

DEEPASHTO
An End-to-End OCR System for Pashto
Cursive Script

(A PIONEERING STUDY)

PhD Thesis

submitted to the Department of Computer Science
of the University of Kaiserslautern

by

Riaz Ahmad

Kaiserslautern, March 5, 2018

Abstract

Optical Character Recognition (OCR) system plays an important role in digitization of data acquired as images from a variety of sources. Although the area is very well explored for Latin languages, some of the languages based on Arabic cursive script are not yet explored. It is due to many factors: Most importantly are the unavailability of proper data sets and complexities posed by cursive scripts. The Pashto language is one of such languages which needs considerable exploration towards OCR. In order to develop such an OCR system, this thesis provides a pioneering study that explores deep learning for the Pashto language in the field of OCR.

The Pashto language is spoken by more than 50 million people across the world, and it is an active medium both for oral as well as written communication. It is associated with rich literary heritage and contains huge written collection. These written materials present contents of simple to complex nature, and layouts from hand-scribed to printed text. The Pashto language presents mainly two types of complexities (i) generic w.r.t. cursive script, (ii) specific w.r.t. Pashto language. Generic complexities are cursiveness, context dependency, and breaker character anomalies, as well as space anomalies. Pashto specific complexities are variations in shape for a single character and shape similarity for some of the additional Pashto characters. Existing research in the area of Arabic OCR did not lead to an end-to-end solution for the mentioned complexities and therefore could not be generalized to build a sophisticated OCR system for Pashto..

The contribution of this thesis spans in three levels, conceptual level, data level, and practical level. In the conceptual level, we have deeply explored the Pashto language and identified those characters, which are responsible for the challenges mentioned above. In the data level, a comprehensive dataset is introduced containing real images of hand-scribed contents. The dataset is manually transcribed and has the most frequent layout patterns associated with the Pashto language. The practical level contribution provides a bridge, in the form of a complete Pashto OCR system, and connects the outcomes of the conceptual and data levels contributions. The practical contribution comprises of skew detection, text-line segmentation, feature extraction, classification, and post-processing. The OCR module is more strengthened by using deep learning paradigm to recognize Pashto cursive script by the framework of Recursive Neural Networks (RNN). Proposed Pashto text recognition is based on Long Short-Term Memory Network (LSTM) and realizes a character recognition rate of 90.78% on Pashto real hand-scribed images. All these contributions are integrated into an application to provide a flexible and generic End-to-End Pashto OCR system.

The impact of this thesis is not only specific to the Pashto language, but it is also beneficial to other cursive languages like Arabic, Urdu, and Persian e.t.c. The main reason is the Pashto character set, which is a superset of Arabic, Persian, and Urdu languages. Therefore, the conceptual contribution of this thesis provides insight and proposes solutions to almost all generic complexities associated with Arabic, Persian, and Urdu languages. For example, an anomaly caused by breaker characters is deeply analyzed, which is shared among 70 languages, mainly use Arabic script. This thesis presents a solution to this issue and is equally beneficial to almost all Arabic like languages.

The scope of this thesis has two important aspects. First, a social impact, i.e., how a society may benefit from it. The main advantages are to bring the historical and almost vanished document to life and to ensure the opportunities to explore, analyze, translate, share, and understand the contents of Pashto language globally. Second, the advancement and exploration of the technical aspects. Because, this thesis empirically explores the recognition and challenges which are solely related to the Pashto language, both regarding character-set and the materials which present such complexities. Furthermore, the conceptual and practical background of this thesis regarding complexities of Pashto language is very beneficial regarding OCR for other cursive languages.

Structure of the Thesis

1	Introduction	7
1.1	Motivation	8
1.2	Goals and Hypotheses	9
1.2.1	Hypotheses	9
1.3	Contributions	10
1.3.1	Conceptual Contributions	11
1.3.2	Data contributions	12
1.3.3	Practical/Implementation Contributions	12
1.4	Thesis Structure	13
I	Background	15
2	Pashto Language	17
2.1	Motivation	17
2.2	Introduction to Pashto	18
2.3	Pashto Characters	19
2.4	Shape Conventions	21
2.4.1	Breaker and Non-Breaker Characters	21
2.4.2	Ligatures	23
2.4.3	Primary Ligature and Secondary Components	23
2.5	Challenges	23
2.5.1	Generic Challenges	24
2.5.2	Pashto Specific Challenges	27
2.6	Related Work	29
2.7	Discussion	29

3	Neural Networks and Deep Learning	31
3.1	Introduction to Neural Networks	31
3.2	Supervised Learning	33
3.2.1	Sequence Classification	34
3.2.2	Segment Classification	34
3.2.3	Temporal Classification	35
3.3	Recurrent Neural Networks	36
3.3.1	Long Short Term Memory (LSTM) units	37
3.3.2	Bi-Directional LSTM (BLSTM)	38
3.3.3	Multi-Dimensional LSTM (MDLSTM)	39
3.4	Connectionist Temporal Classification (CTC)	40
3.5	Deep Learning	41
3.6	Discussion	41
II	Benchmark Pashto Language	43
4	Textual Analysis	45
4.1	Motivation	45
4.2	Pashto Text Acquisition	46
4.2.1	Text Acquisition Approach	47
4.3	Pashto Text Statistics	48
4.3.1	Pashto Ligatures Extraction	48
4.3.2	Pashto Primary Ligatures	49
4.4	Conclusion	50
5	Benchmark OCRs for Arabic Scripts	55
5.1	Motivation	55
5.2	Segmentation-based Approaches	56
5.2.1	Template Matching	56
5.2.2	Over Segmentation based Approaches	57
5.3	Holistic Approaches or Segmentation-free OCRs	58
5.4	Conclusion	60
III	Datasets	63
6	Ligature Based Synthetic Datasets	65
6.1	Motivation	65
6.2	Related Work	66
6.3	Contribution	67

6.3.1	Scale and Orientation variations: Ligature-Based-II	67
6.3.2	Towards Big Database for Pashto Ligatures: Ligature-Based-III	68
6.4	Conclusion	69
7	Real Pashto Imagebase Creation	71
7.1	Motivation	71
7.2	Data Acquisition	72
7.3	Data Acquisition for KPTI	73
7.4	KPTI Description	77
7.5	Conclusion	78
IV	Pre-Processing	79
8	Skew Detection and Correction	81
8.1	Motivation	81
8.2	Related Work	82
8.2.1	Projection Profile (PP)	83
8.2.2	Hough Transform (HT)	84
8.2.3	Fourier Transformation (FT)	85
8.2.4	Axis-parallel Bounding Box (APB)	86
8.3	Datasets	87
8.4	Proposed Method	88
8.4.1	Description of our Proposed Method	89
8.5	Evaluation Metric	91
8.6	Results and Discussion	91
8.7	Conclusion and Future Work	93
9	Textline Segmentation	95
9.1	Motivation	95
9.2	Related Work	96
9.3	Problem Definition	99
9.4	Methodology	99
9.5	Description of the Dataset	102
9.6	Evaluation	102
9.7	Experimental Results	103
9.8	Limitations	104
9.9	Conclusion	105

V	Pashto OCR	107
10	Scale and Rotation Invariant OCR for Pashto	109
10.1	Motivation	109
10.2	Related Work	111
10.3	Dataset	112
10.4	Methodology	112
10.4.1	SIFT Based Ligature Matching	113
10.4.2	LSTM Based Recognition	114
10.4.3	HMM Based Recognition of Pashto Ligatures	116
10.5	Evaluation	116
10.6	Conclusion and Future Work	118
11	Pashto OCR and Deep Learning Benchmark	121
11.1	Motivation	121
11.2	Related Work	122
11.3	Dataset	123
11.4	Methodology	123
11.4.1	Topology of Proposed BLSTM Model	125
11.4.2	Topology of Proposed MDLSTM Model	126
11.4.3	Evaluation Metric	127
11.4.4	Selection of Optimal Parameters	127
11.5	Experiments	129
11.6	Results and Discussion	129
11.7	Conclusion and Future Work	131
12	Post-Processing, Space Recognition Anomaly	133
12.1	Motivation	133
12.2	Problem Definition	135
12.2.1	Impact on Rendering	135
12.3	Related Work	136
12.4	Proposed Solution	138
12.5	Datasets and Evaluation Metric	139
12.5.1	Evaluation Metric	140
12.6	Results and Discussion	141
12.7	Conclusion	141
13	End-to-End OCR	143
13.1	Motivation	143
13.2	Tools and Requirements	144

13.2.1	Back-end Tool	144
13.2.2	Front-end Tool, Qt Designer	145
13.3	Description of GUI	145
13.3.1	Data Preparation Mode	146
13.3.2	Training Mode	149
13.3.3	Testing Mode	153
13.3.4	Closure of End-to-End OCR System	155
13.4	Conclusion	155
14	Conclusion and Future Work	157
14.1	Conclusion	157
14.2	Future Work	160

List of Figures

1.1	Pipeline of Document Image Analysis (DIA).	8
1.2	Thesis's contributions.	11
2.1	Languages belong to Indo-European family	18
2.2	Geographic locations of Pashto speakers.	19
2.3	Arabic like languages, and their character sets	20
2.4	13 breakers characters of Pashto language.	22
2.5	Ligatures, and the role of breaker-characters.	22
2.6	Relationship of Ligatures, Primary ligatures, and Secondary components.	23
2.7	Cursiveness in English and Pashto	24
2.8	Character shapes depend on context (Context dependency).	25
2.9	Space anomalies in Arabic like scripts	26
2.10	A Pashto sentence rendered with 5 different fonts.	27
2.11	The Pashto ی [Yey]s and their five different variants.	28
3.1	Information flow via biological neuron	31
3.2	An Artificial Neuron (AN), and its formulation.	32
3.3	FNN vs RNN	36
3.4	LSTM Block	37
3.5	BLSTM Architecture	38
3.6	MDLSTM & scanning.	39
3.7	Scanning 2D data in all 4 directions	40
4.1	How to split a Pashto word into ligatures.	49
4.2	A cluster of 100 ligature	51
5.1	Template Matching	57

6.1	Images describe the first version of syntactic dataset.	66
6.2	Each ligature has 4 scale and 4 rotation variations	67
6.3	Scale variations for each Pashto ligature.	68
6.4	Each scale variation has 12 rotation variations	69
7.1	The creation of scribe material by a Katib.	73
7.2	Samples from KPTI dataset	75
7.3	An image, its text-lines and Ground-truth conventions.	77
8.1	Illustration of skew correction.	82
8.2	Skew correction by Horizontal Profile method.	83
8.3	Skew correction by Hough Transform method.	84
8.4	Hough transformation of an input image.	85
8.5	Fourier magnitude's spectrum for skew correction.	86
8.6	The main concept of axis-Parallel Bounding box method.	87
8.7	Area of a bounding box is computed by points...	88
8.8	True-lines vs False-line	89
8.9	Failure case of APB.	92
8.10	Document, which could not be bounded in a rectangular form.	93
8.11	Successfully de-skewed images.	94
9.1	Text-line segmentation using Ryu's method.	96
9.2	Failure in text-lines segmentation by Ryu's method.	97
9.3	An input image along with its plot of HPP.	100
9.4	The effect of convolution and HPP's plot.	101
9.5	Results of our proposed method.	104
10.1	Scale and rotation variations in real text.	110
10.2	Pashto ligatures and words.	110
10.3	Rotation variation.	113
10.4	Split of Ligature-Based-III.	114
10.5	MDLSTM Model for scale and rotation variations.	115
10.6	Proposed HMM Model	116
10.7	Results	117
10.8	SIFT; Miss-classification due to shape similarity.	118
11.1	Text-lines from KPTI dataset.	124
11.2	Proposed BLSTM architecture	125
11.3	Proposed MDLSTM architecture.	127
11.4	Results of our proposed OCR system.	130
12.1	Pashto Breaker characters	134

12.2	Space anomalies and their illustration.	134
12.3	Rendering problem	136
12.4	The proposed solution for transforming default ground truth.	138
12.5	Proposed MDLSTM architecture.	140
13.1	Main window-Form	146
13.2	Image scanning	147
13.3	Pre-process data	147
13.4	Splitting of Data	148
13.5	Creation of NetCDF files	149
13.6	Configuration of Network Model	150
13.7	Train a Model	151
13.8	Restart Training	152
13.9	Start New Test	153
13.10	Pagewise OCR	154

List of Tables

2.1	The shapes of Pashto characters and Unicode information.	21
4.1	Websites based on Pashto text.	47
4.2	Statistics of words and ligatures.	48
4.3	Unique Pashto words and their frequencies.	48
4.4	Unique Pashto ligatures and their frequencies.	50
4.5	Shape codes for primary ligatures in Pashto language.	52
4.6	Top ten primary ligatures of Pashto language.	53
5.1	LSTM based related work for the OCR of Arabic scripts.	59
7.1	Statistics of KPTI dataset	76
8.1	Skew correction evaluation on DISEC'13 data.	91
8.2	Skew correction evaluation on Tobacco800, and KPTI.	92
9.1	Results of text-line extraction methods	103
11.1	KPTI statistics.	124
11.2	Empirical analysis for BLSTM	128
11.3	Empirical analysis for MDLSTM	128
11.4	The impact of size of <i>tanh</i> layers on hidden layers	128
11.5	Top 20 confusion related to Pashto alphabets in KPTI dataset.	130
11.6	Results of BLSTM and MDLSTM on KPTI dataset	130
12.1	Top two confusion	135
12.2	Results of our proposed system.	141

Humans can read and comprehend text documents, provided that one has learned the basic knowledge of reading and understanding of a specific language/script. Similarly, Optical Character Recognition (OCR) systems are developed to mimic human's readability skill in computers [Her82]. More technically, OCRs are software applications, which electronically convert images of typed, handwritten or printed text into machine-encoded text. An OCR application plays a very important role in digitization of text documents. The major advantages are storage reduction, ability of searching in the content, and compatibility with other computer applications. These applications could be translation services, text mining for useful analysis, text to speech conversion, and preservation of historical data.

OCR is a research area in the domain of Document Image Analysis (DIA), which is one of the branches of Pattern Recognition (PR). A pipeline of a typical DIA system is comprised of stages like; (i) Acquisition of text images, (ii) classification, (iii) pre-processing, (iv) text recognition (OCR), and (v) post-processing. A Figure 1.1 depicts a pipeline of a typical DIA system.

In the first stage, images are acquired using a scanner, or a camera (photo or video). The documents are first classified into predefined categories [ACM⁺15, AKAL17, AK17]. These acquired images are crude and contain some unwanted artifacts like: noise, skew, graphics, borders, bleed effect, etc. Therefore, in a pre-processing stage, these images are de-skewed, cleaned, and segmented into their primitives. Primitives are titles, paragraphs, text-lines, words, and characters. In most cases, the result of pre-processing should be in a form which is suitable for OCR stage.

In the past decades, there has been a significant research in the field of OCR [ABM95, IOO91]. The OCR applications have been explored to maturity level for the mainstream languages like English, French, German, and for their cursive variations [TJT96], while cursive scripts like Arabic still need further research towards maturity. In general, re-

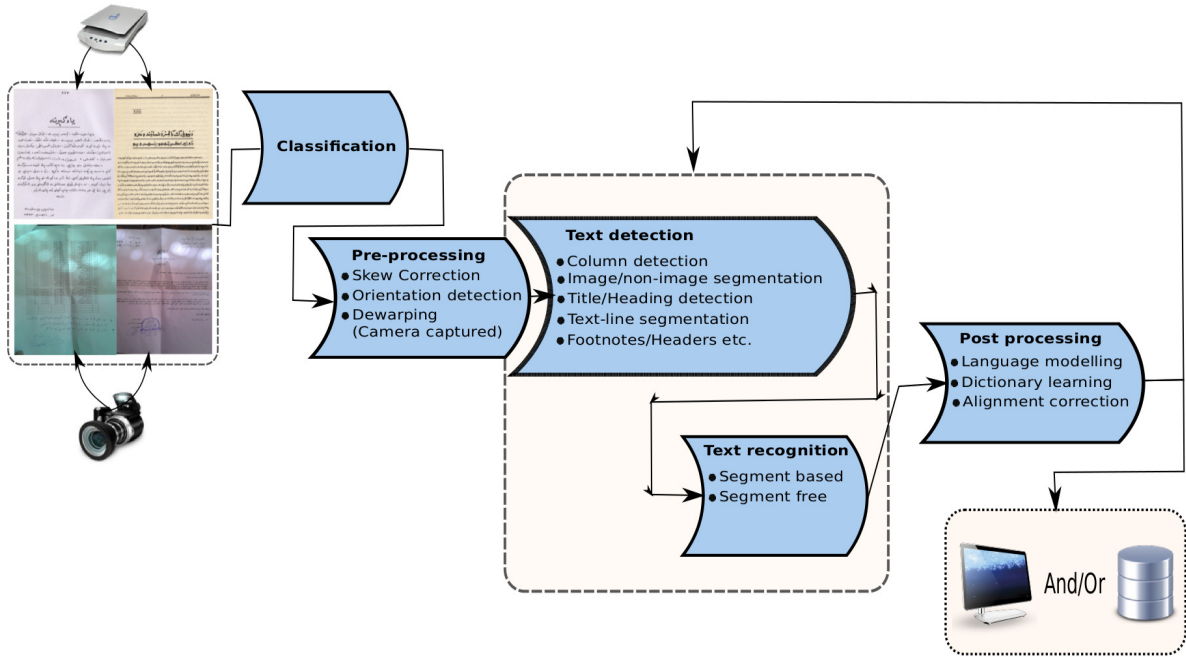


Figure 1.1: *A typical pipeline of document image analysis system (DIA). It is mainly comprised of 5 stages namely, image acquisition, classification, pre-processing, OCR, and post-processing.*

search focuses on partial achievements and presents very little effort toward an end-to-end solution. Such trend makes it hard to generalize the partial achievements across the complete DIA system. This thesis focuses on an end-to-end OCR system and explores Pashto cursive language as a problem domain.

1.1 Motivation

In the era of modern technology and globalization, the demands are equally increasing for understanding different regional languages and translating them into mainstream languages. Pashto is one of such languages (cf. Section 2.2). It has a rich literary heritage. It is not only spoken but also used as a written medium among 50 million people across the world [Pen55]. However, in OCR perspective, the Pashto language has not been studied so far due to limited research and unavailability of data.

The core motivation of this thesis is a lack of research in the field of OCR for the Pashto language. Like other languages, Pashto has its own specificities not only linguistics composition but also character's shape. These specificities make the development of an OCR system more language specific. Consequently, such specificities need a dedicated research to find the conceptual foundation for the challenges posed by the Pashto language in the field of OCR. In most cases, such challenges can be apprehended with the help of

real data. However, the case becomes more difficult when such data is not present. Thus we need an appropriate data to facilitate OCR application for the Pashto language.

Another motivation is the lack of data. Proper data is essential as a benchmark for the investigation of Pashto language in the field of OCR. Unfortunately, there is no such data that could give us insight into understanding challenges in the development of OCR system regarding the Pashto language. This thesis contributes a comprehensive collection of real world data. The data consists of diverse material and contains the most frequent patterns and layouts associated with the Pashto text.

In addition to that, there is an observation that most of the research does not provide an end-to-end system as a final product in the domain of OCR. Lack of such system only guarantees partial achievements in the DIA system. On another hand, an end-to-end system gives the insight to tackle the conceptual issues in the core, connect them with solutions, and provide breakthrough as an application level. Such end-to-end systems provide ease and attract a common man for their utilization. Therefore, this thesis provides an end-to-end system, where all the contributions are integrated into the OCR system to handle Pashto text documents.

1.2 Goals and Hypotheses

Many languages have been explored in terms of OCR applications. Despite this fact, the literature reveals very limited research work regarding the development of Pashto OCR system. Consequently, this research aims to reduce this research gap by conducting empirical research for the Pashto language towards an OCR system. This thesis also aims to find the conceptual reasons behind the challenges posed by the Pashto language in OCR and proposes solutions based on conceptual findings. Such conceptual findings will not only facilitate Pashto language but will also help the researchers to improve OCR systems for other cursive scripts. Another aim of this thesis is to combine all the outcomes and contributions in an end-to-end OCR system, such that the outcomes and benefits of this research are easily available for a common user.

1.2.1 Hypotheses

To reach these goals, we have encountered some intermediate findings during the progression of this study. These are related to noise removal, skew correction, text-line segmentation, OCR, post-processing, rendering, and correction. To investigate and empirically analyze these findings, we have established some research questions, which are listed below.

- *What methodology is helpful for sequential data to address Pashto OCR system?*
- *How a scale and rotation invariant OCR can be developed for Pashto text?*
- *What could help us in developing an efficient and robust skew correction approach?*
- *How could we easily segment the text-lines containing large headings and titles in Arabic script?*
- *What are the characters that conceptually link to some complexities in Pashto language and that could lead to affect the performance of Pashto OCR?*

To achieve the goals in the light of above questions, we can postulate the following hypotheses:

- H1. If we provide enough sequential data to a Recurrent Neural Network (RNN) based on Long Short-Term Memory (LSTM) units, then such model is a better choice to address the problem of Pashto OCR system.
- H2. If we want to develop a scale and rotation invariant OCR for Pashto script, then the use of ligatures and primary ligatures might be a better choice as recognizable units.
- H3. If we could provide segmented ligatures or primary ligatures, then a Multidimensional Recurrent Neural Network (MDRNN), especially with LSTM cells is robust in scale and rotation invariant recognition of a complete shape/ligature.
- H4. If a document contains parallel visual patterns, then based on parallel lines a scanned document could effectively be de-skewed.
- H5. If a text block contains text lines of variable sizes, then convolution with Hanning window filter can give us efficient text-line segmentation.
- H6. If we could identify special characters which cause complexities and morphological transformation, then addressing these special characters with specific treatment could benefit OCR methodologies.

