work neural networks are used as discriminative feature extractor, in combination with novel text line scanning mechanism, to extract discriminative features from unsegmented text lines. The features are processed by HMMs to provide segmentation free text line recognition. The ANN/HMM modules are trained separately on a common dataset by using standard machine learning

procedures. The proposed ANN/HMM OCR system also realizes to some extent several cognitive reading based strategies during the OCR. On a dataset of 1,060 degraded text lines extracted from the widely used UNLV-ISRI benchmark database [TNBC99], the presented system achieves a 30% reduction in error rate as compared to Google's Tesseract OCR system [Smi13] and 43% reduction in error as compared to OCRopus OCR system [Bre08], which are the best open source OCR systems available today.

In addition, this thesis introduces new applications of HMMs and ANNs in OCR and document images preprocessing. First, an HMMs-based segmentation free OCR approach is presented for recognition of low resolution text. OCR of low resolution text is quite important due to presence of low resolution text in screen-shots, web images and video captions. OCR of low resolution text is challenging because of anti-aliased rendering and use of very small font size. The characters in low resolution text are usually joined to each other and they may appear differently at different locations on computer screen. This work presents the use of HMMs in optical recognition of low resolution isolated characters and text lines. The evaluation of the proposed method shows that HMMs-based OCR techniques works quite well and reaches the performance of specialized approaches for OCR of low resolution text.

Then, this thesis presents novel applications of ANNs for automatic script recognition and orientation detection. Script recognition determines the written script on the page for the application of an appropriate character recognition algorithm. Orientation detection detects and corrects the deviation of the document's orientation angle from the horizontal direction. Both, script recognition and orientation detection, are important preprocessing steps in developing robust OCR systems. In this work, instead of extracting handcrafted features, convolutional neural networks are used to extract relevant discriminative features for each classification task. The proposed method resulted in more than 95% script recognition accuracy on various multi-script documents at connected component level and 100% page orientation detection accuracy for Urdu documents.

Human reading is a nearly analogous cognitive process to OCR that involves decoding of printed symbols into meanings. Studying the cognitive reading behavior may help in building a robust machine reading strategy. This thesis presents a behavioral study that deals on how cognitive system works in visual recognition of words and permuted non-words. The objective of this study is to determine the impact of overall word shape in visual word recognition process. The permutation is considered as a source of shape degradation and visual appearance of actual words can be distorted by changing the con-

stituent letter positions inside the words. The study proposes a hypothesis that reading of words and permuted non-words are two distinct mental level processes, and people use different strategies in handling permuted non-words as compared to normal words. The hypothesis is tested by conducting psychophysical experiments in visual recognition of words from orthographically different languages i.e. Urdu, German and English. Experimental data is analyzed using analysis of variance (ANOVA) and distribution free rank tests to determine significance differences in response time latencies for two classes of data. The results support the presented hypothesis and the findings are consistent with the dual route theories of reading.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Dr. Thomas Breuel for his continuous support in my Ph.D. He is always a source of inspiration to me for his patience, motivation, enthusiasm, and immense knowledge in the field of computer science, pattern recognition, image processing and document analysis. I am also obliged to my second advisor, Prof. Dr. Andreas Dengel for his timely and highly up to the mark feedback and support. His grand knowledge and expertise in the field of artificial intelligence is an asset for me life long. Along with my advisors, I am grateful to the chair of my thesis committee, Prof. Dr. Karsten Berns, for his encouragement, insightful comments, and hard questions.

My sincere gratitude to the most adorable Dr. Faisal Shafait for his kind and nourishing supervision of my PhD. Moreover, Dr. Tandra Ghose must be acknowledged for making my basis in cognitive psychology research. A special thanks to Dr. Marcus Liwicki for providing valuable comments in preparation of my thesis defense.

I am blissful to Dr. Marc-Peter Schambach, Dr. Jörg Rottland, Mr. Stephan von der Nüll for offering me internship opportunity at Seimens Konstanz and leading me working on diverse exciting projects.

It is my pleasure to be part of IUPR research group who nurtures my research in different phases. Special thanks to Joost van Beusekom for presenting my work in SPIE HVEI conference, Sheraz Ahmed and Muhammad Imran Malik, my best buddies, for listening and correcting my defense presentation, and Ingrid Romani for helping me in various administrative issues.

Last but not the least, I would like to thank my family: my mother Zubaida Rashid, to whom I owe so much. Thank you mom teaching me the importance of hard work, for giving me the strength during time of adversity and for your constant love & support. There are not enough words to express how grateful I feel to have you as my mom. My

elder brother, Sheikh Muhammad Idrees, who raised me like a father and guided me in every aspect of my life. My wife & best friend, Dr. Iram Ashraf, for love, patience and ability to always keep me balance. You are my rock and constant source of inspiration, and this project not have been completed without your unwavering support. I love you more and more each day of my life. My daughter, Urooj Faisal –my lucky chum–, whose lovely talks are always a refreshing boost for me. And thanks to whole my family for their prayers and well wishes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Replication of human reading process with the help of machines has been an old area of research in the field of pattern recognition and machine learning [Cle65]. In spite of this early start, machine reading or optical character recognition (OCR) of degraded text (broken or merged characters and presence of noise) without human intervention remains an elusive goal. This thesis introduces a new segmentation free OCR approach using a combination of artificial neural networks (ANNs) and hidden Markov models (HMMs) for degraded text recognition. In addition, it provides novel applications of ANNs and HMMs in document image analysis and recognition. The thesis also contributes in the field of cognitive psychology by presenting new psychophysical experiments to determine the impact of overall word shape and importance of letter positions during visual recognition of words.

Researchers are interested in developing OCR systems due to numerous potential applications in business and industry. The usage of OCR varies across application areas. Banking is one of the widely known area where OCR is used for automatic processing of cheques and other forms. The writing on the cheque can be scanned and recognized instantly to reduce the wait times in banks. OCR systems enable form processing tools to extract and read the relevant information from paper based forms. Medical professionals have to deal with a large volume of forms containing important information about the patients. It is useful to keep up with all the information by putting it into a central database digitally, so that the information can be accessed efficiently as required. The large scale digitization projects need efficient OCR system to convert millions of printed books and documents to digital archives. Digital archives provide searchable access to the content, easy backup facilities and eliminate the need for physical storage of printed documents.

OCR technology is widely used in many other fields like mail sorting, education, finance and government or private offices. It automates the reading of addresses on letters and parcels for efficient mail disbursement. It facilitates in digital archiving of conference proceeding and journals to make them available for on-line access. Invoice imaging tools help in many businesses to keep track of financial records. In offices, it simplifies the collection of data from printed documents for analysis and further usage. In short, OCR technology has revolutionized the document management process in a wide range of industries by turning a scanned document image into a computer readable text document.

OCR systems transform a two dimensional image of text, that could contain machine printed or handwritten text, ideally in any script, from its image representation into machine readable text. OCR systems usually work in a pipeline and there are several steps before actual text recognition takes place [Bre08]. A typical OCR system may comprises of preprocessing, layout analysis, character recognition and language modeling. Preprocessing normally includes binarization, noise removal, skew correction and optionally script and orientation detection. Layout analysis identifies text columns, text blocks, text lines and reading order of the text page. Character recognition is responsible for recognition of the text contained in the text line. Statistical language modeling improves the text recognition results by integrating prior knowledge about language, vocabulary and domain of the document. Figure 1.1 shows a block diagram of a typical OCR system.

Despite decades of research and existence of many off the shelve OCR systems, output from OCR processes often contain errors. The ultimate goal of building a reading machine, having the same reading capabilities as humans have, is still unachieved. Over the past years, researchers have been focusing on the problem of isolated character recognition, assuming that sentences are easy to segment into characters prior to the character recognition. Whereas character segmentation is difficult to achieve in case of low resolution and degraded document images. Document degradation leads to many problems such as adjacent characters can touch each other, a character can be split into many pieces, characters can be distorted due to random noise or ink smears. Such degradations may be present due to many physical process like scanning, photocopying, faxing and aging etc. Figure 1.2 shows parts of some degraded document images from UNLV-ISRI document image database [TNBC99]. The database has been developed on diverse collection of document images from scientific, legal and technical backgrounds.

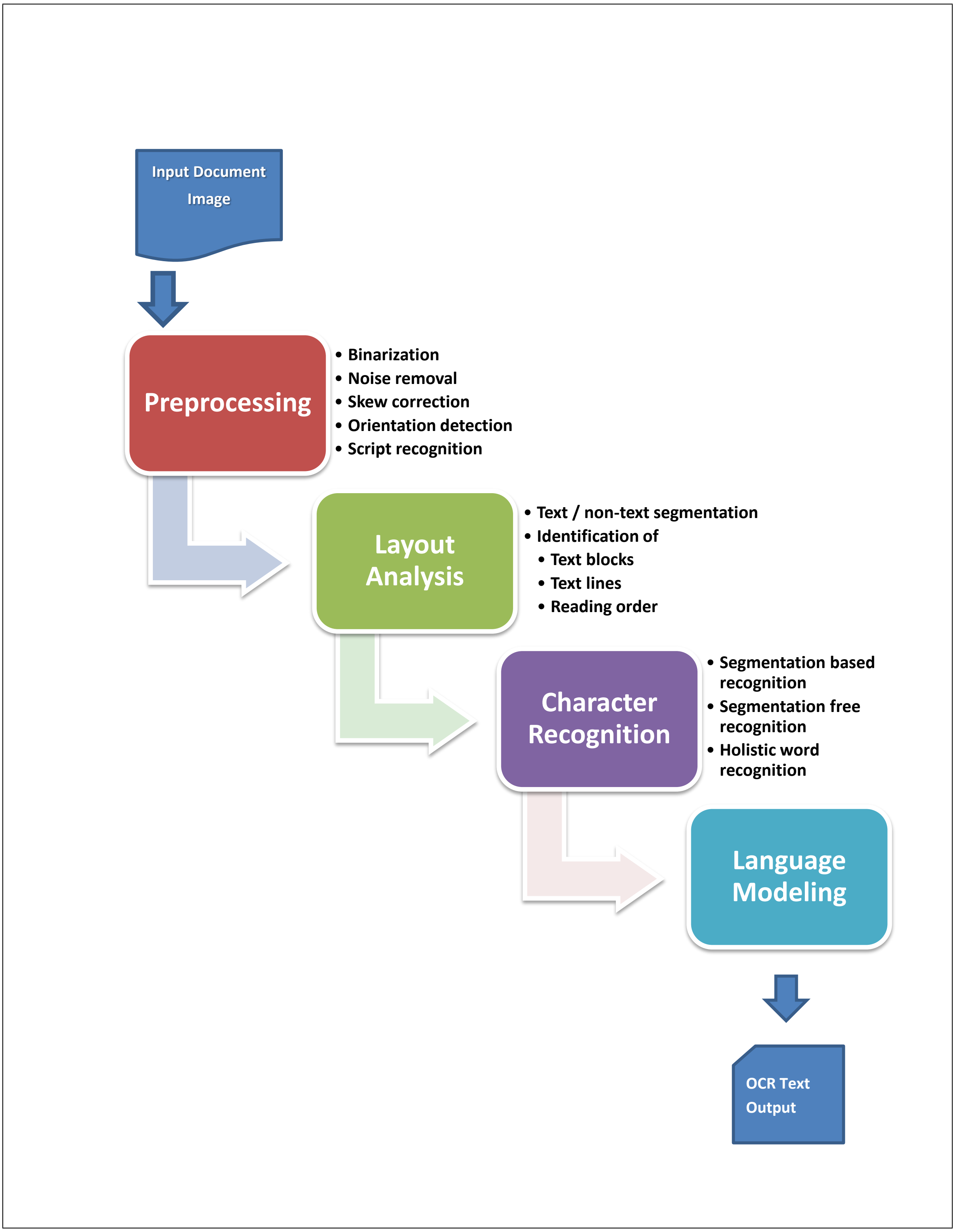
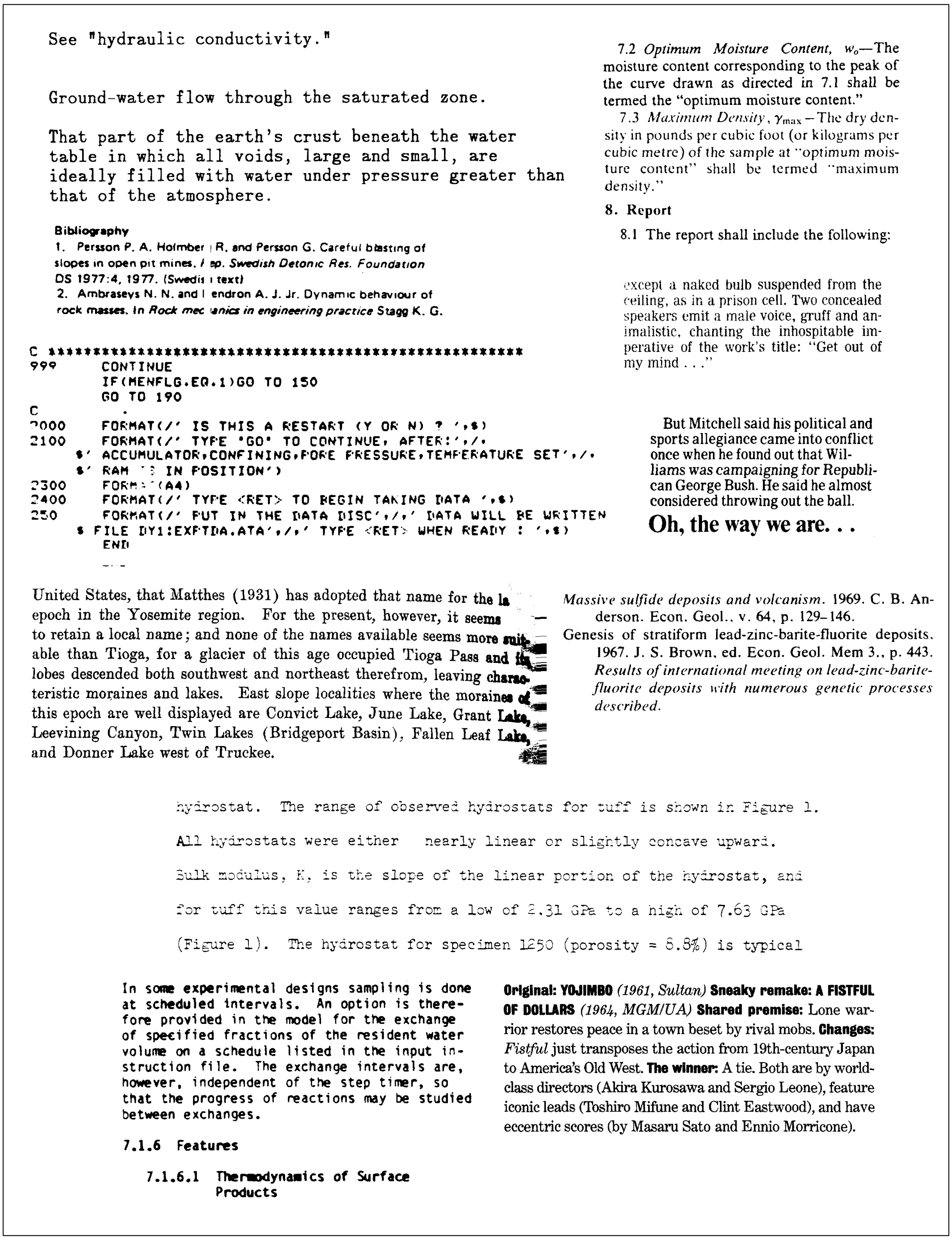


Figure 1.1: A block diagram of a typical OCR System. The diagram shows different intermediate processing steps during OCR process.



See "hydraulic conductivity."

Ground-water flow through the saturated zone.

That part of the earth's crust beneath the water table in which all voids, large and small, are ideally filled with water under pressure greater than that of the atmosphere.

Bibliography

1. Persson P. A. Holmber | R. and Persson G. Careful blasting of slopes in open pit mines. / ep. *Swedish Detonic Res. Foundation* DS 1977:4, 1977. (Swedi: | text)
2. Ambraseys N. N. and | endron A. J. Jr. Dynamic behaviour of rock masses. In *Rock mec anics in engineering practice* Stagg K. G.

```
C ****************************************************
999     CONTINUE
        IF(MENFLG.EQ.1)GO TO 150
        GO TO 190
C        .
?000    FORMAT(/' IS THIS A RESTART (Y OR N) ? ',$)
2100    FORMAT(/' TYPE 'GO' TO CONTINUE, AFTER:',/,
      $' ACCUMULATOR,CONFINING,PORE PRESSURE,TEMPERATURE SET',/,
      $' RAM  : IN POSITION')
2300    FORM: '(A4)
2400    FORMAT(/' TYPE <RET> TO BEGIN TAKING DATA ',$)
250     FORMAT(/' PUT IN THE DATA DISC',/,' DATA WILL BE WRITTEN
      $ FILE DY1:EXPTDA.ATA',/,' TYPE <RET> WHEN READY : ',$)
        END
```

7.2 *Optimum Moisture Content,* w_o—The moisture content corresponding to the peak of the curve drawn as directed in 7.1 shall be termed the "optimum moisture content."

7.3 *Maximum Density*, γ_{max} – The dry density in pounds per cubic foot (or kilograms per cubic metre) of the sample at "optimum moisture content" shall be termed "maximum density."

8. Report

8.1 The report shall include the following:

except a naked bulb suspended from the ceiling, as in a prison cell. Two concealed speakers emit a male voice, gruff and animalistic, chanting the inhospitable imperative of the work's title: "Get out of my mind . . ."

But Mitchell said his political and sports allegiance came into conflict once when he found out that Williams was campaigning for Republican George Bush. He said he almost considered throwing out the ball.

Oh, the way we are. . .

United States, that Matthes (1931) has adopted that name for the la epoch in the Yosemite region. For the present, however, it seems to retain a local name; and none of the names available seems more suitable than Tioga, for a glacier of this age occupied Tioga Pass and its lobes descended both southwest and northeast therefrom, leaving characteristic moraines and lakes. East slope localities where the moraines of this epoch are well displayed are Convict Lake, June Lake, Grant Lake, Leevining Canyon, Twin Lakes (Bridgeport Basin), Fallen Leaf Lake, and Donner Lake west of Truckee.

Massive sulfide deposits and volcanism. 1969. C. B. Anderson. Econ. Geol., v. 64, p. 129–146.

Genesis of stratiform lead-zinc-barite-fluorite deposits. 1967. J. S. Brown, ed. Econ. Geol. Mem 3., p. 443. *Results of international meeting on lead-zinc-barite-fluorite deposits with numerous genetic processes described.*

hydrostat. The range of observed hydrostats for tuff is shown in Figure 1. All hydrostats were either nearly linear or slightly concave upward. Bulk modulus, K, is the slope of the linear portion of the hydrostat, and for tuff this value ranges from a low of 2.31 GPa to a high of 7.63 GPa (Figure 1). The hydrostat for specimen 1250 (porosity = 8.8%) is typical

In some experimental designs sampling is done at scheduled intervals. An option is therefore provided in the model for the exchange of specified fractions of the resident water volume on a schedule listed in the input instruction file. The exchange intervals are, however, independent of the step timer, so that the progress of reactions may be studied between exchanges.

7.1.6 Features

7.1.6.1 Thermodynamics of Surface Products

Original: YOJIMBO *(1961, Sultan)* **Sneaky remake: A FISTFUL OF DOLLARS** *(1964, MGM/UA)* **Shared premise:** Lone warrior restores peace in a town beset by rival mobs. **Changes:** *Fistful* just transposes the action from 19th-century Japan to America's Old West. **The winner:** A tie. Both are by world-class directors (Akira Kurosawa and Sergio Leone), feature iconic leads (Toshiro Mifune and Clint Eastwood), and have eccentric scores (by Masaru Sato and Ennio Morricone).

Figure 1.2: Parts of some document images from UNLV-ISRI document collection showing various kinds of degradations such as breaking characters, ink smearing, poor quality printing, thin strokes, and variable fonts, font sizes and font styles etc.

With some exceptions, most OCR methods are based on segmentation based character recognition approaches, in which words are segmented into isolated characters and characters are recognized individually [DTJT96]. However, in case of degraded and low resolution text, character segmentation is problematic and the performance of character segmentation significantly affects character recognition accuracies. Sophisticated character segmentation techniques have been proposed in past by many researchers [CL96,SREB11], but achieving a good segmentation on various kinds of degradations is still a hard problem. Some other methods take into account word level information and recognize complete words without character segmentation. These methods are generally referred as “holistic word recognition methods” because they do not directly process the characters, but use global word level features like T-junctions, b-loop, d-loop, ascenders and descenders information for the recognition of entire word. However, a drawback of whole word recognition is that it necessarily limited to small vocabularies and are useful only in applications with small static lexicons [MEMY98, CG04]. Recognition based segmentation is another strategy to avoid the segmentation problem during OCR. In these methods input image is divided into many overlapping pieces, resulting into over-segmentation of the image [LLP96, Bre01b, WKJ06]. Usually a hypotheses graph is made using a single character classifier and segmentation boundaries can be drawn using some dynamic programing based procedures.

From last two decades, model based approaches such hidden Markov models (HMMs) have been very popular due to their ability to do segmentation free recognition. HMMs have been proved very successful in automatic speech recognition [Jel97] where segmentation is also a key issue. Over the past years, many HMMs based text recognition approaches has been presented for machine printed, handwritten and cursive script recognition [BSM99,PF09,RRAH^{+}12]. However, HMMs unable to model contextual information completely because of independent observation assumption during state transitions. Another problem of HMMs is the lack of discriminative capabilities, while discriminative models generally give better performance for recognition task [GLF^{+}08]. Artificial neural networks (ANNs) have also been applied to various document image analysis and recognition tasks and good character recognition accuracies have been reported [MGS05]. But most successful application of ANNs are limited to the recognition of isolated characters or digits [LBBH98].

Combination of HMMs and ANNs has also been used effectively in various text and speech recognition problems and is generally referred as *hybrid recognition approach* [BM94].

Development of hybrid systems using HMMs and ANNs usually requires to address two design issues. The first issue concerns with the architectural aspects of ANNs and HMMs, i.e, how structure of the HMMs (discrete or continuous, character or word level HMMs) should be selected, and what kind of ANNs paradigm (multilayer perceptrons, radial basis or time delay neural networks etc.) should be used. The second issue concerns how these two modules can be best trained together. In past many hybrid HMMs/ANNs systems have been revealed using different kinds of HMMs and ANNs structures. In most of the hybrid systems, ANNs are used to augment HMMs either as an approximation of the probability density function or as a neural vector quantizer [BKW^{+}99, MAGD01, KKS00, ECBGMZM11]. Some other hybrid approaches use ANNs to obtain observation probabilities for HMMs [SGH95, BLNB95, KA98]. These hybrid approaches either require combined training criteria for ANNs and HMMs [ECBG-MZM11], or they use complex neural network architectures like time delay neural networks (TDNNs) or space displacement neural networks (SDNNs) [BLNB95].

This thesis addresses OCR problem from various prospects. Main objective of this thesis is to provide segmentation free character recognition of degraded document images. This is done by providing a novel and straightforward combination of artificial neural networks and hidden Markov models to recognize entire text lines without character segmentation. ANNs are best known for their discriminative learning capabilities and are good in static pattern classification, whereas HMMs are powerful statistical tool in learning dynamic time varying properties of a given input signal. By combining these two paradigm, the complementary properties of ANNs and HMMs are used with a view to improve character recognition accuracies in degraded text recognition.

The thesis introduces a simple training mechanism in which ANNs and HMMs are trained separately on a common set of text line images. During recognition, the separately trained modules are combined together to provide segmentation free text recognition of a complete text line. The proposed methods employs a line scanning mechanism using multilayer perceptrons (MLPs) in order to extract discriminative features along with contextual information from the input text line [RSB12]. The features basically represent posterior probabilities of each character class contained in the text line. The posterior probabilities based features are used by already trained HMMs to provide segmentation free character recognition of a complete text line. Interestingly, the proposed approach also realizes to some extent several cognitive reading based strategies like eye fixations, serial scanning of text lines, neural processing, activation of letter units and contextual analysis during OCR. A performance comparison of the proposed method with existing OCR systems and techniques resulted in a 30% reduction in error rate as compared to Google's Tesseract OCR system [Smi13] and 43% reduction in error as compared to OCRopus OCR system [Bre08], which are the best open source OCR systems available today.

Additionally, this thesis evaluates usage of HMMs based segmentation free OCR approaches on low resolution screen rendered text and degraded text line images. Recognition of low resolution text is an interesting OCR problem due to its applications and occurrence of low resolution text in screen-shots, images and videos [RSB11]. Recognition of low resolution text poses many challenges to traditional OCR methods due to touching characters and very small text size. For example, in case of low resolution screen rendered text, x-height of most of the lowercase characters may only be four pixels and width may only be two pixels. Due to anti-aliased rendering process, recognition of low resolution text share the same character recognition problem. The anti-aliased characters are mostly connected to each other and it is difficult to segment them using common character segmentation methods. Therefore, segmentation free text recognition approaches like HMMs are more suitable for this problem. In this work, HMMs based OCR approaches are applied for recognition of isolated screen rendered characters and screen rendered text lines using simple pixel based features. The performance of the proposed method on low resolution text recognition is benchmarked against existing state-of-the-art OCR systems. Evaluation results show that HMMs based approaches reach the performance of other systems in recognition of low resolution text. The proposed HMM based approaches are also evaluated on degraded text line images but these approached are unable to give good character recognition accuracies on degraded text lines.

In addition to OCR, the thesis introduces novel applications of ANNs in document image preprocessing. In this work, ANNs are used for the extraction of discriminative features from raw pixel values of input images. The ANNs based discriminative features analysis is applied to orientation detection and script recognition problems [RSB10b,RBSB09]. Orientation detection and script recognition are two important preprocessing steps in OCR. Orientation detection detects and corrects the deviation of the document's orientation angle from the horizontal direction. Script recognition determines the written script on the page for the application of an appropriate character recognition algorithm. Script recognition is necessary when the OCR system does not have prior knowledge about the language on the page or the page is written in more than one scripts. These useful applications of ANNs in document image preprocessing provides basis for the development of a combined ANNs/HMMs OCR technique, in which ANNs are used as a tool to obtain discriminative features from input text line images.

Another contribution of this thesis is to investigate into cognitive reading process using psychophysical experiments. Human reading is a nearly analogous cognitive process to OCR that involves decoding of printed symbols into meanings. Psychologist are interested to know how readers extract visual information?, what writing is and how it is related to speech and meanings? Is recognition of a word done by recognizing its consists characters or its holistic shape? Even more interestingly, how are humans able to read words even in permuted or jumbled forms. From many years, researchers in the field of cognitive psychology, neurophysiology and linguistics have extensively studied these questions and a large number of theories and reading models exists that explain different aspects of visual word recognition or reading.

This thesis provides a hypothesis about visual recognition of words and permuted non-words [RSB10c]. The hypothesis states that reading of words and permuted non-words are two distinct mental processes and human use different strategies in handling of permuted non-words as compared to normal words. The hypothesis is presented in the context of dual route theories of word recognition and it is observed that dual route theory is consistent in explanation of the hypothesis. The hypothesis is tested for three orthographically dissimilar languages, Urdu, German and English by conducting psychophysical experiments in visual recognition of words and permuted non-words. The outcomes of the experiments lead towards many interesting insights of the reading process that can be used to provide basis for the development of a robust OCR system to recognize an almost innumerable variety of text.

1.1 Contributions of this Thesis

The original contributions of the work detailed in this thesis can be summarized as follows:

- A novel application of state-of-the-art HMMs based OCR methods to recognize low resolution screen rendered text is presented. The proposed method is also evaluated in recognition of degraded text line images.

- A new method for multi-script recognition is described. The method uses convolutional neural networks (CNNs) to extract shape based discriminative features at connected component level. The output of the system is presented in a novel color coded format to distinguish among different script classes in a single line of text. The system is successfully applied in recognition of scripts from Greek-Latin, Arabic-Latin, Antiqua and Fraktur typeface document images.

- A new method to detect document page orientation is proposed. The method employs the same discriminative learning based recognition approach in orientation detection that has been used for the script recognition task. The proposed method is applied to Urdu document images with different page layouts and resulted in 100% page orientation detection accuracy.
- A novel segmentation free OCR method is developed for recognition of degraded printed document images using ANNs and HMMs. The method is based on a novel combination of ANNs and HMMs in which ANNs and HMMs are trained separately on a diverse collection of document image (UNLV-ISRI) text lines to provide best possible character recognition accuracies.
- A new text line image height normalization method is described to preserve typeface characteristics of characters contained in the text line after height normalization.
- A new training methodology is developed for training ANNs on possible character and non-character classes over a complete text line image. The training process also incorporates neighboring contextual information for each character and class using a contextual window of suitable width.
- A text line scanning mechanism is proposed to obtain character class posterior probabilities with the help of specially trained ANNs without character segmentation of the text line.
- A hypothesis about visual recognition of word and non-words is presented. The hypothesis is tested by designing and conducting novel psychophysical experiments in Urdu, German and English languages.

1.2 Thesis Overview

This thesis is divided into six chapters. The current chapter have already provided the aims and contents of the thesis. Chapter 2 describes state-of-the art HMMs based segmentation free OCR approaches and their applications to low resolution screen rendered text and degraded text recognition. Chapter 3 explains ANNs based method to learn discriminative features from input document images. The method is successfully applied to script recognition and orientation detection problems. Chapter 4 outlines the combined ANNs and HMMs based OCR approach for recognition of degraded text. A performance comparison of the proposed method with existing state-of-the-art methods is also provided.

Chapter 5 briefly reviews reading theories and models of visual word recognition and details the hypothesis about visual recognition of words and permuted non-words. It explains the design and implementation of psychophysical experiments for the hypothesis testing and narrates the findings based on statistical analysis of the data.

The final chapter draws the conclusion of the proposed work and provides a brief summary about the achievements in this research.

Chapter 2

Segmentation free OCR using HMMs

This chapter presents a segmentation free OCR approach using hidden Markov models (HMMs). The novel application of the proposed approach is to recognize low resolution screen rendered text. Character segmentation is difficult to achieve in the presence of joined characters because of low resolution. The presented method avoids character segmentation and recognizes complete text line images. The method works directly on gray level images to avoid information loss due to binarization of low resolution text. The method has been evaluated on public and in-house built low resolution text databases. Evaluation results show that the proposed method reach the performance of other methods on low resolution text recognition and yields above 98% character recognition accuracies.

2.1 Introduction

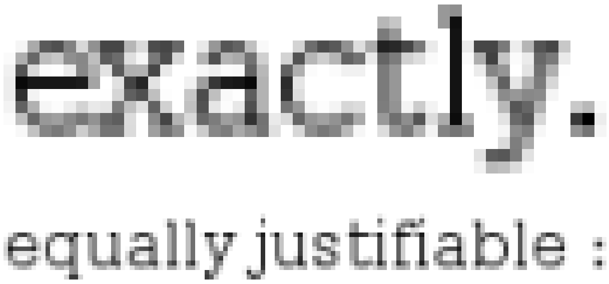
Hidden Markov models (HMMs) have been proved successful in many machine learning and pattern recognition applications for the analysis and modeling of sequential data [Jel97, MB01, PF09]. Researchers have developed efficient pattern recognition systems to recognize speech and handwriting using HMMs [GY07, HBT96, NLS^{+}01, LB06, SZHZ07]. Nowadays HMMs are considered state-of-the-art in statistical methods to provide automatic speech recognition.

Successful applications of HMMs to speech recognition research have provided many useful aspects of the technology to build robust OCR system. For example, an optical character recognition system can be realized without even providing the character segmentation information. Language independence is another attribute, and the same system can be adopted to other languages with no or minor modifications. Today, HMMs are considered powerful tool in text recognition research and many useful systems have been revealed previously. Researcher at BBN Technologies Cambridge, USA, have provided lots of contributions in developing robust HMMs based OCR system. Their work can be considered as one of the pioneers in transforming HMMs based speech recognition system into an OCR system [SLM+96]. The initial work is based on "BYBLOS Continuous Speech Recognition System" [CDK+87] and later the system has been advanced in different directions for system improvements [SLN+99, LSN+99, DNP+05], and for adopting the system to different languages or scripts [BSM99, NSDK03, MNDP04, NMD05]. HMMs have also been very successful for recognizing cursive scripts like Arabic and handwriting.

Another possible benefit of HMMs, which is presented in this chapter, is to recognize low resolution text. Recognition of low resolution text is quite useful due to a wide range of applications. For example, recognition of screen rendered text can facilitate dictionary or language translation tools [Bab97] to provide meanings or translation of text from screen-shots of documents or web images. Other possible applications may include:

- Augmenting screen reading tools for blind or visually impaired people for reading text from screen images where ASCII text is not available on clipboard.
- Recognizing low resolution text in videos [WHL10].
- Automating graphical user interface (GUI) testing tools for correcting the spelling mistakes on GUI screen-shots.
- Useful for correcting web page rendering errors due to bad foreground and background color combination.
- Enabling web indexing tools to capture semantically important information from web images [EIH08].
- Protection against phishing [Wik13] attacks by verifying URLs of potentially important websites against similar looking characters that have different Unicode.

Low resolution text recognition poses several challenges to modern OCR systems due to its small size and anti-aliasing. Low resolution text usually appears in screen-shots, web



exactly.

equally justifiable :

Figure 2.1: Example screen-rendered text images. Character segmentation is difficult due to touching characters and anti-aliasing.

images, video clips and computer screens. The text rendered at a computer screen is often of low resolution (72 or 96 ppi) and has small font sizes. For example x-height of most of the lower case letters may only be four pixels and width may only be two pixels. The rendering process also smooths the low resolution text by means of anti-aliasing. This smoothing process causes another problem of character segmentation. The anti-aliased characters are mostly connected to each other and it is difficult to segment them with the commonly used segmentation methods. However, despite the challenges in recognition of low resolution text by using existing OCR systems, human can easily read low resolution text from various sources. In fact, the smoothing process in rendering low resolution text at computer screens makes the text more legible for human. Figure 2.1 shows some example images taken from screen rendered text lines.

This chapter [1] presents a segmentation free OCR method using HMMs to recognize low resolution text. The proposed method is based on character level hidden Markov models in which character models are trained on text line images without character or word segmentation. The choice of character models gives an advantage of building dictionary free OCR methodology. Text lines are modeled using ergodic HMM topology in which any model can be reached from any other model in finite number of transitions. The method does not perform binarization and works directly on gray scale images. The proposed method uses gray scale pixel features to train character models using written transcriptions of input text line images. The presented method is evaluated for recognition of screen rendered character images, screen rendered text line images, and degraded printed text line images. The printed text line images are taken from UNLV-ISRI document collection that contains documents with moderate degradations and variable fonts.

[1] This chapter is based on the author's work in [RSB11]

The rest of the chapter is organized as follows. A brief review of some existing work on low resolution text recognition is presented in Section 2.2. A short description of HMMs theory is given in Section 2.3. The development of HMMs based OCR approach is outlined in Section 2.4 . The experimental setups and results are discussed in Section 2.5 , followed by a conclusion in Section 2.6.

2.2 Related Work

Mostly OCR approaches are developed to recognize scanned document images, where documents are scanned typically at 150 dpi or higher resolution. These OCR approaches usually segment text lines into individual characters and then recognize the segmented characters individually. The segmentation based approaches are not suitable for the recognition of screen rendered text because the smoothed and small sized screen rendered characters are difficult to segment. HMMs, due to their segmentation free recognition capability, are good choice for this problem. As mentioned before, almost first segmentation free HMM based OCR approach has been introduced by the researchers at BBN Byblos. The BBN OCR system uses the Byblos HMM engine which was originally developed for speech recognition [SLM+96]. The BBN OCR system models each character with 14 states left-to-right HMM with self loops and skips. Each state has an associated output probability density over the features. The probability density is modeled as a weighted sum of Gaussians in the feature space. The BBN OCR system processes the input documents for the extraction of text lines. Features are extracted from extracted text lines using sliding window approach in which text lines are divided into overlapping frames and cells. Each frame is a narrow vertical strip whose width is a small fraction ($\frac{1}{15}$) of the height of the normalized text line. Each frame is further divided into 20 overlapping cells. The frame width, frame overlap and cell overlap are the system parameters. Feature vectors are constructed at each horizontal position along the text line by computing intensity (percentage of black pixels within each cell), vertical derivative of intensity, horizontal derivative of intensity, local slope and correlation in a window of 2 cells square. This resulted in a set of 80 features per frame. The BBN OCR system is trained using Baum-Welch algorithm that aligns the feature vectors with the character models to obtain maximum likelihood estimates of HMM parameters. The HMM parameters are the means and variance of the component Gaussians in the Gaussian mixture model of the state output probabilities, the weight of the mixture component and the state transition probabilities. During recognition process, the system search for the sequence of characters that is most likely given the feature vector sequence and the trained character models, in accordance with the constraints imposed by the language modeling and orthographic rules. The system is trained and evaluated on a set of 2441 text zones taken from University of Washington English Document Image Database I (UWI) [PCH93]. The system is reported to achieve overall 98.6% character recognition accuracy using a closed vocabulary lexicon of about $20,000$ words taken from all the words in the training and test sets. The BBN OCR system is a word based system and it recognizes only the set of words that constitutes the lexicon of the system. To overcome this problem, some further experiments were performed in which character level recognition is evaluated using a lexicon of possible characters instead of words. The language model is the n-gram (bi-gram or tri-gram) on sequence of characters [BSM99]. In these experiments the system is reported to achieve overall 96.7% and 97.1% character recognition accuracies using character bi-gram and tri-gram respectively. In comparison, the same system obtained 99.2% character recognition accuracy using a word bi-gram of $30,000$ words lexicon. The BBN OCR system was also evaluated on faxed documents dataset. The dataset was created by randomly selecting some documents from UWI database. The selected documents were first printed on paper and the printed images were then faxed from one paper fax machine to another. The faxed documents were scanned into images on the computer. This resulted in a faxed documents dataset that mirrors the clean data from UWI database. The system is reported to obtain overall 97.3% and 99.4% character recognition accuracies on faxed documents and corresponding clean documents respectively by using a lexicon extracted from training and testing corpus [BNS$^+$99].

Einsele et al. [EIH07], [EIH08] proposed HMM-based methods to recognize low resolution character and word images taken from screen-shots of web pages. The potential application is to provide an OCR facility for web indexing tools to extract and index semantic information present in the web images. Both methods use character level HMMs with single state topology and self loop. However, the former method employ an additional inter-character model to capture noise between adjacent characters. Words from screenshot images are recognized by combining different character models in ergodic structure. First and second order moment based features are computed by sliding a 2 pixels wide window in writing direction of word images. The method is evaluated on $3,000$ word images synthetically generated by using two font families in variable sizes. This method achieved overall 96% recognition accuracy at 10 pts font size, but the accuracy drops to 90% at 6 pts font size. Despite the use of HMMs, the method has few drawbacks. The method uses only 26 Latin characters in lower case, limiting its use on real word data that may contain words with upper and lower case characters, numerals and punctuations etc. Though the choice of recognizing word images is OK, but HMMs based approaches may be more suitable to recognize complete text lines instead of isolated characters or words. The recognition of completed text lines is more realistic use case to include all complications of segmentation and handling of spaces among constituted words.