1.3 Contributions

This thesis contributes to three levels, (1) conceptual, (2) data, and (3) practical/implementation. Figure 1.2 illustrates the proposed DIA system in the light of thesis's contributions, and the contributions are summarized below:

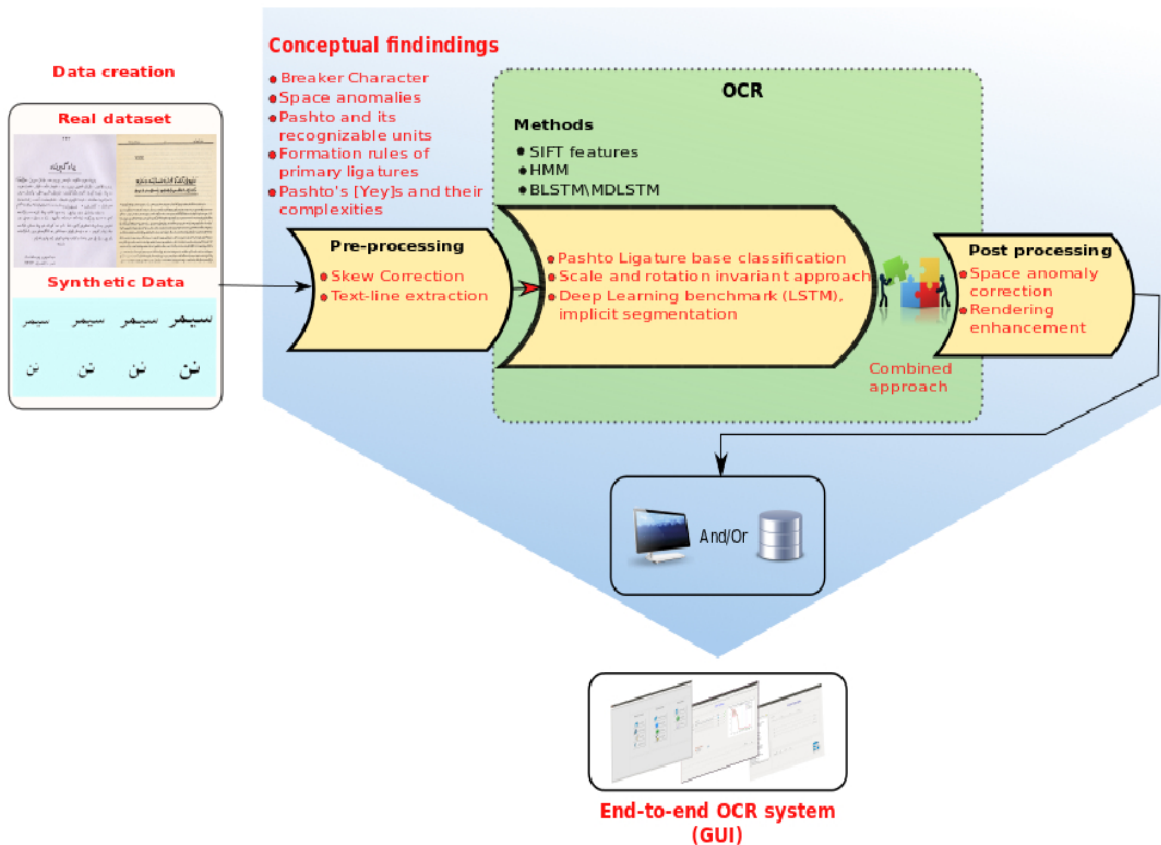


Figure 1.2: *The main contributions include dataset creations, conceptual findings, and OCRs based on deep learning. Finally, the overall contributions are integrated into an end-to-end system. The system is based on GUI and provides ease and support to end users. The red colored texts show the thesis's contributions.*

1.3.1 Conceptual Contributions

- C1.** The first conceptual contribution is the scale and rotation invariant recognition of Pashto ligatures with MDRNN. In this contribution Multi-Dimensional LSTM (MDLSTM) outperforms state-of-the-art descriptor based matching technique i.e., Scale Invariant Feature Transform (SIFT)[Low04] and statistical sequence classifier i.e. Hidden Markov Model (HMM)[Edd96]. This contribution validate hypothesizes H1, H2, and H3.
- C2.** Another conceptual contribution is the identification of the additional characters and some other special characters which produce complexities in the Pashto language. These complexities are cursiveness, space anomaly, and the knowledge of how primary ligatures are formed (cf. Chapter 4). This contribution validates hypothesis H6.

1.3.2 Data contributions

The creation of adequate data is the most laborious task. Because in the supervised learning domain the data needs proper annotations/transcription. This thesis contributes to both synthetic as well as real-world data, that are summarized below:

- C3.** In this thesis, we have created a ligature based synthetic dataset. That initial version contains 1000 unique Pashto ligatures with 4 different scale and 4 dominant rotation variations (cf. Section 6.3.1).
- C4.** The synthetic dataset is further extended by introducing 40 scale and 12 rotation variations for each unique ligature. This dataset is evaluated by MDLSTM, SIFT, and HMM for scale and rotation invariant recognition (cf. Section 6.3.2).
- C5.** Another valuable contribution of this thesis is the creation of first ever real Pashto image-base (KPTI: Katib’s Pashto Text Image-base). The newly created dataset contains contents from diverse resources of Pashto text, which are hand written by scribe/ Katibs¹. The dataset contains 17,015 Pashto text-lines, which are manually transcribed in utf-8 codecs (cf. Chapter 7).

1.3.3 Practical/Implementation Contributions

Practical contributions of this thesis span from pre-processing to post-processing, which covers the entire pipeline of DIA. Other practical contributions are based on conceptual contributions, therefore, has a solid foundation for generalization in all languages derived from Arabic script. Here, the practical contributions are described briefly:

- C6.** Skew detection and correction approach is introduced for scanned documents (Chapter 8). This contribution validates hypothesis H4.
- C7.** The second practical contribution is the introduction of text-line segmentation approach. The proposed method is more focused on the extraction of large titles and headings in Arabic like languages (Chapter 9). This contribution validates hypothesis H5.
- C8.** The third practical contribution of this thesis is the introduction of an OCR based on deep learning; which exploits the power of MDLSTM and Bidirectional LSTM (BLSTM). The proposed system is evaluated on the real data of KPTI dataset (Chapter 11). This contribution validates hypothesis H1.

¹”KATIB” is an Arabic word, means the professional persons, known for their beautiful calligraphic writing style.

- C9.** Another practical contribution of this thesis is related to space anomaly. The space anomaly causes rendering problems in Arabic script. This thesis comprehensively examines this issue and effectively contributes a joint solution. The proposed solution not only achieves the correct rendering but also enhances the overall accuracy (Chapter 12). This contribution validates hypothesis H6.
- C10.** The final practical contribution of the thesis is the integration of all the achievements in one platform. The platform is based on Graphical User Interface (GUI) and provides an end-to-end system for Pashto text recognition.

1.4 Thesis Structure

The rest of the thesis is divided into four parts. The first part presents background knowledge and concepts. It is a vital part and lay down a foundation for understanding the rest of the thesis. It contains three chapters. The first chapter (i.e. Chapter 2) specifically introduces the Pashto language. It presents background information regarding Pashto's history, text contents, character set, and co-relation of Pashto with other languages. The second chapter (i.e. Chapter 3) of this part introduces the basic concepts about Neural Networks and deep learning. This chapter also presents the concepts of sequence classification, RNN, and related work in the domain of OCR. The third chapter (i.e. Chapter 4) of this part presents a textual analysis of Pashto language, it further shares the statistics about the frequency of words, ligatures, primary ligatures, and describes the rules how primary ligatures are formed.

The second part of this thesis is about the datasets that are used in this research. This part contains two chapters. The first chapter (i.e. Chapter 6) is about a synthetic datasets. It introduces a ligature based Pashto synthetic datasets and further explains the extensions that are made to these synthetic datasets. The second chapter (i.e. Chapter 7) of this part describes the creation of real Pashto image-base. This chapter presents the diverse contents of Pashto texts and describes the protocol information about the new KPTI dataset.

The third part of this thesis is dedicated to pre-processing tasks. This part contains two chapters. Chapter 8 presents a novel method to de-skew scanned documents. This chapter explains the proposed methodology used for the detection and correction of skew angle and shows results for some state of the art methods. Chapter 9 describes the detail of line segmentation method. In this chapter, the problem of segmentation of large headings and titles are visualized via real data, and the proposed method is evaluated and compared to state of the art technique.

The fourth part of this thesis mainly covers the OCR portion. This part contains four chapters. The first chapter (i.e. Chapter 10) proposes the scale and rotation invariant OCR system. This chapter explains the related work, describes the proposed methodology and shows detailed results of scale and rotation invariant recognition system using Pashto ligature datasets. The second chapter (i.e. Chapter 11) presents the Pashto OCR and deep learning benchmark. This chapter presents the vital part of the proposed OCR system. It further describes the detail of deep learning based architecture using MDLSTM, and BLSTM models. These models provide the power and real essence of sequence classification under the domain of RNN. The third chapter (i.e. Chapter 12) is related to post-processing, and describes the issue of space anomaly in Arabic script. This chapter has a comprehensive information about space anomaly and provides detail discussions about the complexities posed by space anomaly. Further, It presents a combined approach for solving this issue.

Chapter 13 describes an end-to-end system and explains the GUI environment. This chapter illustrates the training and testing modes and explains the functionality of Integrated Development Environment (IDE) through screenshots.

Finally, Chapter 14 concludes this thesis and summarizes the contributions of the thesis. It discusses the scope and limitation of our proposed methods. It also presents some useful dimensions as future work.

Part I

Background

This chapter provides an overview of the Pashto language. The significance of Pashto language is not only based on the number of speakers but also its uniqueness for containing most of the characters that are used in other Arabic like languages. This chapter first states motivation of choosing the Pashto language that is followed by the demographic distribution of the speakers around the world. Furthermore, a complete description of the Pashto alphabets and their relatedness with other languages has been described. Also, a description of the challenges associated with Pashto particularly and Arabic like languages, in general, are provided. The thesis addresses both types of the issues individually. This chapter further provides a brief description of the related work in Pashto language recognition prior to the thesis in order to complete the overview. The chapter is comprehended with discussion section.

2.1 Motivation

Cursive scripts are known for their complex behaviors and hard nature in the domain of DIA system. Arabic script is one of the major cursive scripts that has a larger scope regarding OCR application. However, being cursive in nature Arabic script provides a lot of challenges in the field of OCR. Despite the significant research, this script still requires effective research towards mature OCR system. More than 70 different languages use this script for their written communication. Pashto is one of such languages that also use Arabic script for its written communication. The Pashto language has extra benefits concerning OCR development. The reason is, as it is evolved under the influence of many other cursive scripts in the surroundings. This evolution caused the creation of many extra characters in the Pashto language while the basic nature of Arabic script is also kept maintained. As a result, the Pashto language has a generic and larger character set, that is the superset of Arabic, Persian, and Urdu languages. Therefore, Pashto

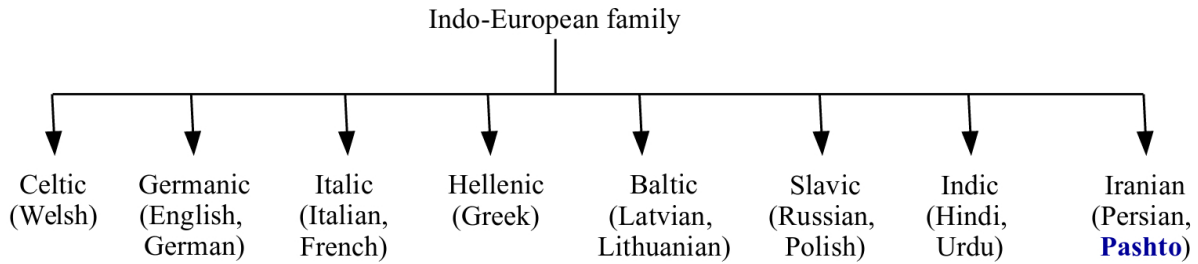


Figure 2.1: *Languages belong to Indo-European family [TR96]*

language not only possesses the inherited complexities of the Arabic script but also its additional complexities, that makes the Pashto language as a concrete case study to explore it regarding OCR application. The contents of this chapter mainly explain the Pashto language, its origin, character set, complexities, etc.

2.2 Introduction to Pashto

Pashto language is originated from Eastern Iranian language and belongs to the Indo-European family[Kai81, TR96]. Figure 2.1 shows the main languages from Indo-European family. Pashto is one of the official languages (Dari and Pashto) of Afghanistan [Ban03]. It is the second largest regional language of Pakistan. People living in the northwest and west side of Pakistan speak the Pashto language. This language is the primary language among the Pashtun diaspora all around the world. The estimate of a total number of Pashto-speakers is 45 – 60 million people worldwide [Pen55, Dav14]. Figure 2.2 shows the main regions of Pashto speakers across the world. The Pashto language has mainly two dialects, *soft* and *hard*. The *hard* dialect is also known as Pakhto [CDM03, Her82].

The Pashto language uses Arabic script for its written communication. It is cursive by nature and written from *right to left*. Only numerals are written from left to right. It is worth mentioning that the grammar and composition of the Pashto language have no links with the Arabic language. A scholar like Abdul Hai Habibi² and others believe that the first ever written book in the Pashto language is tracked back in 8th century [Hab67]. It indicates that the Pashto language has a long association with written text. Most of the written works contain poetry, history, novels, and religious contents. The Pashto is famous for its classic poetry. Some of the renowned Pashto poets and their work which have been realized internationally are Rahman Baba [Sam03]³, Khushal Khan Khatak[Mor60], Amir Hamza Shinwari[Car15] and Ghani Khan[Kha47, Kha13]. The Pashto language has a rich literary heritage that needs preservation. We need a robust

²[http : //www.alamahabibi.com/Biography_Abdul_Hai_Habibi.htm](http://www.alamahabibi.com/Biography_Abdul_Hai_Habibi.htm)

³[http : //news.bbc.co.uk/2/hi/south_asia/4273915.stm](http://news.bbc.co.uk/2/hi/south_asia/4273915.stm)



Figure 2.2: *Pashto speaking zones shown in orange color¹.*

OCR system that can process the Pashto text and converts it to required digital format.

There is another interesting fact, all the characters of Arabic and Persian languages are the subsets of Pashto character set. Further, except two characters in the Urdu language, all other Urdu's characters are also members of Pashto characters. It makes Pashto character set a generic one and could provide a solid hypothesis for transfer learning and domain adaptation due to similar visual clues.

For a better understanding of the Pashto language, next section provides an overview of the Pashto alphabets and their relation with other Arabic like languages.

2.3 Pashto Characters

The Pashto language has 44 basic characters. These characters contain all 28 characters of Arabic language and all characters of Persian language. However, in case of Urdu, all other characters exist in the Pashto language, except; ڈ [da:l], and ڑ [areɪ]. This mutual relationship is shown among these letters in Figure 2.3.

Other characters like; ٹ [t] and گ [gaf] belong to Urdu and Persian languages respectively, but with the passage of time they are now used interchangeably with ټ [t] and ک

Table 2.1: *The shapes of Pashto characters and Unicode information.*

No	Unicode (Hex)	Pashto Chars.	No	Unicode (Hex)	Pashto Chars.	No	Unicode (Hex)	Pashto Chars.
1	u+0627	ا	16	u+069a	ښ	31	u+0631	ر
2	u+0628	ب	17	u+0635	ص	32	u+0693	ړ
3	u+067e	پ	18	u+0636	ض	33	u+0632	ز
4	u+062a	ت	19	u+0637	ط	34	u+0696	ږ
5	u+067c	ټ	20	u+0638	ظ	35	u+0698	ژ
6	u+062b	ث	21	u+0639	ع	36	u+0633	س
7	u+062c	ج	22	u+063a	غ	37	u+0634	ش
8	u+0686	چ	23	u+06a9	ک	38	u+06cc	ی
9	u+062d	ح	24	u+06ab	ګ	39	u+06d0	ې
10	u+062e	خ	25	u+0644	ل	40	u+064a	ي
11	u+0681	ځ	26	u+0645	م	41	u+0641	ف
12	u+0685	ث	27	u+0648	و	42	u+0642	ق
13	u+062f	د	28	u+0646	ن	43	u+06cd	ی
14	u+0689	د	29	u+06bc	ښ	44	u+0626	ئ
15	u+0630	ذ	30	u+0647	ه			

2.4 Shape Conventions

This section defines shape conventions, which are usually related to cursive script languages. These conventions mainly include *breaker and non-breaker* characters, *ligatures*, *primary* ligatures and *secondary components*. These conventions are explained in detail in the following sub sections.

2.4.1 Breaker and Non-Breaker Characters

The concept of the breaker and non-breaker characters inherently exist in almost all languages that use Arabic script. However, Durrani et al. [DH10] referred a term as non-joiner letters. Here we propose the term "breaker-character" instead of non-joiner. The term non-joiner gives an absolute sense that these letters are not able to join. In fact, these letters could join to all others characters except non-joiner. However, they do not allow other characters to join "*after*" them⁴. Almost a small subset of breaker characters exist in all languages that use Arabic script. Mainstream languages of such

⁴ As Pashto is written from right to left. Therefore, please consider the word "after" in the context of Pashto writing direction.

آ ا د ذ ر ز ږ ړ ژ و ے ی

Figure 2.4: 13 breakers characters of Pashto language.

nature are Arabic, Persian, Urdu, Pashto, Punjabi, Sindhi, and Dari etc. However, the number of these characters vary from language to language. We can formally describe a breaker-character as a special character that once occurs inside a word, it breaks the continuity and splits the word into parts. In other words, such character comes either individually or at the end of a ligature or a word. These characters at one side, cause a calligraphic beauty of the script but on the other side, they cause anomalies as well. These anomalies are referred as space insertion and omission anomalies.

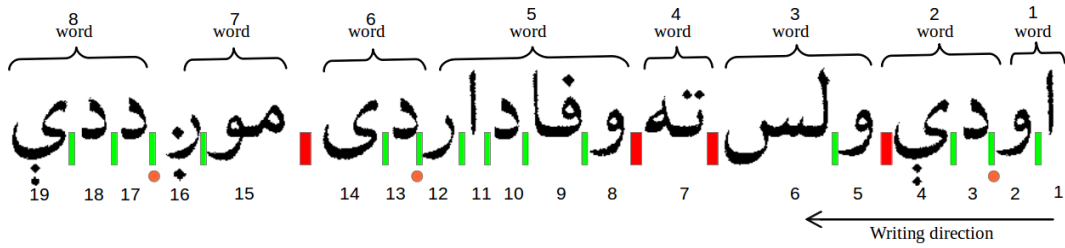


Figure 2.5: Pashto text line with 8 words and 19 ligatures. The red lines indicate the spaces caused by typing space, while the green lines indicate the spaces caused by the breaker-characters. Further, the small orange circles indicate the particular spaces caused by breaker-characters between two adjacent words.

Space Insertion and Omission

Usually, it is a good practice to have space after each word, but in Pashto and other cursive languages, this rule is nonuniformed. This nonuniformity is due to the presence of breaker-characters. Thus the usage of space has two different rules:

- (a) If word/ligature ends with non-breaker character, then the typist must provide the space (*space insertion*).
- (b) If word/ligature ends with breaker character, then the typist may or may not provide the space (*space omission*).

Figure 2.5 illustrates the space insertion and omission anomalies. It should be noted that ligatures are only formed whenever rule *b* is applied.

2.4.2 Ligatures

Any valid combination of characters that maintained in a connected form is known as a ligature. Ligatures are the most important text units in cursive languages and are one of the candidates for recognizable units. Logically, it could be a single character or a combination of characters that must end with breaker character. There is enough literature, in which ligatures are considered as recognizable units [Leh12, SS13, AAK10, Sha02, Hus02]. However, for a particular language, the number of ligatures should be known on a prior basis. Figure 2.5 shows a Pashto text line with 19 ligatures. Where ligatures Number 6, 7, 9 and 15 are the combination of two characters while the remaining all are just individual Pashto characters.

It is worth mentioning that for Arabic language majority of literature refer the ligatures as PAWs (Piece of Arabic Words) [NSBP09, AHK08]. However, in languages like Urdu and Pashto, the term ligature is also used.

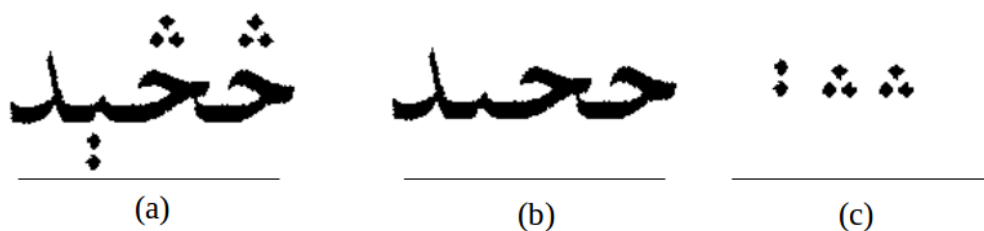


Figure 2.6: A Pashto ligature (a), primary ligature (b) and their secondary components (c).

2.4.3 Primary Ligature and Secondary Components

We can further divide a shape of a ligature into its two main parts. (i) The major connected skeleton in any ligature is known as a *primary ligature*, (ii) and all the other parts are known as *secondary parts* of a ligature, which includes dots and diacritical marks. Figure 2.6 depicts the shapes of a ligature, discrimination ligature, and its constituent secondary parts. In many cases, the primary ligatures are same, while the secondary parts play a distinctive role among the ligatures. We can group these ligatures w.r.t same primary ligatures.

2.5 Challenges

This section defines some basic terms and concepts, which are usually relating to cursive script languages. These terms and concepts are mainly associated with complexities and

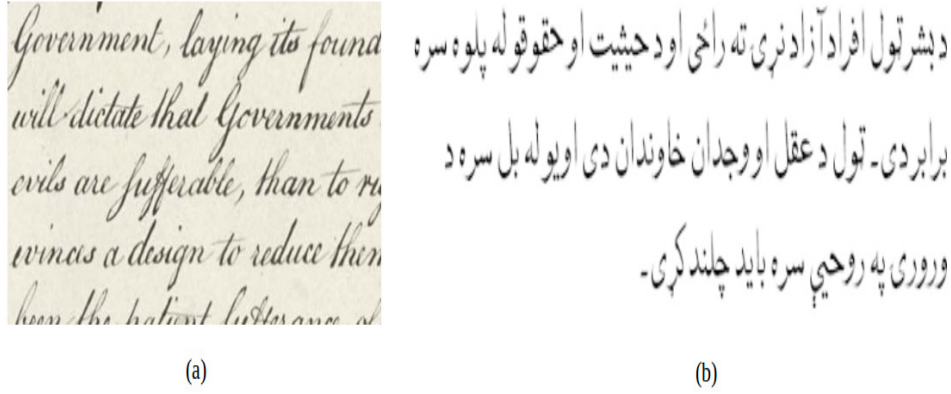


Figure 2.7: Cursiveness in English handwritten (a) and Pashto printed (b).

challenges exist in all languages that are derived from Arabic script. This thesis describes the complexities in two sub-sections. The first sub-section introduces challenges posed by Arabic script. The second sub-section is more specific and describes challenges that are only related to the Pashto language.

2.5.1 Generic Challenges

Generic challenges exist in Arabic like languages are *cursiveness*, *context dependency*, and *space-anomaly*.

Cursiveness

Any script, in which characters in a word are written in a connected form is known as cursive script [Kho02, ANR⁺16a]. Sometimes, this concept is overlapped for some languages. For example, English printed material renders in a non-cursive style. However, English handwritten text shows cursive characteristic. Figure 2.7 shows examples of some cursive scripts. However, in a case of English handwritten word, the characters are connected via cursive strokes, while in a single Pashto word, there may exist one or more connected glimpses, known as ligatures. In addition to that, languages like Arabic, Urdu, Persian, and Pashto are purely cursive in nature. Because both in handwritten as well as in printed form, they shall be written in cursive form.

Context-Dependency

In Latin script, the shapes of the characters mostly retain their salient features, and each character may have two shapes either (upper and lower case). However, in Arabic cursive script, the shape of a character changes according to its position in the word. These

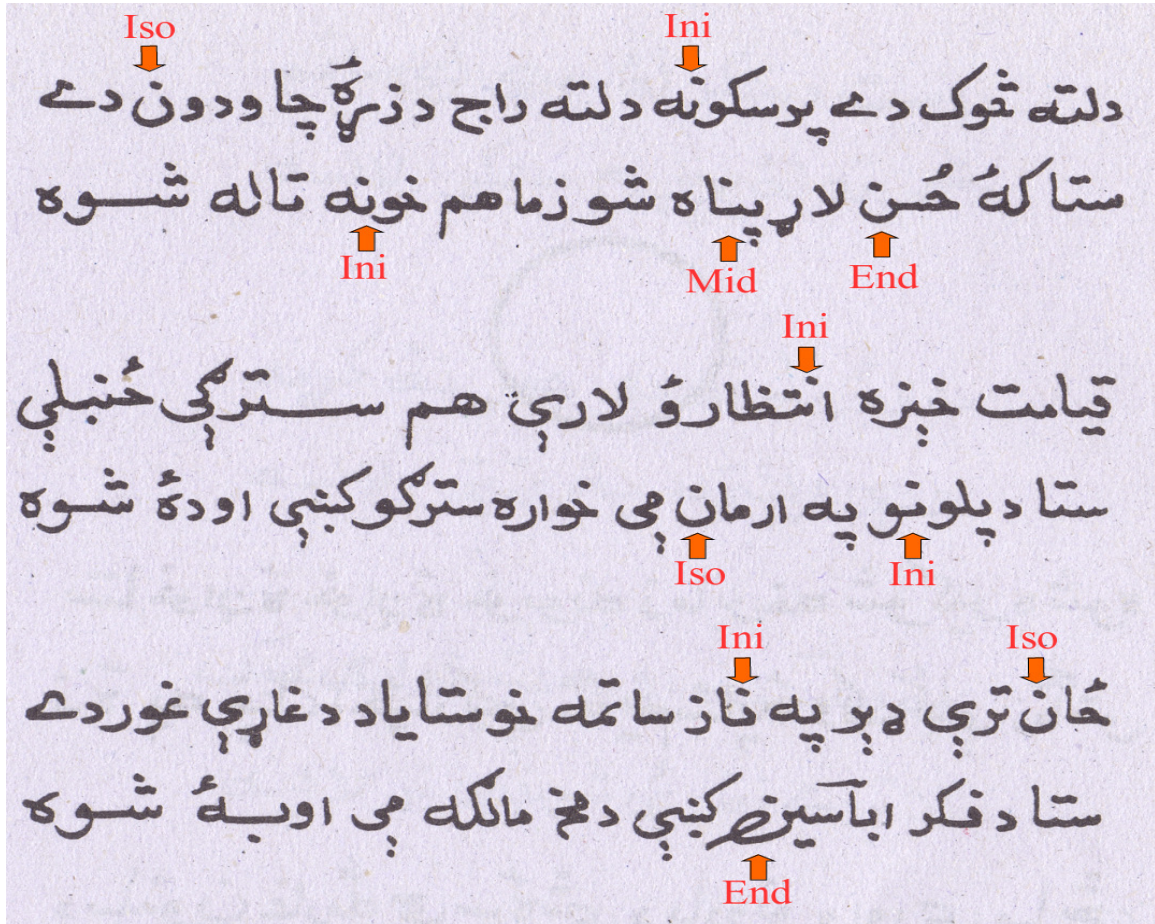


Figure 2.8: *The Pashto character ن [noon] and its four different shapes according to four different positions, that are (Iso: Isolated), (Ini: Initial), (Mid: Middle) and (End: End). The "End" shape of ن [noon] in the last text line is even completely different.*

positions are mainly (I) individual or isolated, (II) beginning or initial, (III) middle, and (IV) end. Thus, each character in a cursive script takes up to four different shapes at four different positions. Figure 2.8 shows one of such examples, where a single Pashto character ن [noon] and its four different shapes are shown in real data.

The issue of context dependency in the cursive script is already investigated in the works like [Kho02, EHLSM05, SCS⁺09]. The existing research concludes that there are two different scenarios while handling the shape variations problem. The first scenario tells that if extracted features are analytical in nature (i.e., hole, dot, curve, and line), then the individual shape of a single character needs a dedicated label for its annotation. Thus a single character may have four different labels for its annotation. The second scenario illustrates that if features are statistical in nature and are learned by a machine under a temporal classification, then assigning extra label will not produce optimal results. Ul Hassan et al. [UHAR⁺13] have proved this, as they have assigned extra labels for four different shapes per character, however, they achieved degraded performance.

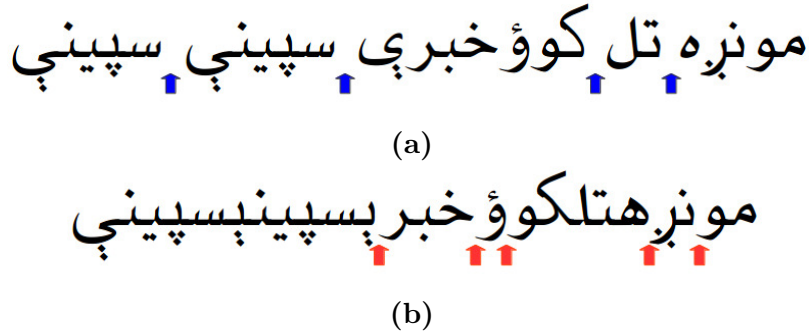


Figure 2.9: *The spaces indicated with blue arrows in (a) are due to explicit space insertion, and the implicit spaces caused by breaker-characters are indicated with red arrows (b). Removing regular spaces will result to render the given text as (b), while the spaces caused by breaker characters will remain the same.*

Space Anomaly

There is an interesting behavior of a cursive script in which some special characters do not allow another character to join them. Such characters are known as non-joiner [AAF10, GETS99] or breaker-characters. These characters break the cursiveness and produce space like impact. Therefore, in this thesis, we refer these characters as breaker characters. Due to this behavior, the conventional style of using or inserting space is affected. Intuitively, if a word (i.e. belongs to one of such languages) ends with a breaker-character, then the insertion of space is not required anymore. Unlike this, if a word ends with non-breaker character, then the use of space becomes mandatory. This mix-up rule of insertion and non-insertion of spaces causes an anomaly, known as space anomaly. Figure 2.9 shows the difference between regular spaces or mandatory insertion with blue arrows and spaces caused by breaker-characters with red arrows. Consequently, a typist should have a knowledge of the respective language that either a word is ending with a breaker or a non-breaker character. This thesis provides a comprehensive insight into this issue and provides a detailed study in Chapter 12.

Font variations

The preceding challenges are very generic in nature. However, as the art of font creation evolves for the sack of calligraphic beauty in printed text, more complex patterns emerge, that subsequently cause to pose more complex challenges to OCR. Such challenges mainly depend on specific fonts rather than a script. In addition to that, a lot of different handwriting styles also make it difficult for an OCR to sustain its accuracy. Figure 2.10 shows a single Pashto text-line with different fonts styles.

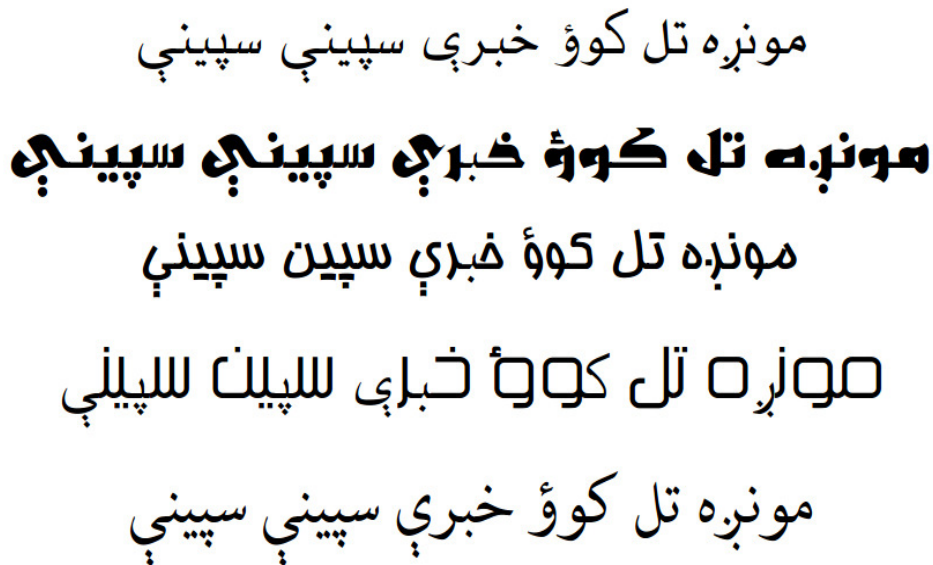


Figure 2.10: A Pashto sentence rendered with five different fonts. The different fonts styles produce an abstraction regarding shape dissimilarity.

2.5.2 Pashto Specific Challenges

OCR for the Pashto language is the main focus of this thesis. Therefore, the challenges specifically associated with the Pashto language are of more interest. For better understanding, we have categorized them into two categories. The first category describes the challenges which are associated with the Pashto characters shapes. The second category describes the challenges which are presented in real Pashto text (i.e. KPTI). In this chapter, we only describe the first category, while Chapter 7 discusses the second category.

Pashto ی [Yey]s

As discussed earlier, the Pashto has a generic and marginally larger character set. The reason is that, with the passage of time, the Pashto writing style is influenced by other languages in the neighborhood. Another, Islam is the main religion of Pashto speakers, and the religious contents are mostly in Arabic script. It caused to evolve many other shapes for some certain Pashto characters. One such example is the variants of ی [Yey]. In the Pashto language, the ی [Yey] has *five* different variants and are frequently used in all type of Pashto text. Figure 2.11 depicts these variants of ی [Yey]. The ی can also be written as ی along with [hamza] above it. It can be observed that not only the shape but also the sound of such variants are very much similar to each other. Therefore, the classification error related to Pashto ی is always one of the candidates in top 20



Figure 2.11: *The Pashto ى [Yey]s and their five different variants.*

confusions.

More Confusion in Characters (د with و and ء and ه; and vice versa)

In addition to Pashto's ى [Yey]s, there are other characters like د, [Dal] و [Waw] and similarly, ء, [Aey] ه [Hey]. They also closely resemble each other and therefore, the recognition system mostly confused them. Although the misclassification in the Pashto's ى [Yey]s could be rectified to some extent using the context information, these characters are subtle and frequently occur in the same context. Therefore, it is a difficult challenge compared to Pashto's ى [Yey]s. The confusion regarding these characters (i.e. د و ء ه) is usually encountered in the top 10.

Interchangeably used Characters

In addition to shape similarity, there are some characters in the Pashto language, which are often used interchangeably. These interchangeable characters cause to extend the target classes for a classifier. The reason is their dedicated Unicode. Consequently, this interchangeable behavior creates hurdles in Pashto OCR. The language itself treats these characters as same, while the classifier deals them completely different. There are no such discrete rules, which could help us in determining that where these particular characters will be used for each other. However, the context information could be used to learn such interchangeable characters. Such characters which are used interchangeably in the Pashto language are given below:

- ك as ك
- گ as گ
- ٹ as ٹ
- In informal texts, ى as well as ى, and ى are sometimes used as the letter ى.

In order to overcome the above mentioned challenges in Pashto language, there are many related works that tried to solve these problems. However, none of the work provides

an end-to-end pipeline for Pashto language recognition. The next section provides an overview of the current state of the art for recognition of Pashto language.

2.6 Related Work

Existing research is quite limited regarding Pashto OCR systems and reports only one reputed work with a reference of BBN Byblos OCR system[DMN04]. The BBN Byblos OCR system is a holistic approach and does not require segmentation during training and testing. It is based on HMM and uses 14 states left to right HMMs. They have used different datasets. However, the datasets are not available publicly.

Another, Wahab et. al. [WAA09] presented a synthetic dataset, having 1000 unique Pashto ligatures. In this thesis, we have extended this dataset for invariant recognition of scale and rotation variations. According to the best of our knowledge, existing research reports only these two references [DMN04, WAA09] that address Pashto OCR. Analyzing these two works, we can conclude that the BBN Byblos OCR system for the Pashto language is the only effort made towards OCR system. The character and word error rates were reported ranging from 2.1% to 26.3%, and 7.1% to 52.3% respectively, which demonstrates the level of complexities and abstraction towards another hand OCR system. On another hand, the work done by [WAA09] is mainly related to the creation of synthetic dataset.

Although the existing research is limited regarding Pashto OCR system, the research on Arabic script can benefit Pashto OCR system. Therefore, in this thesis, Arabic recognition systems are taken as baselines and improvements are made to baseline systems to facilitate Pashto OCR. However, OCR development regarding Arabic scripts itself has been going through intermediate phases and still requires more research to maturity.

2.7 Discussion

In this chapter, we have comprehensively discussed the background information about the Pashto language. This background information includes the origin of Pashto language, Pashto as a cursive script, Pashto's character set, relationship with other Arabic like languages, generic and specific challenges. Another, the related work regarding Pashto OCR system is also summarized.

The important things that are discussed in this chapter are the evolution of the Pashto language, its relationship to Indo-European languages under Iranian sect, information

about current regions around the world where it is spoken, and information about characters and their shapes. More than 50 million people around the world are associated with the Pashto language. The writing style is borrowed from Arabic script, and it is written from right to left using cursive strokes. The existing research is less significant, and could not be generalized for Pashto OCR system.

In this chapter, we have also presented a detailed discussion about the Pashto characters set. It is mentioned, that the character set of the Pashto language is the superset of Arabic and Persian languages. In addition to this 36 out of 38 characters of Urdu language are also available in the Pashto character set. That makes the Pashto character set as generic.

This chapter also explains the generic as well as specific challenges posed by cursive scripts and Pashto text respectively. Generic challenges are discussed in detail, which includes cursiveness, context dependency, and space anomaly. Similarly, in this chapter we have also introduced Pashto specific challenges, these challenges include variants for the Pashto's ع and some of the Pashto characters used as interchangeably. The both generic as well as the Pashto specific challenges make it hard to develop OCR system for the Pashto language.

The next chapter will cover the conceptual background of Neural Networks (NNs). The important topics that are explained in a brief manner are sequence classification, Recurrent Neural Networks (RNNs), and Deep Learning (DL).

Neural Networks and Deep Learning

This chapter presents the very core concepts about Neural Networks (NNs) along with their mathematical foundation. It explains these concepts in the context of sequence classification under supervised learning domain. Further, this chapter links these basics with Deep Learning (DL) and explains the importance of these concepts in the research area of DIA.

3.1 Introduction to Neural Networks

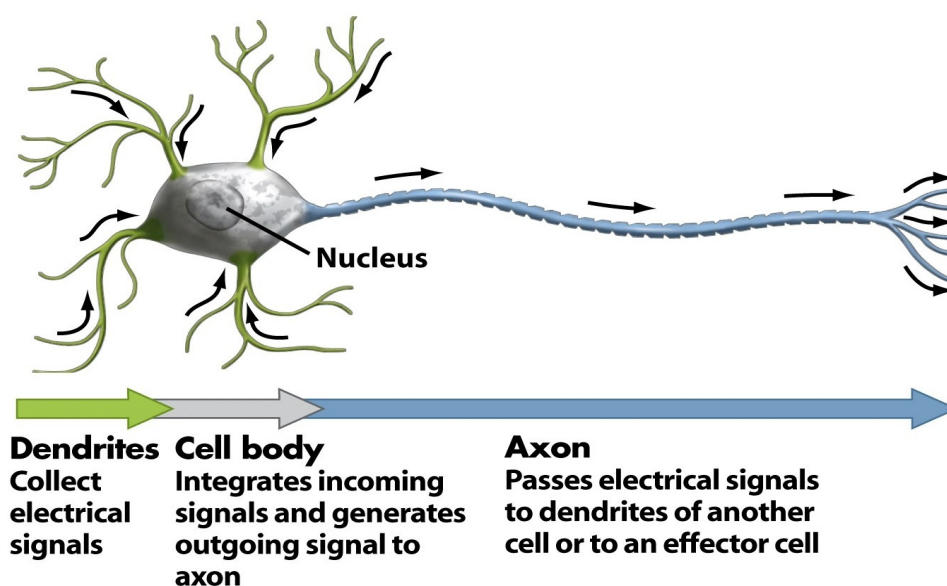


Figure 3.1: *Information flow via biological neuron*,¹.

The conceptual formulation of Neural Networks (NN)s needs to have some basic knowledge about Biological Neurons. The reason is the discovery of NNs from an inspiration of Biological Neurons. Biological Neurons are the core units of the brain. Figure 3.1 shows

¹<http://biology-pictures.blogspot.de/2011/12/information-through-neuron.html>

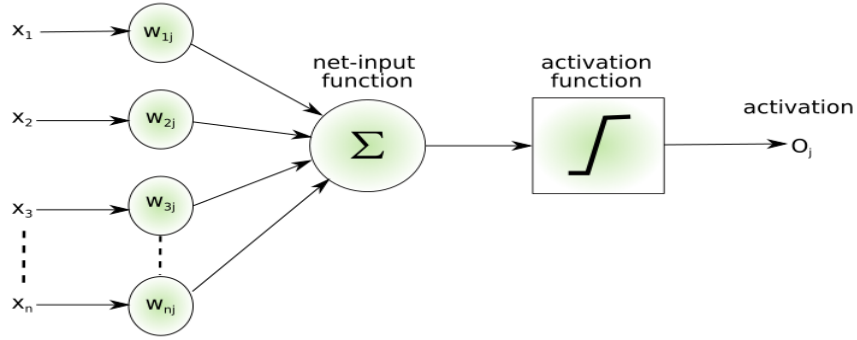


Figure 3.2: Artificial Neuron (AN): Input $X : \{x_1, x_2, x_3, \dots, x_n\}$, and weights $W : \{w_1, w_2, w_3, \dots, w_n\}$, are multiplied and the results are Σ summed, such that if the accumulative response is higher than a threshold θ , the AN outputs 1 otherwise 0.

a simple structure of a biological neuron. It takes inputs via dendrites, integrates them in the nucleus body, and generates output via axon. The axon usually ends with one or more dendrites to link another neuron. In fact, the function of a biological neuron can be imitated by an artificial neuron (AN). An artificial neuron (AN) likely performs as biological neuron by using mathematical formulation[Now06]. Figure 3.2 illustrates the mathematical formulation of AN. The AN takes the input x and path that takes input has a weight w . The product of $x_i \cdot w_i$ is computed and then summed by the summation Σ or net-input function, and finally, the activation function activates the output. Activation function usually uses a sigmoid function where it checks the accumulative response of a summation unit. The sigmoid function either signals 1 or 0 depends on the value of the threshold [Ant01].

To learn and classify an object using ANN, it needs more and more neurons. Usually, we are interested in the number of observations i.e. the number of inputs $|X|$, and the number of output i.e. classes. It mainly defines two layers, an input-layer, and an output-layer. However, to handle abstraction in features and to classify complex patterns, a hidden layer is required. The size of the hidden layer may be one or more, depending on how fuzzy things are to be learned.

As this thesis also explores the DIA system for the Pashto language under supervised learning domain, therefore next section introduces supervised learning and its association with ANNs.