Wachenfeld et al. [WKJ06] presented a hybrid classification and segmentation approach for recognition of screen rendered characters and words. Their method recognizes screen rendered words at a given screen position. The system requires click point coordinates and a screen-shot image for recognition. The recognition result for a given click point is a word candidate list ordered by plausibility. The method uses recognition based character segmentation which was initially proposed by Lee et al. [LLP96]. The main step in this method is the soft segmentation of the word images into sequence of smaller components. This resulted into over segmentation of the words. The over segmentation was performed with the help of dynamic programming based approach that was initially used for segmentation of handwritten text [Bre01b]. In their work, Wachenfeld et al. used a different splitting cost and some additional rules which force or prevent further splitting. For example, they used the knowledge that screen text is often only one pixel thick and connected only by 8-neighborhood. After splitting, the resulting segments were instantly classified by using a gray scale classifier. Further splitting was stopped if a segment was classified with higher plausibility or its width falls bellow a certain threshold. Some other rules forced the over segmentation of the segments that have been classified as a specific character, regardless of a high classification plausibility to avoid segmentation ambiguities among certain classes e.g. “cl” vs. “d” and “rn” vs. “m”. The extracted segments were normalized by a projection of its bounding box onto a uniform square having 1 x 1 dimensions. The projected segment was further divided into 5 x 5 zones. A 25 dimensional feature vector was constructed by computing the average gray value for each zone. The feature vector was used to compute the plausibility of a certain character. The plausibility is the distance between the corresponding feature and the character class, normalized by the maximum distance between any two features in the character database. The distance between the feature of a segment and the class was computed as Euclidean distance to the closest sample of a class. A 98.91% recognition accuracy has been reported on a dataset of 15,808 non-italic, non-bold characters using leave-one-out evaluation technique.

Text images taken by low cost mobile phone or web camera are also resulted in low resolution text. Jacobs et al. [JSVR05] presented a convolutional neural network approach for recognition of low resolution text (10 pts font size) captured by 1024 x 768 resolution web-cam. They trained a convolutional neural network based character recognizer. The recognizer takes a 29 x 29 window of image data as input, and gives a vector containing a probability distribution over the set of characters in the alphabet. The input text images were normalized vertically to the height of 29 pixels. To find the probability distribution for any point on the word, they extracted a 29 x 29 image from the normalized word image, and used that as the input to the character recognizer. This system was trained by taking example characters from 15 document images captured by a web-cam. The documents in the training data included 6 different fonts (both serif and sans-serif) in regular, bold and italic styles, and a variety of font sizes. Word recognition was performed by applying the character recognizer at different locations on the word image, divided into slices to obtain the probabilities of characters on these locations. A dynamic programming based algorithm was used to determine the best possible sequence of characters for the entire word. They used a dictionary of $367,744$ English words and tested the recognizer using data similar to the training data. For alphabetic characters, the character recognizer got 87% accuracy. For alphanumerics, the accuracy dropped down to 83% and with addition of all of the punctuations, the accuracy dropped further to 68%. On the images taken by the 1024 x 768 APLUX camera and 10-point text the system was able to recognize words in the vicinity of $80 - 95\%$ word accuracy.

Yanadume et al. [YMIM04] presented a character recognition algorithm for very low-resolution video data. The proposed method used multi-frame images to integrate information from each image based on a subspace method. The character images used for training and recognition have been captured by a portable camera. Dataset was prepared by printing the characters on a sheet with a fixed print pitch and each character was segmented by using this pitch information. The data was captured at various resolutions by changing the distance between the camera and the sheet in variable sizes. The size of the segmented characters was normalized before training or recognition. They constructed a subspace for each character category by using eigenvectors. The computed eigenvectors from the training data were used to recognize the input characters. They achieved 92% character recognition accuracy with phone camera and 99.9% with DV camera that provides good quality images.

2.3 Hidden Markov Models

Hidden Markov models (HMMs) are statistical models in which system being modeled is considered as a Markov process that has unobserved or hidden states [Rab90]. In hidden Markov models, the state is not directly visible, but only the sequence of observations influenced by the state at a particular instance of time are visible and hence the name hidden Markov Models. A detailed description of HMMs can be found in [Rab90] and [Fin08]. However, a brief theoretical explanation is given in the following subsections.

2.3.1 Definition

Hidden Markov models are two stage stochastic processes with hidden and visible observations. The first stage describes a discrete Markov process that has a finite number of states and state transition probabilities. The transition from one state to the other is restricted and depends on the immediate predecessor state only. This Markov process is said to be of first order i.e.

$$P(s_t|s_1, s_2, s_3, ..., s_{t-1}) = P(s_t|s_{t-1}) \tag{2.1}$$

In the second stage, a series of outputs or emissions is generated for every point in time. In this case, for example at time instance "t", only the output O_t is visible but the sequence of internal states s_t is unknown. These visible outputs are called the "observations" or "emissions" and the overall model is called hidden Markov model.

$$P(O_t|O_1...O_{t-1}, s_1...s_t) = P(O_t|s_t) \tag{2.2}$$

Formally, a *first order* hidden Markov model "λ" can be defined with the following elements: [Rab90, Fin08]:

- a finite set of states $S = \{s_1, s_2, s_3, ..., s_N\}$, where N is the number of states in the model.
- a finite set of distinct observation symbols $O = \{o_1, o_2, o_3, ..., o_M\}$, where M is the number of distinct observations per state.
- a matrix of state transition probabilities $A = \{a_{ij}|a_{ij} = P(s_t = j|s_{t-1} = i)\}$

- a vector of start probabilities $\pi = \{\pi_i | \pi_i = P(s_1 = i)\}$
- the state specific output probability distribution
  $\{b_j(O_t) | b_j(O_t) = p(O_t | s_t = j)\}$ or
  $\{b_j(X) | b_j(X) = p(X | s_t = j)\}$

where $\{b_j(O_t) | b_j(O_t) = p(O_t | s_t = j)\}$ are for discrete emissions and $\{b_j(X) | b_j(X) = p(X | s_t = j)\}$ are for continuous emissions. In most of the cases the generated outputs are discrete in nature like $o_1, o_2, ...o_M$ and the quantities $b_j(O_t)$ represent discrete probability distributions. These emission probabilities can be grouped together in a matrix B:

$$B = \{b_j k | b_j k = P(O_t = o_k | S_t = j)\} \tag{2.3}$$

If the output observations are continuous like $x \in \mathbb{R}^n$ then the output can be represented as continuous density function with $b_j(x) = p(x | St = j)$. An HMM can be described as discrete or continuous based on these output distributions. In case of continuous HMMs, the output probability distributions are usually approximated using state specific mixture of Gaussians [PF09].

2.3.2 Three Basic Problems for HMMs

There are three basic problems that must be solved for effective usage of HMMs in real-world applications [Rab90]. These are:

1. The evaluation problem: Given the observation sequence $O = O_1 O_2 O_3 .....O_T$ and a model $\lambda = (A, B, \pi)$, how can the $P(O|\lambda)$, the probability of observation sequence, given the model be computed efficiently? or that how well a given model matches a given observation sequence? The later view is quite useful and a solution to this problem helps us in finding the model that best matches the observations among several competing models.
2. The decoding problem: Given the observation sequence $O = O_1 O_2 O_3 .....O_T$ and the model λ, how can a corresponding state sequence $S^* = s_1 s_2 s_3 ......s_T$ which is optimal in a sense that it best “explains” the observations be chosen?
3. The optimization problem: How can the parameters of model $\lambda = (A, B, \pi)$ to maximize the $P(O|\lambda)$ be adjusted?

2.3.3 Solutions to the HMMs Problems

Evaluation

It is desired to calculate the probability of the observation sequence, $O = O_1O_2O_3.....O_T$, given the model λ. The brute force approach for this problem is to enumerate every possible state sequence of length T (the number of observations). This solution is not feasible because for N states and T time slots it requires $2TN^T$ computations. Therefore, a more efficient procedure to solve this problem is required. Fortunately, the problem can be solved more efficiently using Forward and Backward algorithms. The algorithms were probably first described in [RJ86] and a summary of the algorithms is presented here.

Forward Algorithm: Consider a forward variable $\alpha_t(i) = P(O_1, O_2, O_3, ..., O_t, s_t = i|\lambda)$ i.e., the probability of generating the partial observation sequence, O_1 to O_t, until time t and reaching at state i.
The solution for the $\alpha_t(i)$ can be given recursively, as follows:

1. Initialization:
   $\alpha_1 = \pi_i b_i(O_1)$
2. Recursion:
   for all points in time $1, ...t, ...T - 1$:
   $\alpha_{t+1}(j) = \sum_i \{\alpha_t(i)a_{ij}\}b_j(O_{t+1})$
3. Finalization:
   $P(O|\lambda) = \sum_i^N \alpha_T(i)$

Where in step 1 the forward variable is initialized to the probability that the first observation O_1 is occurred in state i. And now, the $\alpha_{t+1}(j)$ can be recursively computed by multiplying the $\alpha_t(i)$ with the transition probability a_{ij} for transitioning from state i to state j for all the states i and weight the result with $b_j(O_{t+1})$ to generate the observed emission at time $t+1$. By this way, the observation probability $P(O|\lambda)$ can be computed by summing over the $\alpha_T(i)$]. The complexity of "Forward Algorithm" is N^2T, which is feasible for the practical applications of HMMs.

Backward Algorithm: Similarly, a backward variable can be defined
$\beta_t(i) = P(O_{t+1}, O_{t+2}, O_{t+3}, ..., O_T|s_t = i, \lambda)$ i.e., the probability of the partial observation sequence, from time $t + 1$ to the end, given the state i and the model λ.

Again solution can be given for the $\beta_t(i)$ recursively, as follows:

1. Initialization:
   $\beta_T(i) = 1$

2. Recursion:
   for all points in time $T-1, ..., t, ...1$:
   $\beta_{t+1}(i) = \sum_j a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$

3. Finalization:
   $P(O|\lambda) = \sum_i^N \pi_i b_j(O_{t+1}) \beta_{t+1}(j)$

Where in step 1 the $\beta_T(i)$ was arbitrary initialized to be 1 for all the states i. Step 2 shows that in order to be in state i at time t, and to account for the observation sequence from time $t+1$ on, there is need to consider all possible states j at time $t+1$, all transitions from state j to state i i.e.,a_{ij}, the observation probability in state j i.e., $b_j(O_{t+1})$ and remaining partial observation sequence from state j i.e., $\beta_{t+1}(j)$. Finally, the observation probability $P(O|\lambda)$ can be computed by summing over the $\beta_1(i)$ The complexity of “Backward Algorithm” is also N^2T. The combination of forward and backward algorithms is extensively used to solve the different HMMs problems and it is usually referred as “Forward-Backward-Algorithm” [Rab90]

Decoding

There are several possible solutions to the 2nd problem because of the “optimal” state sequence associated with the given observation sequence and there may exist more than one solutions for that. However, a single best solution can be found by using dynamic programming based procedure called Viterbi algorithm [Vit67, For73]. A detailed description of Viterbi algorithm for decoding problem is described in [RJ86, Rab90]. The algorithm works by finding the most probable hidden state sequence by traversing through a Trellis with traceback information and determine the best overall path by backtracking. Consider the following probability as:

$$\delta_t(i) = \max_{s_1, s_2, ..., s_{t-1}} P(O_1, O_2, ...O_t, s_1, s_2, ..., s_{t-1}, s_t = i|\lambda) \tag{2.4}$$

where $\delta_t(i)$ is the maximum probability along a single path, at time t, for the first t observations and ends in state i. The distribution at time $t+1$ can be obtained as follow,

$$\delta_{t+1}(j) = \max_i \delta_t(i)a_{ij} \cdot b_j(O_{t+1}) \tag{2.5}$$

Now in order to get the most probable state sequence, the above equation is maximized for each t and j. So $\psi_t(j)$ can be defined as "back pointers" along the partial path that define the optimal predecessor state for each $\delta_t(j)$. The complete algorithm can be stated as follows:

1. Initialization:
   $\delta_1(i) = \pi_i b_i(O_1)$
   $\psi_1(i) = 0$

2. Recursion:
   for all points in time $1, ...t, ...T - 1$:
   $\delta_{t+1}(j) = \max_i \delta_t(i)a_{ij} \cdot b_j(O_{t+1})$
   $\psi_{t+1}(j) = \text{argmax}_i\{\delta_t(i)a_ij\}$

3. Finalization:
   $P^*(O|\lambda) = P(O, s^*|\lambda) = max_i\delta_T(i)$
   $s^*_T = \text{argmax}_j\{\delta_T(j)\}$

4. Backtracking:
   for all points in time $T - 1, ..., t, ...1$:
   $s^*_t = \psi_{t+1}(s^*_{t+1})$

Optimization

The third and the most difficult problem of HMMs is to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize the $P(O|\lambda)$. There is no analytical solution exist to estimate the model parameters from finite set of observation sequence or training data. However, a model parameters such that $P(O|\lambda)$ can be locally maximized by using Baum-Welch algorithm [BPSW70]. The algorithm uses the total production probability $P(O|\lambda)$ as optimization criterion and it improves the given model λ depending on training data O in such a way that the optimized model generates the training set with equal or greater probability i.e.,

$$P(O|\lambda') \geq P(O|\lambda)$$

The algorithm used the forward-backward-procedures for improving the estimates of model's parameters by determine the posterior probabilities $\gamma_t(i)$ of being in state i at time t and $\gamma_t(i,j)$ of transitioning from state i at time t into state j at time $t+1$ as follows:

$$\gamma_t(i) = P(S_t = i|O,\lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} \tag{2.6}$$

$$\begin{aligned}
\gamma_t(i,j) &= P(S_t = i, S_{t+1} = j|O,\lambda) && (2.7)\\
&= P(S_t = i, S_{t+1} = j, O|\lambda) && (2.8)\\
&= \frac{\alpha_t(i)a_{ij}(O_{t+1})\beta_{t+a}(j)}{P(O|\lambda)} && (2.9)
\end{aligned}$$

By this way, updated approximations $a_{ij}^{'}$ can be calculated for the transition probabilities by dividing the expected value of transitions from state i to state j by the expected value of all transitions from state j. Now improved emission probabilities $b^{'}(o_k)$ can be obtained by calculating the ratio between the expected value of emission of a specific symbol o_k in state j and the expected value of all the emissions in state j. The new state starting probabilities $\pi_i^{'}$ for state i can simply be set to the probability $\gamma_1(i)$ of being in state i at time 1. The process can be iterated until the model $\lambda^{'}$ does not improve the production probability $P(O|\lambda)$ by more than to a predefined threshold ε. The algorithm is given as follows:

1. Initialization:
   Select of an appropriate starting model λ.
   $\lambda = (\pi, A, B)$

2. Optimization:
   Update the model parameters such that,
   $\lambda^{'} = (\pi^{'}, A^{'}, B^{'})$,
   where,
   $\pi_i^{'} = \gamma_1(i) \qquad a_{ij}^{'} = \frac{\sum_{t=1}^{T-1}\gamma_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \qquad b_j^{'}(o_k) = \frac{\sum_{t:O_t=o_k}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$

3. Termination:
   if:
   $P(O|\lambda^{'})\ P(O|\lambda) + \varepsilon$,

then: $\lambda := \lambda^{'}$ and go to step 2,
else: terminate

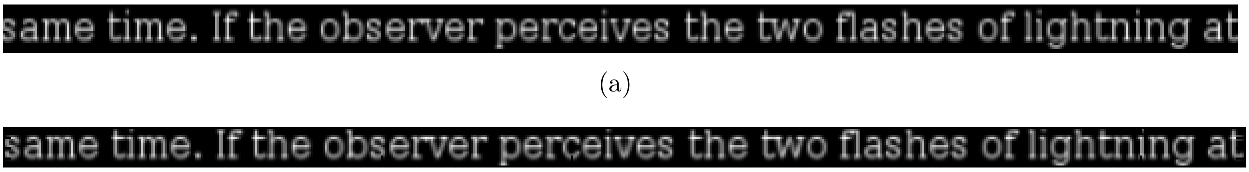
2.4 HMMs based OCR System

This section outlines the proposed HMM based OCR system. HMMs models the variations in printed text as an underlying probabilistic structure, which is not directly observable. As described before, the structure consists of set of states and state transition probabilities. In addition, the system observations on input image are represented as random variables whose distribution depends on a particular state. These observations are taken in form of sequential representation of the input image. HMMs can model the variability of a feature vector as a function of one independent variable. In speech, time is the natural choice of independent variable. However, in OCR, there are two independent variables because text images are two dimensional (2-D). Fortunately, the 2-D text line of image can be represented into 1-D sequence of features by moving a sliding window along the particular text line images. The extracted features are used to train character models on complete text line images without prior character segmentation.

The proposed system consists of preprocessing, feature extraction, HMMs training and recognition. The recognition step is performed to evaluate the trained models on the text lines that are not part of the training dataset. The system is trained and evaluated on two different text line datasets, screen rendered text line images and degraded printed text line images. The execution of the proposed method is same for both kinds of datasets, except different values have been selected for some data dependent parameters. The OCR system can be divided into two functional units i.e., training and recognition. Both training and recognition units use the same preprocessing and feature extractions steps.

2.4.1 Preprocessing

Preprocessing is almost one of the basic step in every text recognition system. Usually it is used to remove noise and different variations in the data. It may include binarization, noise removal, skew correction, image enhancement and data normalization etc. In this work, preprocessing is used to normalize height of text line images. This is done by rescaling image height of each input text line image to a fixed line height. The normlaization



same time. If the observer perceives the two flashes of lightning at

(a)

same time. If the observer perceives the two flashes of lightning at

(b)

Figure 2.2: Text line normalization. a) Original text line image; height = 24, width = 617. b) Normalized text line image; height = 20, width = 622.

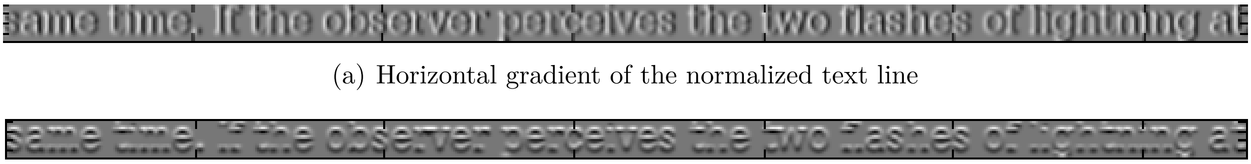
process takes care of the image aspect ratio and the image width is rescaled proportional to the new image height. Figure 4.2 shows an example text line image in original and after image height normalization. Screen rendered text lines are normalized to the height of 20 pixels and this value is determined empirically over the training dataset.

2.4.2 Feature Extraction

HMMs works on sequential data and text line images can be converted into a sequential representation by moving a sliding window along the text line images in writing direction [KCGM93,SLM+96]. Usually the window is only a few pixels wide, which is normally less than the width of a character. During this process small vertical slices or frames are extracted from the text line image. The frames may overlap to some degree depending on movement of the window along the text line. The sequence of image frames provides the basis for subsequent feature extraction. The current method also uses a sliding window to evaluate two different features–gray scale pixel feature and gradient based intensity feature–for the recognition of screen rendered text line images.

Gray scale pixel feature

Gray scale pixel features are raw pixel values taken from input text line images. The input text lines are first normalized as described in Section 3.2.2. A one pixel wide window is moved over the normalized text line and feature vector is constituted by picking up pixel values of each window. The method computes 20 features per window, as the input images are normalized to 20 pixels line height. These pixel based feature are collected for each input text line image and later used for HMMs training along with their ASCII transcriptions.



(a) Horizontal gradient of the normalized text line

(b) Vertical gradient of the normalized text line

Figure 2.3: Horizontal and vertical gradients of the normalized input image are used to compute gradient based intensity features.

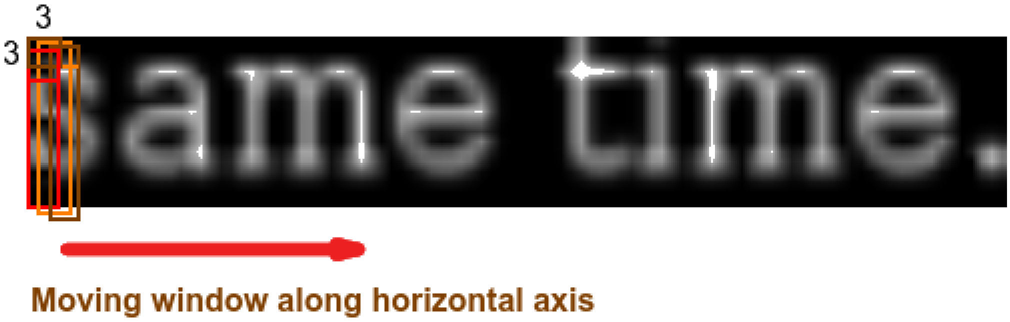


Figure 2.4: Example of sliding window approach: The text line image to be analyzed is shown with overlapping windows and overlapping cells in horizontal and vertical directions. Features are computed from each overlapping (3 x 3) cell, from top to bottom, in every analysis window.

Gradient based intensity features

The gradient based intensity features are extracted from normalized text lines and their corresponding horizontal and vertical gradients. The extracted features consists on three sets of values, the pixels intensity count (representing the blackness in each cell), horizontal gradient of intensity (across overlapping windows), and vertical gradient of intensity (across overlapping cells). The gradients are computed with the help of Sobel operator [GW06]. The gradients capture change in pixel intensities in horizontal and vertical directions of the text line image. Figure 2.3 shows horizontal and vertical gradients of an example normalized text line. The normalized text line and it's corresponding gradients are divided into sequence of small overlapping windows along horizontal direction. Each window is further divided into small overlapping cells along vertical direction. Figure 2.4 shows the example of some overlapping windows and overlapping cells. The window width, cell height and their overlaps are system parameters and their values can be set empirically. In current system, the window width and cell height are fixed to 3 pixels i.e. dimensions of each cell are 3 x 3. A two pixels overlap is used in consecutive windows

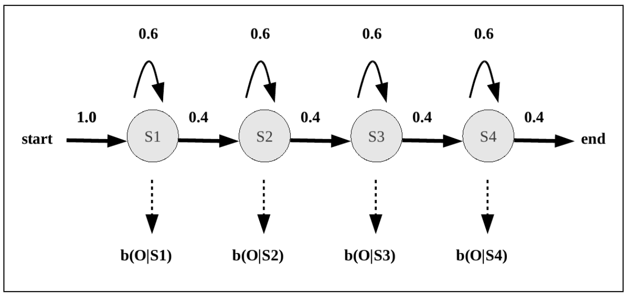


Figure 2.5: Four states left to right HMM topology for modeling characters in screen rendered text lines.

(along horizontal axis) as well as in consecutive cells (along vertical axis). Feature values are computed from top to bottom in each window by counting number of pixel intensities that are greater than zero in each cell. Similarly, the vertical and horizontal gradients of the text lines are divided into overlapping frames and cells, and the change in gray level intensities are measured from top to bottom by counting number of positive and negative gradients in each cell of a particular window. A feature vector is build by combining the values computed from each window of the normalized text line and it's horizontal and vertical gradients.

HMMs Training

The proposed OCR system models each character with a multi-state continuous density hidden Markov model. Transitions from one state to other state are carried out from left to right. Each state in the model is associated with an output probability distribution over the features. The output probability distributions are modeled as a weighted sum of Gaussians, also called Gaussian mixtures. In this method, a Gaussian mixture is parameterized by the means and variances of the component Gaussians and the weight of each Gaussian in the mixture. The number of states and allowable transitions from one state to other are the system parameters. The values of these parameters can be adjusted in start of training process. The number of states that are adequate for each character model depends on the horizontal variability of each character. The number of states may vary from one character model to other, however for sake of simplicity, every character is modeled with a fixed number of states. The work uses 4 state, left to right HMMs with the topology shown in Figure 2.5. The number of Gaussians per state are determined empirically. Initially 2 Gaussian mixture components per state are selected and the value is doubled after every training iteration. A good recognition performance

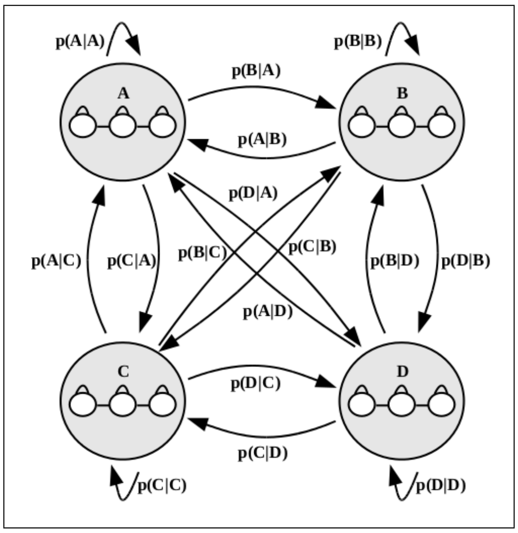


Figure 2.6: Ergodic HMM topology. In this topology any character model can be reached from any other model in finite number of transitions.

is obtained with 256 Gaussians per state. During training process means, variances and weights of the Gaussian mixtures are learned using Baum-Welch algorithm, which is also known as forward-backward or expectation-maximization (EM) algorithm [Rab90, YKO$^+$06, Fin08]. The algorithm iteratively aligns the feature vectors with the character models in order to obtain maximum likelihood estimates of HMMs parameters. The training is performed using a training dataset of text lines and corresponding ground truth files. The ground files are just the sequence of characters in ASCII format. The advantage of EM algorithm is that, no character segmentation information is required and the algorithm automatically aligns the sequence of feature vectors along the text line with the sequence of characters in the ground truth files. In the HMMs topology as shown in Figure 2.5, “start” and “end” are non-emitting states. These states are used to provide transitions from one character model to other character model in the system.

2.4.3 Recognition

During recognition phase input text lines are preprocessed and features are extracted as described in the above sections. The recognition process searches for the sequence of character models that has the highest probability of generating the observed sequence of feature vectors. The process requires the trained character models, a statistical language model and a possible lexicon of words or characters. The recognition is performed using a variant of Viterbi algorithm called “Token Passing Model” [YKO$^+$06] to perform best path search in combinations of different character models. The choice of lexicon and language model is optional and their usage generally results in a lower error rate. In present system a character lexicon is used instead of word lexicon, so that the system will be able to recognize any combination of characters and will not be limited to specific words present in the lexicon. A bi-gram character language model is implicitly learned from the training corpus and is used in the recognition. Character HMMs are combined to provide complex models for the complete text lines using an ergodic HMMs topology. In ergodic topology any model can be reached from any other model in finite number of transitions. The ergodic topology as shown in Figure 2.6 enables to provide lexicon free recognition.

2.5 Experimental Results and Evaluations

The proposed HMMs based OCR system is evaluated on low resolution isolated characters, low resolution screen rendered text lines and degraded machine printed text lines. Evaluation is performed measuring percentage of character recognition accuracy (CRA%) using edit distance [Lev66] as computed in following equation.

$$CRA\% = \frac{N - ED}{N} \times 100 \tag{2.10}$$

where, N = Total number of characters and
ED = Edit distance = deletions + insertions + substitutions (with equal cost)

In addition to the evaluation of the proposed method, current state-of-the-art public and commercial OCR system are also evaluated on the same datasets.

2.5.1 Low resolution characters OCR

Low resolution character recognition experiments are performed using publicly available low resolution character dataset [WKJ07]. This dataset is used to benchmark performance of proposed method against existing low resolution character recognition approaches [WKJ06, WKJ07]. This database contains 28,080 upper and lower case Roman characters in different font styles and sizes. Wachenfeld et al. [WKJ07] reported character recognition accuracy on a subset of 15,808 non-italic and non-bold characters

Table 2.1: Character recognition accuracies for low resolution characters.

Algorithms	**Character Recognition Accuracy (CRA)**
HMMs - Pixel Features	98.35%
Wachenfeld	98.91%
Tesseract	53.78%
ABBYY	52.63%

during evaluation of their method for low resolution text recognition. In this work, the same subset of 15,808 non-italic and non-bold characters is used for training and recognition of proposed HMMs based OCR method. Only pixel based feature are used in this evaluation due to very small size characters. The width of some characters is only two pixels wide like in case of “i” and application of gradient based features (with window size greater than one pixel) is not possible in this case. Table 2.1 presents experimental results of the proposed method on low resolution character images. The table also shows recognition performance of other OCR systems on the same dataset. It can be observed from the Table 2.1 that proposed HMMs based system achieve nearly equal performance to the sophisticated low resolution character recognition approach [WKJ06]. However, public and commercial OCR systems are unable to achieve good recognition accuracies on low resolution characters dataset.

2.5.2 Low resolution screen rendered text lines OCR

Although the proposed HMMs based OCR system provides good recognition accuracy on low resolution characters, it is more appropriate to evaluate the proposed method on complete text lines to benefit from the properties of HMMs. The full text lines can be treated without relying on prior character segmentation. The choice of text lines is also feasible to avoid complexities in layout analysis, as focus of this work is OCR. Due to lack of suitable public dataset for low resolution text lines, an in-house built dataset is used for evaluation of the method. The dataset is built using mixture of public serif and sans-serif fonts. Text lines are rendered at very low resolution (72 ppi) using different font sizes. Sentences for rendering the text lines are taken from a project Gutenberg [Har71] e-book. The dataset comprises of 2,873 text lines having upper and lower case Roman alphabets, numerals, punctuation marks and brackets. Table 2.2 shows evaluation results

Table 2.2: Recognition accuracies for screen rendered text lines.

Algorithms	**Character Recognition Accuracy (CRA)**
HMMs - Pixel Features	98.54%
HMMs - Gradient Features	98.30%
Tesseract	75.32%
ABBYY	99.73%

of proposed HMMs based OCR system using two different features. OCR performances of other OCR systems are also presented. HMM-based OCR system reaches above 98% character recognition accuracy that is much better than existing public OCR systems.

2.5.3 Machine printed text lines OCR

In the previous sections, HMMs based OCR system performance is presented for a special low resolution screen rendered text images. In contrast to the low resolution screen rendered text, OCR of machine printed text is considered relatively an easy task. But this is true only if the underlying document images are very clean and have been scanned at 300 dpi. But, machine printed OCR becomes more challenging in the presence of noise and usage of different fonts, font sizes and font styles. The proposed method is also evaluated on recognition of degraded machine printed document images. The system is trained on 57,600 text lines from UNLV-ISRI document images [TNBC99]. Figure 2.7 shows some example text line images from UNLV-ISRI document database. The machine printed

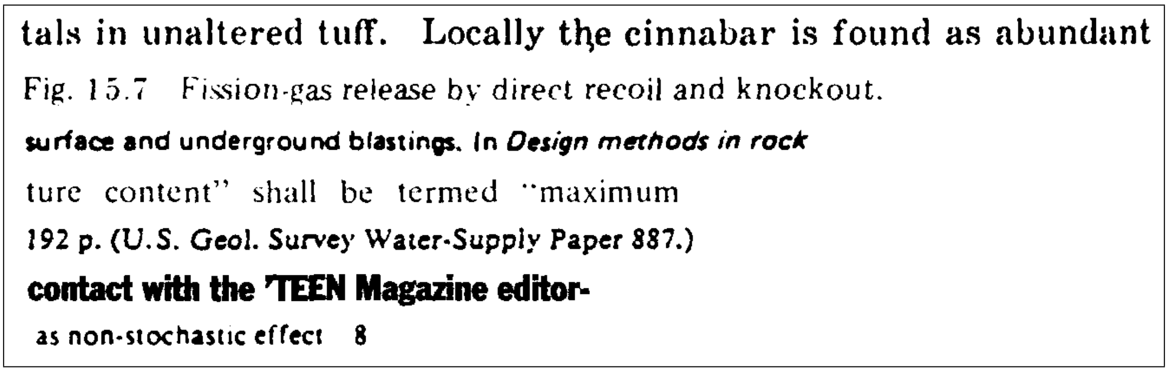

tals in unaltered tuff. Locally the cinnabar is found as abundant

Fig. 15.7 Fission-gas release by direct recoil and knockout.

surface and underground blastings. In *Design methods in rock*

ture content" shall be termed "maximum

192 p. (U.S. Geol. Survey Water-Supply Paper 887.)

contact with the 'TEEN Magazine editor-

as non-stochastic effect 8

Figure 2.7: Sample text lines from UNLV-ISRI document images. The text lines show various degradations like noise, broken or merged characters, variable fonts and font styles etc.

Table 2.3: Character recognition accuracies for UNLV-ISRI text lines.

Algorithms	**Character Recognition Accuracy (CRA)**
HMMs - Pixel Features	95.91%
HMMs - Gradient Features	93.29%
Tesseract	97.66%
ABBYY	99.30%

character set, used in this work, contains 95 character classes, including upper and lower case alphabets, numerals, punctuations and special characters. Recognition is performed on 1060 additional text lines that are not part of the training set. The evaluation of the proposed method and other OCR systems on UNLV-ISRI text lines are presented in Table 2.3. Experimental results show that the proposed HMM based method unable to deliver comparable performance on degraded text line images. The results reported in Table 2.3 are the best results obtained after evaluating various HMMs parameters and topologies on this dataset. It is observed that HMMs are good in learning time varying patterns with uniform variations as reflected by their performance in recognition of screen rendered synthetic text images. But HMMs are unable to capture more dynamic and contextual variations occurred in degraded text line images.

2.6 Discussion

This chapter presented HMM based segmentation free open vocabulary OCR approach for the recognition of low resolution text. The approach achieved character recognition accuracies that are significantly better than the performance of existing OCR systems and reached near to the performance of specialized low resolution character recognition method [WKJ06]. The approach yields above 98% recognition accuracy on screen rendered text lines using both types of features. This is 23% more in comparison with the performance of Tesseract OCR engine [Smi13]. Due to very low resolution and small text size, OCR of screen rendered text usually requires specialized approaches. Applications of prior systems to low resolution text OCR are limited because their high error rates would make the system too unreliable. However, with the use of HMM based techniques a reliable text recognition can be achieved by avoiding the segmentation problem.

The proposed HMM based approach uses the basic methodology of BBN OCR system [SLM$^+$96]. Like BBN OCR, the presented approach modeled each character with multi-state, left to right HMM. However, instead of using 14 states, the proposed system uses four states to model each character in the database because of very small size of the characters. Similarly, the training of the proposed system is done using Baum-Welch algorithm as used by BBN OCR system. The proposed system also uses similar training parameters –means and variance of the component Gaussians, the weight of the mixture component and the state transition probabilities– that have been used in BBN OCR. However, this is the general architecture of more or less every HMM-based recognition system and recognition performance of the systems may vary depending on choice of features, HMM topology, modeling components and usage of different parameters. The proposed approach evaluated on two different features namely gray scale pixel features and gradient based intensity features. The gradient based intensity features are almost similar to the features that have been used in BBN OCR system.

Despite a lot publications describe the usage of BBN OCR system on different scripts and datasets, details of all the parameters and configurations necessary to build a recognition system from scratch are not given. However, in comparison with the reported Latin script BBN OCR system (98.6% character recognition accuracy), the proposed system reached to the similar performance (98.54% character recognition accuracy) using pixel based features. It must be noted that the proposed system is evaluated on low resolution text and did not use any lexicon. However, the performance of BBN OCR system is reported to drop (96.7% character recognition accuracy) while using character bi-grams instead of word lexicon during evaluation on normal printed text. ABBYY provides almost perfect (99.73%) recognition accuracy in the case of low resolution text recognition. The outstanding performance of ABBYY may be attributed to its sophisticated preprocessing and language modeling techniques.

The proposed method is also evaluated on degraded printed document images. Printed text documents usually do not contain the problems that are associated with low resolution text. However, recognition of machine printed text becomes even more challenging when documents have been affected by lots of degradations like noise, smearing, broken or merged characters and have lots of variations in fonts, font styles and sizes. The performance of HMMs based approach drops to around 95% in application to degraded text lines from UNLV-ISRI documents. The reason behind this poor performance may be due to inability of HMMs to learn contextual information and lack of discriminative learning.

Chapter 3

Discriminative Learning for Document Image Analysis using ANNs

This chapter introduces a discriminative learning approach for document image analysis using artificial neural networks (ANNs). The approach works at connected component level in which discriminative shape based variations have been learned with the help of convolutional neural networks (CNNs). The CNNs are design to deal with variability in 2D shapes and they combine the feature extraction and classification process in one step. This eliminates the need to manually design discriminative features for a particular classification problem. The proposed approach is applied to two different document image preprocessing tasks; script recognition and orientation detection. The approach yields above 95% script recognition accuracy on various multi-script document images at connected component level and provides 100% page orientation detection accuracy for Urdu document images. The higher recognition accuracies demonstrate that ANNs have good discriminative learning capabilities and the proposed framework can be used in different document image analysis applications.

3.1 Introduction

Artificial neural networks (ANNs) were originally developed as a mathematical tool to model information processing capabilities of human brain [Bis95]. Although, the complex structure and working mechanism of human brain is not understood completely, it is possible to model the higher level working mechanism of biological neurons in the form of artificial neurons. The basic ANN structure consists of small processing units or nodes, which are connected to each other by weighted connections. In biological terms, the nodes represent neurons and the connection weights represent the strength of synapses between the neurons. An artificial neural network can be activated by providing an input to the input nodes of the network. The input nodes are usually non processing elements and are only responsible to get the input from outside the network. The input spreads throughout the network through the weighted connections. The weights of the network have been learned during a supervised training process and the weighted connections basically represent overall knowledge of the recognition system. The activations of the network resulted in a series of "spikes" at the output layer. The spikes show activations of the specific output neurons that are being activated based on certain input patterns.

Over the past years, many variations to the basic ANN structure are proposed [Hay07]. One of the widely used ANNs structure is feed forward in which units are arranged in layers with connections feeding forward from one layer to next. Well know examples of feed forward neural networks includes perceptrons [Ros58], multilayer perceptrons [DS14], radial basis function networks [KBK$^+$13], Kohnen self organizing maps [Koh01] and convolutional neural networks [LBBH98] etc. Convolutional neural networks (CNNs) are the variants of multilayer perceptrons (MLPs) which are inspired by the early work of Hubel and Wiesel on cat's visual cortex [HW62]. They discovered a complex arrangement of locally sensitive cells in the cat's visual system. The cells are sensitive to small subregions of the input space, called receptive field, and are tilted to the entire visual field. Images have a strong 2 dimensional local structure and the pixels that are spatially near to each other are highly correlated. The local receptive fields are best suited to exploit the local correlations present in the images. The advantages of spatial correlation is to extract and combine local features in terms of edges, points or corners etc. The first implementation of CNNs, called Neocognitron, was proposed by Fukushima to recognize handwritten digits [Fuk88]. The Neocognitron uses the idea of local receptive fields (i.e. each neuron is connected only to a subregion corresponding to some neighboring neurons in the preceding layer) for the extraction of local features from the input patterns. Le-

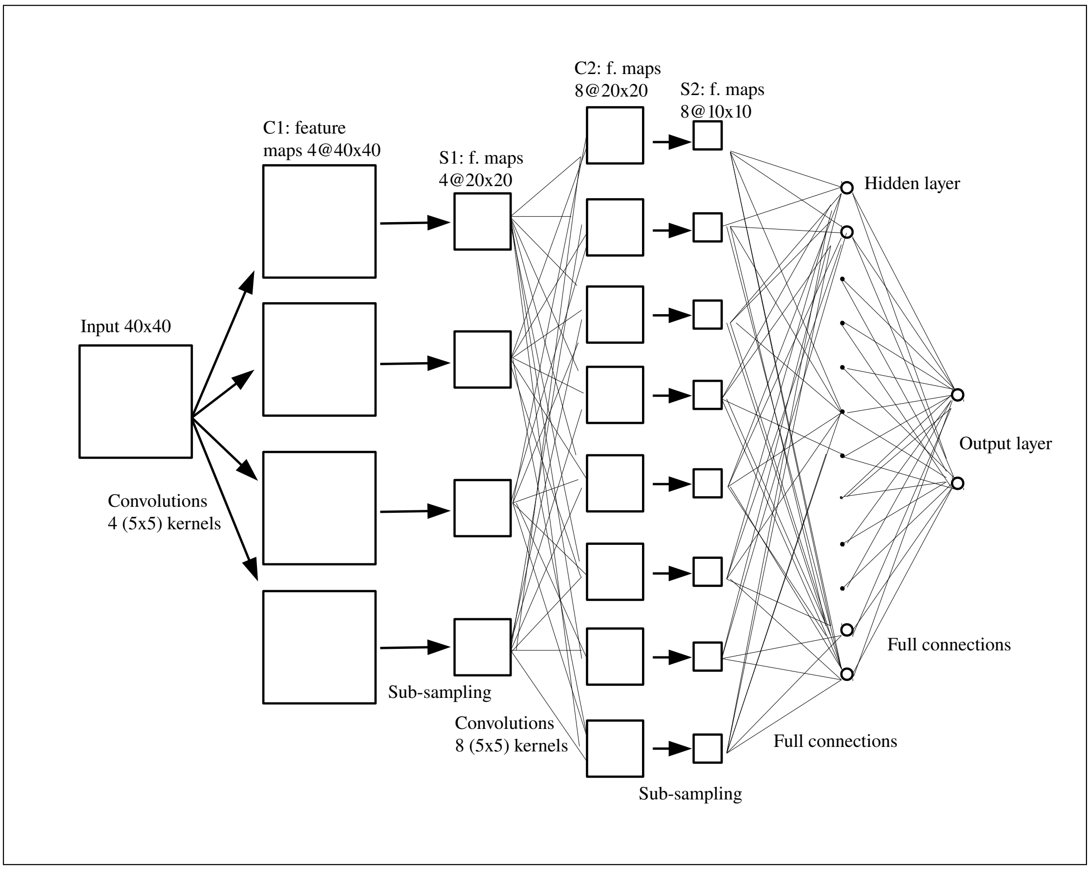


Figure 3.1: The architecture of CNN used for script recognition. C1 and C2 are convolutional layers with 4 and 8 feature maps. Each convolutional layers is followed by sub-sampling layers S1 and S2, containing 4 and 8 feature maps respectively. The output of S2 is feed forwarded to fully connected hidden layer. The final output is obtained at 2 units corresponding to 2 script classes.

cun et al. proposed another CNN architecture, LeNet-5, in which network weights are learned using gradient based backpropagation learning algorithm [LBBH98]. The CNN architecture combine three architectural ideas –local receptive fields, weight sharing or replication and spatial sub-sampling– to obtain some degree of shift, scale and distortion invariance. A typical convolutional neural network (CNN) consists of convolutional layers, sub-sampling layers, one hidden layer and one output layer. Each unit in a layer receives inputs from a small neighborhood (local receptive fields) in the previous layer. With these local receptive fields, neuron can extract initial features like edges, endpoints and corners. The subsequent layers combine these initial features into more higher level features. Units in a layer are organized in planes within which all of the units share the same set of weights. The set of outputs of the units in such a plane constitutes a feature map and units in a particular feature map are constrained to perform the same operation on different parts of the image. A convolutional layer consists of several feature maps, with different weight matrices, so that multiple features can be extracted at each location. The feature layers are connected to the hidden and output layers of the network. The output layer provides the final classification results in terms of the posterior probabilities for specific target classes. Figure 3.1 represents an example architecture of convolutional neural network used in this work for script recognition.