3.2 Supervised Learning

Supervised learning is one of the machine learning tasks. The terms *supervised learning* imitate the very basic instinct of human learning behavior. In supervised learning, entities are labeled with the help of a learned supervisor. We can formally describe this; let suppose there is an input space X and an output space Y , such that for each X_i there is Y_i as the corresponding label. If P is a distribution over $X \times Y$, then in general from distribution P , there exist two datasets as training set and test set respectively. Let Tr is the training set.

$$Tr = (X_i, Y_i)_{i=1}^n \quad (3.1)$$

Thus supervised learning is the process of finding a function f , which is known as classifier from the family of functions $F = \{f : X \rightarrow Y\}$, such that the desired classifier f , can classify, $X \rightarrow Y$. In order to quantify the performance of a classifier, the loss function is used. The loss function imposes a penalty when X is not correctly mapped with its corresponding Y . The total error T_{err} during training can be calculated via loss function.

$$T_{err} \equiv Tr_{(X,Y)}[L(f(x) \rightarrow y)] \quad (3.2)$$

Thus, supervised learning can be described as the recursive process of finding an optimal classifier f' , such that it gives the global minimum value for T_{err} over the training set.

$$f' = \underset{f}{\operatorname{argmin}}(T_{err}) \quad (3.3)$$

During the training process, it is possible to train a classifier with the help of a complex function, such that it could ideally fit overall training data. However, such models are very sensitive to noise and could not be generalized well for similar but unseen data. Such behavior is known as overfitting. Overfitting can be solved by taking a small portion from the main training set as a validation set. The validation set is used to check the learning behavior of a classifier during the training process. The validation process is achieved by regularization, where learning is maintained by the classifier only if it could improve or retain the accuracy on the validation set.

We can use Neural classifiers in handling different classification tasks. However, here in this thesis, we are more interested in sequence classification. In general, a sequence is an ordered list with a finite number of events. We can represent these events as symbolic values i.e. a numerical value, a vector, or any complex data type [XPK10]. The classification of such sequences is one of the important pattern recognition (PR) tasks. Next Section briefly introduces the most relevant types of classifications including

sequence classification, segment classification, and temporal classification.

3.2.1 Sequence Classification

Sequence classification is the most restrictive case in which an input sequence is classified as a single label. For example, classification of a single face and recognition of an individual character. In this type of classification, an entire sequence is processed before its classification. Input sequences with fixed-length size is another requirement of this classification. Therefore, sequences with fixed-lengths are used, or they are easily normalized to fixed-length by zero-padding technique. A well-known example of fixed-length sequences is MINST dataset containing individual digits [LBBH98]. In this thesis, we used this type of classification in Chapter 10, where each Pashto ligature is classified as a single class.

In sequence classification, although fixed-length sequences are classified into a single, the sequential algorithms are proven to be more effective and beneficial. The main reason is their adaptability to learn registration, translation, and distortion issues. This point is validated in Chapter 10 by using sequence classifiers for the recognition of scaled and rotated Pashto ligatures.

3.2.2 Segment Classification

It is often required to classify a single input sequence in multiple class labels. For example, classification of each character in a line of handwriting, or a phoneme in a spoken sentence. If an input sequence has a prior information regarding the positions for each constituent segment and we need to classify such input sequence, then such classification is known as *segment-classification*. The utilization of the context is the distinctive feature of segment classification, which does not exist in sequence classification. However, the classification algorithms that are designed to process a single input have problems in segment classification. The reason is the complexity of feeding the other segments. In other words, a simultaneous access is difficult to other segments in the neighborhood. An approach like *time-window* can solve this problem by collecting the context data on either side as an input pattern. However, time-window approach suffers due to the varying size of segments, as it is unknown in general and determining the size of time-window has always depended on the problem domain.

We can use segment classification in many cases, speech recognition is the one case, where each acoustic frame is considered as a separate segment and is known as *framewise phoneme classification*. Another example is the image segmentation where segmenta-

tion of each pixel or a block of pixels is considered as segments. Therefore, segment classification is mostly used in page/image segmentation problems.

3.2.3 Temporal Classification

In classification tasks mentioned above, we have some assumptions. For example, in sequence-classification, input segment should be classified as a single class, and its size should be of fixed-length. Similarly, in segment-classification, for each segment, the position should be known in advance. There is another classification known as *temporal-classification*, which is more relaxed compared to the sequence and segment classifications. In temporal-classification, we achieve multiclass classification for an input sequence of a variable-length. The more distinctive feature of temporal classification is that we do not care about the position of segments. The classifier/system implicitly aligns the target labels corresponding to the input sequence. However, there is an assumption that the length of a target sequence should be equal or less than the input/sequence.

In OCRs, a text-line is a simple example of a sequence with ordered characters as events. However, in this case, it is more desirable to predict the sequence of characters. This case is a typical example of temporal classification and is also known as *sequence-to-sequence* classification [KS05]. Let suppose a sequence s has n characters in an order $\{l_1, l_2, l_2, \dots, l_n\}$, and L is the set of characters. Then our desired classifier f , such that $f: l_i \rightarrow c, c \in L$ in same order [Kad02].

In this thesis, we also deal our problem under temporal classification, where an input image will be a text-line, and its corresponding ground truth will be a sequence of ordered characters. In Chapter 11, we classified Pashto text-lines under temporal-classification. There exist many techniques that are used to handle temporal-classification however, connectionist temporal classification (CTC) (cf. Section 3.4) is the more important one.

In addition to that, other methods that could be used for handling sequence classification problems in general include *K nearest neighbour* (KNN), *support vector machine* (SVM)[LSST⁺02, LS05], and *hidden Markov model* (HMM)[DEKM98]. However, *Recurrent neural networks* (RNN) based methods are the most effective and dominantly used techniques regarding OCR's research[SVL14, Gra12c]. The RNNs are briefly explained in the following section.

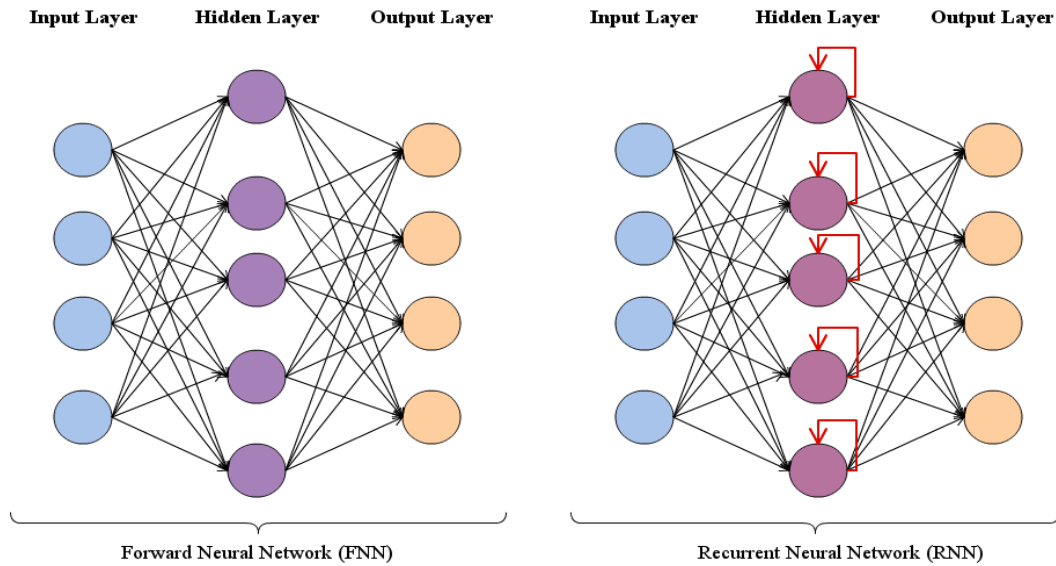


Figure 3.3: *FNN vs RNN: The red lines as circles in RNN make the difference and provide the output of previous time step to hidden layer/s an input for next time step. These cycles cause to utilize the context information in learning.*

3.3 Recurrent Neural Networks

To understand a Recurrent Neural Networks (RNNs), first, we need to explain a simple neural network. As mentioned that a typical ANN has three basic layers like input, hidden, and output layers. The ANN, where an input layer is fully connected to hidden layer and hidden layer is fully connected to the output layer, such that there is no connection which makes a cycle, is known as Feed Forward or Forward Neural Network (FNN). Unlike this, if there are fully connected layers, plus there is also a cycle, such that the output of the hidden layer/s is used as input to the same hidden layer for the next time step, then such ANN is known as Recurrent Neural Network (RNN). A simple architecture of ANN and RNN are shown in Figure 3.3, where the red lines show the difference between FNN and RNN.

The RNNs are powerful as they are capable of exploiting the context information in the vicinity. RNNs achieve context learning by providing the output of the previous observations to the next observations. Due to this characteristic, the research community dominantly uses the RNNs for sequence classification tasks [Pea95]. However, there is a problem in a conventional RNN. For example, an observation that frequently appears in the sequence, the weights of the concern neurons tend to explode during training. Unlike this, an observation that rarely appears after long duration in the sequence, the gradients for that observation tend to vanish after some time. Aforementioned problem is known as *vanishing gradient problem* [Hoc98]. The exploding and vanishing gradient problems can

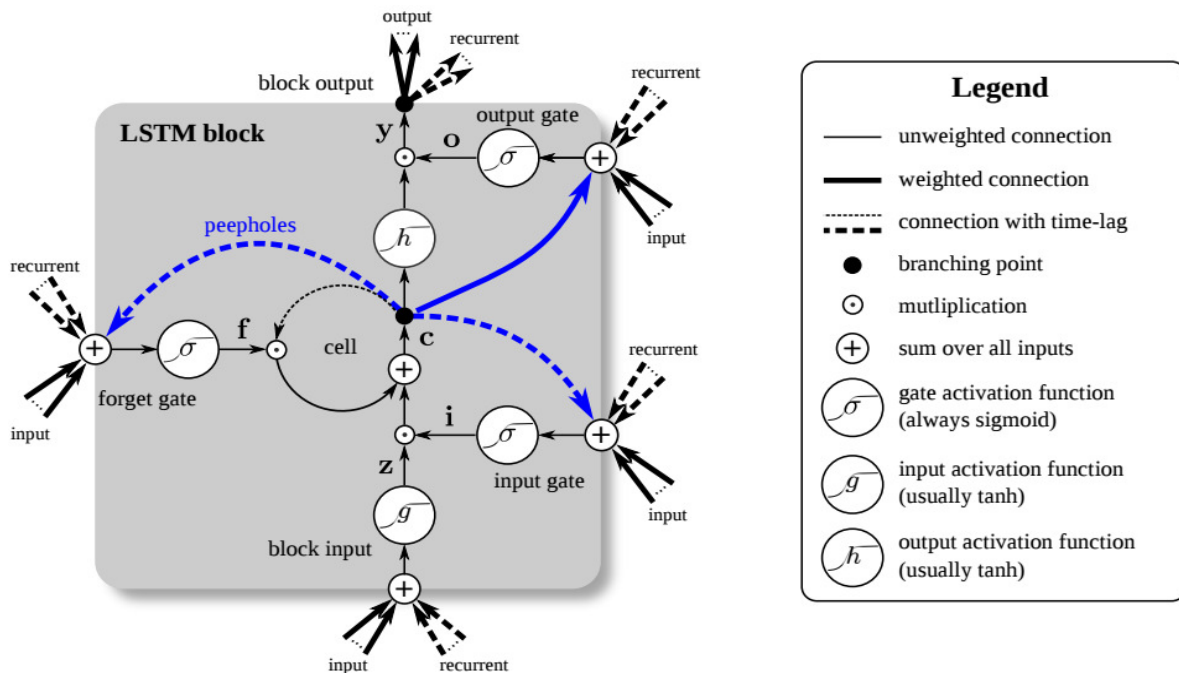


Figure 3.4: *LSTM Block*, [GSK⁺ 16]

be handled by an activation function known as Rectifier Linear Unit (ReLU). Equation 3.4 describes the ReLU function.

$$f(x) = \max(0, x) \quad (3.4)$$

But, the use of ReLU as an activation function does not guarantee that the RNN could now better learn the classes using the context observations. The most effective approach for handling these problems is the Long Short-Term Memory (LSTM) architecture [HS97]. Next Section further explains the LSTM units in more detail.

3.3.1 Long Short Term Memory (LSTM) units

The LSTM architecture consists of LSTM cells, where each LSTM cell has a capability to memorize its best learning weight using observations in the context. The LSTM cell can be thought of as a memory chip and can handle read, write, and reset operations during training. Figure 3.4 shows a typical LSTM cell. The induction of multiplicative gates like forget, input and output gates, makes it possible for a network to decide when to reset when to read and when to write the output. For example, as long as the input gate remains closed, the previously learned values could be used after a long time in the sequence. The usage of LSTM based architectures in pattern recognition is very common. However, in a case of sequence classification, the performance of LSTM networks is

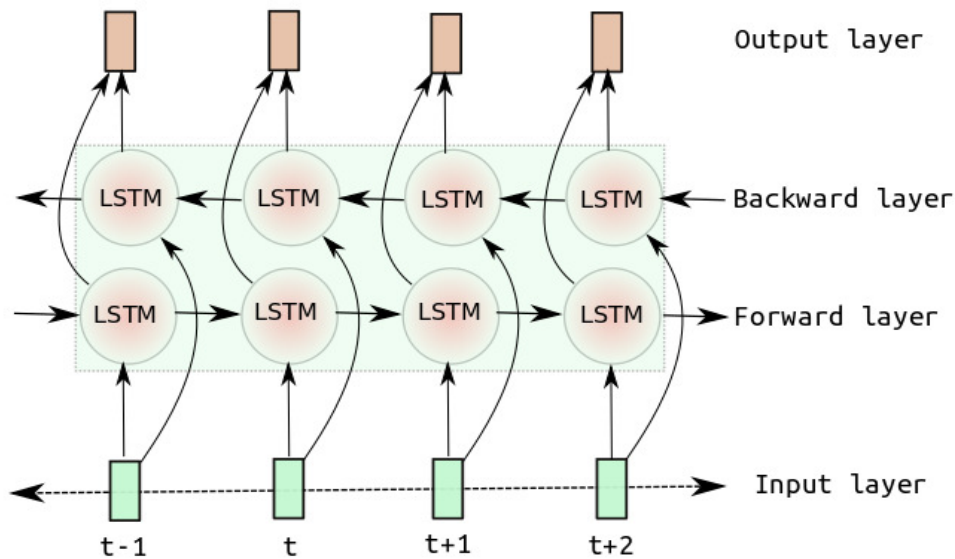


Figure 3.5: *Bidirectional LSTM with "Backward" and "Forward" layers in hidden layer of the network.*

phenomenal. This thesis uses the LSTM based architectures and proposes the Pashto OCR system realizing the most prominent variants like Bi-Directional LSTM (BLSTM) and Multi-Dimensional LSTM (MDLSTM). The following Sections briefly describe the BLSTM and MDLSTM architectures.

3.3.2 Bi-Directional LSTM (BLSTM)

Simple RNNs based models can only access the context information from the *previous* step. However, in OCR, an instance of a single character has a dependency on a previous as well as on a next character. To get benefits not only from previous time step (past) but also from *next* time step (future), RNN can be modified in such a way, that at time step t , the network has access to time steps $t-1$ and $t+1$. The networks that are capable of scanning input in both forward and backward directions are known as Bidirectional RNNs (BRNN). Similarly, the BRNN that contains LSTM blocks as neural units is called Bi-Directional LSTM (BLSTM) [GFS05, FGS08]. This bidirectionality is achieved by introducing two layers in the hidden layer, such that, one layer "*forward layer*" takes an input sequence in a forward (normal) direction, while the second one "*backward layer*" takes the same sequence in a reverse direction. These forward and backward layers are connected with input as well as output layers of the network. The typical example of BLSTM architecture is shown in Figure 3.5.

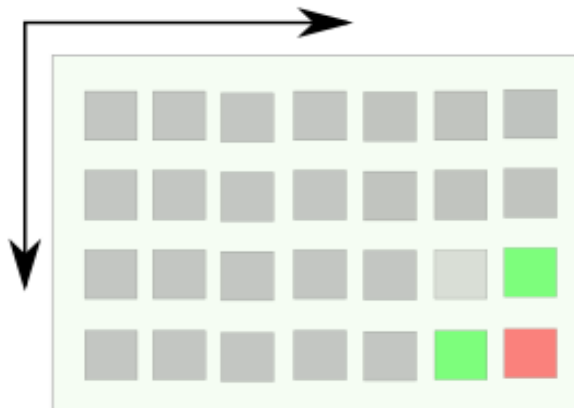


Figure 3.6: *The current pixel (red), and its immediate available context are green, while all the scanned context is in grey color. This is handled by a dedicated layer, that scans from Left-top.*

3.3.3 Multi-Dimensional LSTM (MDLSTM)

In 1-Dimensional data, contextual information can be collected either from past or future, and it can be achieved by BRNNs. However, it is also possible to get an advantage of the context information from all direction. For 1-Dimensional data, we require two special layers (backward and forward) to scan the sequence in both directions. If we want to scan an input in d dimensions, then we require 2^d layers in the hidden layer. Thus for 2D data, we require 4 extra layers, in which each one is dedicated to a certain direction. Suppose, if a network has to fetch a pixel value from a 2D image I , such that $I(i, j)$, then a Multi-Dimensional Network can access to $I(i-1, j)$ and $I(i, j-1)$. It is illustrated in Figure 3.6. Further, scanning 2D data in its 4 directions, where each dedicated LSTM layer can fetch data in a certain direction is also shown in Figure 3.7.

The MDLSTM based networks are more powerful as they exploit the context dependency in all directions. In our case, MDLSTM gives us better results compared to BLSTM. Because in the Pashto text the characters not only have a dependency on left and right characters but also have a vertical dependency due to secondary components.

Here it is worth mentioning, that conventional LSTM based RNNs require input in a segmented form. They need pre-segmented data, which correspond to the target label. It is one of the pre-requisite to train RNNs architectures. However, text-lines written in cursive strokes present segmentation as one of the hard challenges. To avoid segmentation and to treat data as a whole (i.e. under temporal-classification), we need a specialized method. This is solved by Connectionist Temporal Classification (CTC) [GFGS06, Gra12a].

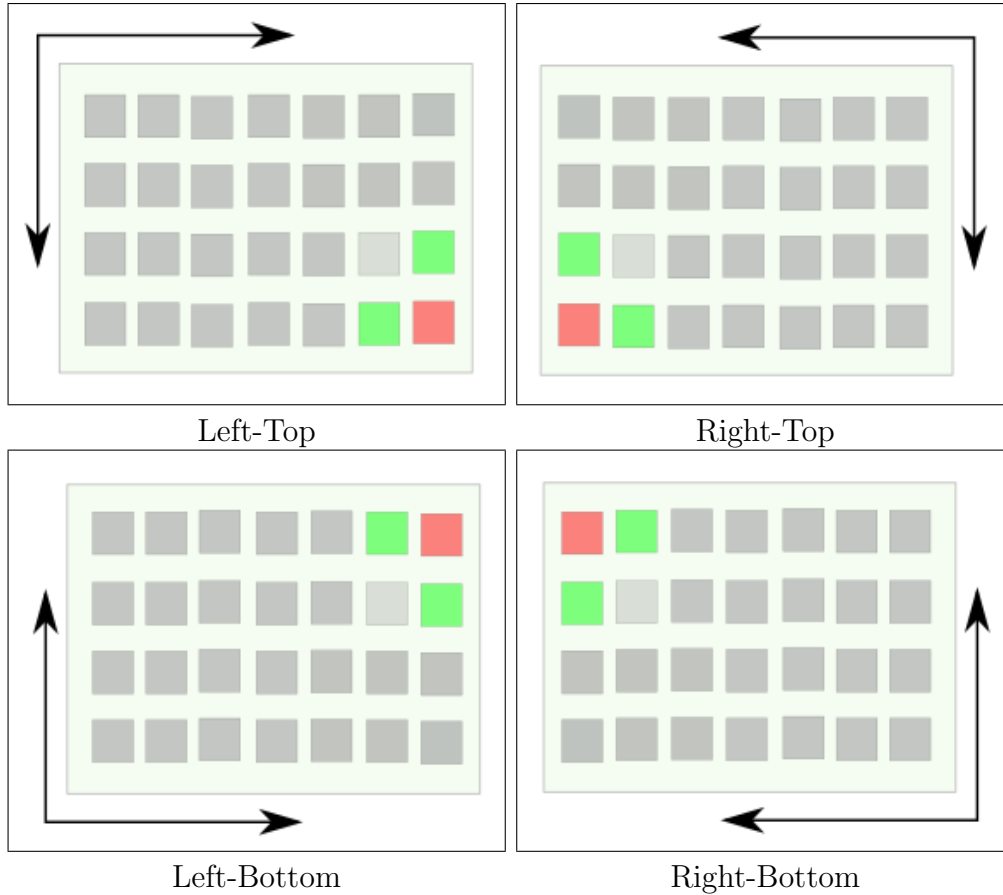


Figure 3.7: *Scanning 2D data in all 4 directions.*

3.4 Connectionist Temporal Classification (CTC)

The CTC rules out the need for pre-segmented data and allows the network to be trained for sequence labeling. This implicit segmentation is achieved by finding the most probable path that has the labels alignment nearly in the same sequence as given in the target sequence. The CTC works as a final layer and fits in almost all variants of RNNs. The performance of CTC in handwritten as well as in speech recognition is exceptional and outperforms state-of-the-art approaches in the domain of sequence classification [Gra12a]. This thesis uses the LSTM and CTC combination for sequence classification, and therefore, avoids the need of pre-segmented data. In next section related work is briefly explained, i.e., mainly done in OCR field using LSTM architecture.

In addition to that, LSTM based architectures are also integrable with deep learning paradigm. It makes the LSTM based models as cutting edge techniques that are capable of exploiting the abstraction in more complex data. Next Section briefly introduces deep learning.

3.5 Deep Learning

Deep learning (DL) has revolutionized the field of artificial intelligence (AI). The other terms like hierarchical machine learning and deep machine learning are also used. In DL approaches, the observations are generated from the distributed data under an assumption, that the observed data can be generated by the interaction of factors organized in layers. Conceptually, these factorized-layers correspond to the level of abstraction that might exist in the data[BCV13]. The DL based architectures learn in a hierarchical manner, where the abstract level concepts are learned from the low-level ones, and each layer detaches one level of abstraction from the data, and let the network to choose the features which are useful for learning.

The LSTM based architectures are also one of the deep learning approaches [Sch15, BAB14]. Therefore, LSTM has the power to learn the abstract level features from the low-level patterns.

3.6 Discussion

This chapter shares basic concepts about supervised learning and sequence classification. These concepts are explained and formulated under the assumption of our problem domain. The types of temporal classification i.e. sequence classification and strong sequence classifications are briefly described. A biological neuron is briefly introduced, and it is also described that how this biological neuron could be formulated as artificial neuron AN. This chapter presents the basic concepts about a simple ANN and RNN. Further, it discusses the limitations of RNN and describes how they are tackled using LSTM units. In addition to that, this chapter presents the breakthrough of LSTM based architectures and discusses their advancement in many sequential problems. Similarly, this chapter describes the emergence of CTC and summarizes its importance in the domain of sequence labeling.

Finally, this chapter explains deep learning (DL) along with its core concepts. It is discussed, that how the depth of neural networks is associated with the level of abstraction in the data. We presented a related work that includes only LSTM based approaches used for the recognition of Arabic like scripts.

Part II

Benchmark Pashto Language

4.1 Motivation

Languages using Arabic-script are considered to be very challenging for atomic segmentation. The reason is its cursive nature. Therefore, most of the researchers avoid segmentation based approaches and prefer holistic approaches for OCR. In last decade, holistic approaches have gained significant attention due to high accuracy. In holistic approaches, ligatures are preferable units for the recognition, because they attain a connected shape in most cases. However, for a particular language, the total number of ligatures defines scalability issue. Because, as large as the ligatures set, more it is difficult to train the system. However, most frequent ligatures used in a language are also limited, and therefore, ligatures are still one of the candidates to be considered as recognizable units. Especially, for the limited domains, like city names, bank names, etc. Furthermore, ligatures can be used more efficiently by finding their primary ligatures.

In this chapter, we present a statistical analysis regarding the choices of recognizable units in Pashto cursive script. The finding and outcomes will ultimately help the researchers in the development of Pashto OCR system. There is enough similar research for other languages like Arabic and Urdu. The most relevant are referred here [PM13, Leh12].

In addition to above facts, it is also important to explore a new language in terms of their frequently used words and how many ligatures could contribute these words. The outcomes of this work will not only be used in OCR application, but obviously, they can be used for linguistics analysis, and for speech recognition as well. In order to conduct statistical analysis, we need Pashto text data. In general Pashto words can be obtained by two convenient methods; (1) to extract Pashto words from a reputed Pashto digital dictionary and (2) to extract Pashto words from Pashto specific web sources. We adapted the second approach because we could not find any authentic digital Pashto dictionary which covers the overall Pashto words. A digital dictionary is found, which contains only

limited (i.e., 1002) Pashto words in its .mdb file¹, while the words are also written in Latin letters. Another, the dictionary could only provide limited information regarding the unique Pashto words instead of their frequencies. Therefore, we extracted the Pashto text data from Pashto based websites.

The analysis has been made on a text corpus that is extracted from 23 different websites. Nowadays, web sources are the most presentable sources which contain enough text material. Such material dynamically changes on a daily basis and therefore provides sufficient variation in a text. Especially when the text that is taken from those websites which are designed to broadcast instant news. In general, such websites contain enough text of diverse nature. This diverse nature of text ensures that the data is unbiased. Therefore, the selection of these web-sources is keenly done ensuring diversity and unbiasedness. These web-sources are filtered only with Pashto characters and numerals. After this filtration, about 2,313,736 words are extracted (in the remaining text in this chapter, we will refer this as 2.3 Million). Then these words are checked for unique words, and about 82,409 unique Pashto words are in the Pashto language. Further, these unique words are split into their constituent ligatures, and about 19,268 unique ligatures are found. Interestingly, the analysis states that only 7,000 ligatures are contributing in 91% of the entire Pashto words. We have also explored the primary ligatures in Pashto script and found 7,681 primary ligatures, that are sufficient to describe the entire Pashto text with some appropriate strategies.

It is worth mentioning that majority of the contents in this chapter are mainly taken from [AAR⁺15b].

4.2 Pashto Text Acquisition

Sufficient data is required to find the statistics about how many words, their constituent ligatures, and primary ligatures exist in the Pashto language. The most convenient method is to crawl different web sources for publicly available text. For this purpose, we have extracted the Pashto text from 23 web-sources. Based on diverse contents the web sources are selected. The contents mainly represent politics, religion, current affairs, sports, poetry, literature, music, and education (science and technology). There are some web sources, aiming to broadcast news and therefore, their contents are frequently changing due to new events, example of such web sources are www.bbc.co.uk/pashto, www.tatobaynews.com, and www.tolafghan.com. In this work, we mainly relied on such web sources, that might influence the extracted data to be unbiased. Table 4.1 shows these web-sources and their corresponding extracted lines and words. Although we have

¹<http://www.yorku.ca/twainweb/troberts/pashto/pashlex1.html>

extracted some reasonable data from the mentioned web-sources, Pashto text based web-sites are very limited compared to other languages like Urdu and Arabic. In the next section, we will explain how the text is extracted from these web-sources.

Table 4.1: *Pashto text-based websites and their corresponding extracted text statistics.*

SNo	Website url	Lines	Words
1	www.tatobaynews.com	11424	202020
2	www.larawbar.net	1976	42619
3	www.khpalapashtu.com	294	1339
4	www.bakhtarnews.com.af	94	4503
5	www.rohi.af	27494	204592
6	www.afghanpost.com	358	1376
7	www.taand.com	28712	152780
8	www.afghanembassy.net	90	2692
9	www.khabarial.com	22186	214431
10	www.gulamkhan.blogspot.de	1748	14684
11	www.pajhwok.com	17555	83770
12	www.pashto.sputniknews.com	12365	81510
13	www.khyber.org	5508	32092
14	www.pushtutarany.wordpress	149870	311616
15	www.sporghay.com	6609	129035
16	www.lekwal.com	6332	86208
17	www.pashtoislamway.blogspot.de	14237	127409
18	www.pa.azadiradio.org	10266	160325
19	www.bbc.co.uk/pashto	2908	40705
20	www.tolafghan.com	18655	211431
21	www.salaamtolana.org	1296	18259
22	www.dw.de	1563	13549
23	rashad.benawa.com	25032	263711
Total		366572	2313736

4.2.1 Text Acquisition Approach

The text has been extracted by using the Python library named Beautiful Soup². A python based script is written particularly for this purpose. The module takes a URL as an input argument and returns a text file, which contains text data.

Extracted text is filtered by only Pashto characters and numerals. In addition to filtering the text is also split into words. The extraction of Pashto words is made by splitting the Pashto text by spaces. However, complexity due to breaker characters has been faced in the real sense. Because, where the typist had never entered the space/s between two

²Beautiful Soup sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree, Source link: <https://pypi.python.org/pypi/beautifulsoup4/4.3.2>

Table 4.2: *Statistics of words and ligatures.*

	Words	Ligatures
Total	2,313,736	286,628
Unique	82,409	19,268

words (see case (b) mentioned in Section 2.4.1), then simply splitting the text into spaces will not work. Although we could not reach an automated solution to this issue, we use an assumption; such that if a word still has more than 15 characters, then it is a potential candidate for further manual checking. Due to this manual checking, if a source word is formed by the combination of two or more words, then it is split accordingly.

4.3 Pashto Text Statistics

A web corpus of 2.3 million Pashto words is considered for the analysis of ligatures and primary ligatures. First, unique Pashto words are extracted, and then the total ligatures are extracted, which constitute the entire unique words for the selected Pashto text corpus. Table 4.2 reports total and unique Pashto words and ligatures respectively. Similarly, Table 4.3 shows the frequencies of Pashto words in 2.3 million words. Second, each unique word is split into their corresponding ligatures. However, before showing the statistics about ligature, it is important to know how ligatures are extracted from Pashto words. A procedure is discussed in the next section, that describes the splitting of Pashto words into their constituent ligatures.

Table 4.3: *Unique Pashto words and their frequencies.*

Number of words	Frequency	% in Corpus
30	593,458	25%
100	826,420	35%
500	1,299,058	56%
1,000	1,528,272	66%
2,000	1,760,491	76%
5,000	2,006,926	86%
14,000	2,168,755	93%
82,409	2,313,736	100%

4.3.1 Pashto Ligatures Extraction

The extraction of the ligatures is achieved by utilizing the presence of breaker-characters. It is now clear that the word segments are now without spaces, and only the implicit

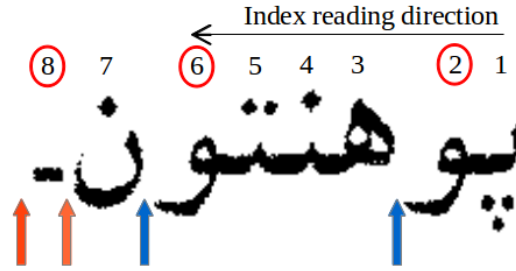


Figure 4.1: *A Pashto word has 7 characters and one full stop -, which constitute 4 ligatures. The arrows in blue color indicate the application of Rule I, while the arrows in orange color indicate the application of Rule II.*

spaces are present that are caused by breaker-character. Technically, after each breaker-character, there is a ligature split, and that is the required position where we can split the word segments. To facilitate the splitting procedure, we have categorized the breaker-characters into two categories. Let say category *A* refers those breaker-characters, which belong to regular Pashto characters and let say category *B* refers those breaker characters which are either punctuations or digits etc. In this work, we have included all Pashto numerals and one punctuation i.e. full stop -. Thus, we define two different rules to handle these categories.

- Rule I: If a breaker character belongs to category *A*, then we have to split the word at one index ahead of that character.
- Rule II: If a character belongs to category *B*, then we have to split the word at two different locations, first at one index ahead and second at one index before the breaker character.

After the application of these rules on each word, we obtained a list of Pashto ligatures. Figure 4.1 illustrates the splitting of a word according to the two different rules. In this way, all unique Pashto words are split into 286,628 ligatures.

Further, to find out the unique ligatures and their frequencies in the Pashto language, the overall set of ligatures is considered. A total of 19,268 unique ligatures is found. The detail about the contribution of these ligatures and frequencies are shown in a Table 4.4.

4.3.2 Pashto Primary Ligatures

The concept of primary ligatures in cursive script is not new. Primary ligatures in the Urdu language are already explored [Leh12]. A recognition system is already presented for the classification of primary ligatures considering Urdu text as a test case, where they have reported 98% recognition rate (see reference [LR13] for detail). The concept of

Table 4.4: Unique Pashto ligatures and their frequencies.

Number of ligatures	Frequency	% in Corpus
30	139,551	48%
100	177,273	61%
500	215,212	75%
1,000	228,727	79%
2,000	240,260	83%
5,000	255,605	89%
7,000	261,605	91%
19,268	286,628	100%

primary ligature in Latin script is also presented with a term like *shape coding*. There is a notable work regarding shape coding, which can be found in these references[LLT08, SS97, BLT09, Spi95, SM99].

Primary ligatures could be used as recognizable units. The reason is that primary ligatures are highly discriminative from each other. Furthermore, the set of primary ligatures is always smaller than all unique ligatures.

We need two types of information to group all unique ligatures into similar groups of primary ligatures. 1: Total unique ligatures of Pashto language (which we have already). 2: Pool labels for many subsets of Pashto characters which represent same base/primary shape according to 4 different positions (Isolated, Initial, Middle, and End position in a ligature). The second requirement needs language-specific knowledge and understanding of different shapes of Pashto characters. Different pools of Pashto characters are formed, and each pool is labeled with an English alphabet. These pools and their characters are shown in Table 4.5. A label from the appropriate pools is then assigned to each character according to its position in the ligature. A total of 7,681 primary ligatures is found, which contribute to a total of 19,268 ligatures. Figure 4.2 shows a cluster/group of 100 ligatures, where all the ligatures have the same primary ligature.

Shapes of top ten primary ligatures, their pool ids, and the total number of ligatures covered by each primary ligature are shown in the Table 4.6.

4.4 Conclusion

In this chapter, a detailed textual analysis is presented regarding the Pashto language. The study provides the statistics about the frequencies of ligatures and primary ligatures. Further, this study also explains the conceptual rules behind the formation of ligatures and primary ligatures in the Pashto language. For this study, a huge corpus has been collected from the worldwide web that contains only Pashto text. The corpus contains

ټيز	تير	تيژ	تير	پيز	نيز	نير	تنز	تنر	بيز	ټيز	بيز	بيز	ټيز
نتر	نتر	ټير	ټير	نير	پير	بیر	بیز	بيز	پير	پير	نيز	پير	بیر
بيز	بیر	بيژ	ييز	پير	پير	نير	نير	پير	بنر	بنز	بیر	بیر	يتز
يتز	پيز	بنر	يتز	نيز	نيژ	بیر	بیر	بيز	بیر	بيژ	بیر	تتر	بيز
ټير	ټيز	ټير	ټيز	ټيز	ټير	ټير	ټيز	ټيز	ټير	ټيز	بیر	ټير	ټير
ينز	ينر	نتر	بيز	بیر	ينز	بيژ	بيز	ينر	ينز	بتر	نير	نيز	نير
پير	نير	بش	بيز	پير	پير	بيز	بيز	ټير	پير	پير	پير	نير	پير
												پير	ټير

Primary Ligature id: TTE,

Shape of Primary Ligature: **سر**

Covering ligatures: 100

Figure 4.2: *A top cluster of Pashto language having the most ligatures, based on a unique primary ligature. It can be seen that despite having a unique primary ligature, an utterance in the secondary component extends the number of ligatures.*

2.3 million Pashto words, in which 82,409 unique words are identified. We found that only 14,000 words can contribute to 93% portion of the corpus.

Further, about 19,268 unique ligatures are identified in the Pashto language. These ligatures are mainly contributing in all shapes of 2.3 million words. It is also found that only 7000 ligatures are sufficient to describe up to 91% of the entire unique words.

Another, potential alternatives like primary ligatures, as recognizable units are also identified. Primary ligatures are produced by the reduction of ligatures into their basic connected shape. Based on our analysis, about 7,681 primary ligatures are discovered, which cover the all 19,268 ligatures.

Besides these findings, we have addressed some issues related to Pashto text. These issues in general cause complexities in recognition of Arabic like scripts. However, the Pashto language presents these complexities with high intensity because Pashto has a larger *breaker-character-set*. Similarly, the term *breaker-characters* is introduced instead of *non-joiners*.

Table 4.5: *Pashto characters are grouped into pools using their various shapes at different positions. The legends are used; Isolated as Iso, initial as Init, middle as Mid, end as End and all as All.*

Pool Id	Member characters	Iso	Init	Mid	End	All
A	آ	-	-	-	-	✓
B	ب پ ث ت	✓	-	-	✓	-
C	چ ج ح خ حْ خ	-	-	-	-	✓
D	د ذ	✓	-	-	✓	-
E	ر ز ژ ږ	✓	-	-	✓	-
F	ښ س ش	-	-	-	-	✓
G	گ ک	-	-	-	-	✓
H	غ ع	-	-	-	-	✓
I	ف ق	-	✓	✓	-	-
J	ف	✓	-	-	✓	-
K	ق	✓	-	-	✓	-
L	ل	-	-	-	-	✓
M	م	-	-	-	-	✓
N	ن ټ	-	✓	✓	-	-
O	ټ	✓	-	-	✓	-
P	پ	✓	-	-	✓	-
Q	ط ظ	-	-	-	-	✓
R	ص ض	-	-	-	-	✓
S	ئ ي ی ی ی ی ی	✓	-	-	✓	-
T	ت ب پ ث ی ی ن ئ ی	-	✓	✓	-	-
U	ږ	-	-	-	-	✓
V	ډ	-	-	-	-	✓
X	ه	-	-	-	✓	-
Y	ه	-	✓	✓	-	-
Z	ن	✓	-	-	✓	-

Table 4.6: *Top ten primary ligatures and their ten covering ligatures along with their pool ids. The final row shows the number of total ligatures covered by each primary ligature.*

Pool Id	TTE	TTS	TCS	TFS	TTA	FTS	TTTA	FTTS	CTE	TTX
Primary Ligature	سر	سى	حى	سى	سا	سى	سا	سى	حر	سو
Top ten ligatures covered by a primary ligature	ئېز	نتى	نخى	نشى	تنا	شنى	نيپا	شيبې	حېز	بيو
	بېز	بېچى	بجى	نسى	ئنا	شنى	بنيپا	بنتې	خېز	تتو
	بىز	تېچى	نجى	يشى	پتا	شتى	بيپا	بنتى	خېز	ئيو
	بىز	يېچى	يجى	بسى	يا	بنتى	بيپا	بنتى	خېز	تتو
	تيز	تىچى	يجى	بسى	تنا	بنتى	بيپا	بنتى	خېز	تبو
	تيز	ينى	يجى	يسى	ثيا	بنتى	تيا	شيني	خېز	بيو
	ينز	ينچى	يجى	يسى	ئا	بنتى	ئاپا	شيني	خېز	ئيو
	بىز	نېچى	يجى	يسى	نا	بنتى	تيا	شيني	خېز	پنو
	ئىز	ئېچى	يجى	يسى	پا	بنتى	تيا	بيني	خېز	بيو
	پېز	ننى	يخى	يشى	يا	بنتى	نتا	شيني	خېز	پبو
ligatures covered	100	94	69	68	63	62	60	59	57	54

Benchmark OCRs for Arabic Scripts

This chapter provides an overview of the existing methods specifically used for the recognition of cursive scripts. It first states the motivation and then discusses the related work regarding the reputed benchmarks. The related work is mostly focused on RNNs with LSTM units for recognizing cursive scripts especially the Arabic scripts. It signifies the work in the thesis in context of existing research and highlights the contribution of the proposed work. The chapter is organized as follows: after the motivation, at first, we briefly introduce the existing techniques that are dominant in the domain of OCR and for better understanding we have categorized them into two categories. The first category refers the traditional approaches based on a segment-and-classify rule. We referred them as **segmentation-based** approaches and are explained in Section 5.2. The second category refers the advanced approaches based on **segmentation-free** techniques. Segmentation-free techniques avoid the segmentation and use *whole* segments, for example, a word or a sentence/text-line as input to OCR. These approaches are also known as holistic approaches and are discussed in Section 5.3. The work in this thesis focuses on the holistic based approaches, therefore, most of the attention is given to this approach. However, segmentation based approaches are mentioned for the sake of completeness.

5.1 Motivation

There are several motivations of benchmarking the existing work for Arabic like cursive script language. The main motivation is based on the fact that Pashto is a superset of Arabic like languages, and benchmarking its related work will not only give an insight into the Pashto OCR application but will also help in understanding the research in OCR for the other Arabic cursive scripts. Though Pashto has a limited research work regarding OCR application, research related to Arabic OCR could be adapted for Pashto OCR. Thus the contents of this chapter will help in understanding the exact status of the

research related to Arabic OCR and their adaptation for Pashto OCR. Another motivation is putting the vast literature that exists for Arabic language recognition in context to highlight the research requirements for Pashto language recognition. Furthermore, the chapter also signifies the need of the proposed work by mentioning the gaps that exist in the current work.

5.2 Segmentation-based Approaches

Segmentation-based approaches are the more conventional approaches used for years in the domain of OCR. In these approaches, the documents are repeatedly segmented until reaching the smallest nonsegmental segment usually known as characters/symbol. One of such examples is Tesseract [Smi07]. Tesseract is one of the well-known open source OCR systems using segmentation-based paradigm. Significant related work can be found both for printed and handwritten text recognition using segmentation-based techniques. The detail of such work can be found in surveys [CL96, Lu95, SSR10].

Segmentation-based approaches can be divided mainly into two classes. The first class refers the approaches which are based on *template matching*. In template matching approaches the smallest segment (i.e. character) is repeatedly checked against the stored templates, and a match is declared successful between a template and a character if similarity measure is high between them. Section 5.2.1 briefly describes template matching techniques.

The second class represents the approaches which are based on *over segmentation* techniques. In general, it is difficult to find the correct cut points for the accurate segmentation. Therefore, in these approaches, a guess has been made to find an approximate cut point for segmentation. Such cuts lead to over segmentation, which is then corrected in the later stages. Section 5.2.2 summarises over segmentation approaches.

5.2.1 Template Matching

Initially, template matching is used in photoelectric and was purely mechanical in nature. Interestingly, some early OCR systems based on mechanical template matching could be found in [Han62, VR77]. However, as the advancement in computer technology became phenomenal, the template matching techniques are also transformed into computer algorithms.