Due to hierarchical learning capabilities, CNNs have been applied in various image classification problems when sophisticated feature extraction is to be avoided and classification is done based on raw image data [Neb98]. This chapter[1] introduces novel applications of CNNs to script recognition and orientation detection. Automatic script recognition and orientation detection are two important preprocessing steps during OCR. Script recognition determines the written script on the page to use an appropriate character recognition algorithm. It is necessary when the OCR system does not have prior knowledge about the language on the page or the page is written in more than one scripts. Orientation detection detects and corrects the deviation of the document's orientation angle from the horizontal direction. This step is also required because if document images are wrongly oriented, the subsequent processing steps like layout analysis and character recognition will fail to work correctly since usually both assume pages to be in the correct orientation.

In this work a discriminative learning based approach is used in application to script and orientation identification. The proposed method works at connected component level in which discriminative shape based features are learned using CNNs. The CNN architecture employed in this work is adopted from Lecun's Lenet-5 [LBBH98]. However, in this work different network topologies and learning parameters are used for the specific image preprocessing task. The CNNs combine feature extraction and recognition process in single step and discriminative features are extracted from the raw input. This eliminates the need for manually defining discriminative features for a particular task. The performance evaluation of the method resulted in above 95% script recognition accuracy at connected component level on Greek-Latin, Arabic-Latin and Antiqua-Fraktur multi-script documents and 100% (page level) orientation detection accuracy on different Urdu documents.

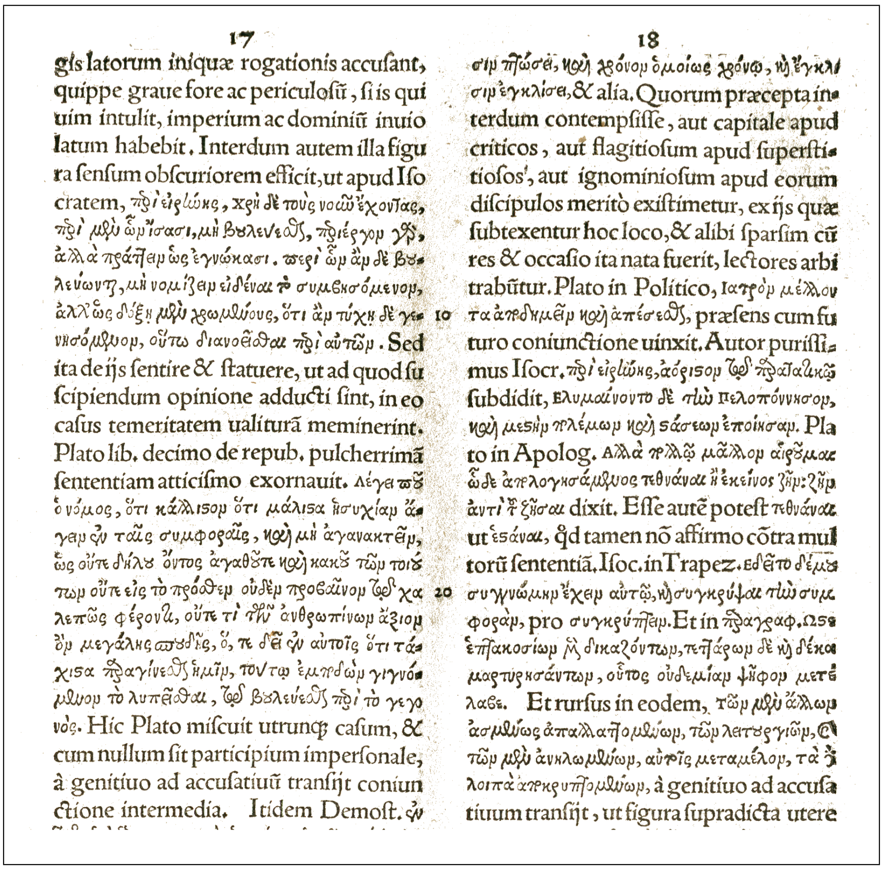
[1] This chapter is based on author's work in [RSB10a, RSB10b] and [RBSB09]

3.2 Script Recognition

Usually OCR systems are developed for a particular script or language, and they can recognize characters that belongs to that particular script or language. A script can be defined as a set of characters used to provide the graphical representation of a certain language or group of languages. Languages in the world are typeset in many different scripts. A script may be used by only one language or it can be shared by more than one languages [GDS10]. A typical example is Arabic script that is used for writing several languages of Asia and Africa, such as Arabic, Persian, Pashto and Urdu [Mir10]. Usually, in multi-script or multilingual environment, OCR systems require to recognize characters irrespectively of their script class. However, building an OCR system that could read characters from all scripts is very difficult. A brute-force solution would be to train an OCR classifier on more than one scripts by adding individual characters from all the scripts in the training process. However, this would lead to more classification errors due to increase in the number of character classes. In addition, the features required for character recognition usually depend on structural properties of the writing which generally differs from one script to other. Another solution is to combine character or word level classifiers for different languages or scripts and recognition of a particular character or word is done by its respective classifier. But, this requires a prior knowledge of the script for application of an appropriate classifiers. Automatic script recognition methods give the prior knowledge about the script or language of a document for selecting a suitable character classifier.

Script recognition is also useful in OCR of multi-script documents in which different paragraphs, text lines or words in a page are written in different scripts. Some examples include ancient multi-script documents, multilingual dictionaries, books with line by line or column wise translation in different languages. Figure 3.2 shows an example document written in two different scripts. In digital libraries, script recognition also helps in indexing, retrieval, and sorting of documents when dealing with multi-script environment.

This thesis presents a novel application of convolutional neural network (CNN) for automatic script recognition in multi-script document images. The method works at connected component level which are considered as characters of a particular script in a document image. The CNN acts as a discriminative learning model, where suitable features for script recognition are automatically extracted and learned. The method is evaluated



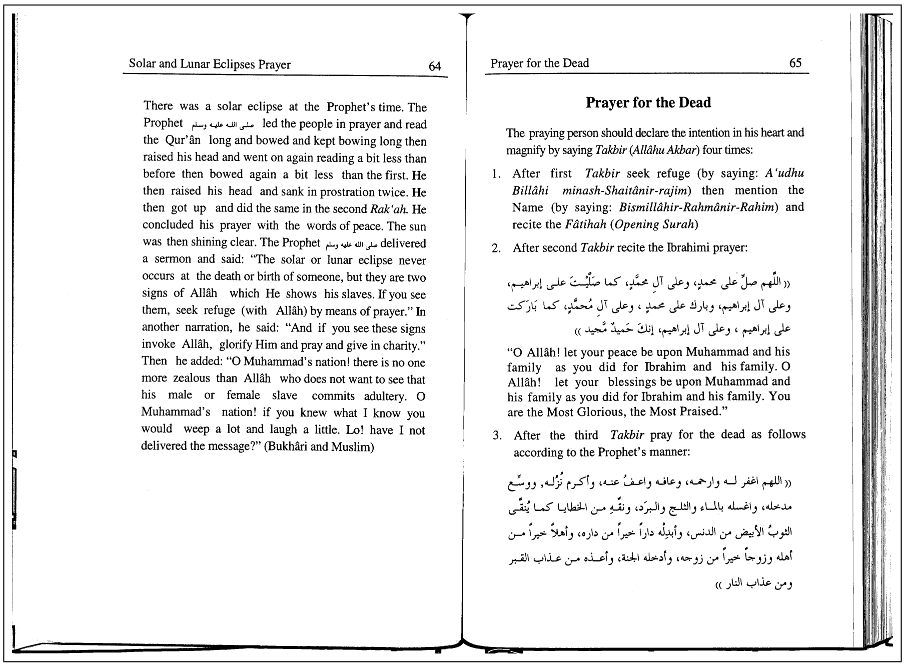
17

gis latorum iniquæ rogationis accuſant,
quippe graue fore ac periculoſũ, ſi is qui
uim intulit, imperium ac dominiũ inuio
latum habebit. Interdum autem illa figu
ra ſenſum obſcuriorem efficit, ut apud Iſo
cratem, περὶ εἰρήνης, χρὴ δὲ τοὺς νοῦν ἔχοντας,
περὶ μὲν ὧν ἴσασι, μὴ βουλεύεσθαι, περιέργον γάρ,
ἀλλὰ πράττειν ὡς ἐγνώκασι. περὶ ὧν ἂν δὲ βου
λεύωνται, μὴ νομίζειν εἰδέναι τὸ συμβησόμενον,
ἀλλ' ὡς δόξῃ μὲν χρωμένους, ὅτι ἂν τύχῃ δὲ γε
νησόμενον, οὕτω διανοεῖσθαι περὶ αὐτῶν. Sed
ita de ijs ſentire & ſtatuere, ut ad quod ſu
ſcipiendum opinione adducti ſint, in eo
caſus temeritatem ualiturã meminerint.
Plato lib. decimo de repub. pulcherrimã
ſententiam atticiſmo exornauit. λέγει που
ὁ νόμος, ὅτι κάλλιστον ὅτι μάλιστα ἡσυχίαν ἄ
γειν ἐν ταῖς συμφοραῖς, καὶ μὴ ἀγανακτεῖν,
ὡς οὔτε δήλου ὄντος ἀγαθοῦ τε καὶ κακοῦ τῶν τοιού
των οὔτε εἰς τὸ πρόσθεν οὐδὲν προβαῖνον τῷ χα
λεπῶς φέροντι, οὔτε τι τῶν ἀνθρωπίνων ἄξιον
ὂν μεγάλης σπουδῆς, ὅ, τε δεῖ ἐν αὐτοῖς ὅτι τά
χιστα παραγίνεσθαι ἡμῖν, τούτῳ ἐμποδὼν γιγνό
μενον τὸ λυπεῖσθαι, τῷ βουλεύεσθαι περὶ τὸ γεγο
νός. Hic Plato miſcuit utrunq; caſum, &
cum nullum ſit participium imperſonale,
à genitiuo ad accuſatiuũ tranſijt coniun
ctione intermedia. Itidem Demoſt. ἐν

18

σιν πτώσει, καὶ χρόνον ὁμοίως χρόνῳ, καὶ ἔγκλι
σιν ἐγκλίσει, & alia. Quorum præcepta in
terdum contempſiſſe, aut capitale apud
criticos, aut flagitioſum apud ſuperſti
tioſos, aut ignominioſum apud eorum
diſcipulos meritò exiſtimetur, ex ijs quæ
ſubtexentur hoc loco, & alibi ſparſim cũ
res & occaſio ita nata fuerit, lectores arbi
trabũtur. Plato in Politico, ἰατρὸν μέλλον
τα ἀποδημεῖν καὶ ἀπέσεσθαι, præſens cum fu
turo coniunctione uinxit. Autor puriſſi
mus Iſocr. περὶ εἰρήνης, ἀόριστον τῷ παρατατικῷ
ſubdidit, ἐλυμαίνοντο δὲ τὴν Πελοπόννησον,
καὶ μεστὴν πολέμων καὶ στάσεων ἐποίησαν. Pla
to in Apolog. ἀλλὰ πολὺ μᾶλλον αἱροῦμαι
ὧδε ἀπολογησάμενος τεθνάναι ἢ ἐκείνως ζῆν: ζῆν
ἀντὶ τοῦ ζῆσαι dixit. Eſſe autẽ poteſt τεθνάναι
ut ἑστάναι, q̃d tamen nõ affirmo cõtra mul
torũ ſententiã. Iſoc. in Trapez. ἐδεῖτο δέ μου
συγγνώμην ἔχειν αὐτῷ, καὶ συγκρύψαι τὴν συμ
φοράν, pro συγκρύπτειν. Et in παραγραφ. ὥστε
ἑπτακοσίων μὲν δικαζόντων, τεττάρων δὲ καὶ δέκα
μαρτυρησάντων, οὗτος οὐδεμίαν ψῆφον μετέ
λαβε. Et rurſus in eodem, τῶν μὲν ἄλλων
ἀσμένως ἀπαλλαττομένων, τῶν λειτουργιῶν, ἐκ
τῶν μὲν ἀνηλωμένων, αὐτοῖς μεταμέλον, τὰ δὲ
λοιπὰ ἀποκρυπτομένων, à genitiuo ad accuſa
tiuum tranſijt, ut figura ſupradicta utere

(a) Example ancient document image in Greek and Latin



Solar and Lunar Eclipses Prayer 64

There was a solar eclipse at the Prophet's time. The Prophet صلى الله عليه وسلم led the people in prayer and read the Qur'ân long and bowed and kept bowing long then raised his head and went on again reading a bit less than before then bowed again a bit less than the first. He then raised his head and sank in prostration twice. He then got up and did the same in the second *Rak'ah.* He concluded his prayer with the words of peace. The sun was then shining clear. The Prophet صلى الله عليه وسلم delivered a sermon and said: "The solar or lunar eclipse never occurs at the death or birth of someone, but they are two signs of Allâh which He shows his slaves. If you see them, seek refuge (with Allâh) by means of prayer." In another narration, he said: "And if you see these signs invoke Allâh, glorify Him and pray and give in charity." Then he added: "O Muhammad's nation! there is no one more zealous than Allâh who does not want to see that his male or female slave commits adultery. O Muhammad's nation! if you knew what I know you would weep a lot and laugh a little. Lo! have I not delivered the message?" (Bukhâri and Muslim)

Prayer for the Dead 65

Prayer for the Dead

The praying person should declare the intention in his heart and magnify by saying *Takbir* (*Allâhu Akbar*) four times:

1. After first *Takbir* seek refuge (by saying: *A'udhu Billâhi minash-Shaitânir-rajim*) then mention the Name (by saying: *Bismillâhir-Rahmânir-Rahim*) and recite the *Fâtihah* (*Opening Surah*)
2. After second *Takbir* recite the Ibrahimi prayer:

« اللّهم صلِّ على محمدٍ، وعلى آلِ محمَّدٍ، كما صَلَّيْتَ على إبراهيم،
وعلى آل إبراهيم، وبارك على محمدٍ ، وعلى آلِ مُحمَّدٍ، كما بَارَكت
على إبراهيم ، وعلى آل إبراهيم، إنكَ حَميدٌ مَّجيد »

"O Allâh! let your peace be upon Muhammad and his family as you did for Ibrahim and his family. O Allâh! let your blessings be upon Muhammad and his family as you did for Ibrahim and his family. You are the Most Glorious, the Most Praised."

3. After the third *Takbir* pray for the dead as follows according to the Prophet's manner:

« اللهم اغفر لـه وارحمـه، وعافه واعـفُ عنـه، وأكـرم نُزُلـه, ووسِّع
مدخله، واغسله بالمـاء والثلـج والبرَد، ونقِّـهِ مـن الخطايـا كمـا يُنقَّـى
الثوبُ الأبيض من الدنس، وأبدِله داراً خيراً من داره، وأهلاً خيراً مـن
أهله وزوجاً خيراً من زوجه، وأدخله الجنة، وأعـذه مـن عـذاب القبر
ومن عذاب النار »

(b) Example document image in Arabic and Latin

Figure 3.2: Example multi-script document images. Documents contain multiple scripts in different text blocks and text lines.

on Greek-Latin, Arabic-Latin multi-script document images and Fraktur, Antiqua single script document images.

3.2.1 Review of related work

Existing methods for script recognition can be broadly grouped into global and local approaches. This categorization is based on feature extraction process employed at the global level (documents or text blocks) or at the local level (characters, words or single text lines) for each individual script. The survey paper of Abirami and Manjula [AM09] presented a precise overview of some of these methods. Another categorization of script recognition methods is made on the basis of visual appearance and structure of the script [GDS10]. Mostly, the script recognition methods work at the global level and assume that different scripts may only exist in certain sections (paragraphs or columns) of the document. Only a few of them consider script recognition at the word or at the text line level.

Hochberg et al. [HKTK97] used cluster based templates for script identification at the document level. Their method developed a set of representative symbols (templates) for each script by clustering textual symbols from training documents and represented each cluster by its centroid. The textual symbols included the characters from Cyrillic script and words or word segments from Arabic script. The script of a new document was identified by matching a subset of symbols from a new image to the templates. A script was selected whose templates provide the best match (based on Hamming distance). The system was trained on thirteen scripts and is reported to correctly identify all the test documents except those printed in fonts that differ from fonts in the training set.

Spitz [Spi97] presented language identification by classifying scripts into two classes, Han-based and Latin-based. The classification was based on the spatial relationship of features related to the upward concavities in character structures. Language identification within the Han script class (Chinese, Japanese, Korean) was performed by the analysis of optical density distribution in text images. Language identification in the Latin script was based on the most frequently occurring word shapes characteristic of the languages. The method built a list of connected components in the image, and the extracted connected components were processed for script and language identification. The system was reported to provide no classification error on Han and Latin based script classification on 240 test samples, containing a minimum of two text lines. In Han-based script, the system was able to classify document language with perfect accuracy on text samples with minimum six lines of text. In Latin-based script, a statistical model of the language categorizations was built using linear discriminant analysis and was tested by cross validation. More than 90% overall language recognition accuracy was reported for 23 languages.

Pal and Chaudhuri [PC01, PC99] separated text lines of different scripts using projection profiles, water reservoir and existence of head-line (a feature specific to Bangla and Devanagari scripts). The method used the head-line information to separate the Devanagari and Bangla script lines in one group, and English, Chinese, and Arabic lines in another group. In the first group, Bangla text lines were separated from Devanagari using zone-wise shape features [PC97]. In the second group, Chinese were separated from English and Arabic using vertical black run information. They computed character-wise maximum vertical black run in English, Arabic and Chinese text lines collected from different books, journals etc. and observed that in Chinese about 57% characters have four or more black runs and the percentage of the characters with one and two vertical black runs is very low (about 14%). They used this as the criterion for separation of Chinese text from English and Arabic. If 57% of the characters in a text line had four or more vertical black run the text line was classified as Chinese. English text from the Arabic was separated using distribution of lowermost points of the components in the text line and water overflow from a reservoir concept. The method was reported to achieve 97.32%, 98.65%, 97.53%, 96.02% and 97.12% identification rates for English, Chinese, Arabic, Devnagari and Bangla script respectively on 700 document images.

Busch et al. [BBS05] described the use of texture as a tool for determining script of a document image. They evaluated a number of commonly used texture features for the script identification task. The method was tested on a database of eight different scripts (Latin, Chinese, Japanese, Greek, Cyrillic, Hebrew, Sanskrit, and Farsi). Classification of input samples was performed by using a Gaussian mixture model (GMM) classifier, which attempts to model each class as a combination of Gaussian distributions in the feature space. In their evaluations, the wavelet log co-occurrence texture feature outperformed the other texture features for script classification, with an overall 1% error.

Joshi et al. [JGS06] used multi-channel log-Gabor filter bank at multiple orientations for identification of ten Indic scripts. They proposed a hierarchical script classifier using globally extracted features. In the first stage, their method grouped scripts into five major classes using global features. At the next stage, a sub-classification was performed based on script-specific features. The system was reported to achieve an overall 97.11% classification accuracy on the test data.

Ramakrishnan and Pati [PR08] reported word level multi-script identification by separately using Gabor and discrete cosine transform (DCT) features. They tested their approach on bi-script, tri-script and eleven-script scenarios. The features had been used to evaluate three different classifiers: nearest neighbor, linear discriminant and support vector machine (SVM). They reported to achieve 98.4% script recognition accuracy for eleven Indic scripts by using a combination of SVM with the Gabor features.

3.2.2 Proposed method

This section describes the whole method in a pipeline from document image preprocessing to script recognition. The output of the system from each processing step is shown in Figure 3.3 and described in the following subsections.

Preprocessing

The preprocessing mainly consists of binarization and noise removal.

Binarization. The complete experimental setup used in this method is based on binarized images. Different state-of-the-art binarization methods can be classified into two groups: (i) global binarization (like Otsu [Ots79]) and (ii) local binarization (like Sauvola [SP00]). Global binarization estimates a single threshold for the complete image, whereas local binarization calculates a threshold for each individual pixel based on the neighborhood information. In general, local binarization works better than global binarization under different types of document image degradations like non-uniform shading or blurring etc. However, local binarization methods are slower than global binarization methods. Shafait's local binarization method [SKB08] overcomes this problem by using integral images [VJ04] for computation of local threshold. This work uses the Shafait's [SKB08] local adaptive thresholding technique for binarization.

Noise Removal. Binarized images may contain small noise (salt and pepper) or big noise (merging components or borders) [FWL02]. The noise is removed by using heuristic

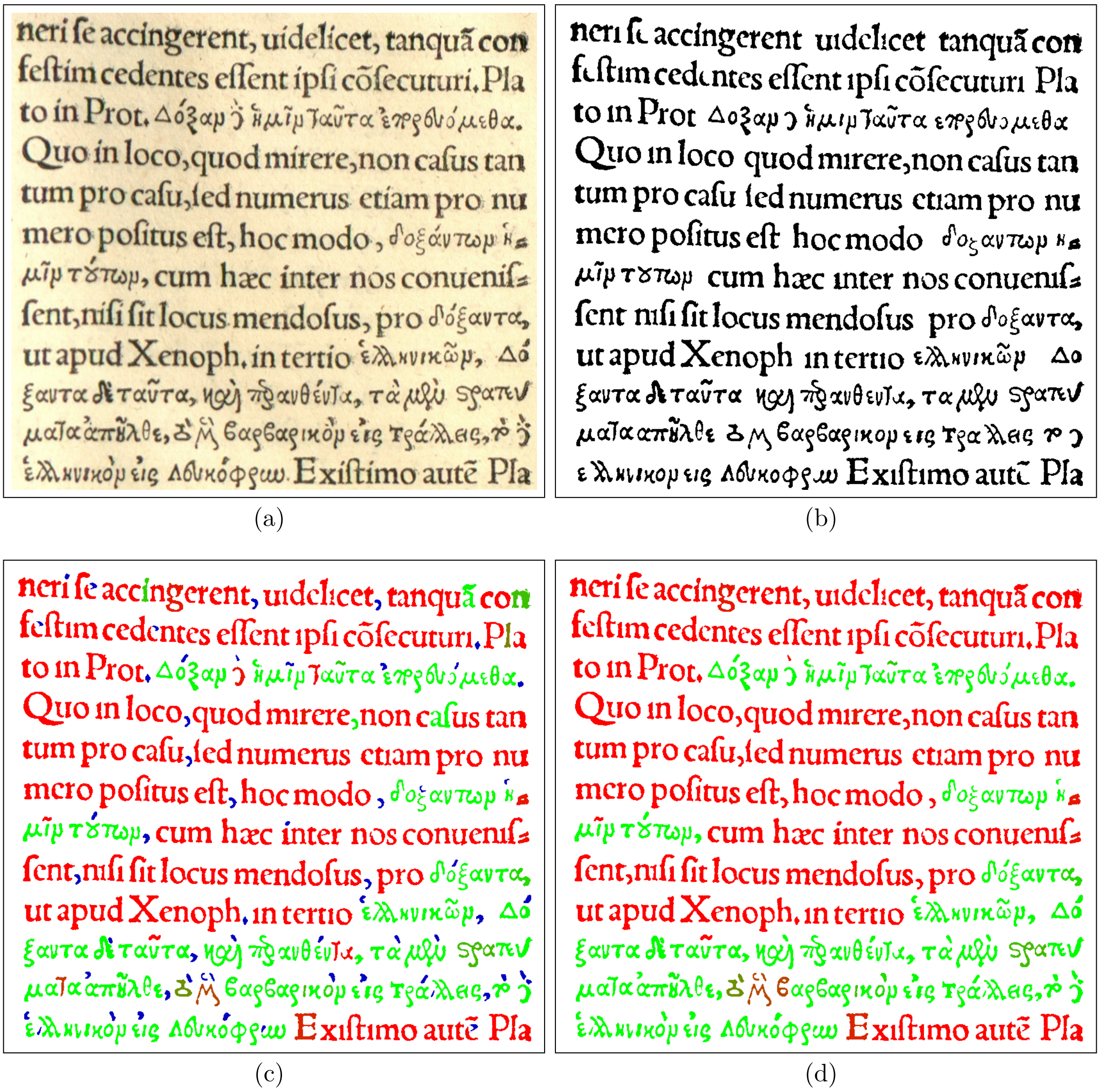


Figure 3.3: Pipeline for multi-script identification system; (a) input image (b) preprocessed image (c) script recognition output (d) post-processing to assign diacritics into their respective script classes. The output is presented in color coded format in which the pixels intensities reflects the probabilistic nature of the output. Darker color shows the higher recognition probabilities for a particular script. Green color represents Greek, red color represents Latin, and blue color represents small components (diacritics or noise).

rules based on size of the connected components. A connected component is considered as a noise if its height and width is less than or equal to 0.3 times the median height and median width, or its height and width is greater than 5 times of the median height and median width of all the extracted components. The noise removal threshold values may vary from one dataset to another. Diacritics and punctuations are small components that occur in both scripts. The small components have been removed from document images using similar heuristic rules i.e. a component is considered as a diacritic or a punctuation if its width and height is equal to or less than 0.7 times the median height and median width of all the components. Figures 3.3(a) and 3.3(b) show the parts of an original image and the preprocessed image taken from an example ancient document image.

Feature vector generation

Instead of extracting complex geometrical, morphological or statistical features, a convolutional neural network (CNN) is used to extract discriminative features from the raw input data. Connected components are extracted from the document image. Each connected component is rescaled to 40 x 40 dimensions while keeping the aspect ratio intact. This is important because change in aspect ratio changes the shape of the connected component. The shape changes may effect the CNN classification accuracy. In the rescaling process, the connected components are downscaled or upscaled depending on their initial width or height. A feature vector is built by normalizing the pixels intensities of the connected components between -1 and 1. The rescaled connected components describe the raw input to the CNN for training and evaluation.

CNN architecture and training

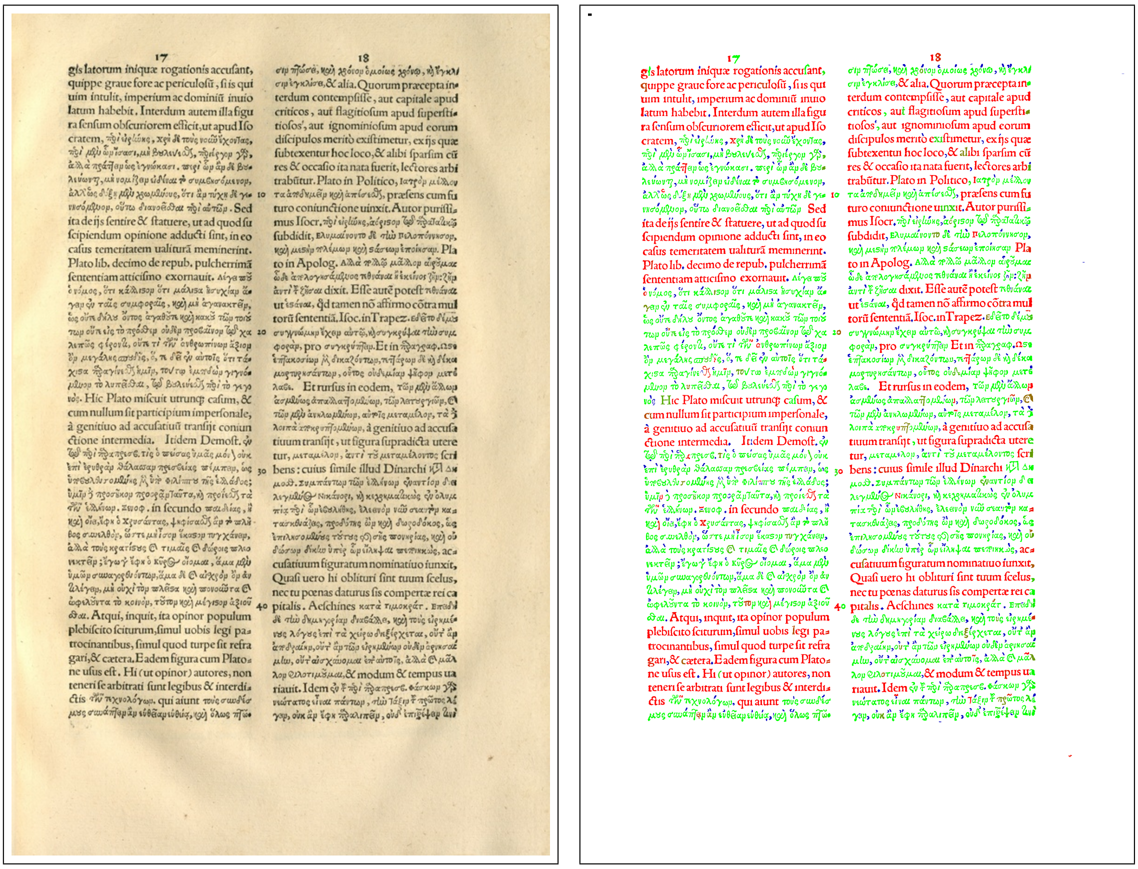
The CNN used in this work consists of two convolutional layers with four and eight feature maps followed by two sub-sampling layers. The input layer of CNN receives the feature vector extracted from each connected component of the document. The values of subsequent feature maps are obtained by convolving the input map with the respective kernels and applying an activation function to the result. The first convolutional layer has 4 (5 x 5) convolutional kernels corresponding to 4 high resolution features. The second convolutional layers has 8 (5 x 5) convolutional kernels corresponding to 8 complex features. Each convolutional layer is followed by sub-sampling layers with a sub-sampling factor of two. The output of the last subsampling layer is forwarded to the fully connected hidden layer with 100 hidden units. The output layer consists of 2 units corresponding to 2 script classes. The first four layers (two convolutional and two subsampling) of this neural network can be viewed as trainable feature extraction layers connected to a trainable classifier in the form of two fully connected layers. The number of hidden units controls capacity and generalization of the overall classifier. The CNN architecture used for script recognition is already shown in Figure 3.1.

The proposed CNN architecture is intended to recognize two scripts present at a time in a single document, but the same network can be used for script identification in single script document images. Different CNN are trained for script recognition of different bi-script document images. The CNN used for Greek-Latin script recognition is trained on $19,600$ training samples and $2,000$ validation samples of Greek and Latin connected components taken from Greek-Latin bi-script documents. The training procedure is run for 200 epochs with 0.1 learning rate. Similar settings are used for training the CNN on other scripts. The details of the number of samples used for training and validation of each script are presented in Tables 3.1, 3.2, and 3.3 along with script recognition results. An on-line error backpropagation algorithm [DHS00] is used for training the CNN in a supervised learning mode.

Script recognition

To determine the script of an input document, the document is first preprocessed and feature vectors are extracted as described in above sections. The extracted features are given to the trained CNN classifier for script identification of each connected component. The output layer of the CNN gives two values corresponding to two script classes. A connected component is classified into a particular script by using the the highest output value from the output layer.

The CNN classification output is also represented by a color code to represent classifier confidence in recognition of a particular script. For example in Greek-Latin script recognition, Greek script is represented in green and Latin script is represented in red colors. The small connected components i.e. punctuations and diacritics were filtered out during the preprocessing step and these are not classified by the CNN. These punctuations and diacritics marks are represented by a default color code (represented as blue color) for all the punctuation marks. The pixel color codes are probabilistic in nature and colors with high intensity values represent more recognition confidence for a particular script class.



(a) Input image

(b) Script recognition output

Figure 3.4: Greek-Latin script recognition results in color coding format. Green color represents Greek, red color represents Latin, and blue color represents small components (diacritics or noise).

Figure 3.4 shows an original document image and the color coded CNN output from the Greek-Latin test dataset.

Post-processing

A post-processing is performed on the CNN output for association of the small components to their respective script class and to improve the script recognition results. In the post-processing stage, small connected components are assigned to the closest neighboring connected component's script. Script recognition results were further improved by extending the bounding box of each connected component to its left and right by a factor of its height or width (whichever is greater) and using the class majority within the neighboring area to relabel the script of that component. Final output after applying the post-processing is shown in Figure 3.3(d).

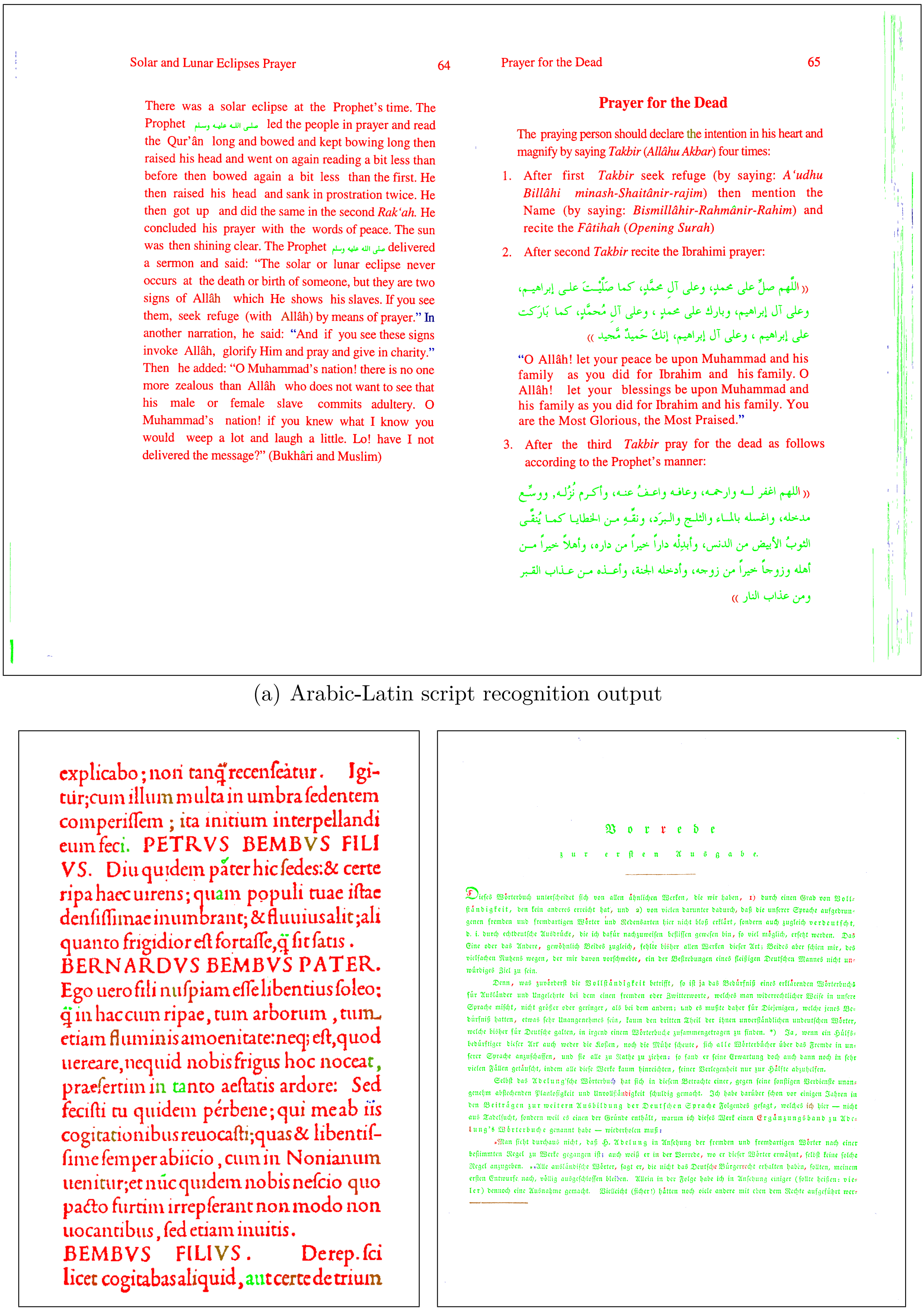
3.2.3 Experimental Results

The presented multi-script recognition method is evaluated on Greek-Latin, Arabic-Latin and Fraktur-Antiqua document datasets. For Greek-Latin script recognition, 19 ancient documents are used for training and testing. The documents have Greek and Latin scripts mixed within sentences and paragraphs. The dataset is manually processed to generate ground truths for the training and testing purposes. The evaluation results for Greek-Latin script recognition are presented in Table 3.1.

In case of Arabic-Latin, scanned pages of an Arabic book and pages from UW-III [GHHP97] dataset are used for training the CNN. The testing is performed on scanned pages from a different book that contains both Arabic and Latin scripts mixed within certain sentences. The evaluation results are presented in Table 3.2. A slightly less recognition accuracy is obtained on Arabic-Latin test dataset because the test dataset is entirely different from the training dataset. However, the method has generalization capabilities to perform well on the unseen data.

For Antiqua-Fraktur recognition, the CNN is trained on three ancient Fraktur document images and training samples for Antiqua are taken from a subset of Greek-Latin document images in Antiqua typeface. The testing is performed on four Fraktur and Antiqua document images. Table 3.3 shows the evaluation results on Antiqua and Farktur recognition task.

The script recognition accuracy is further improved by incorporating a class majority count in the left-right neighboring area of every connected component. As mentioned before, the script recognition output is represented as a color coded image. The color intensities reflect the classification confidence of each script present in the document. This color coded output can be further analyzed by some statistical techniques to improve the script recognition accuracy. Figure 3.5 shows the output of the recognition method on different test documents of different scripts.



Solar and Lunar Eclipses Prayer 64

There was a solar eclipse at the Prophet's time. The Prophet صلى الله عليه وسلم led the people in prayer and read the Qur'ân long and bowed and kept bowing long then raised his head and went on again reading a bit less than before then bowed again a bit less than the first. He then raised his head and sank in prostration twice. He then got up and did the same in the second *Rak'ah*. He concluded his prayer with the words of peace. The sun was then shining clear. The Prophet صلى الله عليه وسلم delivered a sermon and said: "The solar or lunar eclipse never occurs at the death or birth of someone, but they are two signs of Allâh which He shows his slaves. If you see them, seek refuge (with Allâh) by means of prayer." In another narration, he said: "And if you see these signs invoke Allâh, glorify Him and pray and give in charity." Then he added: "O Muhammad's nation! there is no one more zealous than Allâh who does not want to see that his male or female slave commits adultery. O Muhammad's nation! if you knew what I know you would weep a lot and laugh a little. Lo! have I not delivered the message?" (Bukhâri and Muslim)

Prayer for the Dead 65

Prayer for the Dead

The praying person should declare the intention in his heart and magnify by saying *Takbir* (*Allâhu Akbar*) four times:

1. After first *Takbir* seek refuge (by saying: *A'udhu Billâhi minash-Shaitânir-rajim*) then mention the Name (by saying: *Bismillâhir-Rahmânir-Rahim*) and recite the *Fâtihah* (*Opening Surah*)
2. After second *Takbir* recite the Ibrahimi prayer:

«اللّٰهم صلِّ على محمدٍ، وعلى آلِ محمَّدٍ، كما صَلَّيْـتَ على إبراهيـم،
وعلى آل إبراهيم، وبارك على محمدٍ ، وعلى آلِ مُحمَّدٍ، كما بَارَكت
على إبراهيم ، وعلى آل إبراهيم، إنكَ حَميدٌ مَّجيد »

"O Allâh! let your peace be upon Muhammad and his family as you did for Ibrahim and his family. O Allâh! let your blessings be upon Muhammad and his family as you did for Ibrahim and his family. You are the Most Glorious, the Most Praised."

3. After the third *Takbir* pray for the dead as follows according to the Prophet's manner:

«اللهم اغفر لـه وارحمـه، وعافه واعـفُ عنـه، وأكـرم نُزُلـه, ووسِّع
مدخله، واغسله بالمـاء والثلـج والـبرَد، ونقِّـهِ مـن الخطايا كمـا يُنقَّى
الثوبُ الأبيض من الدنس، وأبدِلْه داراً خيراً من داره، وأهلاً خيراً مـن
أهله وزوجاً خيراً من زوجه، وأدخله الجنة، وأعـذه مـن عـذاب القـبر
ومن عذاب النار »

(a) Arabic-Latin script recognition output

explicabo; non tanq̃ recenſeatur. Igitur; cum illum multa in umbra ſedentem comperiſſem; ita initium interpellandi eum feci. PETRVS BEMBVS FILIVS. Diu quidem pater hic ſedes: & certe ripa haec uirens; quam populi tuae iſtae denſiſſimae inumbrant; & fluuius alit; aliquanto frigidior eſt fortaſſe, q̃ ſit ſatis. BERNARDVS BEMBVS PATER. Ego uero fili nuſpiam eſſe libentius ſoleo: q̃ in hac cum ripae, tum arborum, tum etiam fluminis amoenitate: neq; eſt, quod uereare, nequid nobis frigus hoc noceat, praeſertim in tanto aeſtatis ardore: Sed feciſti tu quidem pérbene; qui me ab iis cogitationibus reuocaſti; quas & libentiſſime ſemper abiicio, cum in Nonianum uenitur; et nũc quidem nobis neſcio quo pacto furtim irrepſerant non modo non uocantibus, ſed etiam inuitis. BEMBVS FILIVS. De rep. ſcilicet cogitabas aliquid, aut certe de trium

(b) Antiqua script recognition output

Vorrede

zur erſten Ausgabe.

Dieſes Wörterbuch unterſcheidet ſich von allen ähnlichen Werken, die wir haben, 1) durch einen Grad von Vollſtändigkeit, den kein anderes erreicht hat, und 2) von vielen darunter dadurch, daß die unſerer Sprache aufgedrungenen fremden und fremdartigen Wörter und Redensarten hier nicht bloß erklärt, ſondern auch zugleich verdeutſcht, d. i. durch echtdeutſche Ausdrücke, die ich dafür nachzuweiſen befliſſen geweſen bin, ſo viel möglich, erſetzt werden. Das Eine oder das Andere, gewöhnlich Beides zugleich, fehlte bisher allen Werken dieſer Art; Beides aber ſchien mir, des vielfachen Nutzens wegen, der mir davon vorſchwebte, ein der Beſtrebungen eines fleißigen Deutſchen Mannes nicht unwürdiges Ziel zu ſein.

Denn, was zuvörderſt die Vollſtändigkeit betrifft, ſo iſt ja das Bedürfniß eines erklärenden Wörterbuchs für Ausländer und Ungelehrte bei dem einen fremden oder Zwitterworte, welches man widerrechtlicher Weiſe in unſere Sprache miſcht, nicht größer oder geringer, als bei dem andern; und es mußte daher für Diejenigen, welche jenes Bedürfniß hatten, etwas ſehr Unangenehmes ſein, kaum den dritten Theil der ihnen unverſtändlichen undeutſchen Wörter, welche bisher für Deutſche galten, in irgend einem Wörterbuche zuſammengetragen zu finden. *) Ja, wenn ein Hülfsbedürftiger dieſer Art auch weder die Koſten, noch die Mühe ſcheute, ſich alle Wörterbücher über das Fremde in unſerer Sprache anzuſchaffen, und ſie alle zu Rathe zu ziehen: ſo fand er ſeine Erwartung doch auch dann noch in ſehr vielen Fällen getäuſcht, indem alle dieſe Werke kaum hinreichten, ſeiner Verlegenheit nur zur Hälfte abzuhelfen.

Selbſt das Adelung'ſche Wörterbuch hat ſich in dieſem Betrachte einer, gegen ſeine ſonſtigen Verdienſte unangenehm abſtechenden Planloſigkeit und Unvollſtändigkeit ſchuldig gemacht. Ich habe darüber ſchon vor einigen Jahren in den Beiträgen zur weitern Ausbildung der Deutſchen Sprache Folgendes geſagt, welches ich hier — nicht aus Tadelſucht, ſondern weil es einen der Gründe enthält, warum ich dieſes Werk einen Ergänzungsband zu Adelung's Wörterbuche genannt habe — wiederholen muß:

»Man ſieht durchaus nicht, daß H. Adelung in Anſehung der fremden und fremdartigen Wörter nach einer beſtimmten Regel zu Werke gegangen iſt; auch weiß er in der Vorrede, wo er dieſer Wörter erwähnt, ſelbſt keine ſolche Regel anzugeben. »»Alle ausländiſche Wörter, ſagt er, die nicht das Deutſche Bürgerrecht erhalten haben, ſollten, meinem erſten Entwurfe nach, völlig ausgeſchloſſen bleiben. Allein in der Folge habe ich in Anſehung einiger (ſollte heißen: vieler) dennoch eine Ausnahme gemacht. Vielleicht (ſicher!) hätten noch viele andere mit eben dem Rechte aufgeführt wer-

(c) Fraktur script recognition output

Figure 3.5: Script recognition results for Latin-Arabic Antiqua and Fraktur documents in color coding format. Red color represents Latin/Antiqua, green color represents Arabic/Fraktur, and blue color represents small components (diacritics or noise).

Table 3.1: Script recognition accuracies for Greek and Latin scripts.

	Training set		**Validation set**		**Test set**		
	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Accuracy after left-right neighbor rule (%)
Greek	9,800	99.41	1,000	96.40	11,302	95.16	97.65
Latin	9,800	98.92	1,000	97.80	10,828	97.58	99.15
Average	19,600	99.16	2,000	97.10	22,130	96.37	**98.40**

Table 3.2: Script recognition accuracy for Arabic and Latin scripts.

	Training set		**Validation set**		**Test set**		
	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Accuracy after left-right neighbor rule (%)
Arabic	24,000	97.03	2,000	98.95	6,037	97.80	99.30
Latin	24,000	99.31	2,000	97.70	1,221	90.60	91.92
Average	48,000	98.17	4,000	98.33	7,258	94.20	**95.61**

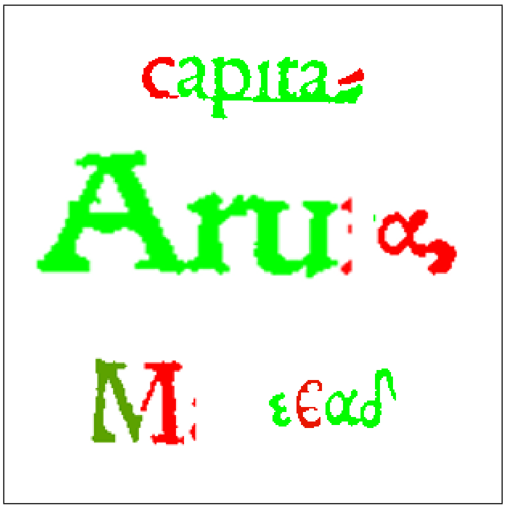


Figure 3.6: Errors due to noise, merged and broken characters.

Table 3.3: Script recognition accuracy for Antiqua and Fraktur scripts.

	Training set		**Validation set**		**Test set**		
	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Nos. of samples	CNN accuracy (%)	Accuracy after left-right neighbor rule (%)
Antiqua	10,600	98.23	2,000	97.90	1,194	87.0	97.40
Fraktur	10,600	98.07	2,000	98.60	4,130	92.27	95.81
Average	21,200	98.15	4,000	98.25	5,324	89.64	**96.61**

3.2.4 Discussion

This section presented a multi-script recognition approach by using discriminative learning at connected component level. The convolutional neural network is used as a discriminative learning model to extract and learn suitable features for the multi-script recognition task. One observation is that due to the appearance based discriminative properties of different scripts, a script can be recognized based on the shape of its individual characters. The use of CNN at connected component level to learn discriminative shape based features is an efficient approach for multi-script recognition as demonstrated by the evaluation results presented in Tables 3.1, 3.2 and 3.3. The approach has generalization capabilities and it gives good results on different target documents that are not part of the training process. The datasets used in these experiments have noise in terms of touching or broken characters and smudge (e.g. ink spots or spread) as shown in Figure 3.6. The noisy components are removed before training the CNN, therefore CNN does not provide recognition accuracy for these components. It is observed that most of the recognition errors are due to the noise and better results may be obtained on clean datasets. Another observation is that CNN is sensitive to character shapes in terms of slight variations e.g, thick or thin writing strokes, and this problem can be overcome by adding more training samples that contain more of these variations.

3.3 Orientation Detection

Orientation detection is an important preprocessing step in large scale digitization projects. The normal flatbed scanners require manual placement of the page on the glass window and page orientation is corrected manually. However, in large scale digitization process, orientation has to be detect automatically, as the use of automatic document feeding scanners often results in wrongly orientated scanned documents. Orientation correction is necessary as many subsequent processing steps (layout analysis, text recognition etc.)

assume the upright position of the document image. The orientation detection process determines the actual orientation of a document image which can be used to transform the document image into right orientation. Orientation detection methods mainly focus on four target orientations 0°, 90°, 180° and 270°. Usually the document scanning process results in these four orientations. The slight variation besides these four orientation can be handled by skew correction techniques [DMMH06].

Most of the existing orientation detection methods are based on computing the ascender to descender ratio. Unfortunately, this cannot be used for the scripts like Arabic in which no consistent ascenders and descenders are present. This chapter describes the application of discriminative learning approach for orientation detection of Urdu document images. The proposed method works at connected component level and a convoultional neural network is used to identify the orientation of components among four major orientations; 0°, 90°, 180° and 270°. The shape of most of the connected components is highly dependent on their orientation and it changes with the change in orientation. Learning the shape of an individual connected component in all four orientations helps in recognition of the correct orientation for a specific connected component. The orientation of the entire document is determined by majority count of connected components in a particular orientation. The system is evaluated on varying layout of Urdu document images and resulted in 100% page orientation detection accuracy.

3.3.1 Related Work

Most of the existing document image orientation detection work can be categorized into two main categories: 1) landscape and portrait detection and 2) up-down orientation detection. Landscape and portrait can be detected using global [AH90] and local [LTW94] projection profiles whereas most of up-down detection techniques are based on the fact that the number of character ascenders are more likely as compared to number of character descenders in Roman script text.

H.B. Aradhye [Ara05] proposed a method for determining the up/down orientation of text in a scanned document of unknown orientation. The method analyzed the “open” portions of text blobs to determine the direction in which open portions face. The method identified direction of the text as a whole by determining the respective densities of blobs opening in a pair of opposite directions right or left. The method was applied on Roman and non-roman (Pahto and Hebrew) text.

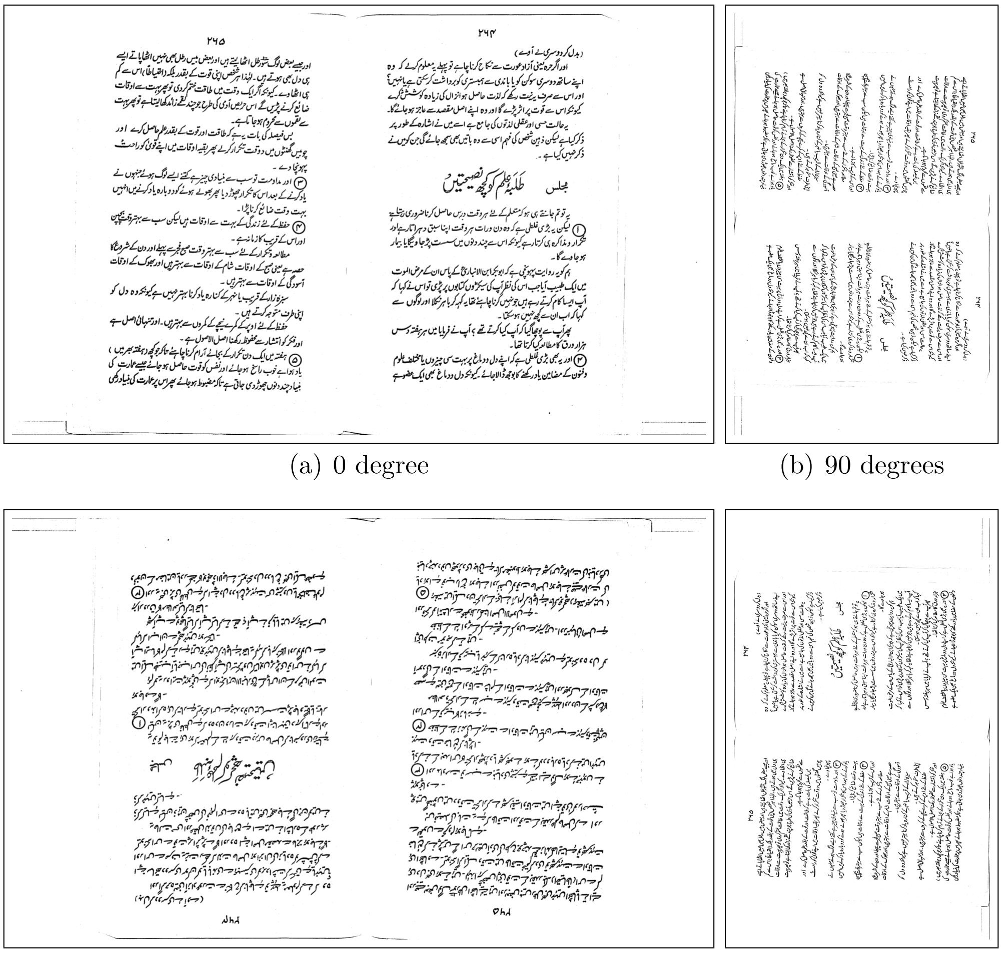
Lu and Tan [LT06] introduced a combined method for script and orientation detection through document vectorization. The method encoded the document orientation and document language information by converting each document image into a document vector through exploitation of the density and distribution of the vertical component runs. Their method was tested on a dataset of 492 document images having text lines ranging from 1 to 12. The method achieved an accuracy of 98.18% for documents with at least 12 text lines.

Beusekom et al. [BSB09, BSB10] proposed a method for combined skew and orientation detection using geometric modeling of Roman script text lines. The method searches for a text line candidates within a skew range of four orientations top-up, top-down, top-left and top-right. The best fit of the model gives the estimate for orientation and skew.

In another work, Beusekom et al. [VBRB10] reported a trainable orientation detection method by using character similarity to compute the correct orientation. They computed a connected component based distance measure to compare the characters of the document image to characters with a known orientation. The orientation of input document was detected with the lowest distance correct orientation characters. Evaluation of their method showed an accuracy of above 99% for Latin and Japanese scripts and an accuracy of 98.9% for Fraktur documents.

3.3.2 Datasets

In this work a dataset of scanned Urdu document images from different Urdu publishing sources is used. Urdu belongs to Arabic script which is very different from Roman. Urdu is written from right to left and characters have connections between each other to form a word. Usually shape of individual characters varies depending on their position in an Urdu word [DH10]. In addition, some characters have special symbols (diacritics) above or below the character. The complete Urdu dataset is categorized into five subcategories: *book*, *novel*, *poetry*, *magazines* and *newspapers* (based on its publishing source), out of which *book*, *novel* and *poetry* has been used in this work. The dataset with these three categories consists of 59 scanned images and each scanned image contains 2 document pages. This dataset is available in 0° orientation, referred as correct orientation, and the examples for other orientations are generated by rotating the original images to other orientations like 90°, 180° and 270°. After these rotations, 236 images having 472 document pages are obtained. Only 128 document pages from *book* dataset are used



(a) 0 degree

(b) 90 degrees

(c) 180 degrees

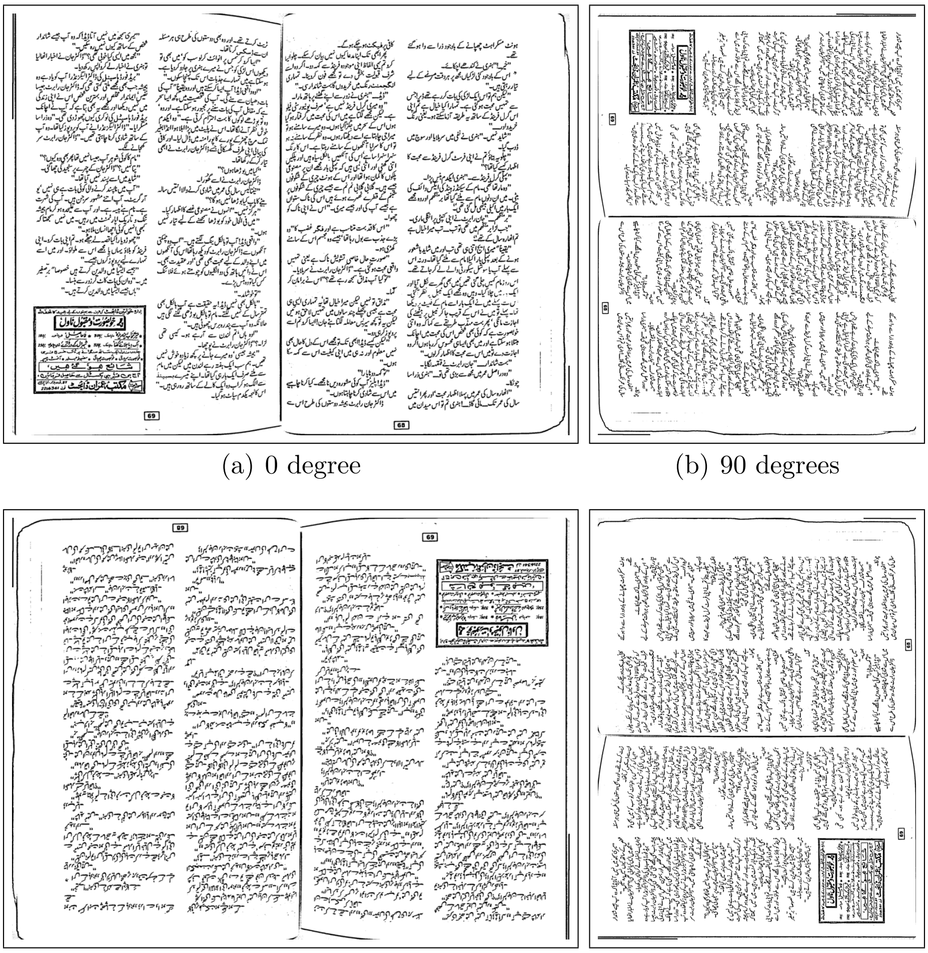
(d) 270 degrees

Figure 3.7: Sample documents from *book* dataset in different orientations.

for training (112 pages) and validation (16 pages) of CNN. The document from other two categories are only used to evaluate the system under slight shape variations of the components. Figures 3.3.2, 3.3.2 and 3.3.2 present some example images of *book*, *novel* and *poetry* documents in four different orientations.

3.3.3 Method description

The method is composed of preprocessing, feature vector generation, CNN training and evaluation steps. Following subsections provide a brief description of each step.



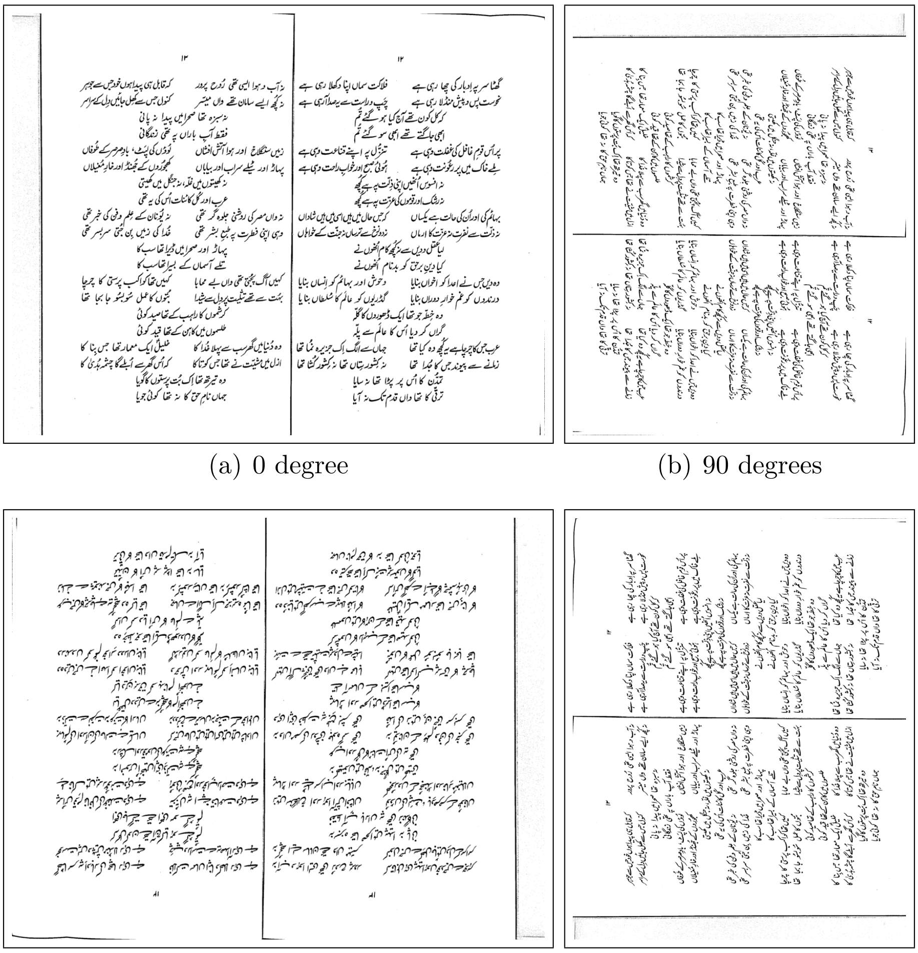
(a) 0 degree (b) 90 degrees

(c) 180 degrees (d) 270 degrees

Figure 3.8: Sample documents from *novel* dataset in different orientations.

Preprocessing

Scanned document images are preprocessed before extracting the feature vectors. Similar preprocessing (binarization and noise removal) is applied that is used in multi-script recognition. Document images are binarized by using a local adaptive thresholding method [SKB08]. Binarized images are further processed for removal of noise. The noise is removed by using heuristic rules, for example, a connected component is considered as marginal noise if its height is greater than 5 times of median height or width is greater than 5 times of median width. Apart from this, most of the Urdu characters consist of small connected components like dots and diacritics. Most of these dots and diacritics have similar shapes in all possible orientations and therefore these are considered as noise and are removed during preprocessing. The diacritics are removed by heuristic rules e.g.



(a) 0 degree

(b) 90 degrees

(c) 180 degrees

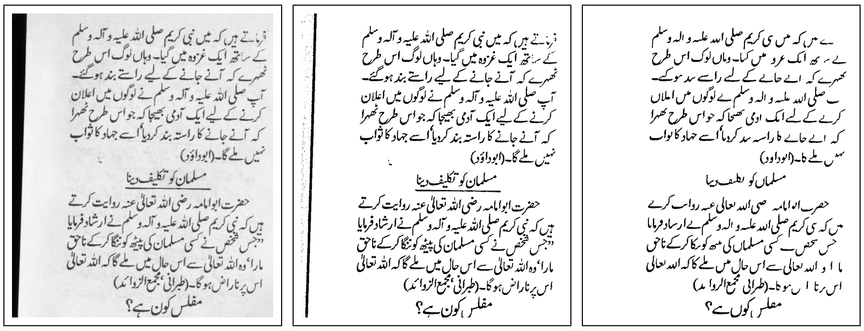
(d) 270 degrees

Figure 3.9: Sample images from *poetry* dataset in different orientations

if the height of a component is smaller than 85% of median height and width is smaller than 85% of median width, the component is considered as noise or diacritic. Figure 3.10 shows an input gray scale image and the images after binarization and noise removal process.

Feature vectors generation

After the application of preprocessing the extracted components are used to build feature vectors for training the CNN. The components are normalized to 40 x 40 dimensions by upscaling or downscaling the width or height of the components. During rescaling, the aspect ratio of each component is maintained so that the shape of the component will not be destroyed. The pixel values of rescaled components are normalized between the



(a) Input gray scale image.

(b) Binarization output.

(c) Cleaned image after noise and diacritics removal

Figure 3.10: Document image preprocessing for orientation detection of Urdu documents. Preprocessing mainly includes binarization and removal of noise and diacritics.

range of -1 and 1. Feature vectors are constructed using these rescaled and normalized components in all four target orientations.

CNN architecture and training

A detailed description of CNNs and their working has already been discussed in the above sections. In this work, a slightly different architecture of CNN is proposed for orientation detection. As described before, a CNN is a kind of multilayer neural network with built-in capability of feature extraction from raw input data. The general working of CNN is the extraction of simple features at high resolution and converting them into more complex features at a coarse resolution. The coarser resolution is obtained by using sub-sampling layers. In this work, the first convolutional layer has 10 (5 x 5) convolutional kernels, the second convolutional layers has 20 (5 x 5) convolutional kernels and each convolutional layer is followed by sub-sampling layers with a sub-sampling factor of two. The network has hidden layer with 100 hidden units and the output layer consists of 4 units corresponding to 4 orientation classes.

For training the CNN, features are extracted from 60% of *book* dataset under different orientations. The network is trained for 200 epochs with 0.1 learning rate. An on-line error backpropagation algorithm [LBBH98] is used to train the CNN.

Table 3.4: Orientation detection accuracies for *book* test dataset.

Document orientation	**Connected component level accuracy(%)**	**Overall connected component level accuracy(%)**	**Page level accuracy(%)**
0	80.54	87.96	100
90	85.04		
180	93.8		
270	92.44		

3.3.4 Experimental Results and Evaluation

The dataset used in the evaluation has two distinctive properties, page layout and font/text-printing technology. In Urdu publishing system, usually *book*, *poetry* etc. are written by 'Katibs' (skilled calligraphers who can write in different calligraphy styles and fonts) and these documents have variability in shape of similar ligatures. However, other document categories like *novel*, *magazine* and *newspaper* etc. are printed by using printing stamps and do not have shape variations for similar ligatures. The page layout of each type of document is also different, for example *poetry* has unique layout of writing words over other words etc. In this work, the CNN is trained only on the *book* dataset but evaluation has also been performed on *novel* and *poetry* document images that vary in shapes of similar ligatures and have different page layouts. The method is evaluated in two experiments as given bellow.

Experiment 1

In first experiment, *book* dataset is used for training and evaluation of the method. As explained above, 60% of the *book* dataset is used for training the neural network and remaining 40% of the *book* data is used for validation (20%) and testing (20%) purposes. The method attained an overall 88% recognition accuracy for the training set, 87% recognition accuracy for the validation set and 87.96% recognition accuracy for the test dataset at connected component level. Recognition accuracies for the test set in all orientations are presented in Table 3.4. However, the method obtained 100% page level accuracy on all documents in all four orientations by a majority count.

Table 3.5: Orientation detection accuracies for *novel* and *poetry* datasets.

Dataset	**Document orientation**	**Connected component level accuracy(%)**	**Overall connected component level accuracy(%)**	**Page level accuracy(%)**
Novel	0	63.64	76.29	100
	90	74.49		
	180	81.17		
	270	85.84		
Poetry	0	74.2	84.63	100
	90	83.42		
	180	93.13		
	270	87.76		

Experiment 2

In second experiment, the already trained CNN is evaluated for *novel* and *poetry* datasets for all possible orientations. Experimental results, given in Table 3.5, show the robustness and generalization capabilities of the method on unseen data. The method is trained only on *book* dataset but the method is capable to categorize the *novel* and *poetry* document images into correct orientations. The method gives less recognition accuracy on *novel* dataset in comparison with *book* and *poetry* because *novel* dataset differs more in terms of page layout and printing method. These variations in printing method cause the variability in shapes of similar Urdu ligatures and characters. The neural network is trained only for one type of printed shapes and therefore it gives less accuracy for other type of printing style. A higher accuracy at component level for *novel* or other printed style category can be obtained by providing more training samples containing all variations (printing style and font) to the CNN. However, the method resulted in 100% accuracy at page level for these different categories of Urdu dataset.

3.3.5 Discussion

This section described a new approach for orientation detection of Urdu document images. A convolutional neural network is used as a discriminative learning model to learn the orientation of Urdu documents varying in layout and printing techniques. Connected components are extracted from Urdu *book* dataset, rescaled to 40 x 40 dimension and used as raw features for CNN training. Orientation of the input document is determined by identifying the orientation of each connected component present in the document, and page orientation is determined by majority count of orientations among connected components. The page level orientation detection accuracy is dependent on number of connected components extracted from the page and it may decrease for pages having few connected components. The proposed method has been evaluated on a subset of publicly available Urdu dataset [SuHKB06] for *book*, *novel* and *poetry* categories. The method reached 100% page level and overall 83% connected component level orientation detection accuracy. Due to discriminative learning behavior of the proposed approach, it can be applied to detect the orientation of other scripts.

3.4 General Discussion

This chapter presented a discriminative learning approach for document image analysis using CNNs. The evaluation of the approach on different image processing applications demonstrated the discriminative learning capabilities of ANNs. The approach worked at connected component level in which a CNN is used as a discriminative learning model to learn discriminative shape based variations from raw pixels of the connected component. The CNN exploits 2 dimensional spatial correlation of neighboring pixel in the image by using the concept of receptive fields and learns the local features in terms of edges, points or corners. The proposed framework is used in application to script recognition and orientation detection of various types of document images. The evaluation of the approach on automatic script recognition resulted in above 95% script recognition accuracy at connected component level. For this particular application, a post-processing procedure is adopted to classify the diacritics based on closest neighboring connected component's script. A different CNN architecture is trained and evaluated for page orientation detection and resulted in 100% page orientation detection accuracy for Urdu documents of variable page layouts.

Chapter 4

A Combined ANN/HMM Approach for OCR

Chapter 2 of this thesis presented a successful application of hidden Markov models to provide segmentation free OCR of low resolution text images. However, the method is unable to produce good recognition accuracies for degraded text images. The reason behind this limiting performance may be contributed to presence of many variations because of noise, variable fonts, different font styles and font sizes, and varying document types. HMMs are incapable to capture all these variations owing to independent observation assumption and lack of discriminative recognition. ANNs are proved to be very good in learning discriminative shape based variations as presented in Chapter 3. This chapter presents a novel and simple integration of ANNs with HMMs to provide improved recognition performance by eliminating the problems associated with HMMs. The combined ANNs/HMMs approach enables to incorporate discriminative learning and contextual knowledge at character level into the system. A text line scanning neural network is developed in which character class posterior probabilities are obtained by scanning a analysis window over a text line. The output of the scanning neural network is decoded by character level HMMs to provide segmentation free OCR of a complete text line. In evaluations on a subset of the UNLV-ISRI document collection [TNBC99], the proposed OCR system achieves 98.4% character recognition accuracy, that is statistically significantly better in comparison with character recognition accuracies obtained from state-of-the-art open source OCR systems.

4.1 Introduction

Optical character recognition of printed document images is one of the most addressed areas of pattern recognition research in the last few decades. OCR systems have reported to achieve high recognition rates and it is ubiquitously considered that OCR of machine printed Latin script is a solved problem. However, error free character recognition is still not possible for documents with moderate degradations, variable fonts, noise and broken or touching characters. Moreover, character recognition accuracy further decreases in case of handwritten or cursive script text. Building OCR systems with improved recognition rates is required for many interesting industrial applications. For example, OCR process has been central in building efficient mail sorting machines, providing text to speech services for blinds or visual impaired people, automatic processing of official documents, machine translation, text reading abilities in robots, and converting historic document archives into digital libraries.

In the past, a number of different approaches to OCR have been introduced. A good overview of the historical development of OCR research can be found in [MSY92]. Fujisawa [Fuj08] presented an overview on the last 40-years of technical advances in the field of character and document recognition. In fact, character recognition has remained a universal benchmark for developing and testing pattern recognition algorithms due to its universal nature. It includes essential problems of pattern recognition that are common to most other topics, while having easy to comprehend inputs and outputs. Some of the most widely used pattern recognition methods today were either developed for, or tested on character recognition problems to demonstrate their strengths. Examples of the first category include Convolutional Neural Networks [LBBH98] and Random Forests [AG97, Bre01a], and the latter category includes Support Vector Machines [SSB+97].

Mostly, OCR approaches rely on recognition of pre-segmented characters or character candidates. The output of these approaches is often a recognition lattice that represents segmentation and recognition alternatives [Bre08,Smi07]. However, these approaches are vulnerable to cases where accurate character segmentation is not possible. For example, in case of degraded printed text or handwritten and cursive script, segmentation of text into characters is problematic and performance of the character segmentation significantly affects character recognition accuracies. Hidden Markov models are common and successful in unsegmented speech recognition [GY07, Rab89] and most of the segmentation free approaches in OCR are employed from speech recognition research. These models are extensively applied to recognize unconstrained handwritten text or cursive scripts [JBT96, Kho07, MB01]. However, HMMs have drawbacks like having independent observation assumption and being generative in nature.

Hybrid approaches, based on combinations of various neural networks and HMMs have also been proposed in application to handwriting, cursive script and speech recognition. In most of the hybrid approaches [BKW$^+$99, MAGD01, KKS00, ECBGMZM11] a neural network is used to augment the HMM either as an approximation of the probability density function or as a neural vector quantizer. Other hybrid approaches [SGH95, BLNB95, KA98] use the neural networks as part of feature extraction process or to obtain the observation probabilities for HMMs. These hybrid approaches either require combined NN/HMM training criteria [ECBGMZM11] or they use complex neural network architecture like time delay or space displacement neural networks [BLNB95]. Recurrent neural networks (RNNs) can be considered as an alternative to HMMs but are limited to isolated character recognition due to the segmentation problem [Bou95]. Graves et. al [GLF$^+$08] combined RNNs with connectionist temporal classification (CTC) to provide segmentation free recognition of off-line and on-line handwritten text.

This chapter details[1] a novel segmentation free OCR method that combines two state-of-the-art pattern recognition paradigms –artificial neural networks and hidden Markov models– to provide improved character recognition performance on degraded document images. The ANNs can be used to learn shape based variations from raw input signals and provide excellent discriminative learning capabilities as demonstrated in Chapter 3 of this thesis. HMMs are good in learning time varying properties of the signal and give good recognition performances on unsegmented text lines (Chapter 2). The proposed method employs ANNs in combination with HMMs to capture discriminative and time varying properties of unsegmented text lines. In the proposed combination, ANNs and HMMs are trained independently on common dataset and the trained models are combined in a simple way to provide segmentation free OCR.

Humans can efficiently and accurately recognize text of varying fonts in the presence of noise and clutter. The unmatched human performance in reading text can be attributed to several cognitive processes and reading strategies. Researchers from the field of cognitive psychology and reading have presented several theories to explain the underlying

[1] This chapter is based on author's work in [RSB12] and was nominated for IAPR best student paper award. `http://www.ict.griffith.edu.au/das2012/awards.html`

cognitive processes during reading and text comprehension. The presented method, to some extent, follows some of the cognitive reading based strategies in order to build a robust OCR system. On a dataset of $1,060$ degraded text lines extracted from the widely used UNLV-ISRI benchmark document collection, the presented system achieves a 30% reduction in error rate as compared to Google's Tesseract OCR system and 43% reduction in error as compared to OCRopus OCR system, which are the best open source OCR system available today.

The rest of the chapter is organized as follows. Section 4.2 gives a brief description of exiting ANNs/HMMs hybrid approaches. Section 4.3 details the proposed ANNs/HMMs approach followed by a short overview of the cognitive reading strategies and their application in the proposed method in Section 4.4. Evaluations of the proposed and current state-of-the-art OCR methods on UNLV-ISRI text line images are presented in Section 4.5. The outcomes of the presented methods are discussed in Section 4.6

4.2 Overview of Related Work

ANNs and HMMs are efficient pattern recognition strategies with different advantages and disadvantages and their combination is an attractive choice in application to develop robust pattern recognition applications. The combination of ANNs and HMMs is interesting and leads to variety of different approaches in order to build hybrid systems. Usually development of hybrid approaches concerns to handle issues related to overall system architecture and estimation of joined system parameters, so that both the paradigms can interact in an optimal way and support each other. Rigoll [Rig02] presented several methods for combining HMMs and ANNs to develop hybrid systems. Most of the hybrid systems are applied to unconstrained speech recognition and handwritten word recognition problems [TG03, ECBGMZM11]. An important connection between HMMs and ANNs is the emission probability component of HMM based systems in which emission probability is assigned to each state. The emission component yields the probability of an observed feature vector when HMM is in a specific state using discrete or continuous emission probabilities. The output modeling component of HMMs can be implemented using a neural network architecture in which ANNs are used to estimate these emission probabilities. In the hybrid systems different kinds of ANNs like multilayer perceptrons [DS14], Kohnen self organizing maps [Koh01] or radial basis function networks [KBK^{+}13] can be used to provide emission probabilities of HMMs [Rig02].

Knerr and Augustin [KA98] presented an ANN/HMM hybrid for handwritten word recognition. In this work, the words were represented as left-right sequence of graphemes. The system modeled the words with ergodic HMMs. The HMM observation probabilities were modeled using a neural network to give observation probabilities for all HMMs. The system employed a complex iterative Expectation-Maximization (EM) like training of the hybrid and HMMs provided the targets for the training of the neural network. The system is designed for small word vocabularies and resulted in around 93% word recognition accuracy for the 30 word vocabulary of legal amount bank cheques. Marukatat et al. [MAGD01] described a similar approach for on-line handwriting recognition using HMM and ANN hybrid. In this method the HMM emission probability densities are approximated using mixtures of predictive multilayer perceptrons. The left-to-right letter level HMMs were used to model words or sentences. The word level output was augmented by a dictionary and language modeling. The method was trained on word images and training consists in several iterations over the training set. Initially character boundaries were estimated using default HMM parameters and in second stage, neural network parameters were re-estimated based on word segmentation achieved in initial step. The method used a dictionary based approach for word recognition and method was extended to recognize sentences by integrating a language model. The system was reported to give around 86% to 90% sentence recognition accuracy using three different bi-grams.