The current template-matching techniques rely on matching pixels values in a CC (i.e. input character segment) against the pixels values presented in the templates. The tem-

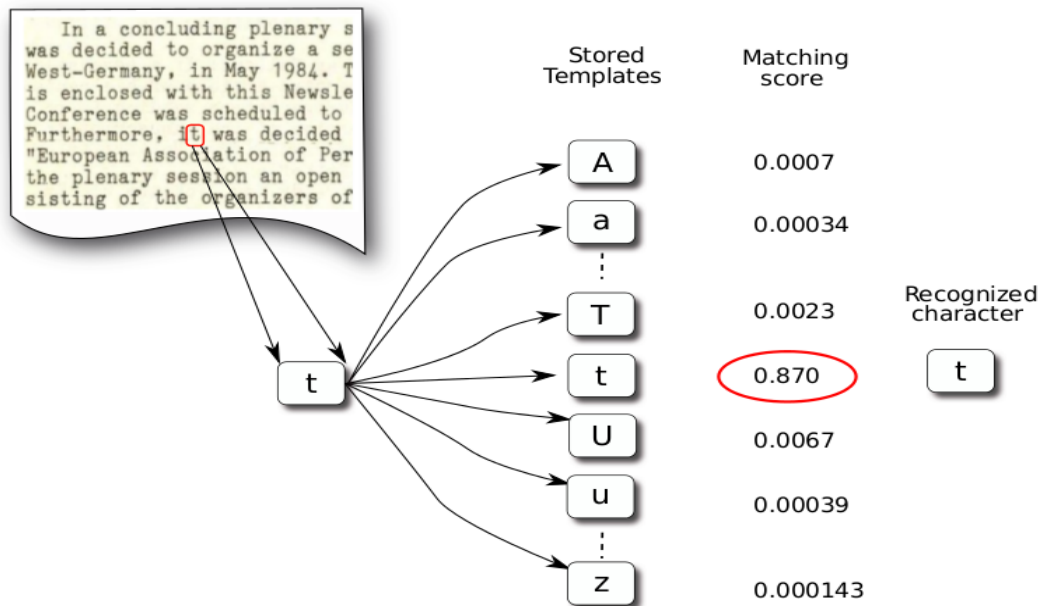


Figure 5.1: *The connected component an input image is matched with stored prototypes of template, and a character is classified on the basis of maximum match score.*

plates are prototypes, stored for all possible shapes of the characters, and each template has a class label. A term *bit-map* is also used for the template in various literature. A matching process is carried out by summing the correspondence of pixel-with-pixel between a CC and a template. This summation results in a matching score, and above a certain threshold matching score decides the candidacy of the target templates. Finally, among the candidates, the one is selected for which the match score is high. Figure 5.1 shows the complete process of template matching technique. More work regarding template matching in the field of OCR could be found in [CP97, CP98, NBK01].

OCRs based on template matching are dominantly used in applications where target text is recognized in a constrained environment. For example, vehicle plate-numbers, ID-card recognition, etc. [KB13, RH15, FAZAQ⁺16]. However, these approaches are very sensitive to noise, character overlapping, font variations, and depending mainly on the quality of segmentation.

5.2.2 Over Segmentation based Approaches

While dealing complex text with touching or overlapping characters, the accurate segmentation becomes a nightmare and becomes more difficult while dealing the cursive scripts. Therefore, to facilitate the segmentation process, the text is initially segmented into imperfect segments. The system combined these imperfect segments into an appropriate representation of a character. The combining process is achieved by classical

pattern recognition techniques like decision trees, nearest neighbor classifiers, Bayes classifiers, or neural networks. Mainly two heuristics are used, (1) how to discard a segment from the candidate list, and (2) how to consider a segment as a candidate for target classification. Defining a heuristic for selecting a candidate segment is one of the toughest factor of OCRs based on over segmentation techniques. The main reason is the presence of broken or overlapped characters. For example, "m" could be confused with 'nn', 'rn', or 'd' could be confused with 'cl', and vice versa. The Tesseract [Smi07] mainly uses this kind of strategy.

The performance of over segmentation based OCRs can be improved by using language modeling. Language modeling helps in combining the over segmented components to a most probable combination that is acceptable under the observation received from the context of a language.

5.3 Holistic Approaches or Segmentation-free OCRs

The inaccurate segmentation is the major weakness of Segmentation based approaches. Therefore, researchers like Nagy [Nag92] argued that the community that dealing document image analysis should move to holistic approaches rather than segmentation based approaches. Holistic approaches or segmentation-free approaches use the complete segment i.e. word, piece-of-word/ligature, or a text-line, and avoid the complexities of segmentation. These approaches not only use features of a character but also use the features from its surrounding context. The context exploitation for better recognition is the peculiar nature of holistic OCRs. Therefore, in last two decades, there is a significant contribution in the improvisation of segmentation-free approaches. The most reputed segmentation-free OCRs are mainly based on HMM. However, ANN and their most advanced versions like RNN based on LSTM (cf. Section 3.3.2) units with a combination of CTC (cf. Section 3.4) have proved their dominance in this field.

The LSTM based models have been effectively used to benefit the state-of-the-art for various applications. These problems include language modeling [ZSV14], translation [LSL⁺14], acoustic modeling of speech [SSB14], speech synthesis [FQXS14], speech recognition [GMH13], analysis of audio [MFE⁺14], video data [DAHG⁺15], and handwriting recognition [GLF⁺09, PBKL14, DKN14]. Besides all the problems mentioned above, here we have to narrow down the scope of the related work, and briefly discuss LSTM based work done for the recognition of Arabic like-script in off-line mode.

The pioneer work that used LSTM for the recognition of offline Arabic text is reported by Graves et al [Gra12b]. They used MDLSTM as training model and raw pixels as features.

Table 5.1: *The target scripts are taken from printed, handwritten, and scribed material. The features that are mainly used are based on the raw pixel, statistical, zoning, and convolved feature, with or without language modelling (LM).*

Reference	Script	Architecture	Features	Dataset	Accuracy%
Graves et al. [Gra12b]	Handwritten Arabic	MDLSTM	Pixels	IFN/ENIT	93.37
Rashid et al. [RSRvdN13]	Printed Arabic	MDLSTM	Pixels	APTI	99
Morillot et al. [MOLS+13]	Handwritten Arabic	BLSTM	Features	NIST	52 (word)
Pham et al. [PBKL14]	Handwritten	MDLSTM	Pixels	RIMES	91.1
	French			IAM	85.6
	English Arabic			OpenHaRT	90.1
Chherawala et al.[CRC13]	Handwritten Arabic	MDLSTM	Pixels	IFN/ENIT	89
Yousefi et al.[YSBS15]	Handwritten Arabic	BLSTM	Pixels	IFN/ENIT	87.4
Hamdani et al.[HDK+14]	Handwritten Arabic	BLSTM +LM	Pixles	OpenHaRT	80.1 (word)
					94.1 (char)
Ul-Hassan et al. [UHAR+13]	Printed Urdu, (Nastaliq)	BLSTM	Pixels	UPTI (10,000)	86.4 94.8
Ahmed et al. [ANR+16a]	Printed Urdu	BLSTM	Pixels	UPTI	88.94 (char)
Ul-Hassan et al. [UHSL15]	Printed Urdu	BLSTM	Curri- culum Pixels	UPTI database	94.25
Naz et al. [NAAR16]	Printed Urdu	MDLSTM	Zoining feature	UPTI database	93.39
Naz et al. [NUA+15]	Printed Urdu	MDLSTM	Statist Features	UPTI database	94.97
Naz et al. [NUA+17]	Printed Urdu	MDLSTM	CNN features	UPTI database	96.4
Naz et al. [NUA+16]	Printed Urdu	MDLSTM	Pixels	UPTI database	98
Ahmad et al. [ANA+17a]	Handwritten Arabic	MDLSTM	Pixels	KHATT database	75.8

The IFN/ENIT¹ dataset is used for evaluation, and a word accuracy of 93% is achieved. Rashid et al. [RSRvdN13] presented an OCR for low-resolution images using MDLSTM and CTC. They used APTI² dataset, and achieved an accuracy of 99%. Morillot et

¹<http://www.ifnenit.com/download.htm>

²<http://diuf.unifr.ch/diva/APTI/>

al. [MOLS⁺13] proposed a system for Arabic handwritten recognition. They used BLSTM considering NIST/OpenHaRT³ data as a test case and achieved 52% word recognition rate. This work was further improved by Pham et al. [PBKL14] by introducing a dropout mechanism with MDLSTM architecture. They have evaluated 3 benchmarks i.e. RIMES⁴ for French, IAM⁵ for English, and OpenHaRT for Arabic handwritten recognition. They have reported character accuracy of 91.1%, 85.6%, and 90.1% for French, English and Arabic texts respectively. The most relevant work regarding Arabic, Urdu, and Pashto OCR system using LSTM architectures are shown in Table 5.1. In that Table, we report the related work that is based on LSTM architectures and addressing OCR for Arabic, Urdu and Pashto languages.

It is concluded from the related work that other cursive languages like Arabic, Urdu, and Persian are mostly explored using either synthetic or ligatures/piece-of-words based datasets. Thus, the non-availability of data is also an issue, and need a comprehensive real dataset, where the language contents present the temporal dependency as well.

In addition to that, a mature OCR does not exist so far that could handle Arabic like languages. As mentioned in Chapter 2, that the Pashto language has a larger and generic character set and if it is considered as a problem domain then certainly it could help the research community. Another, most of the research contributions are in parts, and very less effort is made for an end-to-end OCR system. Therefore, the improvisation of an end-to-end system will significantly improve OCR for all Arabic scripts, considering the generic and marginally more complex script like Pashto.

This thesis focuses the above-mentioned points that need further research and contributes in a way to accomplish an end-to-end OCR system. The contributions of this thesis mainly include language-specific challenges, conceptual findings (cf. see Chapters 2 and 4), real dataset creation (cf. Chapters 6 and 7), skew detection and correction (cf. Chapter 8), text-line extraction (cf. Chapter 9), deep learning baseline for Pashto; a holistic approach (cf. Chapters 10 and 11), space anomaly (cf. Chapter 12), and an end-to-end OCR (cf. Chapter 13).

5.4 Conclusion

We have briefly introduced the most attractive OCRs methodologies dominantly used in document image analysis. Now, we have enough foundation to discuss the contributions of this thesis in the light of existing research work. The OCR systems broadly divided

³<https://www.nist.gov/multimodal-information-group/openhart>

⁴http://www.a2ialab.com/doku.php?id=rimes_database:start

⁵<http://www.fki.inf.unibe.ch/databases/iam-handwriting-database>

into two categories, i.e. segmentation based and segmentation-free. As discussed, that the segmentation based OCR systems mainly depend on segmentation. But, cursive scripts pose more complex behavior in segmentation, therefore, in a broad sense for language like Pashto, the segmentation based techniques are not helpful. On the other hand, segmentation-free or holistic approaches are capable of coping with the Pashto language in the field of OCR. However, the related work is very limited regarding the Pashto language (cf. Section 2.6), and it needs an empirical research that could explore the specificities of this language. Thus, there exists a research gap, that needs to be investigated considering the Pashto language as a test case exploiting the segmentation-free techniques.

Part III

Datasets

Ligature Based Synthetic Datasets

This part of thesis gives an insight to the datasets and their creation for the Pashto language to benefit OCR research. The first Chapter of this part describes a detail about synthetic dataset used in the Pashto OCR system. First, it states motivation and then introduces the related work regarding Pashto existing datasets. Further, it explains the contribution of this thesis regarding the synthetic data for the Pashto OCR. It also describes how we came to extend the existing Pashto datasets. In short, this chapter covers the aspects of the creation of the synthetic dataset in the domain of OCR for the Pashto language.

6.1 Motivation

Datasets play a vital role in the evaluation of benchmarks in the field of OCR. However, the creation of a dataset is a very tedious job, and it becomes more difficult while dealing a language, whose research just started from scratch. Pashto is one of the languages, that does not have any appropriate data and needs proper data to evaluate and investigate OCR domain. Therefore, the core motivation of this work is to create a synthetic dataset to evaluate the existing benchmarks. A creation of a synthetic dataset is a common tradition in the field of OCR. The first attempt always goes for the creation of synthetic dataset. The reason is the time-saving factor, as there is no such effort involve for annotating images as required in real world data. Although a model trained on a synthetic data could not generalize well to novel data, there is still a need for synthetic data. In the initial evaluation, synthetic data is required to get the very generic clues and analysis regarding the proposed model/OCR, and about the data itself.

Furthermore, it is important to first go through the existing research work about Pashto datasets. Therefore, the next section reports the related work regarding Pashto datasets.

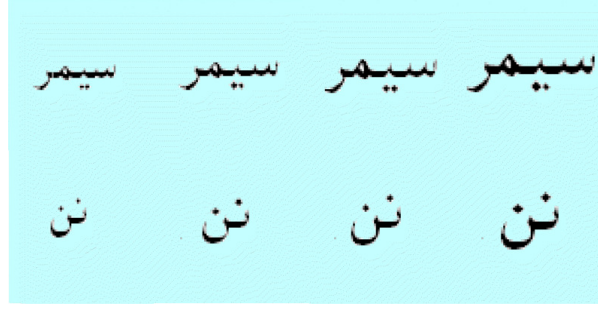


Figure 6.1: *Two Pashto ligatures and their 4 different font sizes i.e. 12, 14, 16, and 18. Image source: [WAA09]*

6.2 Related Work

In fact, there is a limited work regarding the Pashto OCR as well as datasets. The first ever work regarding Pashto OCR is presented by Decerbo et al. [DMN04]. They have reported their results on some Pashto datasets, but unfortunately, those datasets are not available publicly. The first ever synthetic dataset for Pashto OCR is created by Wahab et al. [WAA09]. This dataset is available on request from the principal authors.

The dataset developed by Wahab et al. [WAA09] contains 1000 unique Pashto ligatures. They selected each ligature from a Pashto novel *Da Jwand aw Da Ceray*. First, all these ligatures are typed in a word editor known as Aisha Soft¹. Then for each ligature, they created four images for font sizes 12, 14, 16, and 18 respectively. The Pashto font named *Karor* is used to render the text. The size of each image is fixed and is 100×100 pixels. Initially, this dataset contained overall 4,000 images, comprised of 1000 unique Pashto ligature with 4 scale variations. Figure 6.1 shows an example of two ligatures and their corresponding 4 sizes.

The annotation scheme used for the initial dataset mainly relied on the file name used to refer an image. For example, for each image, file name looks like *L11.bmp*. In this example, the first 1 determines what is the *label* of the ligature, while the second 1 represents the size of the ligature. Note that, the values for label determiner is ranging from 1 – 1000, and the values for size determiner is ranging from 1 – 4, (i.e. 1 for font size 12, 2 for font size 14, 3 for font size 16, and 4 for font size 18). The file name for a ligature having a label: 103, and size: 2 would be *L1032.bmp*.

Initially, these images are acquired in RGB format and then they are converted to monochromatic (black and white). The reason behind such method of image creation is that training data for OCR often suffer from different factors which include font size variations, different font styles, smeared and broken texts, scaling, skew, and noise. These factors can cause poor recognition rate.

¹An editor developed by Pashto Academy, University of Peshawar

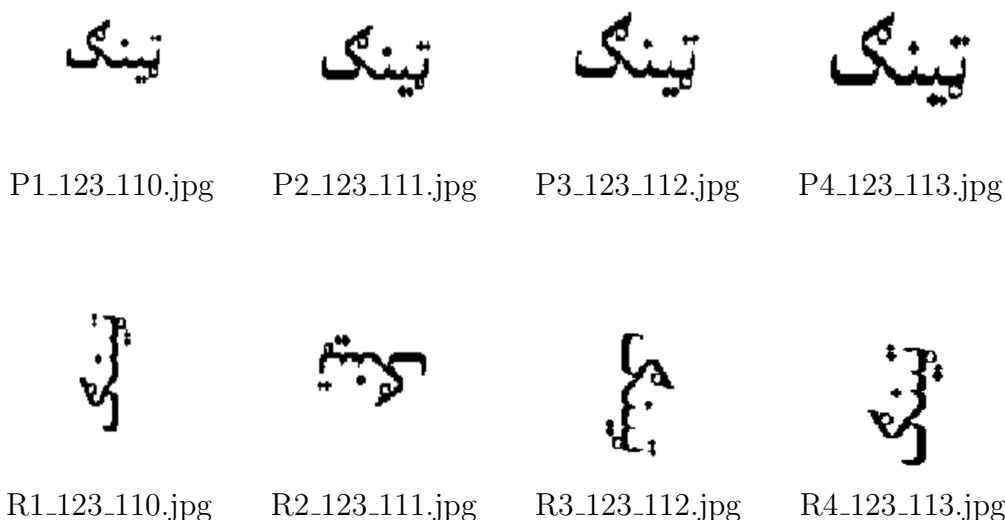


Figure 6.2: *Each ligature has 4 scale and 4 rotation variations. The file names are given below each ligature, for all these ligatures only one text file exists as ground truth i.e. P123.txt.*

This dataset provides annotations in numeral form instead of machine readable codes i.e. (utf-8), and only transcribes the ligatures instead of characters. Therefore, the use of this dataset is only applicable for ligature-based evaluations, and character level evaluation is not possible using the default annotation scheme. To address such deficiencies we have made some enhancements to the dataset. Next section includes these enhancements.

6.3 Contribution

6.3.1 Scale and Orientation variations: Ligature-Based-II

In our first contribution, we extended the initial dataset by introducing 4 dominant rotation variations. We needed this extension to investigate some issues regarding invariant recognition of Pashto ligatures. The annotations are also updated to UTF-8 codecs. The updated version contains 8,000 images of 1,000 unique Pashto ligatures with 4 scale and 4 rotation variations. The extended dataset is investigated using SIFT features for scale and rotation variations [AAK10]. Figure 6.2 shows an example from the extended version depicting scale and rotation variations. As this dataset is used in chapter 10 and the proposed recognition system is published in [ANA⁺15]. Therefore, in the remaining text, we refer the updated version of the synthetic dataset as Ligature-Based-II.

In addition to the extension in rotation variations, we modified annotation scheme for



Figure 6.3: *Scale variations for each Pashto ligature introduced in Ligature-Based-III.*

Ligature-Based-II. We provide 1000 *.txt* files, where each file contains ground truth for corresponding Pashto ligature. The new ground truth is in UTF-8 codecs and therefore, provides compatibility for the evaluation on character level as well. Further, we also changed the file naming scheme for an image the new file name could be *P1_103_12.jpg*. The new file name can be split into three parts. The first part refers the image size, here *P1* means that it is a plain (without rotation) image of size 1. The second part refers the label of the Pashto ligature, here 103 means it is the label of the Pashto ligature, and the final part just presents a counter. Figure 6.2 shows an example of 8 different instances along with their file names.

6.3.2 Towards Big Database for Pashto Ligatures: Ligature-Based-III

To benchmark, the most advanced techniques based on deep learning, we further extended the Ligature-Based-II dataset. We have introduced 40 scales variations for each ligature and 12 rotation variations per scale. This extension resulted in 480,000 images. The purpose of this enhancement is to make the dataset suitable for DL/LSTM based approaches. Chapter 10 covers the detail of such approach that is subsequently published in [AAR⁺15a]. In the remaining text, the enhanced version is referred as Ligature-Based-III.

We have introduced scale variations in two steps process. In the first step, we took a ligature of font size 12 (let say it S_1) as a base scale, and have cropped the ligature in a tight bounding box. In the second step, we scaled up the cropped ligature by 10 pixels keeping the aspect ratio locked. The process of scaling up the images is repeated for 40 times until to get images like S_1, S_2, S_3 , up to S_{40} . Figure 6.3 shows 40 scales for each

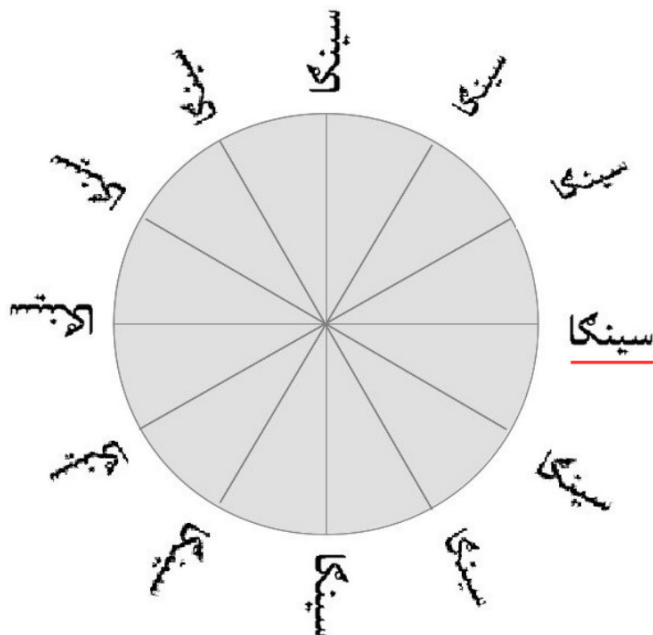


Figure 6.4: *Each scale of unique Pashto ligature has 12 rotation variations; source [AAR⁺15a].*

ligature. Similarly, we have rotated each scale to get 12 different rotations by applying a step of $+30^\circ$ in anti-clockwise. Figure 6.4 illustrates the rotation variations for a single ligature.

The transcription and naming conventions for Ligature-BasedIII dataset is different from Ligature-BasedII dataset. In Ligature-BasedIII dataset, the file name could be like *R0_S12_P120.jpg*, we can split the file name into three parts. The first part refers the nature of the rotation. For example, if it is *R0*, then the ligature will be without any rotation. This first part can take values like *R0*, *R30*, *R60*, *R90*, etc. Where the integer values describe the rotation angle of the rotated ligature. The second part refers the scale variations, for example, the *S12* represents the 12th scale. The final part describes the label id and refers a text-file which contains ground-truth information.

6.4 Conclusion

This chapter provides detail about the syntactic datasets, which are required for the Pashto language in the domain of OCR development. We can conclude that there are mainly 3 versions of synthetic datasets for Pashto OCR. These three datasets are based on 1000 unique Pashto ligatures. The first version was created by Wahab et al. [WAA09], which contained 4 scales variations for 1000 unique Pashto ligatures. The dataset was annotated with a number from 1 to 1000 as unique labels for each unique Pashto ligature. The second version i.e. Ligature-Based-II contains the same 1000 unique Pashto ligatures

with 4 scale and 4 additional rotation variations. Finally, Ligature-Based-III is the 3rd version of Pashto synthetic datasets, it contains 40 scales and 12 rotation variations for each scale. The latter two versions are the outcome of this thesis as contributions and are subsequently evaluated in work reported in [AAK10, ANA⁺15] respectively.