Bengio et al. [BLNB95] presented a NN/HMM hybrid for on-line handwriting recognition using convolutional neural networks and hidden Markov models. In this system neural network and HMMs were jointly trained to minimize an error measure defined at the word level. The neural network was used to recognize characters and the HMMs were used to model the word candidates from the neural network output. The system used the fixed size multi layer convolutional neural network (MLCNNs) with multiple input, multiple output structure, and the output layer of the system consists of radial basis function units. These MLCNNs are also called space displacement neural networks (SDNNs). The SDNNs was used to provide scores associated to characters from different segments of the word and a graph of character candidates was built using character level HMMs. The three state character HMMs was used to model the sequence of network output observed for each character. The observation graph of the input word was obtained by connecting these character models.

Kim et al. [KKS00] developed a HMM-MLP hybrid model for cursive handwritten scripts.

In this system they trained HMMs and MLPs using two different set of features. The HMM model was built using implicit segmentation based word level HMMs in which four different kinds of features were used for training the HMMs. The MLPs were trained using mesh and crossing features. The outputs of the trained models were combined using three different combination schemes such as conventional voting, linear confidence accumulation (LCA) and weighted multiplication method. The HMM and MLP hybrid is based on the idea that two different classification paradigms can compliment each other by using different feature sets. The hybrid of HMM and MLP system resulted in 92.2% recognition rate on a subset of CENPARMI database.

Jaeger et al. [JMRW01] also reported development of an hybrid system for on-line handwriting recognition. The system is based on multi-state time delay neural networks (MS-TDNNs). MS-TDNNs is an extension of time delay neural networks (TDNNs) that is combined with a nonlinear time alignment method to find an optimal alignment between strokes and characters in handwritten words. In the proposed MS-TDNNs, words are represented as sequence of characters and each character is modeled with three states that represent first, middle and last part of the character. Hence, the MS-TDNNs structure can be considered as an ANNs/HMMs hybrid. The MS-TDNN is trained in three steps with backpropagation algorithm. Initial training procedures require segmented word data for alignment using Viterbi algorithm. However, in final training step, the Viterbi training is replaced by force alignment procedure and can be performed on unsegmented words. The recognition system is based on a predefined lexicon of words in which a search tree is built for every character representing all words starting with this specific character. The nodes in every tree are HMMs that represent individual characters. For training the overall system, the word images have been gone through a series of sophisticated preprocessing steps, and a number of different features were computed. The system was evaluated on different datasets by using several dictionaries. The system is reported to achieve 96% recognition rate for a 5,000 word dictionary when trained with printed and handwritten data from different databases.

Recently, España-Boquera et al. [ECBGMZM11] proposed HMM/ANN hybrid model for recognition of unconstrained off-line handwritten text. In the proposed system left-to-right HMMs were used to model graphemes. A neural network was employed to estimate the emission probabilities instead of Gaussian mixtures. The estimates of the posterior probabilities of the neural network were divided by prior state probabilities, resulting in scaled likelihoods which were used as emission probabilities in the HMMs. The training of the complete hybrid system was performed by an iterative EM algorithm, where the training cycles of ANN and HMMs were altered. Training of ANNs usually requires labeling of every feature vector, and this could be done either by providing hand crafted label data or by using a previously trained system. The system was reported to improve the error rate by 42% over the base line HMM based system on IAM database.

4.3 Proposed Method

This section explains the building blocks of the proposed OCR methodology. The system consists of five modules:

1. Preprocessing
2. Feature extraction
3. Auto tunable multilayer perceptron (AutoMLP)
4. Text line scanning
5. Hidden Markov models

4.3.1 Preprocessing

Preprocessing is one of the basic steps in almost every pattern recognition system. It is applied to transform the input data to a uniform format that is suitable for the extraction of discriminative features. Preprocessing normally includes noise removal, background extraction, binarization or image enhancement etc. In the present system, preprocessing consists of skew correction and text line normalization. These two steps are important for reliable text recognition. In the following, a brief detail of these two steps is provided.

Skew correction

Printed documents originally have zero skew, but when a page is scanned or photocopied, (nonzero) skew may be introduced. A skew correction algorithm aligns the text line with the x-axis of the image. The operation corrects any rotation that may have occurred during scanning. For skew correction, it is required to determine the skew angle of the text line image. It is assumed that the lower baseline of the text line can be approximated by a straight line as described in [MB01] and the skew angle of the base line can be further used for alignment. The lower baseline is determined by computing the lowest white pixel for each column of the text line. In this system, the writing is represented in the form of white pixels and background in the form of black pixels. The set P of white pixels, obtained by the process, approximate the lower baseline of the given text line. Formally, the set P can be represented as follows

$$P = \{p_i = (x_i, y_i) | \text{lowest white pixel in column } x_i\} \tag{4.1}$$

Linear regression is applied on the set of points P, and the objective is to compute the slope "m" and y intercept "b" of a straight line, as given in the following equation

$$y = mx + b \tag{4.2}$$

Let's suppose that the if set P has n number of values then the mean value of x and y can be computed as:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.3}$$

The line parameters m and b are computed by following two formulas

$$m = \frac{\sum_{i=1}^{n} x_i y_i - n\mu_x\mu_y}{\sum_{i=1}^{n} x_i^2 - \mu_x^2}, \qquad b = \mu_y - m\mu_x \tag{4.4}$$

Linear regression minimizes the error between the give set of points and the line, but character's descenders may influence the regression line. In case of lower baseline estimation, character's descenders can be considers as outliers in the set of points P. To reduce this influence on the regression line, the summed squared error between the line and the set P can be computed as follows

$$e = \sum_{i=1}^{n}(mx_i + b - y_i)^2 \tag{4.5}$$

If the total error e is greater than a predefined threshold t, the point p_i with larger value from the sum is eliminated from set P. The whole process is repeated until the error e is less than threshold t. When the parameters for the text line are estimated, the skew angle is determined as below:

$$\alpha = mtan(m) \tag{4.6}$$

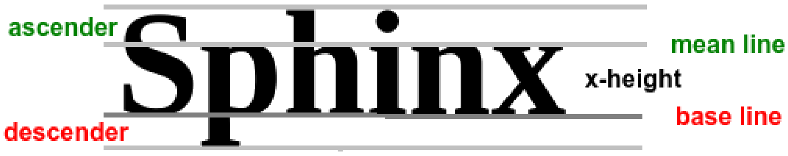


Figure 4.1: Typographical arrangements of letters in an example text line. The figure shows ascender and descender regions of the text line. The x-height is typically refers to height of letter “x” in a particular font.

The skew of the text line is corrected by rotating the text line with $-\alpha$ angle.

Text line normalization

Text line normalization is an essential step for training the neural network classifier. The neural network takes the fixed dimensional input and text lines usually differ significantly with respect to height and width of the characters. In this system, text line normalization is based on vertical and horizontal scaling of the text line. This work introduced a novel text line normalization technique while preserving typeface characteristics of the text line. The main objective is to transforms the printed text into a standard position along the vertical axis of the text line.

In English typography, a baseline is the line upon which most letters “sit” and a mean-line is at half the distance from the baseline to the cap height, where cap height refers to the height of a capital letter in a particular typeface. The portion of a letter that extends below the baseline is called descender, and the portion of a letter that extends above the mean-line is called ascender. Ascender is part of a letter that is taller than the font’s x-height [Typ12]. Typically, x-height is the height of letter x in a particular font [Wik11]. Figure 4.1 illustrates the typographic arrangements of characters in a typical text line. For Latin scripts, the typographic arrangements preserve the major characteristics that define appearance of a typeface.

In the proposed method, instead of normalizing height of the input text line to a certain value, the text line contents are normalized to get a standard typographic arrangement for all the characters in the text line. In this regard, the x-height of the text line is normalized to a specific height and the ascenders and descenders are rescaled accordingly. During the normalization process the input text line is first decomposed into three regions; an ascender region, a descender region and a middle region. These regions are extracted by estimating the mean-line and baseline of the input text line. The baseline is already known during skew correction, and the mean-line is determined by similar linear regression method, that is used to estimate the baseline. After estimating the baseline and mean-line, the average point of the points that estimated the baseline and mean-line is calculated. The region between the top most vertical point and the average mean-line point is considered as an ascender region, the region between the average mean-line point and the average baseline point is considered as a middle region, and the region between the average baseline point and the bottom most vertical point is considered as a descender region. Figures 4.2(a), 4.2(b) and 4.2(c) show an example text line image, its marked baseline, mean-line, top most and bottom most points, and three extracted regions from this text line image.

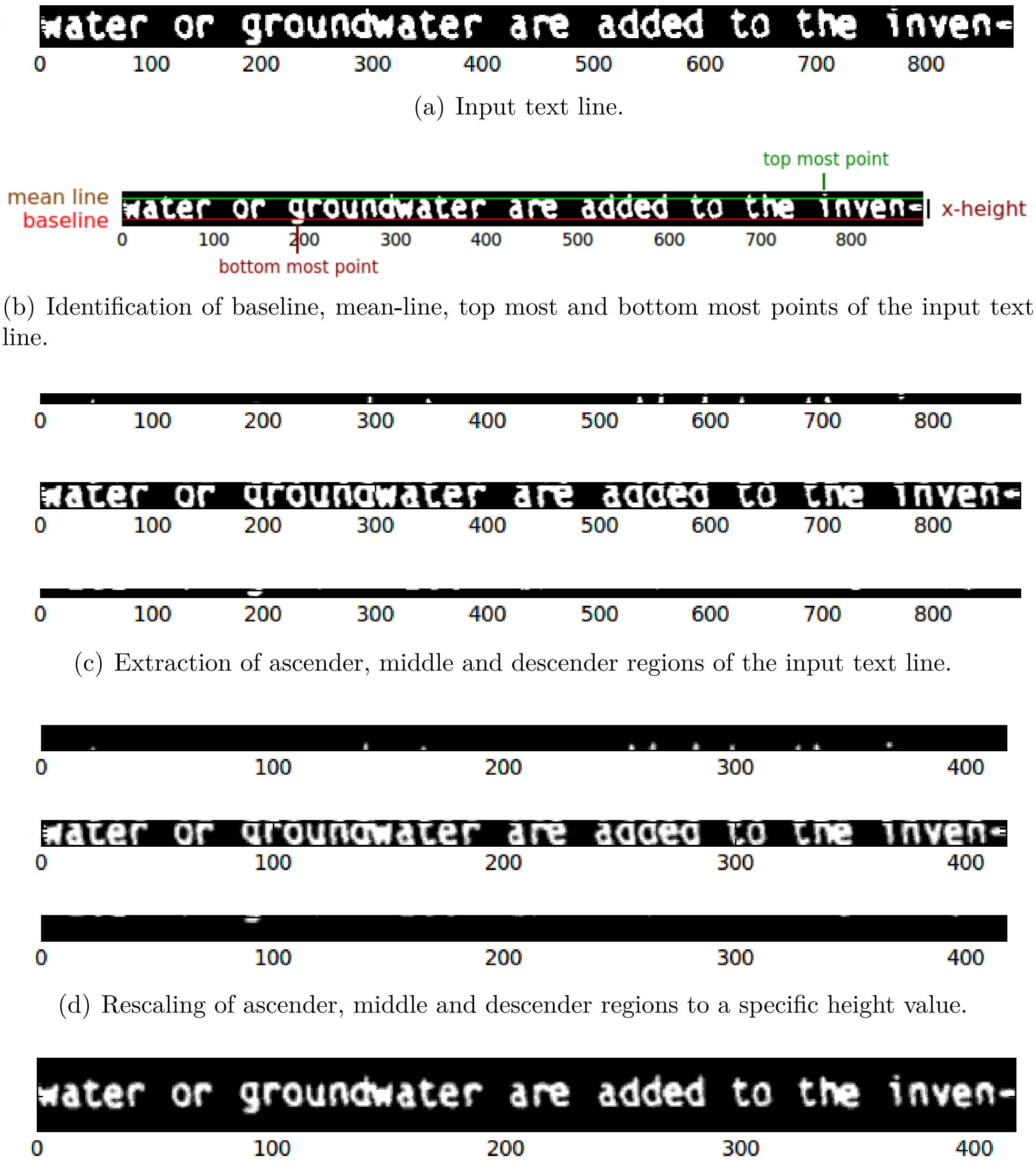
The heights of ascender and descender regions depend on the existence of ascenders or descenders in a particular text line. If the height of ascender and descender regions do not exceed from a particular threshold, then it is assumed that the text line does not have the ascender or descender portions in the text. After extraction of ascender, descender and middle regions, the input text line is normalized by following the below mentioned steps.

1. Heights of the ascender and descender regions are made equal to the height of middle region, by padding with extra pixels, or by cropping the extra pixels.

2. Each of these regions are rescaled independently to 10 pixels height.

3. The normalized text line is obtained by joining the scaled regions to each other vertically.

Figure 4.2(d) shows rescaled regions and figure 4.2(e) shows the normalized text line image obtained after joining these region vertically. The main goal of this normalization step is to ensure that the characters have approximately the same height for every character instance in the database and also to obtain the similar height text lines. The width of the text lines are determined relatively to the original image height and text lines are rescaled without affecting the ratio of x-height to the body height (one of the major characteristics that defines the appearance of a typeface). The input text line is normalized to 30 pixels height and each extracted region of the text line is rescaled to 10 pixels height as a fraction desired height ($\frac{desiredheight}{3}$).



(a) Input text line.

(b) Identification of baseline, mean-line, top most and bottom most points of the input text line.

(c) Extraction of ascender, middle and descender regions of the input text line.

(d) Rescaling of ascender, middle and descender regions to a specific height value.

(e) Normalized text line after vertical joining of the rescaled regions of the input text line.

Figure 4.2: Text line normalization process. The text line is divided into three regions. Each region is rescaled to a specific common height value. The rescaled regions are combined vertically to give a normalized text line image.

4.3.2 Feature Extraction

Feature extraction is an important step in building any recognition system. Features are extracted to get discriminative information from raw input. Selection of appropriate features is difficult task. In most pattern recognition applications, features are manually described and vary from application to application. It is commonly assumed that characters are recognized by analyzing the sub-orthographic features like lines, angels, and curves. Mostly, character recognition methods employed pixel based features, pixel distribution and structural features like dots, wholes, ascenders, descenders, and t-bars etc. In the current system, handcrafted features for the classification are not defined, rather artificial neural networks are used to learn the features directly from raw input. The neural network based features are generated from the entire text line for possible character and non-character or "garbage" classes. These extracted features are later used in the recognition process of the text line. The following subsections detail the steps during features extraction process.

Character segmentation

Character segmentation is required only to train the artificial neural network on character and non-character class instances. The possible character candidates are obtained by using a dynamic programming based character segmentation method [Bre01b]. In this method, text lines are over-segmented into sub-character images by curved pre-stroke cut segmentation. The algorithm evaluates a large set of curved cuts through the input image with the help of dynamic programming. A small optimal subset of cuts is selected for final text line segmentation. The algorithm is part of OCRopus OCR engine [OCR08] for character segmentation of Latin text. The output of the algorithm is a color coding that separates the characters from each other using a unique color code. Figure 4.3 shows the character segmentation results of an example text line.

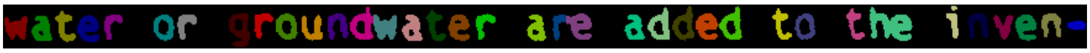


Figure 4.3: Output of character segmentation algorithm in color coded format. Pixels of each segmented character are assigned a unique intensity value and each segmented character is represented by a different color code.

Mapping function

The color coded text lines are used as basis for the identification of character and non-character positions on the normalized text lines. The pixels in the color coded text lines and the normalized text lines have different Cartesian coordinates due to change in image size during normalization process. A mapping function is defined to map the character positions from color coded text line image to the normalized text line image. The mapping function β maps the mass center point of each character in color coded text line to a point in the normalized text line. The mass center point of each character in a color coded text line and the mapping factors are computed by using following equations:

$$P(x_{mid}, y_{mid}) = (\frac{x_{top} + x_{bottom}}{2}, \frac{y_{top} + y_{bottom}}{2}) \tag{4.7}$$

Where x_{top}, y_{top}, x_{bottom}, and y_{bottom} are the top and bottom (x, y) positions of each character in color coded text line.

$$x_{mapping\_factor} = \text{height}_{normalized} * \frac{1}{\text{height}_{colour\_coded}} \tag{4.8}$$

$$y_{mapping-factor} = \text{width}_{normalized} * \frac{1}{\text{width}_{colour\_coded}} \tag{4.9}$$

Where $\text{height}_{normalized}$, $\text{height}_{color\_coded}$, $\text{width}_{normalized}$, and $\text{width}_{color\_coded}$ are heights and lengths of the normalized and color coded text lines.

The corresponding point on line normalized text line image was determined as following:

$$P(nx_{mid}, ny_{mid}) = (x_{mid} * x_{mapping\_factor}, y_{mid} * y_{mapping\_factor}) \tag{4.10}$$

The above equation provides an estimate of the center point for each character in the normalized text line.

Feature vector generation

As described above, the feature vectors are generated for character and non-character elements of the text line. A 30 × 20 (*height* × *width*) window is used for this purpose. The window is named as “contextual window” because it also captured the neighboring

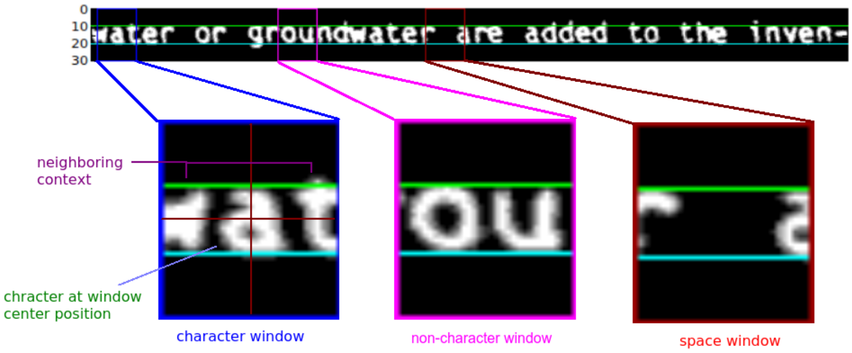

0
10
20
30
water or groundwater are added to the inven-
neighboring context
chracter at window center position
character window
non-character window
space window

Figure 4.4: Contents of different contextual windows on a text line. The figure shows an example character window, an example non-character window and an example space window on the text line.

context of the targeted character. The non-character elements are treated as a single "garbage" class that includes the examples in which two consecutive characters or their parts are appeared in the contextual window. The window is placed at the center of first possible character on the normalized text line, so that the base line is at $y = 20$, x-height is at $y = 10$ and the character is at the center of the window (taking $y = 0$ at top of the text line). The window width is adjusted to 20 pixels. This value is sufficient to enclose the widest character in the dataset and, in addition, to enclose the pixels from the left and right neighboring characters. This neighboring context is important to adapt the network in various degradations and noise. The contextual window moved from one possible character to another possible character to extract the feature vectors for valid characters. Feature vectors for non-character or garbage class are obtained by placing the window at the center of two consecutive characters as shown in Figure 4.4. Spaces are considered as valid characters and distinction between a space and a garbage class is made by computing the distance between two consecutive characters. If the distance is less than a specific threshold value then it is considered as garbage, otherwise it is considered as a space. Due to variations in inter-character spaces, the threshold was computed for every text line. A mean distance between all characters in a text line was computed and standard deviation was added to the mean. The sum of mean and standard deviation provides the threshold value for spaces. At each 30 × 20 contextual window, gray scale pixel values are used to construct the feature vector $x_i \in R^{600}$.

4.3.3 Auto Tunable Multilayer Perceptron (AutoMLP)

Multilayer perceptrons [Bis95] are used for the identification of character units on the input text lines. Multilayer perceptrons (MLPs) are feed forward artificial neural networks and consist of multiple layers of interconnected processing nodes, called neurons. The most common network architecture contains a input layer, a hidden layer and an output layer. The nodes at input layer are the non-processing elements and are only responsible to take the input from the environment. The nodes at hidden and output layers are the main processing elements with nonlinear activation function. The architecture is called multilayer and feedforward due to multiple layers of processing elements, and flow of information is forwarded form one layer to the other. Typically, in MLPs the layers are fully connected, this means that all the units in one layer are connected with all the units in the next layer. Multilayer preceptrons are usually trained by using supervised learning technique. In supervised learning, the correct results or the desired outputs from the network are known in advance and are used to adjust the network weights during training process.

Determining the suitable neural network topology and training parameters is not a straightforward task in most of the pattern recognition applications. Usually, the number of input and output nodes are determined from the application, but selection of appropriate number of hidden nodes requires lots of experimentation. Too few hidden nodes will prevent the network to fully learn the desired function and too many hidden units may cause the network to overfit the training data. In addition, the choice of learning rate and number of epochs also effect the performance of the trained network. In this work, auto tunable multilayer perceptron (AutoMLP) [BS10] is used to avoid the above mentioned problems during training. The AutoMLP works by combining the ideas from genetic algorithms and stochastic optimization. It maintains a small ensemble of networks that are trained in parallel with different learning rates and different numbers of hidden units using gradient based optimization algorithms [Bis95]. After a small fixed number of epochs, the error rate is determined on a validation set. The worst performer neural networks are replaced with copies of the best networks, modified to have different numbers of hidden units and learning rates.

The AutoMLP has been trained on character and non-character classes. In this work, 94 character classes -upper and lower case Latin characters, numerals, punctuation marks and white space– are used along with one extra garbage class. The network, in result, has total 95 output nodes corresponding to each class. The input layer got the input from the pixels enclosed in a 30×20 window, and it has 600 input nodes. The network is trained using supervised learning and a class label is provided for each class in batch mode. Training data is obtained from the text lines extracted from UNLV ISRI database. The text lines are preprocessed as described above, and features are extracted by placing the contextual window at each possible character and non-character positions on the text line. In this work, $57,600$ text lines are used that provide $3,869,327$ character and non-character instances for training the AutoMLP.

4.3.4 Text Line Scanning

The trained neural network can now be used to obtain the neural features for final recognition. The activations at the output layer can be interpreted as the probabilities of observing the character or non-character classes at a particular position on the text line. This leads us to the idea of text line scanning neural network (NN). The neural network is now able to provide the output in terms of posterior probabilities of observing a certain character or non-character at an arbitrary position on the text line. The text line scanning neural network operates by moving a contextual window, from left to right, on a normalized text line. The output of the line scanning neural network is a vector of posterior probabilities (one element for each character class) and it can be consider as a frame. The probability of a specific character class depends on the position of the contextual window on the text line. A maximum probability value for a particular character class is obtained when the character is fully aligned at the center of the window. The frame-wise output of the network is illustrated in figure 4.5. This kind of output is similar to the output generated by Graves et al. [GFGS06, GLF+08] using RNN and CTC architecture.

4.3.5 Hidden Markov Models

Hidden Markov Models (HMMs) have been successfully applied to continuous speech, handwritten and cursive text recognition [Jel97], [BSM99], [MB01] due to their ability to recognize time varying and dynamic patterns. HMMs can be realized as a stochastic finite state automata with several states. Transitions among the states are done by state transition probabilities. In addition to these transition probabilities, an emission probability is associated to each state, which provides the probability of observing a

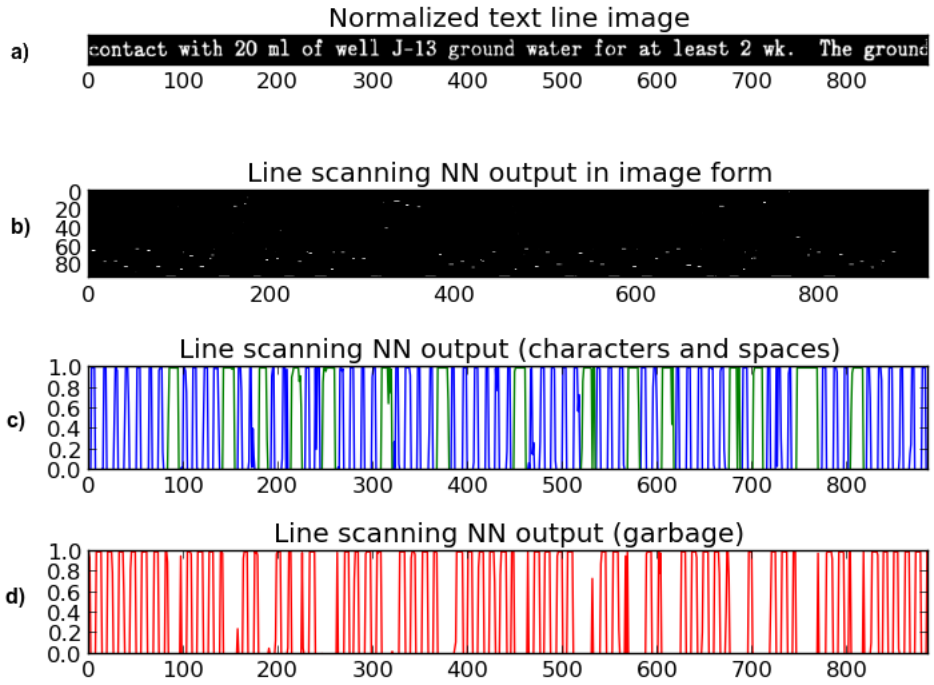


Figure 4.5: An illustration of the line scanning NN output. Each vertical slice obtained from scanning neural network is referred as a *frame*. (a) normalized input text line, (b) frame-wise output from line scanning NN in image form, (c) a graph of posterior probabilities for top classes (blue), for spaces (green), (d) a graph of posterior probabilities for garbage class (red).

feature vector at this state. The emission probabilities are the only outcome to the external environment and the states of the HMMs are hidden to the outside world, hence name as hidden Markov models [Rab90]. In HMMs based systems, the recognition process is done by aligning the feature vector sequences to the HMMs states and computation of the observation probabilities for observed patterns at each particular state. For more details about HMMs and their applications readers are referred to [Rab90] or to the Section 2.3 of this thesis.

In this work, hidden Markov models are employed for the recognition of entire text lines. The posterior probabilities of character candidates, obtained from the scanning neural network, are used as feature vectors for training the HMMs. The main idea is that the output of line scanning neural network can be interpreted as a left-to-right sequence of signals that are analogous to the temporal sequence of acoustic signals in speech. HMMs

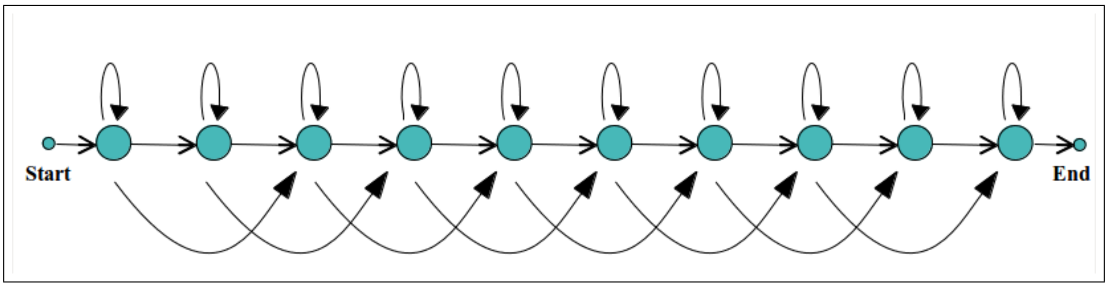


Figure 4.6: 10 states left-to-right HMM topology with self loops and one state skip.

provide adequate framework for modeling these temporal sequences into character units and decoding the sequences of characters with the help of bi-gram language modeling. The output frames, generated by scanning neural network, are treated as the observations for Gaussian mixture based HMMs. The output probabilities from scanning NN have very skewed distribution. The probabilities are smoothed with a Gaussian kernel ($\sigma = 0.5$) and are converted to negative logs before passing to HMMs.

In small vocabulary systems, it is possible to build an individual HMM for every text line or every word in the database. But in case of systems with large vocabulary, this method requires lots of training data and computational efficiency to provide a reasonable recognition rate. In addition, the choice of word level HMMs limits the scope of the recognition system to only recognize the words that exist in the lexicon. In this system, character level HMMs are used. The use of character models allows to provide an open vocabulary OCR system by using a fixed number of hidden Markov models, one model for each character in the database. In the following sections, hidden Markov models topology and training process are briefly explained.

Hidden Markov model topology

A simple left-to-right HMM topology is taken in modeling all the character classes. Each model in the system has left-to-right transitions, multi-states, self loops to the states, and one state skip. The HMM topology used in this system is shown in figure 4.6. This topology is followed because it suits to capture the intra-character variations and intra-character similarities during the scanning process. The self loops are provided for common intra-character features and skips are necessary to capture the abrupt intra-character variations. Moreover, this topology has also been applied in various HMMs based text recognition systems developed for recognition of multiple scripts [DNP$^+$05]. Selection of

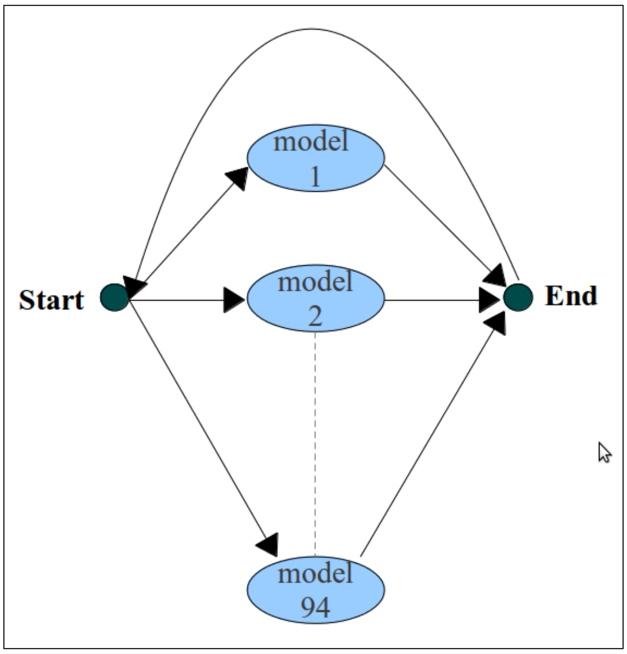


Figure 4.7: Ergodic HMM topology. In this topology any character model can be reached from any other model in finite number of transitions.

number of states for each character model is a design parameter [Sch03]. Usually, the optimal number of states varies from one character model to other, and depends on the representing features for that character. In most HMMs based OCR system, pixels based features are extracted by using a frame of certain width. Characters with small width may require less number of states in comparison to the characters that have larger width. However, in this system, a very different kind of features (posterior probabilities) are used with the help of a fixed size contextual window and a single number to represent the internal states of all character models can be chosen. Each character model in the database has 10 states and the number of states is empirically determined over a small set of validation data. In addition, two extra non-emitting states, "start" and "end", are used to provide the transitions from one character model to other character model. 94 character models are provided to represent all the characters in the database. Text lines are modeled by concatenating the character models in an ergodic HMMs topology [Sch09]. The ergodic topology, as shown in figure 4.7, enables the transitions from one character model to other character model. Spaces are treated as a valid character in the system and a separate HMM is used to represent the spaces during recognition process.

This system uses continuous density HMMs and the output probabilities are expressed with mixture of Gaussians. The basic function of HMM output modeling is to compute the probability of a feature vector if it is assumed that the underlying HMM is in particular state while the feature vector is being generated [Rab90]. Good recognition accuracy is obtained by using 256 Gaussian mixtures per state. This number is determined empirically over a small set of validation data.

Hidden Markov model training

Hidden Markov models have been trained iteratively over $57,600$ text lines from UNLV ISRI database. Features are extracted from all the text lines with scanning network approach. HMMs are trained by using the extracted features and textual transcriptions of the text lines. The training process estimates the parameters –state transition probabilities and feature probabilities distribution– of each character HMM. Training or estimating the HMM parameters is performed using Baum-Welch re-estimation algorithm [YKO+06], which iteratively aligns the feature vectors with the character models in maximum likelihood sense. The training algorithm guarantees a convergence of a local maximum of the likelihood function. The feature probability distributions are characterized by the *means*, *variances*, and *weights* of the Gaussian mixtures. Training is not performed at character level, rather entire text line is used for training.

4.3.6 Text Line Recognition

The recognition step shares the same normalization procedure as used for training the scanning neural network. Input text line is normalized to 30 pixels height. A 30x20 window is moved in the direction of writing for the extraction of neural network based features. The window is moved pixel by pixel and contents of the window are passed to the specially trained neural network (as illustrated in section 4.3.3). The frame-wise output of the neural network is collected and is passed to the trained hidden Markov models for final recognition. A global HMM is built for the whole text line by concatenating the final states of the character models to the first states of all the other character models. The Viterbi algorithm [Rab89] is applied to determine the best path in the global HMM. The best path represents the sequence of characters with the maximum probability, given the frame-wise scanning neural network features. Figure 4.8 shows the architecture of the proposed OCR system. The architecture makes it possible to recognize the input text lines by avoiding the difficult task of segmentation a text line into individual characters. In fact the basic input to the HMM are the neural features, extracted with the help of a artificial

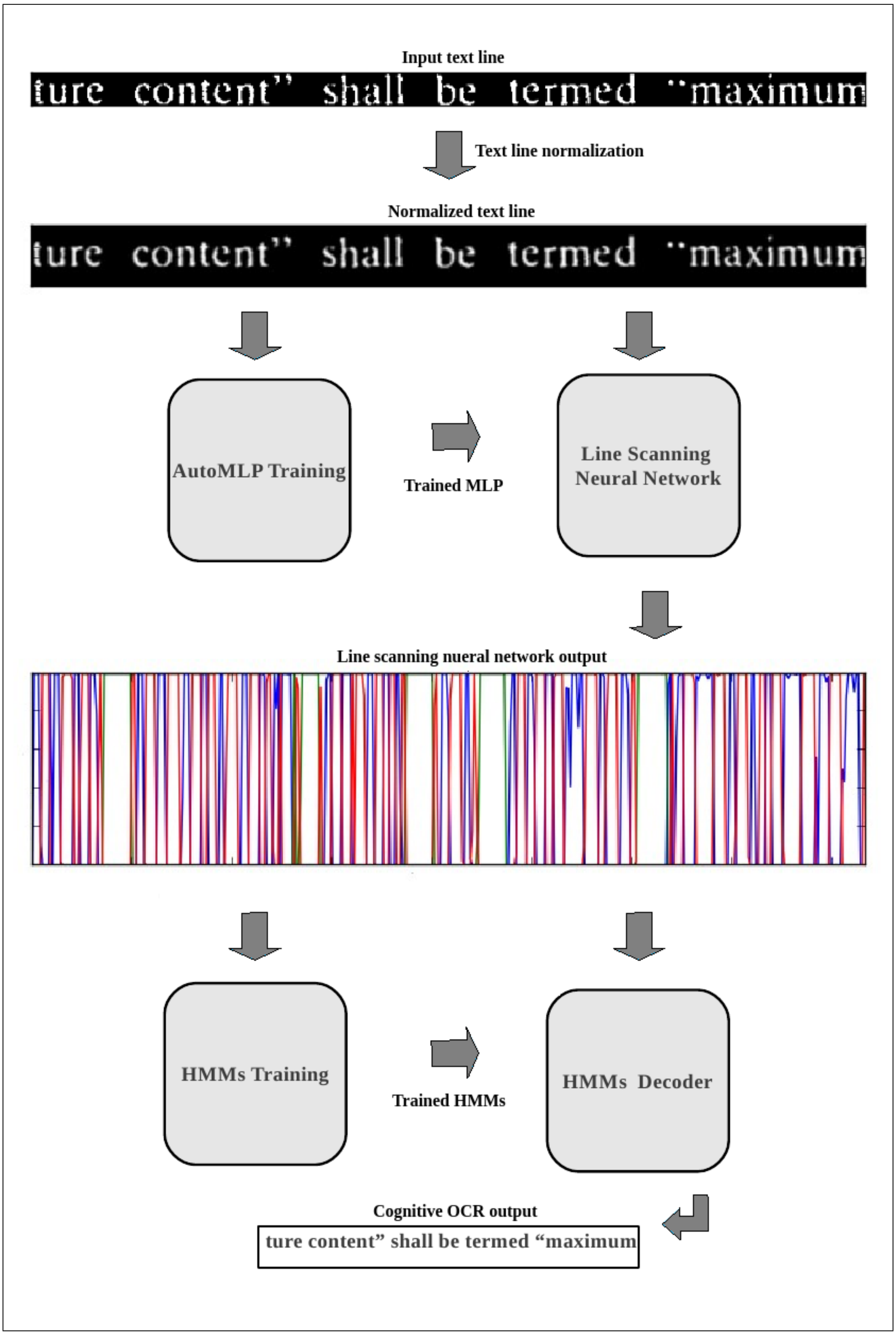

Input text line
ture content" shall be termed "maximum
Text line normalization
Normalized text line
ture content" shall be termed "maximum
AutoMLP Training
Trained MLP
Line Scanning Neural Network
Line scanning nueral network output
HMMs Training
Trained HMMs
HMMs Decoder
Cognitive OCR output
ture content" shall be termed "maximum

Figure 4.8: A combined ANN/HMM OCR system architecture.

neural network, from a complete text line. The output of the scanning neural network represents the neuronal signals in form of spikes for the detected character classes. The output can also be decoded into detected character sequences by using a peak detection (local maximum) algorithm.

Character bi-grams

The HMM based recognition process can be augmented by using the language level contextual knowledge. The language level contextual information can be introduced by using prior probabilities of the characters in the language considered. There are several ways to introduce the contextual knowledge into the recognition system. For example, the probabilities on a succession of characters like bi-grams or trigrams of characters can be considered to provide the contextual knowledge. Or a word lexicon or a dictionary of words extracted from some large text corpus can be used for this purpose. However, the lexicon based recognition puts a hard constraint on the system and it can only recognize the words that are available in the dictionary. In this work, the character bi-grams are introduced to augment the Viterbi decoding during recognition process. The character bi-grams have been trained explicitly during HMMs training from the entire training data.

4.4 Cognitive Reading Strategies

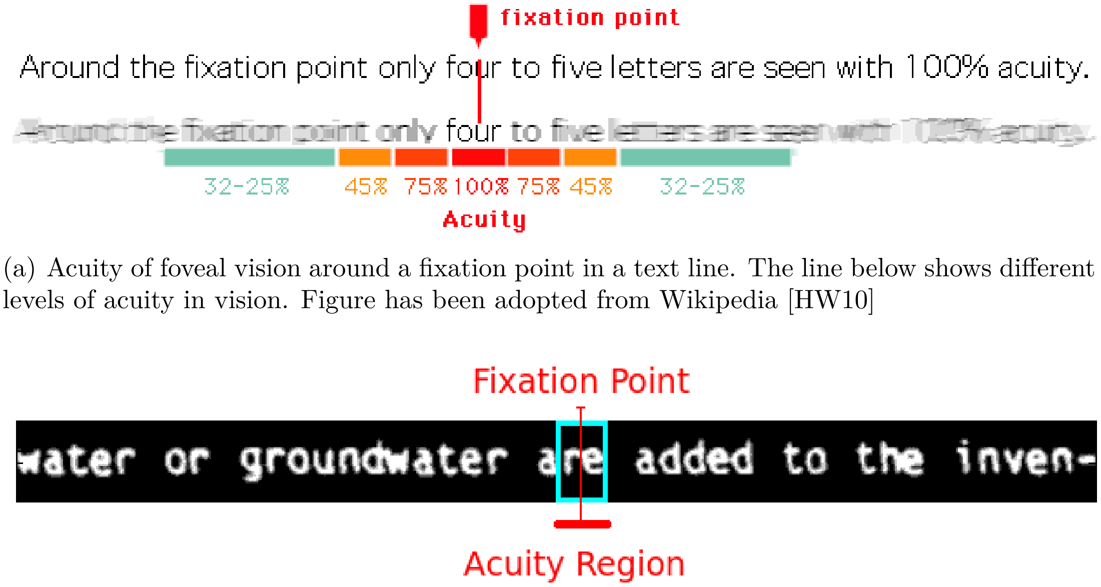
Human reading is a nearly analogous cognitive process to OCR that involves decoding of printed symbols into meanings. The reading process includes complete visual perception of the text, recognition of the iconic symbols on the page, and extraction of meanings from the text. During this process, people utilize their existing knowledge of the language and a mixture of information to derive meanings from the written text. For example, knowledge of the text orientation, syntax, semantics, morpheme, context and phonology are the important features for skilled reading. This section puts forward some cognitive reading strategies that have been used as a baseline for design and development of the proposed OCR method.

4.4.1 Eye Fixations and Retina Level Processing

The human visual system enables individuals to perceive information from the environment. People see things when the eye lens focuses an image onto a light sensitive membrane in the back of the eye. This membrane is called "retina" and it acts as a transducer to convert the patterns of light into neuronal signals. In the reading process, the retina level processing corresponds to eye fixations and perceptual acuity. For example, in case of reading English, eyes move forward from left to right across the text line with alternate "stable" and "moving" phases, called fixations and saccades. Fixations typically last for 250 to 500 milliseconds. Saccades are fast and ballistic movements that bring the gaze to another location on the text line. The average saccade size in reading is about 7-8 letter spaces and most saccades are forward movements. It is believed that vision is suppressed during saccades and visual perception from text is only obtained during fixations [Ray98].

In looking straight ahead, the visual field can be divided into foveal, parafoveal, and peripheral regions. Fovea is comprising the central 2° of the visual field and it is the area where visual acuity is very good. Acuity drops off markedly in the parafoveal region, that extends out to 5° on either side of the fixation point and it is even poorer in the peripheral region that is outside of the parafoveal range [RJP07]. The saccadic eye movements are mainly because of acuity limitation and the purpose of eye movements during reading is to place the to-be-processed text in the foveal region, where it can be most easily recognized.

Eye-contingent techniques make it possible to examine the eye movements during ordinary reading. The techniques rely on eye tracking equipments and computers that manipulate the text during a saccade. The reader is unable to perceive these text manipulations if the change is completed before the saccade has finished. For example, in "moving window technique" a portion of the text is appeared normally at the center of reader's fixation point and all of the text outside this "window" or text portion is replaced by something meaningless like "xxxx". Each time the fixation point changes the window is also moved so that there is a portion of normal text surrounded by some meaningless text. The theory behind this technique suggests that if the size of the window is large enough, reading will not be affected. McConkie and Rayner [MR75] are first who used this moving window technique and examined how many letters are required to provide normal reading experience. According to this study the perceptual span is around 15


fixation point
Around the fixation point only four to five letters are seen with 100% acuity.
four to five
32-25%
45%
75%
100%
75%
45%
32-25%
Acuity
Fixation Point
water or groundwater are added to the inven-
Acuity Region

(a) Acuity of foveal vision around a fixation point in a text line. The line below shows different levels of acuity in vision. Figure has been adopted from Wikipedia [HW10]

(b) Acuity region around a fixation point in a text line

Figure 4.9: Acuity in foveal vision and in the proposed system.

letters and if the reader is given 15 letters past the fixation point, then reading speed is just as fast as normal reading without moving window.

In this work, retinal level processing is incorporated with the help of the moving window technique. A window is defined that corresponds to the foveal region, same as in the human visual field, and the information lying at the center of the window is processed by higher level processing units. This system mimics the eye fixation behavior and is similar to fixating at the region of a visual field that need to be processed. The perceptual span is defined by the width of the window which is an adjustable parameter. Currently, window width is fixed to 20 pixels that is suitable to enclose every single English alphabet available in the database. In this work, saccadic movements are not used, rather pixel by pixel scanning of the text line is employed. The saccadic eye movements are only due to the acuity limitations and saccades are made to perceive the visual stimulus that is closer to fovea. In the present system, the whole window is considered as foveal region and therefore the saccade span is limited to one pixel only. However, the window can scan across the text line by more than one pixel transitions. In reference to human reading, the visual span is roughly 15 letter spaces and the average saccade length is about $7-8$ letters. So according to these finding the window can be move forward by making saccades of $8-10$ pixels. Figure 4.9 shows the foveal acuity and eye fixations during reading process, and formation of acuity region (a scanning window) in the present system.

4.4.2 Serial Order Scanning

Some psychologists believe that word recognition is based on the shape of the visual stimulus and words are recognized as a single unit. Many psychologist and word recognition models disagree and suggest that words are recognized analytically using letters as abstract identities for processing. Among these models, there is another debate on how brain processes these letter identities, serially or in parallel. In parallel input models, orthographic components of input stimulus are processed in parallel, but in serial models, components are processed in sequential order [Dav99].

Left to right serial scanning has been justified by many models of reading. Sperling [Spe60] presented the idea of scanning visual information in a short term visual store. He argued that the presentation of visual stimulus forms a two dimensional representation in visual information storage. The information in visual storage remains only for few seconds unless it is overwritten by new information due to saccade to a new location. In some other experiments, Sperling [Spe63] examined how rapidly a letter could be read out of this visual storage. He employed visual masking procedure in which he showed strings of random letters to the participants. These random strings had been shown in between the display of a forward and a backward mask (random dot patterns). The display duration of random strings was varied from 0 to 60 milliseconds. It had been observed that there was a linear trend between letter identification and time, and participants were able to report one additional letter for each additional 10 milliseconds of delay.

In consistence with Sperling's [Spe63] findings, Gough [Gou72] suggests that the reading process starts with the formation of an icon. (In this process) When the eye fixates on a text line, the image in the visual field is registered in an iconic memory. The iconic memory is of a shorter duration but it roughly contains 20 letter spaces of the text line during fixation. The iconic memory is then internally scanned letter by letter, into a character register which is more durable form of storage. However, the iconic scan does not correspond to the visual scan involving in physical eye movements.

Models that include the word length effect during word recognition process also support the serial letter recognition hypothesis. The obvious implication is that shorter words are recognized faster than longer words. Cattell [Cat86] found that long words took more time to name than shorter words. This result has been observed frequently in the standard naming and in perceptual identification studies [Dav99]. The word length effect is also observed in the visual word recognition experiments that have been presented in Chapter 5 of this thesis, both for words and non-words, even though these experiments do not require the production of a pronunciation.

In contrast to serial processing of letter strings, some psychologists believe in parallel processing of letter strings [MR81, HS99, CRP$^+$01]. It has been assumed that if there is no effect of reaction/response time (RT) on the numbers of items to be processed, it can be considered as the basis of parallel processing. However, Whitney and Lavidor [WL04] argued that serial processing could fail to yield a length effect if increased length also have counter balancing facilitative effect. An increase in numbers of letters in the word may reduce the time for the settlement of lexical network as compared to the shorter words. The increased letter processing time and the decreased lexical settlement time may cancel each other and therefore there is no length effect in serial processing. Some recent electroencephalography (EEG) studies in lexical decision also support this scenario and show that word length have no effect on RT but it yields complementary effects on EEG amplitude at different time periods [HP04]. A more recent study [NFPB06] provides the analysis that once the effects of frequency, number of syllables and orthographic neighborhood size were factored out, RT shows a reverse behavior and RT decreases with increase in length for three to five letter words and it remains constant for five to eight letter words.

In this work, the serial letter scan approach is adopted. The serial letter scan is considered as a more obvious phenomenon in reading as concluded by above mentioned studies. Some other implications in English reading include

- Reading is inherently left-to-right process
- Eyes skip from left-to-right across the page
- Sequences of words are read serially left-to-right and the sequences of letters are also read from left to right

Therefore, in the proposed OCR system, text lines are processed from left-to-right in search of possible letter candidates.

4.4.3 Neural Processing

People constantly receive different kinds of sensory inputs from the environment and are able to process this information with great ease. People can easily distinguish among several object categories, recognize persons, distinguish between sounds, identify different kinds of tastes and smells, and are able to perform computationally demanding perceptual task effortlessly. The reason behind this amazing performance is the massive (and) parallel neural structure of human brain. The human brain is composed of more than 10 billion basic processing units called neurons. Neurons are connected to each other and work in parallel to perform a specific cognitive task. The network of these connected neurons is referred as biological neural network.

The complex structure and working mechanism of human brain is not understood completely, but it is possible to model the higher level working mechanism of biological neurons in the form of artificial neurons. Connectionist or neural models of reading emerged largely to describe the explicit behavior of human brain during the reading process [Pla05]. In these models, the cognitive processes involved in reading are implemented in the form of a cooperative and competitive interaction of simple neurons. Typically, each neuron is modeled using a real valued activity function. Interaction among the neurons is governed by weighted connections that represent the knowledge of the system. These weights are gradually learned through experience. The units are organized into layers or groups, and activities of different layers are processed hierarchically among the layers in a feed forward or a bi-directional way. In this scenario, activities at some group of nodes may encode the input to the system and activities at some other groups of nodes may show the output or system's response based on this input. In this way, a group of layers can be built to encode different kind of knowledge like, written form or orthography, spoken form or phonology, and meaning or semantics of the word.

Most of existing connectionist models of reading focus on the processing of a single word. However, these models differ widely in employing underlying knowledge representation and processing mechanism. The representation of a word in a particular model can be localist (i.e. one processing unit per word) or distributed (multiple units for each word). McClelland and Rumelhart presented the first localist, non-linear connectionist model of reading, "the interactive activation and competition (IAC) model of letter and word perception" [MR81, RM82]. Later, Seidenberg and McClelland [SM89b], and Plaut et al. [PMSP96] further progressed in the development of neural network based models of

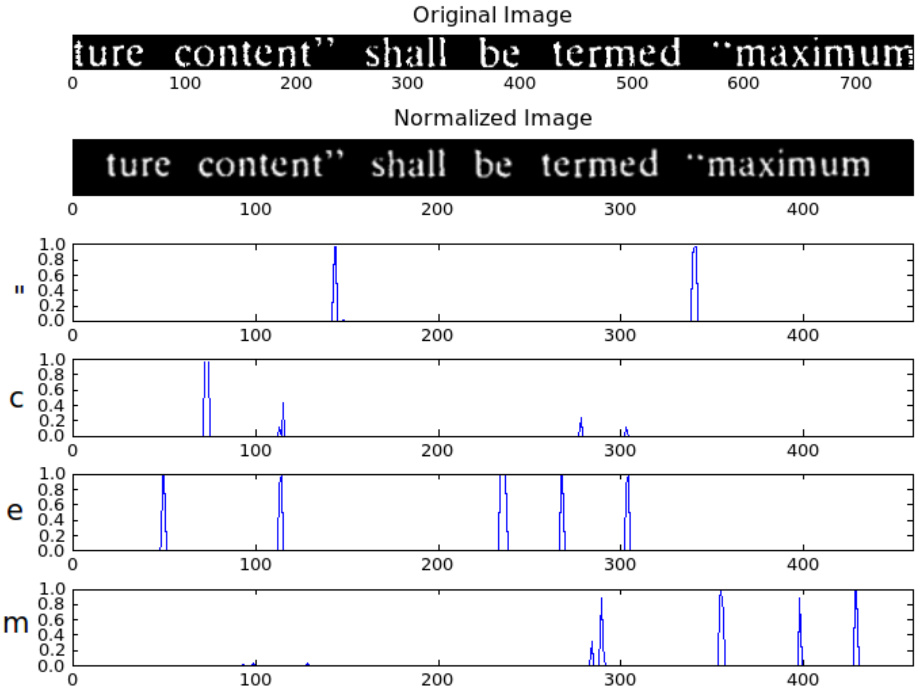


Figure 4.10: Output activations of the line scanning neural network at different character classes. Peaks represent the higher posterior probability of particular character classes on the text line image.

reading to capture more human reading behaviors. Mozer [Moz87, Moz91] presented a connectionist model of visual object recognition and spatial attention. The model is called MORSEL (Multiple Object Recognition and Attentional Selection) and it has been trained to recognize letters and words at various positions on its "retina." The system is hierarchically organized in the form of layers. Each layer in the recognition system consist of units with spatially restricted receptive fields. At the top of the system are position independent units, that respond to specific letter triples like '#HO', 'OUS', 'USE', 'SE#' for the word 'HOUSE'.

This work utilizes a connectionist neural architecture in the form of multilayer perceptrons to recognize letter units present in the input text line. Multilayer perceptron [Bis95] architecture is composed of a large number of interconnected neurons and these neurons work in parallel to solve a specific problem, similar to biological neurons. The neurons are connected to each other in hierarchical layers of nodes. The input nodes contain the activities in the form of pixel intensities. The pixel intensities are enclosed in a window which slides across the input text line. The input layer is referred as retina layer because it corresponds to the visual stimulus in the foveal region of the system. The activities at retina layer are passed to the next processing nodes (hidden nodes) via weighted connections. The activities at hidden nodes are computed by weighted sum of the input nodes and are passed to the final output layer. The final nodes also compute the weighted sum of values coming from hidden nodes and provide the output after applying a "squashing" function. The weights have been learned during a supervised training process, and these weighted connections represent the knowledge of a character recognition system. The activities at the output layer are dependent on the activities at the retina layer and the hidden layer. Whenever the retina layer finds a possible character candidate in the visual span of the window, a high spike is generated on output nodes corresponding to that character class. The spike correspond to the posterior probability of the top candidate character within the window as shown in Figure 4.10.

4.4.4 Local Contextual Analysis

Most researchers believe that reading is much more than recognizing isolated characters. People use various contextual and semantic sources to understand the written text efficiently. Context can be considered at different levels of knowledge representations and is often required for reliable recognition. In order to build a recognition system based on human visual perception, local contextual coding may play a role in determining the contextual relation of individual characters.

The usage of local context is initially adopted from the work of Wickelgren [Wic69] who proposed the concept of 'wickelphone' to encode positions of of phonemes in speech. The idea is first used by Seidenberg and McClelland [SM89a] in the form of 'wickelgraphs' or letter triples for encoding of letter strings. According to their proposed model, word 'BLACK' can be coded as '#BL', 'BLA', 'LAC', 'ACK', and 'CK#' where # represents a space. This kind of arrangement is unique and the five wickelgraphs cannot be re-arranged to make any other word of English.

Most recent schemes for local context coding are based on bi-gram contextual analysis. For example, the SERIOL model [Whi01] uses the idea of 'open bi-grams' in orthographic processing of the input letter string. According to the model, bi-gram nodes can recognize ordered pair of letters, corresponding to the neuronal units that fire only if 'A' is followed by 'B'. The activation of bi-gram nodes depend on the activation of letter nodes representing the constituent letter and the difference of time between the firing of those nodes. Later, Whitney [Whi08] proposed several refinements to the model based on the evidence for a special role of external letter and sequential behavior of letter activations. For example, the input string 'CART' first activates the bi-gram '*C' (when letter node 'C' fires), then 'CA' (when 'A' fires), 'AR', 'CR', 'RT', 'AT' and 'CT' and then 'T*', where '*' shows the presence of 'space'. Due to the effect of temporal separations, bi-grams '*C, CA, AR, RT, and T*' get the maximum activation level.

Bi-gram encoding is a basic and simple way to include the contextual knowledge. In the present work, the contextual knowledge is included by using letter bi-grams. The bi-gram knowledge is implicitly built during character level hidden Markov models (HMMs) training. A brief description about selection of HMMs topology and training is already provided in Section 4.3.5.

4.5 Experimental Results and Evaluations

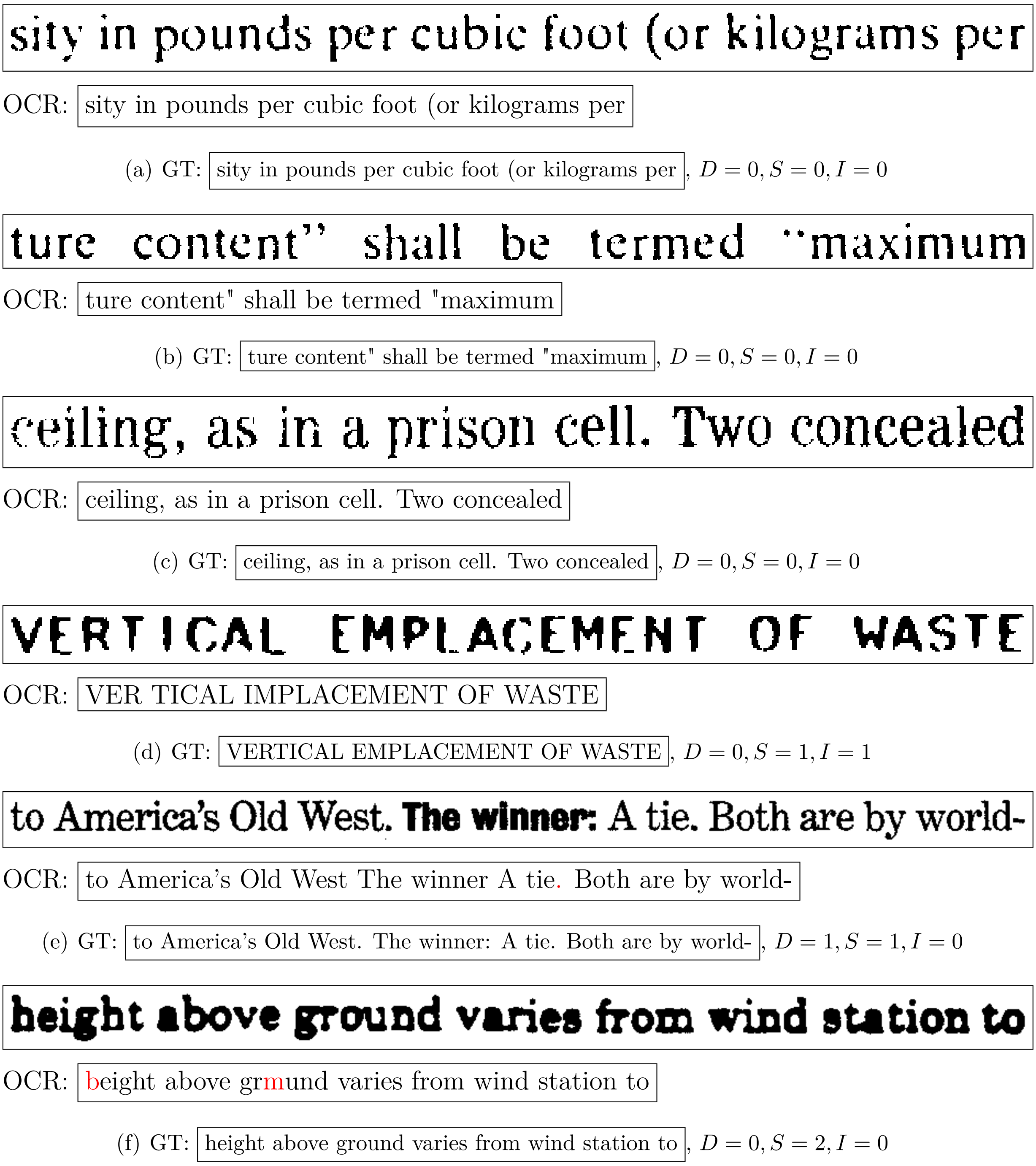
The proposed OCR system is evaluated on degraded text lines extracted from UNLV-ISRI document database. A subset of $1,060$ text lines is used in evaluation. The text lines contains major degradations like noise, broken or merged character instances and lots of variations in terms of fonts, fonts sizes and font styles. Some representative input text line images and OCR output of the proposed method are given in figure 4.11.

The performance evaluation is carried out by computing percentage of character recognition accuracy (CRA%). The CRA is determined using edit distance operations [Lev66]. Edit distance counts the number of insertions, deletions and substitutions by comparing the OCR output with the ground truth text. The CRA% is computed using the following equation.

$$CRA\% = \frac{N - ED}{N} \times 100 \tag{4.11}$$

$$\text{Edit distance} = ED = I + D + S \tag{4.12}$$

where N = total number of characters, I = *insertions* , D = *deletions*, and S = *substitutions* (with equal cost).


sity in pounds per cubic foot (or kilograms per
ture content'' shall be termed ``maximum
ceiling, as in a prison cell. Two concealed
VERT ICAL EMPLACEMENT OF WASTE
to America's Old West. The winner: A tie. Both are by world-
height above ground varies from wind station to

OCR: sity in pounds per cubic foot (or kilograms per

(a) GT: sity in pounds per cubic foot (or kilograms per, $D = 0, S = 0, I = 0$

OCR: ture content" shall be termed "maximum

(b) GT: ture content" shall be termed "maximum, $D = 0, S = 0, I = 0$

OCR: ceiling, as in a prison cell. Two concealed

(c) GT: ceiling, as in a prison cell. Two concealed, $D = 0, S = 0, I = 0$

OCR: VER TICAL IMPLACEMENT OF WASTE

(d) GT: VERTICAL EMPLACEMENT OF WASTE, $D = 0, S = 1, I = 1$

OCR: to America's Old West The winner A tie. Both are by world-

(e) GT: to America's Old West. The winner: A tie. Both are by world-, $D = 1, S = 1, I = 0$

OCR: beight above grmund varies from wind station to

(f) GT: height above ground varies from wind station to, $D = 0, S = 2, I = 0$

Figure 4.11: Representative text lines and proposed OCR performance. OCR = output from the proposed OCR method, GT = ground truth, D = deletions, S = substitutions and I = insertions.

Table 4.1: Confusion matrix for top confusions during OCR.

	n	**,**	**a**	**m**	**.**	**t**	**O**	**l**	**S**	**1**	**d**	**s**	**$**	**o**	**P**	**i**	**~**	**r**	**8**
n	2774	0	0	56	0	0	2	0	0	0	0	2	0	1	0	0	0	0	0
,	0	438	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	3320	0	0	1	0	0	0	0	0	2	0	15	0	0	4	0	0
m	13	0	0	947	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
.	0	12	0	0	509	1	0	0	0	0	0	0	0	0	0	0	1	0	0
t	0	0	1	0	0	3558	0	1	0	0	1	1	0	0	0	2	0	10	0
O	0	0	0	0	0	0	37	0	0	0	0	2	0	9	0	0	0	0	0
l	0	0	0	0	0	1	0	1631	0	0	0	0	0	0	0	5	0	2	0
S	0	0	0	0	0	0	0	0	132	0	0	5	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	5	0	124	0	0	0	0	0	0	0	0	0
d	0	0	1	0	0	0	0	1	0	0	1468	0	0	0	0	4	0	1	0
s	0	0	0	2	1	0	0	0	3	0	1	2653	0	0	0	1	0	4	0
$	0	0	0	0	0	0	0	0	3	1	0	0	2	0	0	0	0	0	4

Table 4.2: Top deletions and insertions during OCR.

Character	**Deletions**	**Insertions**
space	**106**	**49**
l	**11**	1
i	**9**	2
I	**7**	3
.	**7**	2
,	**5**	3
:	**4**	0
f	**4**	2
S	**4**	0
m	**4**	2
~	1	**4**

The system achieves 98.41% recognition rate ($N = 51,261$) on the test set. There are total of 817 ($D = 207, S = 525, I = 85$) errors. Most of the errors are due to deletions (106 times) and insertions (49 times) of *space*. Other top confusions and deletions are given the Table 4.1 and Table 4.2.

To compare the recognition performance with other contemporary OCR systems, ABBYY FineReader 10 professional [ABB11], Tesseract 3.1 OCR engine [Smi07] and OCRopus 0.4 [Bre08] OCR systems are evaluated on the same test dataset. ABBYY FineReader is evaluated by using “English“ settings in batch mode for all the text lines. Tesseract is evaluated in line-wise mode with default options. For OCRopus, the evaluation is performed using segmentation based recognizer. The recognizer internally uses the same AutoMLP classifier for recognition of segmented characters.

Additionally, two segmentation free HMMs based OCR systems are developed as a baseline to compare the performance of the proposed method with standard state-of-the-art HMM based segmentation free OCR approaches. The baseline HMM OCR systems are trained on the same training set that has been used for training the proposed ANN/HMM OCR method. Two different features, plain pixel values and gradient intensity values, are used in training the baseline HMM OCR systems. The baseline systems share the methodology that has been used in the development of HMMs based low resolution screen OCR. A complete description of the features and HMM based OCR methodology is already presented in Chapter 2. Several experiments are conducted to optimize the HMM parameters of baseline OCR systems, and only the best obtained recognition results are reported for comparison.

Figure 4.12 shows the character recognition accuracies obtained from the proposed OCR system, baseline OCR systems and current state-of-the-art OCR systems on a common test set of $1,060$ UNLV-ISRI text lines. The proposed ANN/HMM approach gives statistically significantly better performance in comparison with the baseline HMM OCR and state-of-the-art public OCR systems at character level. The proposed system resulted in an impressive 61% reduction in the error rate of the baseline HMM OCR and it reduces the error rates of OCropus and Tesseract by 43% and 30% respectively. However, ABBYY fine reader (a commercial OCR system) gives higher recognition accuracy as compared to the proposed method. This may be due to the use of sophisticated preprocessing and post-processing techniques, and application of the language modeling in ABBYY OCR system.

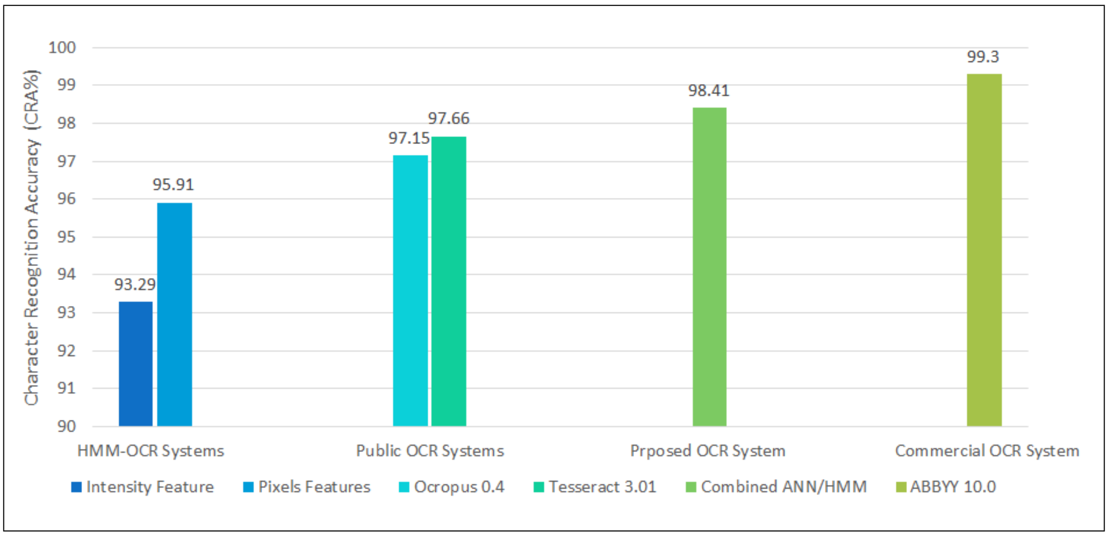
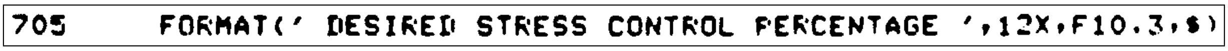


Figure 4.12: Character recognition performances of the proposed OCR approach and other state-of-the art OCR systems. The CRA% represents percentage of character recognition accuracy for each OCR system.

4.6 Discussion

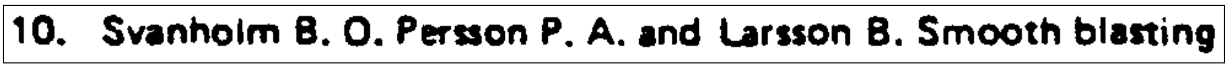
This chapter introduced a combined ANN/HMM OCR approach for degraded text line images. The proposed method avoided complex neural network architectures and complicated global hybrid system training. Rather, the proposed method employed a simple hybrid architecture in which ANNs and HMMs are trained independently on a common dataset and the trained models are combined to give segmentation free OCR. The method used widely adopted multilayer perceptrons to extract discriminative features from degraded text line images for HMMs training. The neural network is trained on 95 character classes including lower and upper case characters, numerals, punctuations, brackets, empty space and a special "garbage" class. The novelty of the proposed method is to provide a line scanning mechanism in which text lines are traversed serially from left-to-right and neural features –as posterior probabilities– are extracted by analyzing contents of a contextual window. The neural features are further processed by character level HMMs to give segmentation free open vocabulary OCR. In the approach, input text lines were normalized using a novel normalization method in which typeface characteristics of the constituent letters are maintained after normalization of the text line. Language context is incorporated into the system using character bi-grams that have been learned implicitly from ground truth text. Interestingly, the proposed approach also realizes several cognitive reading based strategies like eye fixations, serial scanning



705 FORMAT(' DESIRED STRESS CONTROL PERCENTAGE ',12X,F10.3,$)

OCR: 705 FORRAT(' DESIRED STRESS CONTROL FERCENTAGE" r 12 ~~~ F10 ~~~ 9

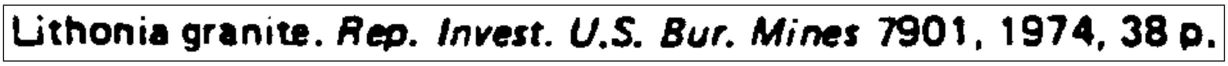
(a) GT: 705 FORMAT(' DESIRED STRESS CONTROL PERCENTAGE ',12X,F10.3,$, $D = 0, S = 11, I = 4$



10. Svanholm B. O. Persson P. A. and Larsson B. Smooth blasting

OCR: 10. Svanhoirn B. O. Permson P. A. and Larsson B. Srnooth blarting

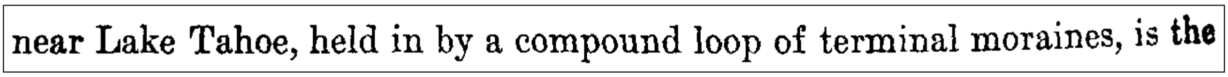
(b) GT: 10. Svanholm B. O. Persson P. A. and Larsson B. Smooth blasting, $D = 0, S = 5, I = 2$



Lithonia granite. Rep. Invest. U.S. Bur. Mines 7901, 1974, 38 p.

OCR: Lithmis vanine. Ae. tmyer. U.3. aur wser 7901,1974, 38. 8.

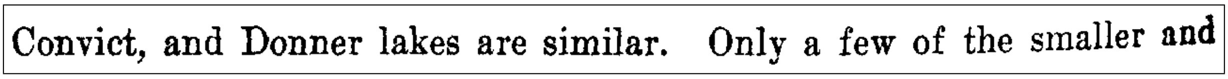
(c) GT: Lithonia granite. Rep. Invest. U.S. Bur. Mines 7901, 1974, 38 p., $D = 6, S = 16, I = 0$



near Lake Tahoe, held in by a compound loop of terminal moraines, is the

OCR: near Lake Tahoe, held in by a compound loop of terminal DiOtilBPR 19799

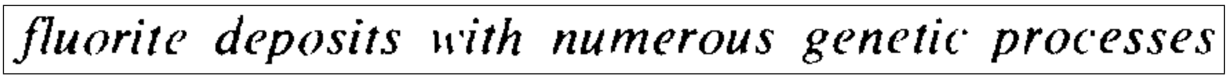
(d) GT: near Lake Tahoe, held in by a compound loop of terminal moraines, is the, $D = 1, S = 13, I = 0$



Convict, and Donner lakes are similar. Only a few of the smaller and

OCR: Convict, and Donmer lakes are similar. Only a few of the Sth8280~~~~

(e) GT: Convict, and Donner lakes are similar. Only a few of the smaller and, $D = 0, S = 12, I = 0$



fluorite deposits with numerous genetic processes

OCR: fivorire deposirs ~ith munerous generic processes

(f) GT: fluorite deposits with numerous genetic processes, $D = 0, S = 8, I = 0$

Figure 4.13: Proposed OCR system performance limitations. (a) to (c) severely degraded text lines, (d) and (e) nonuniform skew at the end of text lines, and (f) italics with degradations.
OCR = output from the proposed OCR method, GT = ground truth, D = deletions, S = substitutions and I = insertions.

of text lines, neural processing and contextual analysis during OCR. The system did not use any post-processing or language modeling and evaluation results are reported in terms of character recognition accuracy. Evaluation results reveal that the proposed OCR system outperforms the state-of-the-art open source OCR systems and baseline HMM-based OCR systems, and gives statistically significant improved recognition accuracy. Most of the recognition errors are due to insertions (49 times) and deletions (106 times) of *space* and n confused with m (56 times). Usually, *space* is not consider as an important alphabet in other OCR systems because most OCR methods recognize the isolated characters and *space* can easily be ignored. But the presented approach recognizes the complete text line without character segmentation and *space* is treated just like other alphabets in the system. The confusion of n with m is mostly due to the disambiguates caused by broken "m" from upper left corner of the character. Generally, the recognition errors are occurred due to sever degradations, presence of nonuniform skew and degraded italic styles in the text lines. Figure 4.13 shows such example text line images and recognition output of the proposed system.

Chapter 5

Visual Recognition of Permuted Words

Previous chapters of this thesis presented approaches to solve different problems in OCR from engineering perspective. In this chapter, in contrast, a behavioral study that deals on how cognitive system works in visual recognition of words is presented. Visual recognition of words is a nearly analogous cognitive process to OCR that involves decoding of printed symbols into meanings. Studying the cognitive reading behavior may help in building a robust machine reading strategy. The work is motivated by an Internet meme that suggests that people can read permuted or jumbled words as easily as normal words, provided the first and last letters of permuted words are at their original position. This chapter empirically investigates whether the reading of such permuted words and normal ones are based on the same underlying cognitive mechanisms. A hypothesis is proposed that reading of words and permuted non-words are two distinct cognitive processes and people use different strategies in reading normal and permuted text. The hypothesis is tested by conducting psychophysical experiments in visual recognition of words and permuted non-words. The experiments involved a slightly modified lexical decision tasks of Latin and Cursive scripts languages (English, German and Urdu). Participants had to categorize permuted or normal words (animal names) as birds or non-birds. A response time and recognition error rate analysis show that it takes longer to recognize permuted non-words compared to normal words. The error rate is also higher for permuted non-words. The results support the presented hypothesis and the findings are consistent with the dual route theory of reading.

5.1 Introduction

An important distinction in building OCR system is the use of segmentation based or segmentation free character recognition approaches. Segmentation based or analytical approaches work by segmenting words into characters, recognizing the individual characters and providing a word level interpretation using a dictionary or lexicon. In contrast, the segmentation free approaches or more precisely holistic approaches treat the words as single entity and words are recognized on the basis of features from the overall word shape. The holistic approaches are partly inspired from early psychological studies of human reading and models of visual word recognition that suggest that words are cognitively recognized as a whole. This chapter investigates a bit more in this direction and presents some psychophysical experiments in visual recognition of words and permuted non-words.

The study [1] is motivated by following meme circulated over the Internet [Dav09]:

> "Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it doesn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe."

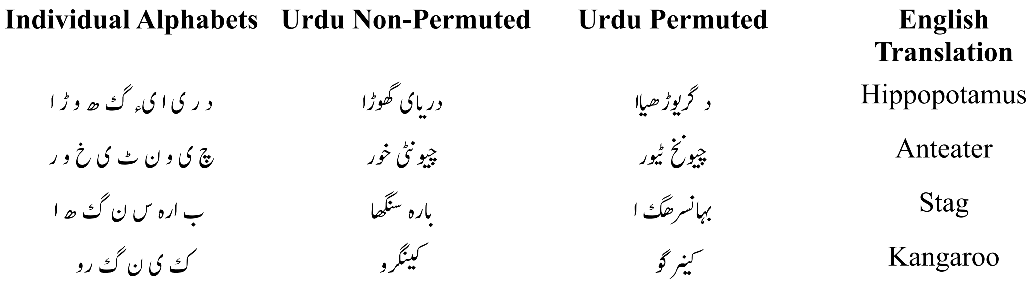
The meme does not have any existing bases in research and does not belong to University of Cambridge. However, an empirical investigation about the underlying cognitive behavior of reading permuted text may leads to formulate basis for the improvements in OCR technology. It is somehow true that humans, as general pattern solvers, are able to read permuted text but underlying cognitive process of reading permuted and normal text may or may not be the same. One of the goals of this work is to prove or disprove the statement presented in the meme. A hypothesis is formulated that recognition of permuted non-words ("Aoccdrnig" is misspelled –three letters have been permuted– but people are still able to read it as "According") and normal words are two distinct cognitive processes and people use different strategies in handling permuted words as compared to normal words. The study is presented in the context of dual route theory of reading and it is observed that the obtained results are consistent with the dual route theory. The data supports presented hypothesis that there are distinct processes underlying the cog-

[1] A part of the study is based on the author's work in [RSB10c]

nitive behavior for reading words and permuted non-words. It is suggested that the word recognition process by a skilled reader is carried out via the direct route i.e., orthographic to recognition while the recognition process for permuted non-words follows the indirect route i.e., orthographic to phonology to recognition.

In the past years, many interested experiments have been reported to study the jumbled or transposed letter (TL) effect in various tasks like lexical decision, naming, semantic categorization, orthographic priming, and silent reading [And96], [Cha79], [PL03a], [PL03b], [PL04], [DB04]. Rawlinson [Raw07] conducted a series of experiments to determine the significance of letter positions in word recognition during his Ph.d. One of the objectives of his research is to examine the current theories of the time, which included the idea that words are recognized through shape. Recent experiments in masked priming effect showed the robust form orthographic priming effects when primes and target words shared the same letters but in different order. The phenomenon is referred to as transposition priming which shows that primes build by transposing two adjacent letter in the target word (e.g. “gadren” is used as a prime for target word “garden”) facilitate word recognition compared with the other controlled primes [PL03b, SG04]. Interestingly, Perea and Lupker [PL04] found that the transposed letter priming effects are not restricted to the transposition or permutation of adjacent letters in the word and non-adjacent transposed letter non-word prime gives robust priming effect in comparison to the orthographic controls, e.g., caniso-CASINO vs caviro-CASINO. Furthermore, they analyzed that the non-words created by permutation of adjacent letters are highly like to the base words. For example it is also concluded from various experiments that the transposed letter “jugde” is much more similar to its base word “judge” than the substituted letters non-word “junpe” [JRP07]. Perea & Lupker [PL03a, PL03b] also examined the difference between transposed letter internal “uhser” and transposed letter final “ushre” non-words in priming base word “usher”. They found that the transposed letter internal primes have strong effect in comparison to the transposed final primes and substituted letter primes. They also concluded that the transposed letter final non-words are not effective at activating semantic information from their base words. Rayner et al. [RWJL06] reported that a change in the position of first or last letter of words in a sentence causes delays in reading in comparison with normal text (26% decrement in speed of reading for transposition of first letter and 36% decrement in speed of reading for transposition of last letter).

In this work, three different languages –varying in script, writing directions, fonts and



Individual Alphabets	Urdu Non-Permuted	Urdu Permuted	English Translation
د ر ی ا یء گ ھ و ڑ ا	دریای گھوڑا	د گریوڑھیاا	Hippopotamus
چ ی و ن ٹ ی خ و ر	چیونٹی خور	چیونخ ٹیور	Anteater
ب اره س ن گ ھ ا	بارہ سنگھا	بھانسرھگ ا	Stag
ک ی ن گ رو	کینگرو	کینرگو	Kangaroo

Figure 5.1: Urdu words in permuted and non-permuted forms. The visual appearance of each alphabet changes with the change of its position inside the word resulted in overall word shape degradation.

orthography– are selected to compare different visual aspects in reading behavior of people for permuted non-word strings. Psychophysical experiments are designed to determine the effect of letter permutation (shape distortion) in visual recognition of words taken from English, German and Urdu. The German and Urdu subsequently add more complexities relative to English in terms of grammar and orthographic structure. For example, German has many more grammatical endings and compound words, and Urdu has entirely different orthographic structure. Urdu is a cursive script language which is written from right to left, letters have connections to each other and individual letters lose their basic shape when they merge with other letters to form a word. The final shape of each letter is dependent on its position (beginning, middle, or end) within the word and shape of the letter changes with the change in its internal position. Therefore, in case of Urdu, global word shape is highly dependent on component letters and their positions inside the word. The change in letter positions entirely changes the overall shape of the word. Figure 5.1 shows the shape variation of of Urdu words in permuted and non-permuted forms along with individual alphabets.