The synthetic datasets are more specific as they contain only Pashto ligatures, and are more suitable for the classification of the Pashto ligatures. Another, they are more specialized regarding scales and rotation variations and provide a study case for scale and rotation invariant recognition systems. However, due to syntactic nature, they have a deficiency of lacking the real world artifacts. In addition to that, the specificity of a language itself could not be learned only from ligatures based dataset. Thus, we need a dataset that contains real world instances of Pashto text, and also contains temporal observation and patterns that describe the structure of the language itself. Therefore, next chapter presents a detail about the creation of real dataset for the Pashto language.

Real Pashto Imagebase Creation

This chapter provides an overview of the creation of the real Pashto dataset in the field of OCR. At first, it presents a motivation for the creation of real dataset. Second, it provides the general information about image acquisition in the field of OCR. It describes the way we have acquired the Pashto text images. Further, it demonstrates the procedure for transcription creation. This chapter also exhibits the importance of the target data regarding its nature and hardness.

7.1 Motivation

It is a common observation that synthetic data does not cover all the real world artifacts that could pose real challenges to the proposed OCR system. Further, classifiers trained on synthetic data also lack generalization toward real data. Synthetic datasets regarding the Pashto language contained only ligatures and are specific regarding scale and rotation variations, therefore, could not be used for the robust Pashto OCR system. In contrast, real data presents more challenges, these challenges are due to different stages, include data creation, image acquisition, transcription generation, preprocessing, etc. These factors increase the level of difficulty for the robust recognition.

Although researchers frequently use synthetic datasets, the usage of real data always has much importance in the field of OCR. Regarding Pashto OCR, to the best of our knowledge, there is no real data that could be used to explore real world challenges. Furthermore, an effective-digitization needs a mature OCR system, and to achieve a sophisticated OCR system for the Pashto language, we need a comprehensive dataset for benchmarking state-of-the-art methods/models. Therefore, the motivation of this chapter is to achieve a real dataset for the Pashto language in the field of OCR. The contents of the real dataset are taken from the Pashto language and are represent **scribed/handwritten** materials. It is one of the major contributions of this thesis concerning data level.

7.2 Data Acquisition

The collection of relevant contents for the dataset creation is also an important factor. The researchers are more attentive to things like; acquisition methods (scan based, camera based, etc.), contents layout, unbiasedness, and much more. There are many articles, having a comprehensive discussion regarding image acquisition, some important ones are referred here [RMM⁺94, JKJ04].

In general, it is a good idea to start with a scanner based acquisition. Scanner based acquisition is abundantly used, and considered as a standard approach for the digitization of books and historical stuff. However, digitization of camera captured data also gained significant importance. It is due to the recent hype in mobile technology in shape of abundant smart phones, gadgets, digital cameras, etc.[Doe98, LDL05].

It is worth mentioning that camera captured data mostly suffered due to artifacts like; blur, perspective distortion, warping, skew, shadow, and light reflection. In contrast, data acquired via scanner only contain skew, and are free of other aforementioned artifacts. However, the artifacts that really associate with the contents itself are preserved in scanner based acquired data. The following text describes the important factors that we considered specifically in Pashto data acquisition.

A comprehensive dataset should cover the very frequent contents and the most occurring patterns for a target language. An inclusion of these patterns in the dataset makes it ensure that the dataset is more suitable for addressing the most frequent challenges. In this context, we have analyzed the Pashto text and have found that the Pashto text materials can be categorized into two periods i.e. before 1995, and after 1995.

The first period describes the text materials that are written by scribes, locally known as *Katibs* or calligraphers. Nowadays, the use of *Katibs* is very rare to scribe text. Figure 7.1 shows the way how *Katib* wrote text document. The second period refers that materials that are written by computer typist using a composing software. The materials are printed via computer printers, and subsequently, many copies are generated by modern press machines. Therefore, these materials are also known as printed text. In most cases, the contents from the second period have their digital copy as well. However, the contents from the first period do not have their digital copy, and hence are more valuable for research regarding OCR system towards digitization.

Therefore, to get maximum benefits regarding Pashto text digitization, the creation of a new and real imagebase is mainly focused on the material of the first period. There are many reasons, but the most important ones are given below.

- The collection of the scribed materials is abundant.
-



Figure 7.1: *The creation of scribe material by a Katib. Image source¹.*

- Most of the historical and renowned books of Pashto literature are written in scribed form.
- As they don't have digital copies, they need preservation.
- They present the genuine artifacts like calligraphic beauty and its non-uniformity due to human nature.
- There are variations in quality as well as in a composition of the paper used by Katibs.
- Variations in the size of a tip of the pen used by Katib.

In the next section, we describe how the images for real imagebase of Pashto language are acquired.

7.3 Data Acquisition for KPTI

In general, Pashto literature contains a variety of text layouts. These variations mainly exist due to the nature of the contents. The contents of Pashto literature can be classified as poetry, essay, novel, reports, news, religion, etc. Therefore, the real instances of text images are acquired via scanning six different Pashto books. The titles of these books are

¹<http://www.gettyimages.ae/photos/urdu-writings?excludenudity=true&sort=mostpopular&media-type=photography&phrase=urdu%20writings>

(1) *Abaseen*, (2) *Ghwarzangona*, (3) *Hagha-Dagha*, (4) *Kachkool*, (5) *Da-Khatir-Kulyat*, and (6) *Pukhtana-Shuara*. These books are chosen for their diverse contents. For example, the contents of *Abaseen* are mainly news, reports, and essay. Similarly, *Kachkool* is a novel, and *Pukhtana – Shuara* contains history contents. In short, these books contain all the mentioned variations. As we acquired these images from calligrapher *Katib* written materials, we named this collection as *Katib’s Pashto Text Imagebase (KPTI)*. Figure 7.2 depicts a representative text image of each book. All the books are scanned with a resolution of 300dpi (dots per inch). After the acquisition of images, a default skew is encountered in a majority of images due to manual scanning. The skew is corrected by our own method[ARA⁺16] (presented in chapter 8).

Further, the creation of real datasets is famous for their time-consuming factor regarding the creation of quality ground truth/transcription/annotations. We have observed the same case in the creation of KPTI dataset. For the transcription of each acquired image, we hired the services of professional experts. They used word editor for transcribing each page/image. Each image has exactly one .txt file that contains ground-truth in *utf-8* codecs. The ground-truth is further validated for each text-line, that is successfully extracted. Further, description of the KPTI dataset is given in next section.



Figure 7.2: KPTI dataset contains different layouts of Pashto text and the source name of the Pashto books; Image source [AAR+ 16]

Table 7.1: KPTI statistics regarding source books, total pages scanned, extracted text lines, split detail(Train, Test, and Validation sets) and layout.

Book Name	Prefix	Total Pages	Total Extracted Lines	Train Set 70%	Test Set 15%	Valid Set 15%	Contents Layout
Abaseen	Abaseen	82	1,714	1,201	257	256	News, Report
Ghwarzangona	GhorZa	205	3,586	2,509	538	539	Report, Poetry
Hagha Dagha	HaqDag	75	1,765	1,235	265	265	Religion
Kachkool	Kach	50	865	606	129	130	Novel
Khatir Kulyat	Khatir	434	5,738	4,017	860	861	Poetry, Essay
Pukhtana Shuara	PasPo	180	3,347	2,342	503	502	Essay
Total		1,026	17,015	11,910	2,552	2,553	—

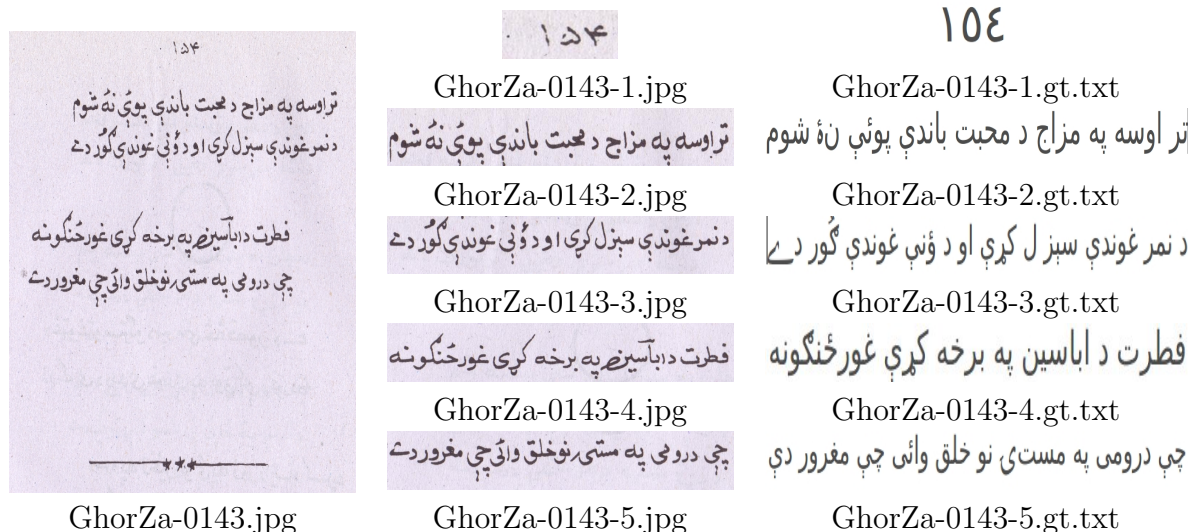


Figure 7.3: An input image with its file name below it (left), their corresponding extracted text lines along with their file names (middle) and ground truth and their naming convention for files (right). Image source: [AAR⁺16]

7.4 KPTI Description

The KPTI contains total 1,026 images. The names of the source books, the number of their corresponding acquired pages, the total number of extracted text lines and layout information are shown in Table 7.1. The text lines are extracted with a method, that is also one of the contributions of this thesis and is presented in chapter 9. Then text lines are extracted from the image, and each text line image is then referred as *Prefix - XXXX - YYY.jpg*. Now, *YYY* represents a certain line in a page number *XXXX*, which is taken from the source book, that is identified by *Prefix*. The KPTI thus contains a total of 17,015 images of Pashto text lines. Currently, the ground truths are only available for text line's images. The file which contains ground truth for a certain text line is named as *Prefix - XXXX - YYY.gt.txt*. The KPTI dataset contains overall 96 characters/symbols, which include all characters of Pashto, Pashto numerals, punctuation, etc. Very few instances of English characters also existed. However, we have removed them from KPTI. Figure 7.2 shows an input image taken from Ghwarzangona along with the extracted text lines and their corresponding ground truth. Hence, the file names conventions are also illustrated in Figure 7.2.

The split for *Train*, *Test*, and *Validation* sets, is done with 70%, 15%, and 15% for each set respectively. The extracted text-lines from each source (book) are first shuffled and then split accordingly. To make it sure, that the training set exactly presents 70% contents from each and individual book, and similarly, the test set and validations set also present 15% from each and separate source respectively. Thus, the overall text-lines

i.e. 17,015 are split in 11,910 as training set, 2,552 as test set, and 2,553 as validation set.

7.5 Conclusion

In this chapter, we have presented the detail about the creation of real imagesbase. We named it as KPTI. To the best of our knowledge, the existing research does not have any clue for real dataset regarding the Pashto language in the domain of OCR. The KPTI is the first ever real dataset regarding Pashto text. It contains the calligraphic material written by Katibs. The contents in the KPTI dataset mainly represent the text material written before 1995, and hence cover an immense collection of layouts and most frequent patterns in those documents. As the feeding units for OCR are text-lines, and text-line not only preserve the temporal patterns but also provide rich information for research community regarding the Pashto language itself. Furthermore, the character set of the Pashto language is the super-set of Arabic, Persian and Urdu languages, therefore, provide a generic benchmark for investigating the shared complexities among these languages.

The statistics of the KPTI include a total of 1,026 images of Pashto scanned documents. These images are then segmented in text-lines, and a total of 17,015 text-lines are successfully extracted. The ground-truths are verified manually for each text-line. All text-lines are then split in Training, test, and validation sets according to the proportion of 70%, 15%, and 15% respectively.

The skew in the scanned documents is inevitable. The images in KPTI dataset also have a skew in the range of $\pm 3^\circ$. The skew detection and correction is essential for accurate text-line extraction. Therefore, in next chapter, we present skew detection and correction approach.

Part IV

Pre-Processing

Skew Detection and Correction

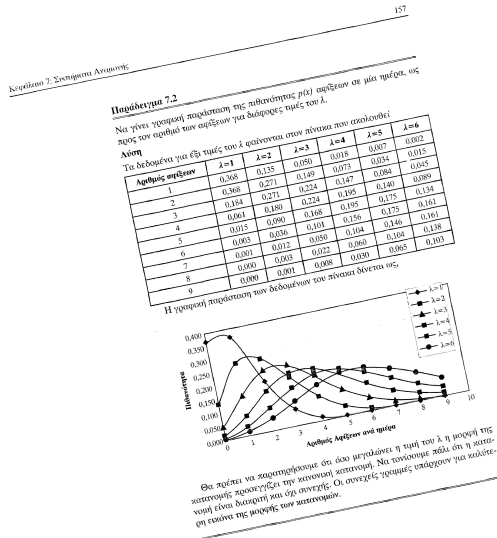
This part of thesis mainly focuses the contributions made in the stage of pre-processing for DIA system, and particularly for Pashto OCR system. In this chapter, we present our method for skew detection and correction in scanned documents. This chapter explains the motivation, discusses the need for another skew correction method, and compares the proposed method with the state-of-the-art methods. The next chapter of this part contains the detail description of our contribution concerning text-line segmentation.

8.1 Motivation

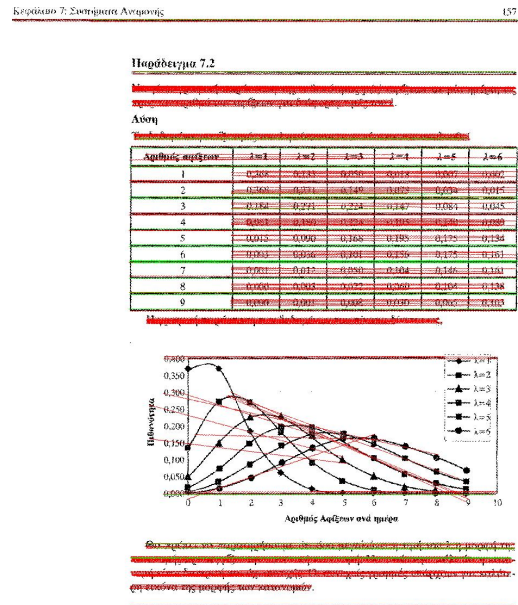
The creation of KPTI dataset subsequently resulted in the issue of skew in scanned documents. To correct the skew in Pashto text images, we have explored the current state-of-the-art methods. There is significant work in the field of skew detection and correction. However, we noticed that the most sophisticated methods have high time complexity [Fab14]. Another, heuristics based method like [SSA15] is also found dependent on noise and boundary removal. Therefore, a novel method is needed to de-skew scanned documents, and which is not only efficient but also has better accuracy compared to the state-of-the-art methods.

An optimal skew detection approach should be fast, reliable, script independent and robust to noise and borders. Additionally, it should also work for an acceptable range of angles. A de-skew method should be equally applicable to different scripts as well as different layouts. Keeping all these points in mind, we present a novel skew detection and correction approach, which is simple but robust and outperforms state-of-the-art methods using both accuracy and efficiency as evaluation metrics. The short version of this work is already acknowledged and published in [ARA⁺16]. A typical illustration of a skew document before and after skew correction is shown in Figure 8.1.

In this chapter, we present novel method and evaluate its performance on three different



(a)



(b)

Figure 8.1: A skewed document (a) and after skew correction (b).

datasets. These datasets are Document Image Skew Estimation Contest (DISEC’13)¹, Tobacco800², and 150 images from KPTI. The proposed method is thoroughly investigated, and a comprehensive analysis is given regarding skew detection and correction using scanned documents. This method is one of the contributions of this thesis, and it is integrated into a complete pipeline of DIA system.

Next section reports the related work regarding skew detection and correction and explains the base techniques sharing their pros and cons.

8.2 Related Work

The related research mainly represents the skew detection and correction regarding many approaches, which are based on the following main techniques. (1) Projection Profile (PP) (2) Hough Transform (HT) and (3) Fourier Transformation (FT) [Pos86, PT97]. In addition to these approaches, 1st nearest neighbor clustering (1st-NN) [HYR86] and heuristics based approach like axis-Parallel-Bounding-Box (APB) [SSA15] also address this issue. Each main technique is described in the following sections.

¹<http://users.iit.demokritos.gr/~alexpap/DISEC13/>
²<http://www.umiacs.umd.edu/~zhugy/tobacco800.html>

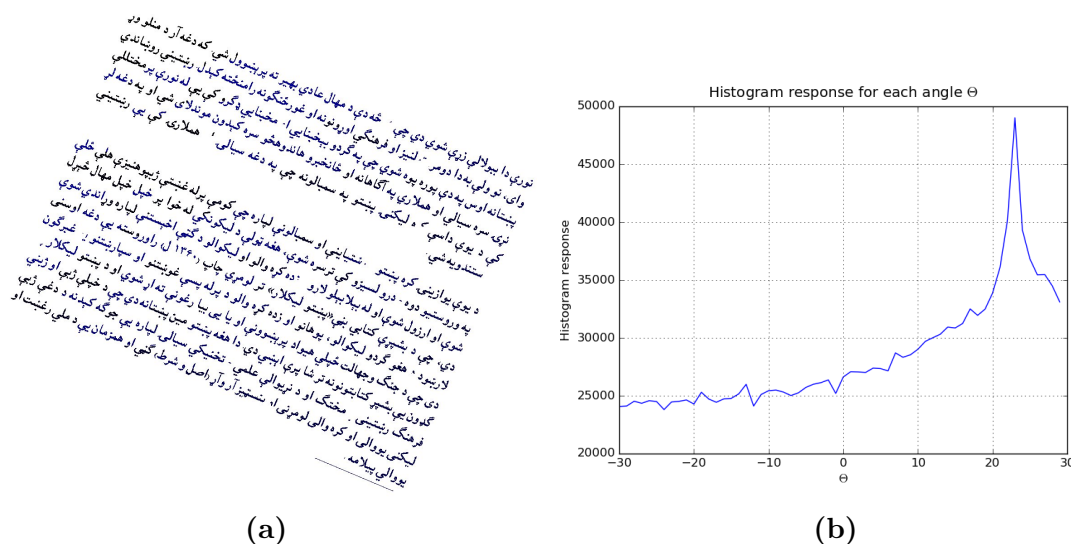


Figure 8.2: Document (a) has a skew angle of 23° , and the histogram response for the highest peaks for each angle ranging from -30° to 30° (b). The response is maximum at 23° .

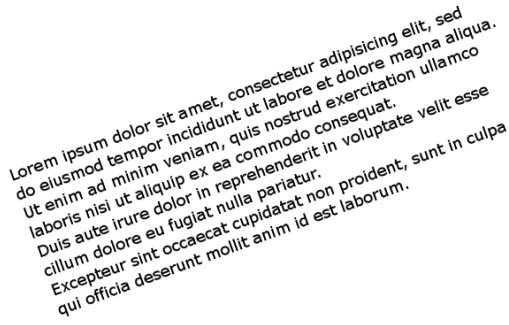
8.2.1 Projection Profile (PP)

Projection Profile (PP) based approaches use horizontal projection profile. It is observed that if a document has no skew, then the histogram of horizontal projection profile has maximum peaks. This characteristic of text documents was first realized by Postl [Pos86]. Figure 8.2 shows the highest peaks obtained from the different histograms of the horizontal PPs. The PPs are computed from a rotated document using a required range of angles. The highest peak represents the corresponding skew angle. The main issue with PP's based approaches is the computational cost because for a given range of angles one has to repeatedly rotate the entire document and check the peaks in the histograms again and again.

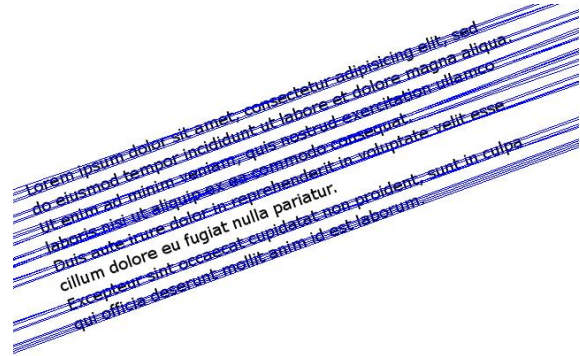
Ciaddiello et al. [CSD⁺88] presented a method to reduce the computational cost, where a sub-region is selected using the maximum density of black pixels per row, and that sub-region is tested for PP and histogram computations. Baird [Bai95] proposed a method, in which connected components were found in the first step, and then the bottom-centers of the connected components were projected for a set of angles in the second step. Finally, the PP was used as a criterion function for each angle.

Similarly, Bloomberg [BKD95] introduced the $2\times$ reduction of a full page using sub-sampling to speed up the overall process. Ishitani [Ish93] and Kavallieratou [KFK02] have also introduced methods, which are based on the detection of skew angle for a sub-region instead of full document/page. Besides all these improvements, methods based on PP are still very sensitive to noise and mainly depend on the contents of the documents. Documents containing graphics provide a challenge in skew detection and correction for

PP based approaches.



(a)



(b)

Figure 8.3: *The input image (left) has a skew angle of -19° , the Hough lines are detected (right), the average angle of these lines is obtained along the x -axis, which is a skew angle of the input image.*

8.2.2 Hough Transform (HT)

Another base technique that has been used for skew detection and correction in documents is the Hough Transform (HT). In this technique, the accumulative impact of each black pixel is represented in the form of a sinusoidal function by using equation 8.1.