5.2 Literature Review

The early theories of visual word recognition suggest that words are recognized as a whole, on basis of their shapes, and not in terms of their component letters. Cattell [Cat85, Cat86] is considered to be the first to propose the word shape model. This model posits that words are recognized as complete units by visualizing the ascending, descending and neutral patterns of individual characters. These patterns only exist in lowercase text due to the presence of ascending and descending portions of text. Word shape model is also supported by studies done by Woodworth [Woo38], Smith [Smi69], and Fisher [Fis75], who found that people can read lowercase text 5-10 % faster than uppercase text.

Another theory is that words are formed by letters in a word and this letter based information is used to recognize the whole word. Gough [Gou72] proposed that words are recognized serially letter by letter from left to right. Analytical models such as the search model [For76], the interactive-activation model [MR81], the activation-verification model [PNMS82] and the multiple read-out model [GJ96] assume that information about visual word shape is lost early in the process of word recognition, therefore the particular word shape is irrelevant to this process. However, Rayner and Pollatsek [RP89] have concluded from the results of many studies that both types of processing (holistic and analytic) are involved in visual word recognition. Besner and Johnston [BJ89] proposed a multiple-route model and they suggested that a lexical decision response can be achieved by three routes:

- using a visual familiarity assessment i.e via global word shape
- using an orthographic familiarity assessment based on overall lexical activation in the orthographic lexicon
- word identification on the basis of letter-level codes

There is another debate, whether words are read via one of the following routes: one, on visual basis where meaning is computed from orthographic patters or spelling and two, on phonological basis where a spelling leads to an internal phonological pattern followed by meaning. These two routes of reading have traditionally been termed direct route (orthography-to-meaning) and phonologically mediated lexical access or dual route (orthography-to-phonology-to-meaning). Marshall and Newcombe [MN73] were first to express these ideas in box-and-arrow model of reading and later on Baron [Bar77] provided a further explanation.

Despite of all these models and debates some researchers argued that visual objects (words) are recognized by spatial frequencies rather than collection of visual features for example, Gervais et al. [GHJR84] showed that letter confusions are better predicted by spatial frequency rather than visual features. Allen [AWW95] proposed the holistic model and suggested that words can be formed either via letter-level codes or via word-level codes in which âĂIJthe spatial frequency pattern of whole wordâĂİ is the basic unit of analysis. Recently, Allen [PAL$^+$09] proposed a multi-stream model having a lexicon for decision about word recognition based on spatial frequency information. According to this model different channels respond to different aspects of stimulus, i.e. words are recognized from lower frequency components and letters are recognized from higher spatial frequency components. The word recognition process tends to be based on holistic channel, however recognition can be based on information from analytical channels if stimulus information is unfamiliar.

How letter identities and letter positions are encoded during word recognition is an interesting question for existing computational models [DB06]. It is generally considered that the information about the abstract letter identities and letter positions both are necessary in visual word recognition process, otherwise people are unable to distinguish between anagrams like stop, spot, post, tops, pots, and opts. Different theories have been presented in past to describe how letters are encoded during visual word recognition and existing computational models adopted these theories with no or little modification to the base theories. Broadly we can categorize these theories into three major classes: slot-based coding, local-context coding and spatial coding [Gv03].

In slot-based coding scheme, word recognition models assume the channel specific or position specific coding. In this coding scheme letters are tagged to their positions within words and each letter is processed independently within its own channel or slot. For example, word "ACT" has 'A' at first, 'C' at second and 'T' at third position while in word "CAT" all these letters have different codings based on their positions. This coding scheme is mainly incorporated by interactive activation (IA) [MR81], multiple readout [GJ96], and the dual-route cascaded [CRP$^+$01] models. Relative position coding can be introduced by adding anchor points to the slot-based coding. Colthert et al. [CRP$^+$01] and Jacobs et al. [JRZG98] proposed two different relative position coding schemes and proposed to adopt one or more anchor points based on initial or final letters. For example by Jacobs et al. [JRZG98] the word string "ACT" can be encoded as I = 'A', I+1 = F-1 = 'C', and F = 'T', where "I" and "F" correspond to initial and final letters.

Local-context coding is mainly inspired from the work of Wickelgren [Wic69]. Wickelgren proposed the usage of "wickelphone" to encode the phoneme positions in speech. Seidenberg and McClelland [SM89a] adapted the concept in form of "wickelgraphs" or letter triples. According to this scheme word string "ACT" would be represented by "#AC", "ACT", and "CT#", where "#" represents the space or word boundary. This scheme is more consistent with the importance to letter order than the channel specific schemes. Mozer [Moz87] includes the nonadjacent letter to the local context as well and gives the concept of open-trigrams. According to his BLIRNET model the word “ACT” can be activated with trigrams “_CT”, “_AT”, “AC_” etc. and the underscore tells that any letter can be inserted at this position. Whitney [Whi01, WP08] presented an encoding based on open-bigrams that consists on ordered pair of letters. For example the input “take” is represented by activation of units representing “TA”, “TK”, “TE”, “AK”, “AE” and “KE”. The activated units do not contain the precise information about the position of each letter or which letter is next to which. The similar coding scheme is used by Grainger and van Heuven [Gv03] in which retinotopic letters are converted into an open-bigram encoding. The open-bigram activations are either 0 or 1 and are activated by letter pairs having at most two intervening letters. Dehaene et al. [DCSV05] presented a model that starts with a noisy retinotopic letter array and bi-gram models are built on top of letter activations. The final processing layer connects the bi-gram letter units to the word units. Grainger et al. [GGF$^+$06] presented the Overlap Open-Bigram model (OOB). The first two layers of the model are similar to the layers as proposed by Dehaene et al. [DCSV05] i.e. retinotopic letter representations with noise and bi-grams. The retinotopic bi-grams activate abstract bi-grams and these abstract bi-grams activate words at the next processing level. The model provide improvements in previous models by incorporating the location invariant encoding and bi-gram activation mechanism. The model also includes the activations of bi-grams corresponding to letter transpositions.

In spatial coding, that has been used in the SOLAR model [Dav99,DB06], all letter units are not dependent of position context. In this case the node that represent the letter “A” should be activated when the input string contains an “A”, irrespective of its serial position in which this letter occurs in the input word string. The relative order of the letters in the given input string is encoded by the relative activities of set of letters nodes. In this scenario different letter ordering results in different spatial patterns of activities for example the transposed letter words “ACT" and “CAT” share the same set of letter nodes but have different spatial patterns. In the overlap model presented by Gomez, Ratcliff and Perea [GRP08], the identities of the letters in a given word string are assumed to be normally distributed over position. For example in the word “trail”, the letter ‘a’ will be associated with position 3 but based on the standard deviation it is also associated with the positions 1, 2, 4 and 5 with lesser degree.

5.3 Experiments Overview

Various psychophysical experiments were conducted in which visual stimuli of words and permuted non-words were presented to native speakers of English, German and Urdu languages. The visual stimuli were common animal names which people usually learn during their childhood. The task of the subject was to categorize the visual stimuli as a bird or non-bird. The stimuli were presented for 5 seconds, a duration long enough to allow word recognition. Permuted non-words were generated by pseudo random transposition of internal letters of a word. The letter transposition was done by generating new position indexes for each internal letter using normally distributed random numbers. The permutation factor was controlled by a parameter *sigma*. A higher *sigma* value led to letter position indexes with greater distance from its original location as compared to a lower *sigma* value. In this work, *sigma* value of 5 was used for each experiment. Visual stimuli for selected words and permuted non-words were rendered with the help of python imaging library [PIL12] and Pango [Pan12] in selective fonts and font sizes. The presentation of visual stimuli and calculations of responses was done by integrating the routines from “PsychoPy” library [Pei08] into python code.

5.3.1 Experiments in English: Cursive and Times Fonts, Uppercase and Mixed-case

Method

Participants. Eighteen students at the University of California, Merced, volunteered to take part in the experiment for partial course credit in an undergraduate psychology course. All participants had normal or corrected-to- normal vision and no dyslexia. They were naive to the purpose and nature of the experiment, and gave informed consent in accord with the policies of the University of California, Merced, committee for the protection of human subjects, which approved the experimental protocol. Subjects were divided into two groups for two different font types “URW Chancery L” and “Times”. “URW Chancery L” is a cursive font and is selected to match the flow of Urdu font. Nine people with mean age of 20 years participated in “URW Chancery L” experiments and nine people with mean age of 22 years participated in “Times” font experiments.

Material. Twelve native speakers of English provided names of animals that they learned as a child and 98 most common names were selected for this study. The length of words

Table 5.1: Response times for English words and permuted non-words in mixed-case, "Times" font. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	Std	N	$E\%$	M	Std	N	$E\%$
	0.764	0.397	303	8.91	0.987	0.632	573	17.63
Syllables								
<2	0.784	0.444	204	6.86	0.912	0.608	134	12.69
>=2	0.723	0.273	99	13.13	1.010	0.638	439	19.13
Letters								
<6	0.772	0.4185	258	9.30	0.956	0.634	160	18.13
>=6	0.721	0.236	45	6.67	0.999	0.631	413	17.43

ranged from 3-11 letters. All the words were passed to the word permutation algorithm that randomly produces 64 permuted non-words for mixed-case data and 77 permuted non-words for uppercase data.

Procedure. Procedure similar to the Urdu language experiment was adopted for this experiment. Two sets of experiments were conducted in English language with two distinct groups of participants. In first group participants were tested for "Times" font and in second group for "Chancery" font. Visual stimuli were rendered in uppercase and mixed-case for both the experiments. Each participant was tested on both mixed-case and uppercase in two different rounds. The order of the case type was counterbalanced across the participants.

Data Analysis. Data analysis and plotting of graphs were done in a manner similar to Urdu. Additional ANOVAs were performed to see the effect of letter permutation in cursive and normal fonts for both letter cases.

Results

Tables 5.1, 5.2, 5.3 and 5.4 show the mean response time M, standard deviation Std, total number of samples N, and error percentage $E\%$ for normal words and permuted non-words for "Times" and "Chancery" fonts of English in both letter cases. Summaries of response times are plotted in Figures 5.2 and 5.3 with respect to number of syllables. The ANOVA shows comparison of "Times" font with F_{Times_mixed}, F_{Times_upper} and of "Chancery" font with $F_{Chancery\_mixed}$, $F_{Chancery\_upper}$.

Table 5.2: Response times for English words and permuted non-words in uppercase, "Times" font. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	*Std*	*N*	*E%*	*M*	*Std*	*N*	*E%*
	0.841	0.510	187	14.43	0.962	0.596	688	11.33
Syllables								
<2	0.815	0.485	152	9.21	0.849	0.456	188	7.97
>=2	0.955	0.602	35	37.14	1.004	0.637	500	12.6
Letters								
<5	0.836	0.493	151	15.23	0.846	0.419	81	8.64
>=5	0.862	0.583	36	11.11	0.978	0.615	607	11.69

Table 5.3: Response times for English words and permuted non-words in mixed-case, "Chancery" font. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	*Std*	*N*	*E%*	*M*	*Std*	*N*	*E%*
	Mean	Std	N	E%	Mean	Std	N	E%
	0.873	0.411	306	7.19	1.149	0.612	572	18.71
Syllables								
<2	0.830	0.367	207	6.76	1.072	0.548	135	12.59
>=2	0.961	0.479	99	8.08	1.173	0.628	437	20.59
Letters								
<6	0.866	0.425	261	8.43	1.088	0.560	161	18.01
>=6	0.910	0.315	45	0.0	1.1732	0.629	411	18.98

Table 5.4: Response times for English words and permuted non-words in uppercase, "Chancery" font. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	*Std*	*N*	*E%*	*M*	*Std*	*N*	*E%*
	0.868	0.410	188	12.76	1.147	0.660	691	17.65
Syllables								
<2	0.843	0.411	152	7.89	1.007	0.497	189	11.11
>=2	0.973	0.395	36	33.33	1.202	0.705	502	20.11
Letters								
<5	0.873	0.429	152	13.81	0.995	0.560	81	12.34
>=5	0.849	0.319	36	8.33	1.169	0.670	610	18.36

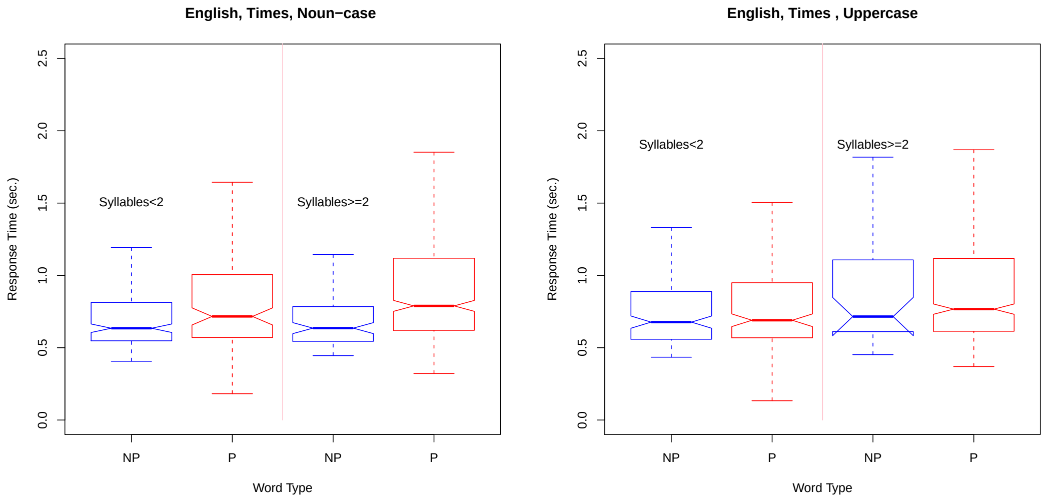


Figure 5.2: Boxplot of 'Response Time' and 'Word Type' with respect to number of syllables for English "Times" font, uppercase and mixed-case. NP = Non-permuted, P = Permuted.

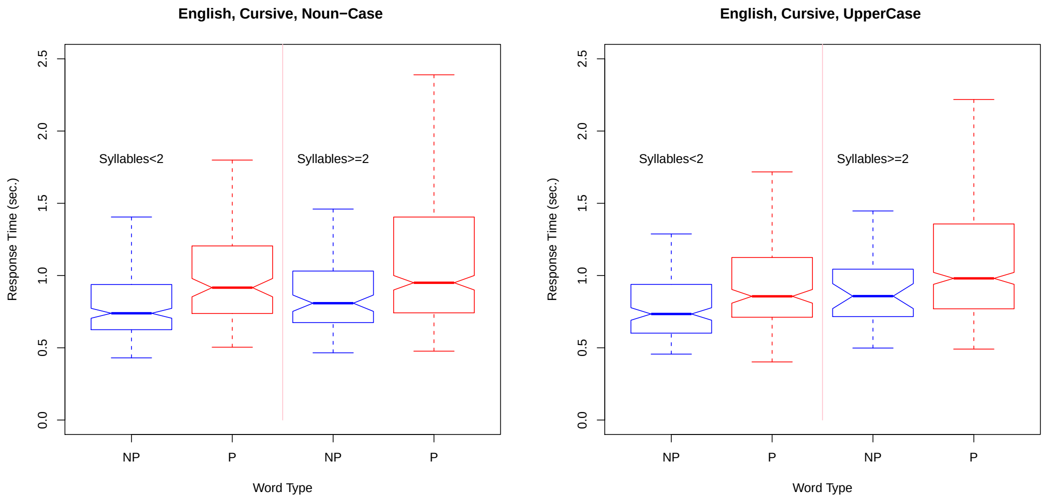


Figure 5.3: Boxplot of 'Response Time' and 'Word Type' with respect to number of syllables for English "Chancery" font, uppercase and mixed-case. NP = Non-permuted, P = Permuted.

Response Time Analysis. Significant differences were found in mean response latencies between normal words and permuted non-words in "Chancery" and "Times" fonts in mixed-case, $[F_{Chancery\_mixed}(1,8) = 79.86, p < 0.001]$, $[F_{Timess\_mixed}(1,8) = 19.4, p < 0.001]$. Significant difference were also found in uppercase scenario for "Chancery" font $[F_{Chancery\_upper}(1,8) = 81.56, p < 0.001]$ but this difference was less significant in case of "Times" font $[F_{Timess\_upper}(1,8) = 4.30, p < 0.1]$. The participants took longer to recognize permuted non-words than normal words in all the cases as shown by mean response time latencies in Tables 5.1, 5.2, 5.3 and 5.4. A significant effect of number of syllables (< 2 or ≥ 2) was found in "Times" uppercase for non-words. This is visible by comparing the mean response time of permuted non-words with number of syllables ≥ 2 ($M_{Times\_upper} = 1.173$), number of syllables < 2 ($M_{Times\_upper} = 0.961$), and the output of the ANOVA $[F_{Times\_upper}(1,686) = 9.34, p < 0.001]$ but there was less significant effect of word length (< 5 or ≥ 5) $[F_{Times\_upper}(1,686) = 3.50, p < 0.1]$. There was no effect of number of syllables or word length for normal words in "Times" uppercase. No main effect of syllables (< 2 or ≥ 2) $[F_{Times\_mixed\_non-words}(1,571) = 2.49, p > 0.5]$, $[F_{Times\_mixed\_words}(1,301) = 1.56, p > 0.5]$ and word length (< 6 or ≥ 6) $[F_{Times\_mixed\_non-words}(1,571) = 0.53, p < 0.5]$, $[F_{Times_{m}ixed_{w}ords}(1,301) = 0.63, p < 0.5]$ was found for permuted non-words and normal words in "Times" mixed-case. A significant effect of syllables (< 2 or ≥ 2) in "Chancery" uppercase for non-words was found. This is visible by comparing the mean response time of permuted non-words with number of syllables ≥ 2 ($M_{Chancery\_upper} = 1.202$), number of syllables < 2 ($M_{Chancery\_upper} = 1.007$) and the output of the ANOVA [$F_{Chancery\_upper}(1, 689) = 12.16, p < 0.001$]. A significant effect of word length ($<$ 5 or $\geq$ âĽě5) was also found in "Chancery" uppercase for non-words and it is visible by mean response time with word length ≥ 5 ($M_{Chancery\_upper} = 1.169$), word length < 5 ($M_{Chancery\_upper} = 0.995$), and the output of the ANOVA [$FChancery\_upper(1, 689) = 4.97, p < 0.01$]. No significant effect of syllables or word length was found in case of normal words for "Chancery" uppercase. No main effect of number of syllables (< 2 or ≥ 2) [$F_{Chancery\_mixed}(1, 570) = 2.81, p > 0.5$] and word length ($< 6$ or ≥ 6) [$F_{Chancery\_mixed}(1, 570) = 2.24, p > 0.5$] was found for permuted non-words in "Chancery" mixed-case. However, an effect of syllables [$F_{Chancery\_mixed}(1, 304) = 6.87, p < 0.01$] for normal words in "Chancery" mixed-case was found but, no main effect of word length [$F_{Chancery\_mixed}(1, 304) = 0.42, p > 0.5$] was found in this case.

Error Rate E%. Error rate shows that participants responded more accurately to normal words ($E_{Times\_mixed}\% = 8.91$, $E_{Chancery\_mixed}\% = 7.19$, $E_{Chancery\_upper}\% = 12.76$) than to permuted non-words ($E_{Times\_mixed}\% = 17.63$, $E_{Chancery\_mixed}\% = 18.71$, $E_{Chancery\_upper}\% = 17.65$) except in the case of "Times" font uppercase as shown in Tables 5.1, 5.2, 5.3 and 5.4. In recognition of permuted non-words, the overall recognition accuracy is decreased by 8% for "Chancery" and 3% for "Times" in both letter case types.

5.3.2 Experiments in German: Uppercase and Mixed-case

Method

Participants. Ten volunteer students of Technical University of Kaiserslautern participated in this experiment. They were naive to the purpose and nature of the experiment. All participants were male and native speaker of German with mean age of 24.3 years. All participants had normal or corrected-to-normal vision and no dyslexia.

Material. Three native speakers of German provided names of animals that they learned as a child and 90 most common names were selected for this study. The length of words ranged from 4-15 letters. All selected words were passed to the word permutation algorithm that randomly produces 46 permuted non-words.

Procedure. "Times" font was used to generate the visual stimuli for German words and non-words in uppercase and mixed-case letter case types. A procedure similar to the Urdu language experiment was employed. Each participant was tested on both mixed-case and

Table 5.5: Response times for German words and permuted non-words in uppercase. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	*Std*	*N*	*E%*	*M*	*Std*	*N*	*E%*
	1.041	0.497	440	1.36	1.684	1.256	460	10.21
Syllables								
<3	1.053	0.538	320	1.87	1.457	1.042	310	7.41
>=3	1.008	0.364	120	0	2.152	1.515	150	16
Letters								
<10	1.040	0.516	390	1.53	1.489	1.088	380	7.63
>=10	1.044	0.309	50	0	2.607	1.573	80	22.5

Table 5.6: Response times for German words and permuted non-words in mixed-case. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	*Std*	*N*	*E%*	*M*	*Std*	*N*	*E%*
	0.995	0.557	440	2.27	1.571	1.160	460	10.22
Syllables								
<3	0.974	0.539	320	1.87	1.395	0.960	310	7.41
>=3	1.053	0.602	120	3.33	1.935	1.426	150	16
Letters								
<10	0.971	0.514	390	2.05	1.387	0.939	380	7.10
>=10	1.181	0.803	50	4.0	2.447	1.626	80	25

uppercase in two different rounds. The order of the case type was counterbalanced across the participants.

Data Analysis. The procedure for data analysis and plotting of graphs were similar to Urdu except that word length < 10 or ≥ 10 was used for grouping the German language words. Additional ANOVAs were performed to see the effect of case type for both permuted non-words and normal words.

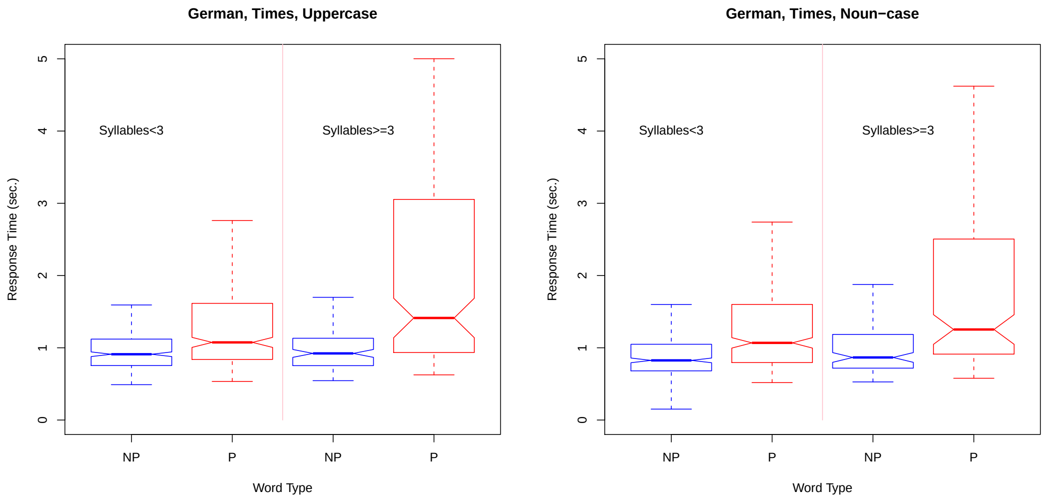


Figure 5.4: Boxplot of ‘Response Time’ and ‘Word Type’ with respect to number of syllables for German uppercase and mixed-case. NP = Non-permuted, P = Permuted.

Results

Tables 5.5 and 5.6 show the mean response time M, standard deviation Std, total number of samples N, and error percentage $E\%$ for normal words and permuted non-words in upper and mixed cases. Summaries of response times are plotted in Figure 5.4 with respect to number of syllables. The ANOVA shows comparison of uppercase with F_{upper} and of mixed-case with F_{mixed}.

Response Time Analysis. Significant differences were found in mean response time latencies for German permuted non-words and normal words for both case types, $[F_{upper}(1,9) = 59.51, p < 0]$, $[F_{mixed}(1,9) = 66.536, p < 0]$. The participants took longer to recognize permuted non-words than normal words as shown by mean response time latencies given in Tables 5.5 and 5.6. There was main effect of number of syllables (< 3 or ≥ 3) for permuted non-words in both upper and mixed cases. This is visible by comparing the mean response time of permuted non-words with number of syllables ≥ 3 ($M_{upper} = 2.152, M_{mixed} = 1.935$), number of syllables < 3 ($M_{upper} = 1.457, M_{mixed} = 1.395$) and the output of the ANOVA $[F_{upper}(1,458) = 33.03, p < 0]$, $[F_{mixed}(1,458) = 22.93, p < 0]$. Similarly, the permuted non-words with word length < 10 or ≥ 10 have the significant effect on response time latencies i.e. the mean response time of permuted non-words with word length ≥ 10 ($M_{upper} = 2.607, M_{mixed} = 2.447$), word length < 10 ($M_{upper} = 1.489, M_{mixed} = 1.387$) and the output of the ANOVA, $[F_{mixed}(1,458) = 62.55, p < 0]$.

The differences in response time for permuted non-words are also shown in Figure **??**. There was no main effect of number of syllables $[F_{upper}(1, 438) = 0.39]$, $[F_{mixed}(1, 438) = 0.18]$ and word length $[F_{upper}(1, 438) = 0.0021]$, $[F_{mixed}(1, 438) = 6.3772, p < 0.05]$ in case of normal words for both upper and mixed cases.

Error Rate E%. Error rate shows that participants responded more accurately to normal words ($E_{upper}\% = 1.36, E_{mixed}\% = 2.27$) than to permuted non-words ($E_{upper}\% = 10.21, E_{mixed}\% = 10.22$). In case of permuted non-words the error rate is more for words having syllables ≥ 3 ($E_{upper}\% = 16, E_{mixed}\% = 16$) as compared to words with syllables < 3 ($E_{upper}\% = 7.41, E_{mixed}\% = 7.41$). Similarly, recognition error is greater for words having word length ≥ 10 ($E_{upper}\% = 22.5, E_{mixed}\% = 25$) than words with word length < 10 ($E_{upper}\% = 7.63, E_{mixed}\% = 7.10$). The overall recognition accuracy is decreased by 11% in recognition of permuted non-words for both cases.

5.3.3 Experiment in Urdu

Urdu [Wik12b] is orthographically entirely different from Latin script. It is mostly written in Nastaleeq [Wik12a] style and has its root in Arabic and Persian languages.

Method

Participants. Ten students of Technical University of Kaiserslautern volunteered to participate in this experiment. They were naive to the purpose and nature of the experiment. All participants were male and native speakers of Urdu with mean age of 25 years. All participants had normal or corrected-to-normal vision and no dyslexia [Dys10].

Material. Three native speakers of Urdu provided names of animals that they had learned as a child and 75 most common names were selected for this study. The length of the words ranged from 3-12 letters. All the selected words were passed to the word permutation algorithm that randomly produces 43 permuted non-words. Visual stimuli of the words and permuted non-words were generated using Nafees Nastaleeq [Cru09] font in 36 points font size.

Procedure. Each participant was tested by himself in a quiet room. Before starting the experiment, brief instructions were presented and verbally described to the participant. Participants started the experiment by pressing the key “S” on a standard computer keyboard. Visual stimuli consisting of words and permuted non-words were randomly presented at the center of computer screen with black background and white foreground.

Table 5.7: Response times for Urdu words and permuted non-words. M = mean response time, Std = standard deviation, N = total number of samples, and $E\%$ = error percentage

	Words				**Permuted non-words**			
	M	*Std*	*N*	*E%*	*M*	*Std*	*N*	*E%*
	1.479	0.889	316	6.65	2.231	1.311	429	20.75
Syllables								
<3	1.480	0.895	306	6.86	2.134	1.254	349	19.19
>=3	1.436	0.713	10	0	2.652	1.470	80	27.5
Letters								
<5	1.526	0.929	256	7.81	1.948	1.099	79	25.31
>=5	1.278	0.662	60	1.66	2.294	1.347	350	19.71

Participants responded to the stimulus by pressing the key “B” for birds or key “N” for non-birds. After each stimulus presentation a wait screen was displayed and the next stimulus was presented when participant pressed any key from the keyboard. The purpose of the wait screen was to provide a brief pause during the experiment and wait time was not added to the response time of the user. Each stimulus was presented for a maximum of 5 seconds. If the participant failed to respond within this time period then the response was counted as “NULL” response. The response and the response time were recorded for each stimulus.

Data Analysis. Data analysis and plotting of graphs were done using the R [Hor11] software package. Null responses (responses for which participants were unable to respond within 5 seconds) were considered as wrong responses. The trials in which response time was less than 100 msec and/or response key was other than ‘B’ or ‘N’ were excluded from analysis. This resulted in exclusion of less than 1% of total response data. Primary analysis was to determine the recognition accuracy for words and permuted non-words and then to analyze the response latencies for each kind of stimulus. Analysis was performed on the whole set of data making two major groups (words, permuted non-words)l; data was further grouped based on number of syllables and word length. Analysis of variance (ANOVA) was performed to determine the significant differences in response time latencies for two classes of data. There were three factors in ANOVA: word type (permuted or non-permuted), number of syllables and word length for Urdu language.

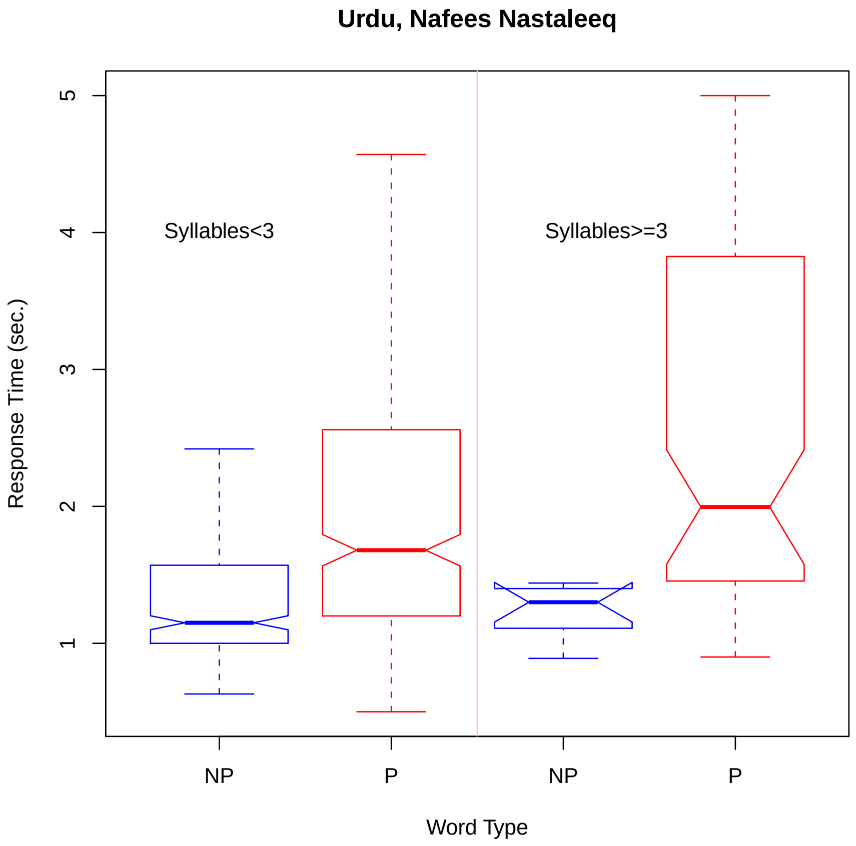


Figure 5.5: Boxplot of 'Response Time' and 'Word Type' with respect to number of syllables for Urdu. NP = Non-permuted, P = Permuted

Results

Table 5.7 shows the mean response time *M*, standard deviation *Std*, total number of samples *N*, and error percentage *E%* for permuted and non-permuted words of Urdu language. For Urdu language, the words were divided into groups based on number of syllables (< 3 or ≥ 3) and word length (< 5 or ≥ 5).

Response Time Analysis. It can be observed from the data that the participants required more time to recognize permuted non-words than to recognize the normal words. This difference is clearly visible by response time mean for permuted non-words ($M = 2.231$) and response time mean for normal words ($M = 1.479$). A significant difference was found in within subjects ANOVA of response time with respect to permuted non-words and normal words [$F(1, 9) = 79.66, p < 0$]. A main effect of the number of syllables and word length was also found in recognition of permuted non-words. This is visible by comparing the mean response time of permuted non-words with number of syllables < 3 ($M = 2.134$), mean response time of permuted non-words with number of syllables ≥ 3 ($M = 2.652$) and the output of the ANOVA [$F(1, 427) = 10.38, p < 0.001$]. The difference in mean response time for permuted non-words having word length < 5 ($M = 1.948$) and ≥ 5 ($M = 2.294$) [$F(1, 427) = 4.53, p < 0.01$] also showed a similar pattern. This difference is also shown in Figure 5.5. Any significant effect of numbers of syllables [$F(1, 314) = 0.024$] and word length [$F(1, 314) = 3.84, p < 0.05$] was found in case of normal words.

Error Rate E%. Error rate showed that participants responded more accurately to normal words ($E\% = 6.65$) than to permuted non-words ($E\% = 20.75$). In case of permuted non-words the error rate increased for words having ≥ 3 syllables ($E\% = 27.5$) as compared to words with < 3 syllables ($E\% = 19.19$). The overall recognition accuracy for permuted non-words was lower by 31%.

5.4 Discussion

In this study the reading behavior, the effect of number of syllables and the effect of word length in visual recognition of words and permuted non-words were analyzed. The difference in mean response time latencies of permuted non-words and normal words reflects a difference in reading behavior of people for these two cases. This difference supports the presented hypothesis that there are two distinct processes involved in the cognitive processing of permuted non-words and normal words. It can be concluded that people use direct route i.e. spellings to meaning in recognition of non-permuted words showed by fast response time and they use the indirect phonological mediated route (spelling to sound to meaning) in recognition of permuted words. The reason is that when there is no direct visual pattern to map to mental lexicon then phonology helps in recognition of this orthographically new pattern by mapping it to some already familiar phonologically correct word and then mapping it to the mental lexicon. The extra processing involved in using phonological route to map the visual input is supported by the delayed responses of participants for permuted words. Recognition accuracy and mean response time latencies were also significantly affected by the number of syllables and word length for permuted words but this effect was not found in case of normal words. Error rate was significantly higher for permuted non-words in comparison with normal words. These observations reflect the significance of letter positions within a word and the extra processing involved in recognizing permuted non-words. This suggests that individual letter positions have significant impact on reading and any change of letter position within a word makes it difficult to read and understand, resulting in a change of cognitive strategy to read permuted non-words.

It is observed that people showed similar behavior for orthographically different languages i.e. people were slower and made more errors in visual word recognition of permuted non-words in comparison to normal words for three different languages used in this study. This observation points towards the similarity of cognitive processes when reading different languages. It is also observed that reading of Urdu is comparatively slower than reading of German and English, in both permuted and non-permuted forms. This difference in reading may be due to the visual characteristics of the Urdu script. These characteristics suggest that there can be a possibility of division of labor [HS04] among two working components of brain (orthography and phonology) and in case of Urdu, orthography to phonology to meaning components may be more activated as compared to orthography to meaning components, resulting in delay of visual word recognition even for normal words. Another observation is that the error rate in recognition of Urdu permuted words is significantly increased with the increase in number of syllables or word length. This is because greater number of syllabic units or number of letters in a word caused more shape degradation in permuted form. This shape degradation resulted in difficulty of reading permuted non-words that have more number of letters or syllables. The effect of global word shape is also observed in reading of mixed-case and uppercase words for both German and English. The results obtained in this study are also in accordance with the presence of Visual Word Form Area (VWFA) [CDN+00] in the rear left-hemisphere occipital lobe to recognize familiar words and phonological processing besides the visual word recognition of words and anagrams [KPM+04].

Chapter 6

Conclusions

This thesis presented a number of contributions in the field of document image analysis, optical character recognition (OCR) and cognitive reading research. The main objective of this thesis is to develop an OCR system for the recognition of degraded text with varying fonts, font sizes and font styles. This is achieved by building a novel segmentation free OCR methodology using a combination of hidden Markov models (HMMs) and artificial neural networks (ANNs). In addition, the thesis presented novel applications of convolutional neural networks (CNNs) and HMMs for document image preprocessing and OCR of low resolution text. Furthermore, the thesis introduced novel psychophysical experiments to determine the effect of letter permutation in visual word recognition of Cursive and Latin script languages.

First, the thesis presented novel applications HMMs-based segmentation free OCR approach for the recognition of low resolution screen rendered text. The approach achieved character recognition accuracies that are significantly better than the performance of existing OCR systems and reached near to the performance of specialized low resolution character recognition method [WKJ06]. The approach yields above 98% recognition accuracy on screen rendered text lines using two types of features. This is 23% more in comparison with the performance of Tesseract OCR engine [Smi13]. Then, the thesis presented a discriminative learning approach for document image analysis using CNNs. The evaluation of the approach on different image processing applications demonstrated the discriminative learning capabilities of ANNs. The proposed framework is used in application to script recognition and orientation detection of various types of document images. The evaluation of the approach resulted in above 95% script recognition accuracy at connected component level for ancient multi-script documents and 100% page orientation detection accuracy for Urdu document of variable page layouts.

Based on the performance of ANNs and HMMs in different image analysis and recognition applications, a combined ANN/HMM OCR approach is introduced for the recognition of degraded text. The approach used a text line scanning mechanism in which a specially trained ANN is employed for the extraction of discriminative features from the input text line. The output of the ANN was decoded by HMMs to provide segmentation free recognition of complete text line image. The system achieved significantly better recognition accuracy at character level and resulted in a 30% reduction in error rate as compared to GoogleâĂŹs Tesseract OCR system [Smi13] and 43% reduction in error as compared to OCRopus OCR system [Bre08], which are the best open source OCR systems available today.

Further, the thesis presented a hypothesis about visual recognition of words and permuted non-words inspired by a circulated Internet meme [Dav09]. The hypothesis was tested by conducting psychophysical experiments in Latin and Cursive languages with an objective to determine the effect of letter permutation and shape distortion during visual word recognition process. Experimental results supported the hypothesis that recognition of words and permuted non-words there are two distinct mental processes and people use different strategies in handling permuted words as compared to normal words. The results also concluded the importance of letter unit identities and their positions within the word during reading, and deduced that change in letter positions within word (letter permutation) requires more information to be utilized in visual word recognition process.

Bibliography

[ABB11] ABBYY. ABBYY FineReader OCR software. http://finereader.abbyy.com/, 2011. [Online; accessed 01-June-2011].

[AG97] Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.

[AH90] T. Akiyama and N. Hagita. Automated entry system for printed documents. *Pattern Recognition*, 23(11):1141–1154, 1990.

[AM09] S. Abirami and D. Manjula. A survey of script identification techniques for multi-script document images. *International Journal of Recent Trends in Engineering*, 1(2):255–257, May 2009.

[And96] S. Andrews. Lexical retrieval and selection processes: Effects of transposed-letter confusability. *Journal of Memory and Language*, 35:775–800, 1996.

[Ara05] H. B. Aradhye. A generic method for determining the up/down orientation of text in roman and non-roman scripts. *Pattern Recognition*, 38(11):2114–2131, 2005.

[AWW95] P. A. Allen, B. Wallace, and T. A. Weber. Influence of case type, word frequency, and exposure duration on visual word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 21(4):914–934, Aug 1995.

[Bab97] Babylon. Babylon translation software. http://www.babylon.com/, 1997. [Online; accessed 01-June-2011].

[Bar77] J. Baron. *Basic processes in reading: perception and comprehension*, chapter Mechanisms for pronouncing printed words: use and acquisition. Hillsdale, NJ: Erlbaum, 1977.

[BBS05] A. Busch, W. W. Boles, and S. Sridharan. Texture for script identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1720–1732, November 2005.

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Inc., New York, NY, USA, 1995.

[BJ89] Derck Besner and James C. Johnston. *Reading and the mental lexicon: on the uptake of visual information.* MIT Press, Cambridge, MA, USA, 1989.

[BKW$^+$99] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, and G. Rigoll. Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data. In *Proc. of Fifth International Conference on Document Analysis and Recognition*, pages 446–449, September 1999.

[BLNB95] Yoshua Bengio, Yann Lecun, Craig Nohl, and Chris Burges. LeRec: A NN/HMM hybrid for on-line handwriting recognition. *Neural Computation*, 7(6):1289–1303, November 1995.

[BM94] Herve A Bourlard and Nelson Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer, 1994.

[BNS$^+$99] Issam Bazzi, Premkumar Natarajan, Richard Schwartz, Andras Kornai, Zhidong Lu, and John Makhoul. Ocr of degraded documents using hmm-based techniques. In *Proceedings Symposium on Document Image Understanding Technology*, page 149, 1999.

[Bou95] N. G. Bourbakis. Handwriting recognition using a reduced character method and neural nets. In *Proc. of SPIE Nonlinear Image Processing VI*, volume 2424, pages 592–601, February 1995.

[BPSW70] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of

Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[Bre01a] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Bre01b] Thomas M. Breuel. Segmentation of handprinted letter strings using a dynamic programming algorithm. In *Proc. of Sixth International Conference on Document Analysis and Recognition*, pages 821–826, September 2001.

[Bre08] Thomas M. Breuel. The OCRopus open source OCR system. In *Proc. of SPIE Document Recognition and Retrieval XV*, volume 6815, page 68150F, January 2008.

[BS10] Thomas Breuel and Faisal Shafait. AutoMLP: Simple, Effective, Fully Automated Learning Rate and Size Adjustment. In *The Learning Workshop*, April 2010. Extended Abstract.

[BSB09] J. Beusekom, F. Shafait, and T. M. Breuel. Resolution independent skew and orientation detection for document images. In *Proceedings of SPIE Document Recognition and Retrieval XVI*, Jan. 2009.

[BSB10] Joost Van Beusekom, Faisal Shafait, and Thomas M Breuel. Combined orientation and skew detection using geometric text-line modeling. *International Journal on Document Analysis and Recognition*, 13(2):79–92, June 2010.

[BSM99] Issam Bazzi, Richard Schwartz, and John Makhoul. An omnifont open-vocabulary ocr system for english and arabic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):495–504, 1999.

[Cat85] J. M. Cattell. The inertia of the eye and brain. *Brain*, 8:295–312, 1885.

[Cat86] J. M. Cattell. The time it takes to see and name objects. *Mind*, 11:63–65, 1886.

[CDK$^+$87] Y. Chow, M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz. Byblos: The bbn continuous speech recognition system. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, volume 12, pages 89–92. IEEE, 1987.

[CDN+00] L. Cohen, S. Dehaene, L. Naccache, Stephane Lehericy, Ghislaine Dehaene-Lambertz, MarieAnne Henaff, and Francois Michel. The visual word form area: Spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients. *Brain*, 123(2):291–307, 2000.

[CG04] Rhandley D Cajote and Rowena Cristina L Guevara. Global word shape processing using polar-radii graphs for offline handwriting recognition. In *IEEE Region 10 Conference TENCON 2004*, pages 315–318. IEEE, 2004.

[Cha79] S. M. Chambers. Letter and order information in lexical access. *Journal of Verbal Learning and Verbal Behavior*, 18:225–241, 1979.

[CL96] Richard G Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):690–706, 1996.

[Cle65] Jon Kaufmann Clemens. *Optical character recognition for reading machine applications*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering, August 1965.

[CRP+01] M. Coltheart, K. Rastle, C. Perry, R. Langdon, and J. Ziegler. DRC: A Dual Route Cascaded model of visual word recognition and reading aloud. *Psychological Review*, 108(1):204–256, Jan 2001.

[Cru09] Crulp. Nafees Nastaleeq v1.02 beta. `http://www.crulp.org/software/localization/Fonts;http://nafeesnastaleeq.html`, 2009. [Online; accessed 18-April-2012].

[Dav99] C. J. Davis. The self-organising lexical acquisition and recognition (SOLAR) model of visual word recognition. Unpublished doctoral dissertation, 1999.

[Dav09] Matt Davis. MRC Cognition and Brain Sciences Unit. `http://www.mrc-cbu.cam.ac.uk/people/matt.davis/Cmabrigde`, 2009. [Online; accessed 18-April-2012].

[DB04] C. J. Davis and J. S. Bowers. What do letter migration errors reveal about letter position coding in visual word recognition? *Journal of*

Experimental Psychology: Human Perception and Performance, 30:923–941, 2004.

[DB06] C. J. Davis and J. S. Bowers. Contrasting Five Different Theories of Letter Position Coding: Evidence From Orthographic Similarity Effects. *Trends in Cognitive Science*, 32:335–341, 2006.

[DCSV05] Stanislas Dehaene, Laurent Cohen, Mariano Sigman, and Fabien Vinckier. The neural code for written words: a proposal. *Trends in Cognitive Sciences*, 9(7):335 – 341, 2005.

[DH10] Nadir Durrani and Sarmad Hussain. Urdu word segmentation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 528–536. Association for Computational Linguistics, 2010.

[DHS00] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2000.

[DMMH06] B. V. Dhandra, V. S. Malemath, H. Mallikarjun, and R. Hegadi. Skew detection in binary image documents based on image dilation and region labeling approach. In *18th International Conference on Pattern Recognition*, volume 2, pages 954–957, 2006.

[DNP$^+$05] Michael Decerbo, Premkumar Natarajan, Rohit Prasad, Ehry MacRostie, and Arun Ravindran. Performance improvements to the bbn byblos ocr system. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 411–415. IEEE, 2005.

[DS14] Ke-Lin Du and MNS Swamy. Multilayer perceptrons: Architecture and error backpropagation. In *Neural Networks and Statistical Learning*, pages 83–126. Springer, 2014.

[DTJT96] Øivind Due Trier, Anil K Jain, and Torfinn Taxt. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[Dys10] Dyslexia. National Institute of Neurological Disorders and Stroke. `http://www.ninds.nih.gov/disorders/dyslexia/dyslexia.htm`,

2010. [Online; accessed 01-June-2012].

[ECBGMZM11] S. España-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):767–779, April 2011.

[EIH07] Farshideh Einsele, Rolf Ingold, and Jean Hennebert. A HMM-based approach to recognize ultra low resolution anti-aliased words. In *Proceedings of the 2nd international conference on Pattern recognition and machine intelligence*, pages 511–518, 2007.

[EIH08] Farshideh Einsele, Rolf Ingold, and Jean Hennebert. A language-independent, open-vocabulary system based on HMMs for recognition of ultra low resolution words. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 429–433, 2008.

[Fin08] G.A. Fink. *Markov Models for Pattern Recognition.* Springer-Verlag Berlin Heidelberg, 2008.

[Fis75] D. F. Fisher. Reading and visual search. *Memory and Cognition*, 3:188–196, 1975.

[For73] G. D. Forney. The Viterbi algorithm. *Proc. of the IEEE*, 61:268–278, March 1973.

[For76] K. I. Forster. *New approaches to language mechanisms*, chapter Accessing the mental lexicon, pages 257–287. Amsterdam: North-Holland, 1976.

[Fuj08] Hiromichi Fujisawa. Forty years of research in character and document recognitionâĂŤan industrial perspective. *Pattern Recognition*, 41(8):2435–2446, 2008.

[Fuk88] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

[FWL02] K. C. Fan, Y. K. Wang, and T. R. Lay. Marginal noise removal of document images. *Pattern Recognition*, 35(11):2593–2611, 2002.

[GDS10] Debashis Ghosh, Tulika Dube, and Adamane P Shivaprasad. Script recognitionâĂŤa review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2142–2161, 2010.

[GFGS06] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proc. of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[GGF$^+$06] J. Grainger, J. P. Grainer, F. Farioli, Eva Van Assche, and W. J. B. Van Heuven. Letter position information and printed word perception: The relative-position priming constraint. *Journal of experimental psychology. Human perception and performance*, 32(4):865–884, 2006.

[GHHP97] I. Guyon, R.M. Haralick, J.J. Hull, and I.T. Phillips. *Data sets for OCR and document image understanding research*, pages 779–799. World ScientiïňĄc Singapore, 1997.

[GHJR84] M. J. Gervais, L. O. Harvey, Jr., and J. O. Roberts. Identification confusions among letters of the alphabet. *Journal of Experimental Psychology: Human Perception & performance*, 10:655–666, 1984.

[GJ96] J. Grainger and A. M. Jacobs. Orthographic processing in visual word recognition: A multiple read-out model. *Psychological Review*, 103:518–565, 1996.

[GLF$^+$08] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, May 2008.

[Gou72] P. B. Gough. *Language by ear and by eye*, chapter One second of reading. Cambridge, MA: MIT Press, 1972.

[GRP08] Pablo Gomez, Roger Ratcliff, and Manuel Perea. The Overlap Model: A Model of Letter Position Coding. *Psychological Review*, 115:577–600, 2008.

[Gv03] J. Grainger and W. J. B. van Heuven. *Modeling letter position coding in printed word perception*, chapter In The Mental Lexicon, pages 1–23.

Nova Science New York, 2003.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., 2006.

[GY07] Mark Gales and Steve Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, Jan 2007.

[Har71] Michael Hart. Free ebooks - Project Gutenberg. `http://www.gutenberg.org/`, 1971. [Online; accessed 01-June-2011].

[Hay07] S. S. Haykin. *Neural networks: a comprehensive foundation.* NJ: Prentice Hall, Englewood Cliffs, New Jersey, USA, 2007.

[HBT96] Jianying Hu, Michael K. Brown, and William Turin. HMM Based On-Line Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):1039–1045, October 1996.

[HKTK97] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns. Automatic script identification from document images using cluster-based templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2), February 1997.

[Hor11] Kurt Hornik. The R FAQ. `http://cran.r-project.org/doc/FAQ/R-FAQ.html`, 2011. ISBN 3-900051-08-9.

[HP04] O. Hauk and F. Pulvermuller. Effects of word length and frequency on the human ERP. *Neuropsychologia*, 115:1090–1103, 2004.

[HS99] M. W. Harm and M. S. Seidenberg. Phonology, reading acquisition, and dyslexia: Insights from connectionist models. *Psychological Review*, 106(3):491–528, 1999.

[HS04] M. W. Harm and M. S. Seidenberg. Computing the meanings of words in reading: cooperative division of labor between visual and phonological processes. *Psychological Review*, 111(3):662–720, 2004.

[HW62] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[HW10] Hans-Werner34. This picture shows the acuity of foveal vision in reading (during one eye stop). http://en.wikipedia.org/wiki/File:EyeFixationsReading.gif, 2010. [Online; accessed 08-October-2012].

[JBT96] Jianying Hu, M. K. Brown, and W. Turin. HMM based online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, October 1996.

[Jel97] Frederick Jelinek. *Statistical methods for speech recognition.* MIT Press, Cambridge, MA, USA, 1997.

[JGS06] G. D. Joshi, S. Garg, and J. Sivaswamy. *Script Identification from Indian Documents*, pages 255–267. Springer Berlin/Heidelberg, 2006.

[JMRW01] Stefan Jaeger, Stefan Manke, Jürgen Reichert, and Alex Waibel. Online handwriting recognition: the npen++ recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–180, 2001.

[JRP07] Rebecca L. Johnson, Keith Rayner, and Manuel Perea. Transposed-letter effects in reading: Evidence from eye movements and parafoveal preview. *Journal of Experimental Psychology: Human Perception and Performance*, 33:209–229, 2007.

[JRZG98] A. M. Jacobs, A. Rey, J. C. Ziegler, and J. Grainger. *MROM-p: An Interactive activation, multiple read-out model of orthographic and phonological processes in visual word recognition*, chapter Localist connectionist approaches to human cognition, pages 147–188. Lawrence Erlbaum Associates Inc Mahwah NJ, 1998.

[JSVR05] Charles Jacobs, Patrice Y. Simard, Paul Viola, and James Rinker. Text Recognition of Low-resolution Document Images. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 695–699, 2005.

[KA98] S. Knerr and E. Augustin. A neural network-hidden Markov model hybrid for cursive word recognition. In *Proc. of Fourteenth International Conference on Pattern Recognition*, volume 2, pages 1518–1520, August 1998.

[KBK$^+$13] Rudolf Kruse, Christian Borgelt, Frank Klawonn, Christian Moewes, Matthias Steinbrecher, and Pascal Held. Radial basis function networks. In *Computational Intelligence*, pages 83–103. Springer, 2013.

[KCGM93] A. Kaltenmeier, T. Caesar, J.M. Gloger, and E. Mandler. Sophisticated topology of hidden markov models for cursive script recognition. In *Proceedings of the Second International Conference on Document Analysis and Recognition, 1993.*, pages 139–142, 1993.

[Kho07] M. S. Khorsheed. Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK). *Pattern Recognition Letters*, 28(12):1563–1571, September 2007.

[KKS00] Jin Ho Kim, Kye Kyung Kim, and Ching Y. Suen. An HMM-MLP Hybrid Model for Cursive Script Recognition. *Pattern Analysis & Applications*, 3(4):314–324, 2000.

[Koh01] Teuvo Kohonen. *Self-organizing maps*, volume 30. Springer, 2001.

[KPM$^+$04] Pammer Kristen, C Hansen Peter, L Morten, Kringelbach, Ian Holliday, Gareth Barnes, Arjan Hillebrand, D Singh Krish, and L Cornelissen Piers. Visual word recognition: the first half second. *NeuroImage*, 22:1819–1825, 2004.

[LB06] M. Liwicki and H. Bunke. HMM-based on-line recognition of handwritten whiteboard notes. In *Proceedings 10th International Workshop Frontiers in Handwriting Recognition*, pages 595–599, 2006.

[LBBH98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), November 1998.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[LLP96] Seong-Whan Lee, Dong-June Lee, and Hee-Seon Park. A New Methodology for Gray-Scale Character Segmentation and Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):1045–1050, October 1996.

[LSN$^+$99] Zhidong Lu, Richard Schwartz, Premkumar Natarajan, Issam Bazzi, and John Makhoul. Advances in the bbn byblos system. In *Document*

Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on, pages 337–340. IEEE, 1999.

[LT06] S. Lu and C. L. Tan. Automatic document orientation detection and categorization through document vectorization. In *Proceedings of the 14th annual ACM International Conference on Multimedia*, pages 113–116, 2006.

[LTW94] D. X. Le, G. Thoma, and H. Weschler. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, 27(10):1325–1344, 1994.

[MAGD01] Sanparith Marukatat, Thierry Artières, R Gallinari, and Bernadette Dorizzi. Sentence recognition through hybrid neuro-markovian modeling. In *Proceedings of Sixth International Conference on Document Analysis and Recognition, 2001.*, pages 731–735. IEEE, 2001.

[MB01] Urs-Viktor Marti and Horst Bunke. Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):65–90, February 2001.

[MEMY98] Côté M., Lecolinet E., Cheriet Mohamed, and Suen Ching Y. Automatic reading of cursive scripts using a reading model and perceptual concepts The PERPECTO system. *IJDAR*, 1(1):3–17, 1998.

[MGS05] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):23–35, 2005.

[Mir10] Mahinnaz Mirdehghan. Persian, urdu, and pashto: A comparative orthographic analysis. *Writing Systems Research*, 2(1):9–23, 2010.

[MN73] J. C. Marshall and F. Newcombe. Patterns of paralexia: a psycholinguistic approach. *Journal of Psycholinguistic Research*, 2:175–199, 1973.

[MNDP04] Ehry MacRostie, Premkumar Natarajan, Michael Decerbo, and Rohit Prasad. The bbn byblos japanese ocr system. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 650–653. IEEE, 2004.

[Moz87] M. Mozer. *Early parallel processing in reading: A connectionist approach*, chapter Attention and Performance XII: The psychology of reading, pages 83–104. Lawrence Erlbaum Associates Ltd Hove, UK, 1987.

[Moz91] Michael C. Mozer. *The perception of multiple objects: a connectionist approach*. MIT Press, Cambridge, MA, USA, 1991.

[MR75] G.W. McConkie and K. Rayner. The span of the effective stimulus during a fixation in reading. *Perception and Psychophysics*, 17:578–586, 1975.

[MR81] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88:375–407, 1981.

[MSY92] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.

[Neb98] C. Nebauer. Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 9(4):685–696, Jul 1998.

[NFPB06] B. New, L. Ferrand, C. Pallier, and M. Brysbaert. Re-examining word length effects in visual word recognition: New evidence from the English Lexicon Project. *Psychonomic Bulletin and Review*, 13:45–52, 2006.

[NLS$^+$01] Premkumar Natarajan, Zhidong Lu, Richard M. Schwartz, Issam Bazzi, and John Makhoul. Multilingual Machine Printed OCR. *International Journal Pattern Recognition and Artificial Intelligence*, 15(1):43–63, 2001.

[NMD05] Premkumar S. Natarajan, Ehry MacRostie, and Michael Decerbo. The bbn byblos hindi ocr system. In *Electronic Imaging 2005*, pages 10–16, 2005.

[NSDK03] Prem Natarajan, R Schwartz, M Decerbo, and T Keller. Porting the bbn byblos ocr system to new languages. In *Symposium on Document Image Understanding Technologies*, pages 47–52, 2003.

[OCR08] OCRopus. The OCRopus(tm) open source document analysis and OCR system. `http://code.google.com/p/ocropus`, 2008. [Online; accessed 01-April-2012].

[Ots79] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[PAL$^+$09] A. Allen Philip, F. Smith Albert, Mei-Ching Lien, P. Kaut Kevin, and Angie Canfield. A multistream model of visual word recognition. *Attention, Perception and Psychophysics*, 71:281–296, 2009.

[Pan12] Pango. Pango information page. `http://www.pango.org`, 2012. [Online; accessed 18-April-2012].

[PC97] U. Pal and B.B. Chaudhuri. Automatic separation of words in multilingual multi-script indian documents. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1997.*, volume 2, pages 576–579, 1997.

[PC99] U. Pal and B.B. Chaudhuri. Script line separation from indian multi-script documents. In *International Conference on Document Analysis and Recognition*, pages 406–409, 1999.

[PC01] U. Pal and B.B. Chaudhuri. Automatic identification of english, chinese, arabic, devnagari and bangla script line. In *International Conference on Document Analysis and Recognition*, pages 790–794, 2001.

[PCH93] Ihsin T Phillips, Su Chen, and Robert M Haralick. Cd-rom document database standard. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 478–483, 1993.

[Pei08] Jonathan W. Peirce. Generating stimuli for neuroscience using PsychoPy. *Frontiers in Neuroinformatics*, 2, 2008.

[PF09] Thomas Plötz and Gernot A. Fink. Markov models for offline handwriting recognition: a survey. *Int. Jour. Doc. Anal. Recognit.*, 12(4):269–298, November 2009.

[PIL12] PIL. Python Imaging Library. `http://www.pythonware.com/products/pil`, 2012. [Online; accessed 18-April-2012].

[PL03a] M. Perea and S. J. Lupker. Does jugde activate COURT? Transposed-letter similarity effects in masked associative priming. *Memory and Cognition*, 31:829–841, 2003.

[PL03b] M. Perea and S. J. Lupker. *Transposed-letter confusability effects in masked form priming*, chapter Masked priming: State of the art, pages 97–120. Psychology Press Hove, UK, 2003.

[PL04] M. Perea and S. J. Lupker. Can CANISO activate CASINO? Transposed-letter similarity effects with nonadjacent letter positions. *Journal of Memory and Language*, 51:231–246, 2004.

[Pla05] David C. Plaut. *The Science of Reading A Handbook*, chapter Connectionist approaches to reading. Blackwell Publishing, 2005.

[PMSP96] D. C. Plaut, J. L. McClelland, M. Seidenberg, and K. E. Patterson. Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, 103(1):56–115, 1996.

[PNMS82] K. R. Paap, S. L. Newsome, J. E. McDonald, and R. W. Schvaneveldt. An activation-verification model for letter and word recognition: The word superiority effect. *Psychological Review*, 89:573–594, 1982.

[PR08] Peeta Basa Pati and A.G. Ramakrishnan. Word level multi-script identification. *Pattern Recognition Letters*, 29(9):1218 – 1229, 2008.

[Rab89] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[Rab90] Lawrence R. Rabiner. Readings in speech recognition. chapter A tutorial on Hidden Markov Models and selected applications in speech recognition, pages 267–296. 1990.

[Raw07] Graham Rawlinson. The significance of letter position in word recognition. *Aerospace and Electronic Systems Magazine, IEEE*, 22(1):26–27, 2007.

[Ray98] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372–422, November 1998.

[RBSB09] S.F. Rashid, S.S. Bukhari, F. Shafait, and T.M. Breuel. A discriminative learning approach for orientation detection of urdu document images. In *13th IEEE Int. Multi-topic Conference, INMICâĂŹ09*, Islamabad, Pakistan, Dec 2009. IEEE.

[Rig02] Gerhard Rigoll. Combination of hidden markov models and neural networks for hybrid statistical pattern recognition. *SERIES IN MACHINE PERCEPTION AND ARTIFICIAL INTELLIGENCE*, 47:113–144, 2002.

[RJ86] Lawrence Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[RJP07] K. Rayner, B. J. Juhasz, and A. Pollatsek. *The Science of Reading: A Handbook*, chapter Eye Movements During Reading. Blackwell Publishing Ltd, Oxford, UK, 2007.

[RM82] D. E. Rumelhart and J. L. McClelland. An interactive activation model of context effects in letter perception: Part 2. The contextual enchancement effects and some tests and extensions of the model. *Psychological Review*, 89:60–94, 1982.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[RP89] Keith Rayner and Alexander Pollatsek. *The Psychology of Reading.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[RRAH$^+$12] Abdullah M Rashwan, Mohsen A Rashwan, Ahmed Abdel-Hameed, Sherif Abdou, and Ahmed Husien Khalil. A robust omnifont open-vocabulary arabic ocr system using pseudo-2d-hmm. In *IS&T/SPIE Electronic Imaging*, pages 829707–829707, 2012.

[RSB10a] S.F Rashid, F. Shafait, and T.M. Breuel. Connected component level multiscript identification from ancient document images. In *9th IAPR*

Workshop on Document Analysis Systems, (short paper), Boston, MA, USA, June 2010.

[RSB10b] S.F. Rashid, F. Shafait, and T.M. Breuel. Discriminative learning for script recognition. In *17th IEEE International Conference on Image Processing (ICIP), 2010*, pages 2145–2148, 2010.

[RSB10c] Sheikh Faisal Rashid, Faisal Shafait, and Thomas M. Breuel. Visual Recognition of Permuted Words. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Human Vision and Electronic Imaging XV - part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-21, 2010, Proceedings*, volume 7527 of *SPIE Proceedings*, page 752715. SPIE, 2010.

[RSB11] Sheikh Faisal Rashid, Faisal Shafait, and Thomas M. Breuel. An Evaluation of HMM-Based Techniques for the Recognition of Screen Rendered Text. In *Proc. of Eleventh International Conference on Document Analysis and Recognition*, pages 1260–1264, September 2011.

[RSB12] Sheikh Faisal Rashid, Faisal Shafait, and Thomas M. Breuel. Scanning neural network for text line recognition. In *10th IAPR International Workshop on Document Analysis Systems (DAS), 2012*, pages 105–109. IEEE, 2012.

[RWJL06] K. Rayner, S. J. White, R. L. Johnson, and S. P. Liversedge. Raeding wrods with jubmled lettres: There is a cost. *Psychological Science*, 17:192–193, 2006.

[Sch03] Marc-Peter Schambach. Model length adaptation of an hmm based cursive word recognition system. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 109–113, 2003.

[Sch09] Marc-Peter Schambach. Recurrent hmms and cursive handwriting recognition graphs. In *10th International Conference on Document Analysis and Recognition, 2009. ICDAR'09.*, pages 1146–1150, 2009.

[SG04] Sofie Schoonbaert and Jonathan Grainger. Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language and Cognitive Processes*, 19(3):333–367, 2004.

[SGH95] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time delay neural networks and hidden Markov models. *Machine Vision and Applications*, 8(4):215–223, 1995.

[SKB08] F. Shafait, D. Keysers, and T. M. Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Proc. SPIE Document Recognition and Retrieval XV*, pages 101–106, San Jose, CA, USA, January 2008.

[SLM+96] Richard Schwartz, Christopher LaPre, John Makhoul, Christopher Raphael, and Ying Zhao. Language-independent ocr using a continuous speech recognition system. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 99–103. IEEE, 1996.

[SLN+99] Richard Schwartz, Zhidong Lu, Prem Natarajal, Lssam Bazzi, Andras Kornai, and John Makhoul. Recent improvements in the bbn ocr system. In *Proceedings 1999 Symposium on Document Image Understanding Technology*, page 245. UMD, 1999.

[SM89a] Mark S. Seidenberg and James L. McClelland. A distributed developmental model of word recognition and naming. *Psychological Review*, 96:523–568, 1989.

[SM89b] S. M. Seidenberg and J. L. McClelland. A distributed, developmental model of word recognition and naming. *Psychological Review*, 96:523–568, 1989.

[Smi69] F. Smith. Familiarity of configuration vs.determinability of features in the visual identification of words. *Psychonomic Science*, 14:261–262, 1969.

[Smi07] R. Smith. An Overview of the Tesseract OCR Engine. In *Proc. of Ninth International Conference on Document Analysis and Recognition*, pages 629–633, 2007.

[Smi13] R. Smith. History of the Tesseract OCR engine: what worked and what didn't. In *Proc. SPIE 8658, Document Recognition and Retrieval XX*, pages 865802–865802–12, 2013.

[SP00] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[Spe60] G. Sperling. The information available in brief visual presentations. *Psychological Monographs*, 74, 1960.

[Spe63] G. Sperling. A model for vision memory tasks. *Human Factors*, 5:19–31, 1963.

[Spi97] A. L. Spitz. Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3), March 1997.

[SREB11] Tanzila Saba, Amjad Rehman, and Mohamed Elarbi-Boudihir. Methods and strategies on off-line cursive touched characters segmentation: a directional review. *Artificial Intelligence Review*, pages 1–20, 2011.

[SSB$^+$97] Bernhard Scholkopf, Kah-Kay Sung, Chris JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765, 1997.

[SuHKB06] F. Shafait, Adnan ul Hasan, D. Keysers, and T. M. Breuel. Layout analysis of urdu document images. In *10th IEEE International Multitopic Conference (INMIC 2006), Islamabad, Pakistan.*, Dec 2006.

[SZHZ07] T.-H. Su, T.-W. Zhang, H.-J. Huang, and Y. Zhou. HMM-Based Recognizer with Segmentation-free Strategy for Unconstrained Chinese Handwritten Text. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 133–137, 2007.

[TG03] Edmondo Trentin and Marco Gori. Robust combination of neural networks and hidden markov models for speech recognition. *Neural Networks, IEEE Transactions on*, 14(6):1519–1531, 2003.

[TNBC99] Kazem Taghva, Tom Nartker, Julie Borsack, and Allen Condit. UNLV-ISRI document collection for research in OCR and information retrieval. In *Proc. of SPIE Document Recognition and Retrieval VII*, volume 3967, pages 157–164, December 1999.

[Typ12] Typographical. Glossary of (some) typographical terms. `http://pfaedit.sourceforge.net/glossary.html`, 2012. [Online; accessed 13-October-2012].

[VBRB10] Joost Van Beusekom, Yves Rangoni, and Thomas M Breuel. Trainable multiscript orientation detection. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7534, page 31, 2010.

[Vit67] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[VJ04] P. Viola and M. J. Jones. Robust real-time face detection. *Int. Journal of Computer Vision*, 57(2):137–154, 2004.

[Whi01] C. Whitney. How the brain encodes the order of letters in a printed word: The SERIOL model and selective literature review. *Psychonomic Bulletin & Review*, 8:221–243, 2001.

[Whi08] Carol Whitney. Supporting the serial in the SERIOL model. *Language and Cognitive Processes*, 23(6):824–865, Feb 2008.

[WHL10] X. Wang, L. Huang, and C.P. Liu. A video text location method based on background classification. *Int. Jour. on Document Analysis and Recognition*, 13(3):187–207, Sep. 2010.

[Wic69] W. A. Wickelgren. Context sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review*, 76:1–15, 1969.

[Wik11] Wikipedia. X-height — Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/w/index.php?title=Cap_height&oldid=369297561`, 2011. [Online; accessed 13-October-2012].

[Wik12a] Wikipedia. Nastaleeq Script. `http://en.wikipedia.org/wiki/Nastaleeq`, 2012. [Online; accessed 18-April-2012].

[Wik12b] Wikipedia. Urdu. `http://en.wikipedia.org/wiki/Urdu`, 2012. [Online; accessed 18-April-2012].

[Wik13] Wikipedia. Phishing — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Phishing&oldid=558875672, 2013. [Online; accessed 09-June-2013].

[WKJ06] Steffen Wachenfeld, Hans-Ulrich Klein, and Xiaoyi Jiang. Recognition of Screen-Rendered Text. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 02, pages 1086–1089, 2006.

[WKJ07] Steffen Wachenfeld, Hans-Ulrich Klein, and Xiaoyi Jiang. Annotated Databases for the Recognition of Screen-Rendered Text. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, volume 02, pages 272–276, 2007.

[WL04] C. Whitney and M. Lavidor. Why words length only matters in the left visual field. *Neuropsychologia*, 42:1680–1688, 2004.

[Woo38] R. S. Woodworth. *Experimental Psychology.* Holt, New York, 1938.

[WP08] Carol Whitney and Cornelissen Piers. SERIOL Reading. *Language and Cognitive Processes*, 23(1):143–164, Jan 2008.

[YKO$^+$06] Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[YMIM04] Shinsuke Yanadume, Yoshito Mekada, Ichiro Ide, and Hiroshi Murase. Recognition of Very Low-Resolution Characters from Motion Images Captured by a Portable Digital Camera. In *PCM (1)*, pages 247–254, 2004.

Sheikh Faisal Rashid

CONTACT INFORMATION

b4value.net (GmbH)
German Research Centre for Artificial Intelligence (DFKI)
Trippstadter Str. 122, 67663 Kaiserslautern, Germany

Mobile:+4917672463890
*E-mail:*rashid@iupr.com

OBJECTIVE

Placement in a position that allows for advanced research in artificial intelligence and machine learning

RESEARCH INTERESTS

Machine Learning, Konwledge Management, Statistical Pattern Recognition, Document Analysis and Information Extraction

EDUCATION

Technical University Kaiserslautern, Germany

Doctor of Engineering (Dr.-Ing.), Computer Science

- Grades: Magna cum laude (1.0)
- Thesis Topic: *Optical Character Recognition - A combined ANN/HMM Approach*
- Candidacy: Research problems in OCR of degraded and cursive documents
- Advisor: Professor Dr. Thomas Breuel
- Area of Study: Statistical Pattern Recognition, Machine Learning and Cognitive Psychology

University of Engineering and Technology, Lahore Pakistan

M.Sc., Computer Science, 2007

- Grades: A, 1st Div.
- Thesis Topic: *Image Processing to Capture Actor Movements for Generating Animated Videos*
- Advisor: Professor Dr. Shaiq A. Haq
- Area of Study: Computer Vision

B.Sc.(Hons.), Computer Science, 2003

- Grades: A 1st Div
- Final Year Project: *Arabic Speech Recognition and Synthesis System*

PROFESSIONAL APPOINTMENTS

b4value.net (GmbH)
Application Developer October 2013 – To date

Siemens AG, Mobility, Logistics and Postal Automation
Research Associate November 2012 – July 2013
Research Project: Optimizations of Recurrent Neural Networks (RNNs) in application to postal automation

Image Understanding and Pattern Recognition Research
Research Associate October 2009 – October 2012
Research Project: TextGrid – Virtual Research Environment for the Humanities

Fraunhofer-Gesellschaft, Kaiserslautern, Germany
Research Assistant August 2008 – September 2009
Research Project: Industrial Mathematics and Image Processing software development

Department of Computer Science and Engineering, University of Engineering and Technology, Lahore Pakistan

- **Assistant Professor** February 2004 – January 2008
- **Lecturer** August 2003 – January 2004

Lahore University of Management Sciences (LUMS), Lahore Pakistan

- **Research Assistant** April 2003 – September 2003
- **Teacher Assistant** March 2003 – May 2003

Adzze IT Solutions, Lahore Pakistan
Project Manager November 2004 – January 2005

Lahore University of Management Sciences (LUMS), Lahore Pakistan
Oracle Programmer & Administrator August 2001 – May 2002

Systems & Solutions (Pvt.) Limited, Lahore Pakistan
Oracle Developer May 2001 – July 2001

Publications

Sheikh Faisal Rashid, Marc-Peter Schambach, Jörg Rottland, Stephan von der Nüll. ***Low Resolution Arabic Recognition with Multidimensional Recurrent Neural Networks***. 4th International Workshop on Multilingual OCR (MOCR2013). Washington, DC August 24, 2013.

Marc-Peter Schambach, **Sheikh Faisal Rashid**, ***Stabilize Sequence Learning with Recurrent Neural Networks by Forced Alignment***. 12th International Conference on Document Analysis and Recognition, ICDAR'13. Washington, DC August 25-28, 2013.

Adnan Ul-Hasan, Saad Bin Ahmed, **Sheikh Faisal Rashid**, Faisal Shafait , Thomas M. Breuel. ***Offline Printed Urdu Nastaleeq Script Recognition with Bidirectional LSTM Networks***. 12th International Conference on Document Analysis and Recognition, ICDAR'13. Washington, DC August 25-28, 2013.

Adnan Ul-Hasan, Syed Saqib Bukhari, **Sheikh Faisal Rashid**, Faisal Shafait , Thomas M. Breuel. ***Semi-Automated OCR Database Generation for Complex Scripts***. 21st International Conference on Pattern Recognition, Tsukuba International Congress Center Tsukuba Science City, JAPAN, November 11-15, 2012.

Sheikh Faisal Rashid, Faisal Shafait, Thomas M. Breuel. ***Scanning Neural Network for Text Line Recognition***. 10th IAPR International Workshop on Document Analysis Systems, DAS'12. Gold Coast, Queensland, Australia, March 2012. (**The IAPR Best Student Paper Award**)

Sheikh Faisal Rashid, Faisal Shafait, Thomas M. Breuel. ***An evaluation of HMM-based Techniques for the Recognition of Screen Rendered Text***. 11th International Conference on Document Analysis and Recognition, ICDAR'11. Beijing, China, September 2011.

Sheikh Faisal Rashid, Faisal Shafait, Thomas M. Breuel. ***Discriminative Learning for Script Recognition***, 17th International Conference on Image Processing, ICIP'10. Hong Kong, China, September 2010.

Sheikh Faisal Rashid, Faisal Shafait, Thomas M. Breuel. ***Connected Component level Multiscript Identification from Ancient Document Images***, 9th IAPR Workshop on Document Analysis Systems, DAS'10. Boston, MA, USA, June 2010.

Sheikh Faisal Rashid, Faisal Shafait, Thomas M. Breuel. ***Visual Recognition of Permuted Words***, SPIE Human Vision and Electronic Imaging XV, HVEI'10. San Jose, CA, USA, January 2010.

Sheikh Faisal Rashid, Syed Saqib Bukhari, Faisal Shafait, Thomas M. Breuel. ***A Discriminative Learning Approach for Orientation Detection of Urdu Document Images***, 13th IEEE International Multi-topic Conference, INMIC'09. Islamabad, Pakistan, December 2009.

M. Shoaib, **F. Rasheed**, J.Akhtar, M.Awais, S. Masud, S.Shamai1. ***A Novel Approach to Increase the Robustness of Speaker Independent Arabic Speech Recognition***, 7th IEEE International Multi-topic Conference, INMIC03. Islamabad Pakistan, December 2003.

Muhammad Shoaib Bashir, **Sh Faisal Rasheed**, Shahid Masud, M. Muhammad Awais, Shafay Shamail ***Simulation of Arabic Phoneme Identification through Spectrographic Analysis*** , International Bhurban Conference on Applied Sciences & Technology. Bhurban Pakistan, June 2003.

Courses Taught

MS Courses

- *Applications of Artificial Intelligence*, summer semester 2014, TU Kaisersluatern, Co-lecturer with Prof. Dr. habil. Marcus Eichenberger-Liwicki

- *Document and Content Analysis*, summer semester 2010, TU Kaiserslautern, Co-lecturer with Prof. Dr. Thomas Breuel

- *Distributed Systems*, session 2006, UET Lahore

BS Courses

- Programming in Dot Net Frame Work, session 2005
- Data Communication & Networks, session 2005
- Dot Net Frame Work, session 2004
- Data Base Management Systems (ORACLE 10g), session 2003 (Fall)
- Micro Controllers & Micro Processors, session 2003
- Dot Net Frame work, session 2003 (Fall)
- Data Communication & Networks, session 2004
- Operating Systems, session 2003
- Expert Systems, session 2002
- Distributed Object Oriented Technologies, session 2001
- Expert Systems, session 2001
- Introduction to Computer Science, session 2003 (Fall)
- Management Information Systems, session 2000

Projects Supervised

MS Project

- Evaluation of Recurrent Neural Networks on Cursive and Non-Cursive Datasets, year 2012, TU Kaiserslautern

BS Projects

- RobiBall A Robot for Balls Posting in a Specific Arena, session 2003
- MYRSH Expert system for voice synthesis , session 2002
- Analysis & Design of 3G Platform based CDMA-WLL for Wireless Communication, session 2002
- Optimization of PEZW in VTC Encoder in MPEG4 by Using MMX/SSE Instructions Set, session 2001
- Internet Cafe Commander, session 2001
- TRENDAn Information Retrieval System for Medical Diagnosis Expert System, session 2001
- Mega WAP, session 2001

Workshops & Conferences

10th IAPR International Workshop on Document Analysis Systems, 27 - 29 March 2012, Gold Coast, Queensland, Australia.

Brain-Computer-Interface (BCI) Workshop, 13 March 2012, DFKI, Kaiserslautern, Germany.

ENS/INRIA Visual Recognition and Machine Learning Summer School, 25 - 29 July 2011, Paris, France.

Workshop on "Human Computer Interaction & Visualization" (HCIV), 1 - 2 March 2010, Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, Germany.

International Bhurban Conference on Applied Sciences & Technology, June 2003, Bhurban, Pakistan.

Dynamic Learning & Management Skills Workshop, 18 - 21 January 2001, Lahore University of Management Sciences (LUMS), Pakistan.

Oracle Hands-on Training Workshop 26 February - 11 May 2001, Punjab Information & Technology Board Lahore, Pakistan.

Honors and Awards

First place in ICDAR 2013 competition on multi-font and multi-size digitally represented Arabic text recognition

The IAPR Best Student Paper Award, DAS 2012, Gold Coast, Queensland, Australia

HEC-DAAD Ph.D Scholarship, 2008–2012

Best Teacher's Award for the years: 2004–2005 and 2003–2004
All Pakistan Software Competition Awards:

- 1st prize @ GIKI 2002
- 3rd prize @ BAHRIA 2002
- 3rd prize @ SOFTEC FAST 2003

OCP Scholarship Award from Pakistan Information Technology Board (PITB)

Distinction Scholarship Award at Higher Secondary School

Certifications

Oracle Certified Professional (OCP 2000 / 6*i*)

Software Skills

Programming Languages: Python, Matlab, R, C#, C/C++, Tcl/Tk, Assembly Language for IBM PC & 8051 Microcontroller, UML

Tools: MS Visual Studio (Dot Net Framework), MS Project, V Tune Performance Analyzer, Rational Rose, Visio 2000

Databases: SQL, PL/SQL, MS Access, Oracle Develope, Erwin, Toad

References

Prof. Dr. Prof. h.c. Andreas Dengel
Professor, Department of Computer Science
Technical University of Kaiserslautern, Germany
Director, German Research Center for Artificial Intelligence (DFKI)
Email: andreas.dengel@dfki.de

Dr. Thomas M. Breuel
Research Scientist, Google Brain Team
Former Professor, Department of Computer Science
Technical University of Kaiserslautern, Germany
Fomer Director, Image Understanding and Pattern Recognition Research (IUPR)
Email: tmb@iupr.com

Dr. Mian Muhammad Awais
Associate Professor
Department of Computer Science
Lahore University of Management Sciences, Lahore Pakistan
Email: awais@lums.edu.pk