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (8.1)$$

An accumulator is populated with votes corresponding to the points of intersections in the Hough space. Points of intersections in a Hough space represent a line. Due to the capability of line detection, HT is one of the potential methods for skew detection and correction. In HT based methods, lines are detected, and the angles of these lines along the horizontal axis are computed. Thus, the final angle is calculated by averaging all the angles, which have been detected by HT. A comprehensive survey about HT and it's used in image processing and computer vision is presented by Illingworth [IK88]. Hinds [HFD90] presented a method to reduce the time complexity of HT by computing the horizontal and vertical black run-length on the input image, which has reduced the document data by a factor of 7.

Srihari [SG89] introduced a skew detection method by computing PP of the HT for each value of θ and the angle at which the histogram of PP gives prominent valleys is considered as skew angle. Similarly, to reduce the time complexity of converting black pixels to Hough space is presented by Pal et al. [PC96]. They considered the lowermost and uppermost pixels of some characters, and then those selected pixels are subjected to HT for skew angle detection.

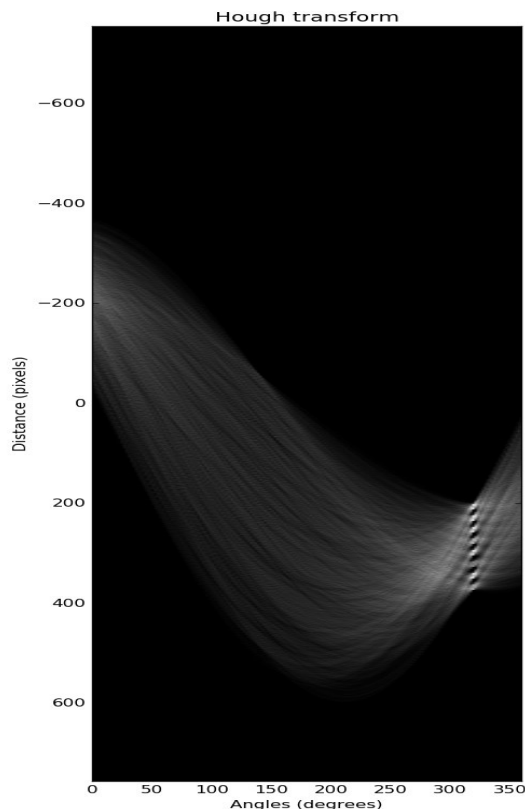


Figure 8.4: *A Hough space of input image; depicted in the Figure 8.3. The maximum peaks can be seen along 341° which is equivalent to -19° in a clockwise direction.*

These approaches are known for high accuracies but are relatively slow. The computational cost is high because each black pixel needs to be transformed to the Hough space. In addition to that, in the presence of noise, these methods become very slow, and in the case of sparse text, it is very difficult to detect correct skew angle. Furthermore, in the presence of graphics and figures, false lines are contributed as candidate lines for final skew detection. Singh [SBK08] presented a pre-processing step using block adjacency graph (BAG). The method is efficient but only works better for Roman scripts. Figure 8.3 depicts an input image (a) and detected Hough lines (b), while the Hough space of the same input image is shown in Figure 8.4.

8.2.3 Fourier Transformation (FT)

Fourier Transformation (FT) has also been used for skew detection and correction. The idea of Postl [Pos86] was extended by Peake [PT97] for skew detection and correction. In this method, an input image is first divided into 4 quadrants. Then each quadrant is transformed to a 2D Fourier's magnitude spectrum. Figure 2.10 shows the 4 blocks/quadrants of an input image and their respective Fourier's spectrums. Peake [PT97] mentioned that

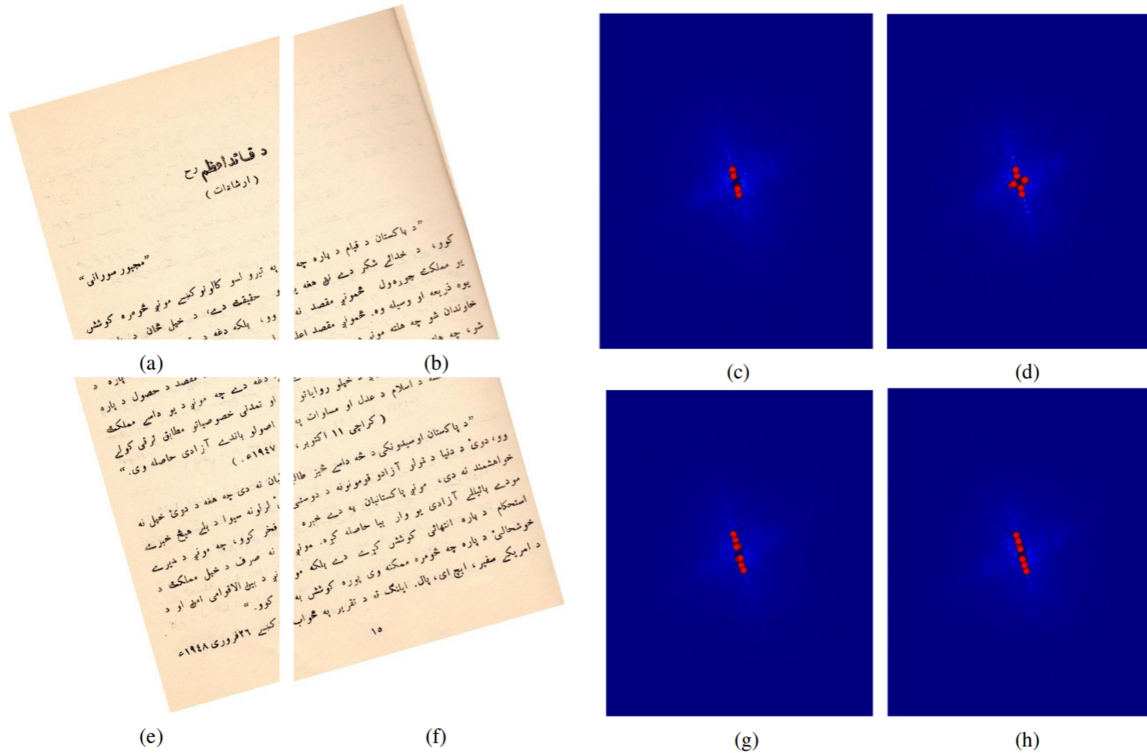


Figure 8.5: *The figure (a), (b), (e), and (f) are the four quadrants of an input image. While, (c), (d), (g) and (h) are their corresponding Fourier magnitude's spectrum. The red color represents the peaks in the FT's spectrum.*

the intensities alignment in the Fourier space represents the skew of the image. Thus, a directional vector is computed, in which the sum of the magnitudes from the magnitude spectrum for each angle is stored [KJ13, WR14, Fab14]. The maximum value in the directional vector represents a skew angle.

FT based methods are also computationally expensive, and in the presence of graphics contents, the skew angle detection becomes very difficult.

8.2.4 Axis-parallel Bounding Box (APB)

An approach presented by Shafii [SSA15] based on axis-parallel bounding box (APB) has reported good results for skew detection and correction. The APB method is conceptually based on the heuristic, that if the contents of a document can be *bounded in a rectangular form*, then the area of the bounding box will be minimum, only if there is no skew in the document. Figure 8.6 illustrates this concept. In the APB method, the area of the bounding box is computed by putting the values of the points in using Equation 8.2,

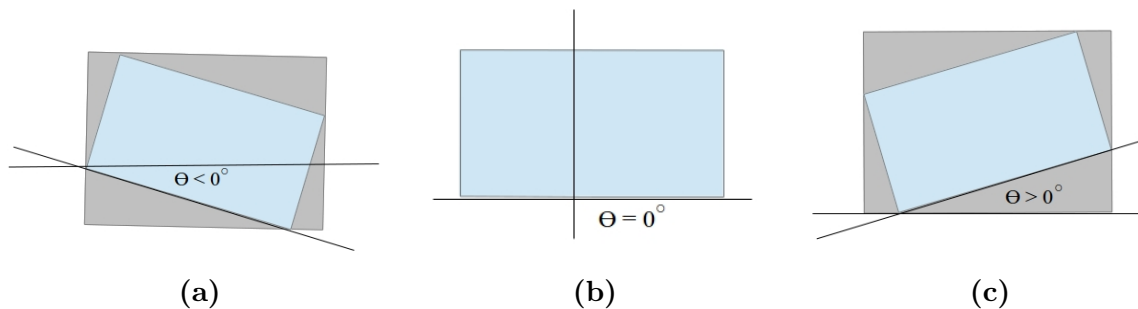


Figure 8.6: *The light blue area represents text contents in a document. Figure (a) has a document with a skew in a clockwise direction. Similarly, figure (c) has a document with a skew angle in counter-clock direction, while document (b) has no skew at all. As shown in figure (b), if there is no skew in the document, then the area of the bounding box might be reduced to gray area.*

These points are minRow, maxRow, minCol, and maxCol.

$$Area = (maxCol - minCol).(maxRow - minRow) \quad (8.2)$$

Figure 8.7 also depicts these points, where for this particular example minRow and minCol are just one point. Thus, if the area of the bounding box is reducing in a certain direction, then it means that the skew angle exists in that particular direction; otherwise there is no skew in the image. After this, the process of rotation and checking the area of the bounding box in a detected direction is repeated, until to get the minimum area for the bounding box. The angle, at which the minimum area is obtained, is the exact skew angle. This method has been found comparatively reliable for text documents which can be bounded in a rectangle box. Further, APB method is also independent of scripts and could work both for graphics plus text contents. However, this approach is highly dependent on noise and mainly requires boundary/border removing strategies as a prerequisite.

8.3 Datasets

The experiments are carried out on three different datasets; (I) Document Image Skew Estimation Contest (DISEC'13)[PGLS13], (II) Tobacco800 [LAA⁺06][AAF⁺06] and (III) Pashto real images. The DISEC dataset contains 200 skewed images along with corresponding skew angles as ground truth. The DISEC dataset is considered as a rich and specialized dataset having skewed images of different scripts and contents variations. However, almost all images are noise free and do not have borders artifacts.

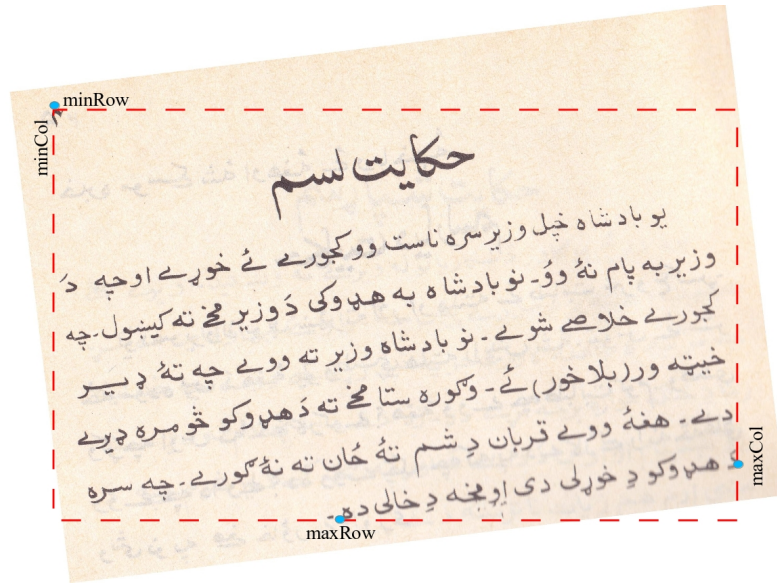


Figure 8.7: Area of a bounding box is computed by points *minRow*, *minCol*, *maxRow*, and *maxCol*.

The other two datasets Tobacco800 and KPTI are also evaluated to check how our proposed method performs on diverse contents. There are 50 images selected from Tobacco800 dataset, while from Pashto dataset 150 images are evaluated for skew detection and correction. Pashto real images are taken from Katib’s Pashto Text Imagebase (KPTI) [AAR⁺16]. The contents of Tobacco800 and KPTI datasets are having a lot of noise and border artifacts. The presence of noise and border artifacts is essential for checking the robustness of our proposed method towards noise and borders.

8.4 Proposed Method

We address this issue by exploiting the logical structure of the scanned documents. Figure 8.8 (a), for example, shows an image containing text and graphics. In almost all cases, text lines are parallel to each other. It means, that despite being having a skew in a document, if the detection of lines comes from the region where a text is located, then the detected lines will also be parallel to each other. Unlike this, if the response of the detected lines comes from other than text regions, then these lines will have random values for their corresponding slopes. Intuitively, the response of parallel lines from the text region gives us a clue, such that maximum parallel lines can represent the set of true-line. In Figure 8.8 (b), the red lines indicate all lines detected initially by our proposed method (before suppressing false-lines). Thus, using parallelism, we can distribute these lines into different clusters, and we found, that the cluster that contains maximum lines, represents true-lines. Here in the Figure 8.8 (c);(i.e. after suppressing false-lines). We can see, that the lines detected in text region and a line detected in the lower portion

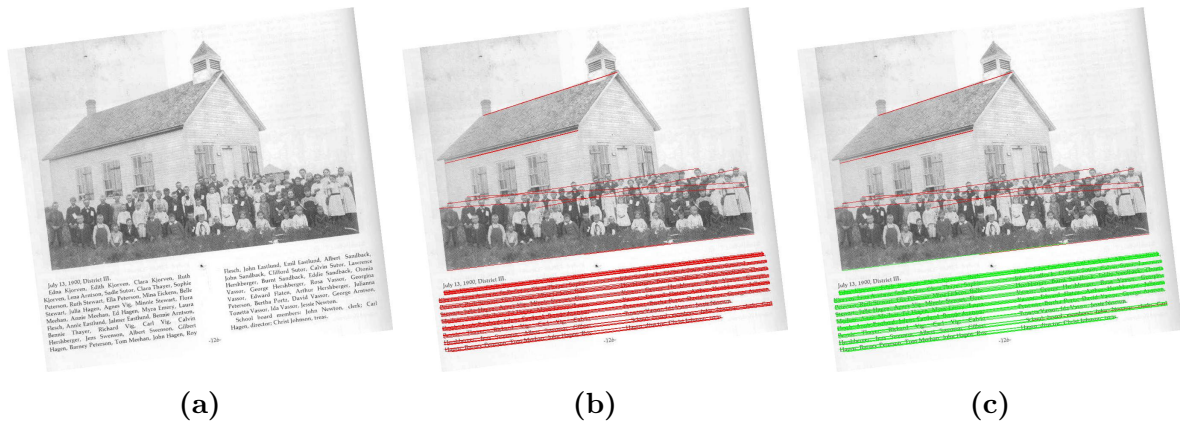


Figure 8.8: *An input image (a), with all detected lines (b) and the figure (c) with true-lines as green and false-lines as red.*

of the graphics in the document are strong candidates for the true-lines. It is the result of our proposed method. To visualize the results, we marked true-lines in *green* color and all other lines in *red* color.

8.4.1 Description of our Proposed Method

Initially, the input image is normalized with respect to its width by fixing its width to 1250 pixels, while the aspect ratio is maintained. If w and h are width and height of the original image, and $nWidth = 1250$ (*normalized new width*) then $nHeight$ (*normalized new height*) is achieved using equation 8.3.

$$nHeight = \frac{(nWidth * h)}{w} \quad (8.3)$$

After that, we apply Otsu binarization [Ots75] method to binarize the normalized image. Although there are many advance techniques [PPEBZM⁺15, APPS⁺15, LFA15] but for our purpose Otsu is sufficient. Concerning noise removal, we do nothing as our proposed method is significantly robust against noise/borders. To detect lines, we used Probabilistic Hough Transform PHT [Ste91]. PHT is the optimized version of simple HT and thus is very fast compared to HT. The parameters used in PHT are $\rho = 1.0$ (which is the distance resolution of the accumulator in pixels), $\theta = \pi/360$ (which is the angle resolution of the accumulator in radians), $minimum_line_length = nWidth/4$, and $max_LineGap = 50$. These parameters are fixed for the entire experiments. The detected lines are then grouped into clusters using their same *gradients or slopes* value. It means, for each cluster we will have a set of lines parallel to each other. Let suppose, a line has two points (x_1, y_1) and (x_2, y_2) , then the slope or gradient " m " can be computed using

equation 8.4, providing ($x_1 \neq x_2$).

$$m = (y_2 - y_1)/(x_2 - x_1) \quad (8.4)$$

If the absolute difference of slopes is less than 0.0001, then their lines are grouped into one cluster. The cluster having the largest number of lines is chosen as the set of true-lines. Such true-lines are shown in the Figure 8.8 (c) with green color. Among the true-lines, the largest line is chosen. Let suppose the largest line has two points (x_1, y_1) and (x_2, y_2), then its corresponding angle along the horizontal axis can be computed by equation 8.5.

$$A = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (8.5)$$

The resultant angle is a skew angle. Algorithm 1 illustrates this process through a pseudo code.

Algorithm 1 : The algorithm takes an input image I and return a skew angle as A .

```

1: function GETSKEWANGLE( $I$ )
2:    $cI = \text{getCannyEdges}(I)$ 
3:    $hPLines = \text{getHoughProbalisticLines}(cI)$ 
4:    $chPLines = \text{getCleansLines}(hPLines)$  ▷ remove tiny lines.
5:    $allClusters = []$ 
6:   for  $ln1$  in  $chPLines$  do
7:      $vote = 0$ 
8:      $cluster = []$ 
9:      $cluster.append(ln1)$ 
10:     $m1 = \text{getSlope}(ln1)$  ▷ using equation 8.4.
11:    for  $ln2$  in  $enumerate(chPLines)$  do
12:       $m2 = \text{getSlope}(ln2)$  ▷ using equation 8.4.
13:      if  $abs(m1 - m2) \leq 0.0001$  then
14:         $vote ++$ 
15:         $cluster.append(ln2)$ 
16:      end if
17:    end for
18:     $allCluster.append(vote, cluster)$ 
19:  end for
20:   $trueCluster = \text{getTrueCluster}(allCluster)$  ▷ get the cluster having maximum votes.
21:   $line = \text{getLine}(trueCluster)$  ▷ Get any line from true cluster.
22:   $return A = \text{getAngle}(line)$  ▷ calculate the angle of the given line wrt x-axis.
23: end function

```

Table 8.1: Evaluation on DISEC [PGLS13] dataset.

	GEA%	Av-Time [s]
PP	48.0%	9.57
HT	13.0%	2.18
FT	4.0%	3.93
APB	77.0%	8.79
Our Method	82.5%	1.83

8.5 Evaluation Metric

The evaluation protocol (only for DISEC dataset) is considered as good estimation accuracy (GEA) in %. The GEA is obtained by error deviation (ED). ED means, that to what extent the detected angle deviates from the ground truth value. The GEA for a de-skewed image will be 1 if the absolute value of ED is less than 0.2° , otherwise 0. Time complexity is also evaluated as average time taken in seconds [s] to de-skew an image.

Further, as the other two datasets (Tobacco800, Pashto images) do not have skew angles as ground truth, therefore, the evaluation has been made on the basis of visual appearance.

8.6 Results and Discussion

The results show that our proposed method outperforms on all three datasets. The results on DISEC dataset are shown in the Table 8.1, where our proposed method achieved an accuracy of 82%, while APB method came as runner-up and have achieved an accuracy of 77%. It is also observed that if we relax the threshold for ED that is 0.2° (default) up to 0.5° ; then the accuracy reached to $90 + \%$ for our proposed method. Figure 8.11 shows images from DISEC dataset that are successfully de-skewed by our method.

Results on Tobacco800 and Pashto datasets show that the overall success rate is 99.3%, and the average time taken by our method to de-skew an image is 0.98 seconds. In this context, our proposed method gives better results meeting the both success rate and average time as evaluation criteria. The results are shown in Table 8.2.

As mentioned in section 8.2.4, the APB method is very sensitive to borders and noise, while our proposed method is robust to such noise. A typical example of a failure of APB method is shown in Figure 8.9, where a failure is encountered due to the very small black spot exists at the top left of the input image. Due to such noise, the minimum area for a bounding box is obtained with an angle, which is not the exact skew angle. Another,

Table 8.2: The success rate in % and average time to de-skew an image in seconds [s] on Tobacco800 and Pashto text documents.

	Success Rate%	Av-Time [s]
PP	58.0%	10.21
HT	80.0%	1.68
FT	66.34	2.42
APB	90.0%	6.15
Our Method	99.3%	0.98

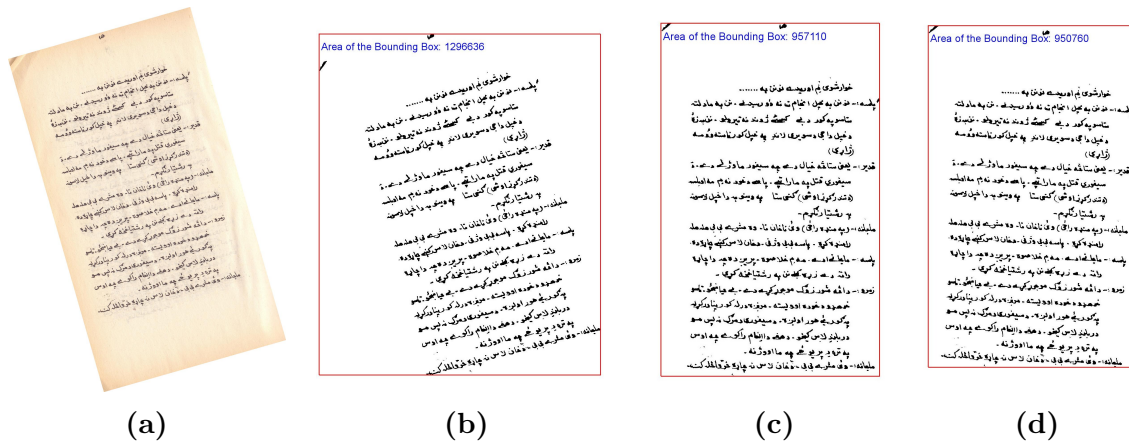


Figure 8.9: (Failure case of APB): An input image (skewed image) 8.9a which has an exact skew angle of -14° . In this case, the smallest area obtained in Figure 8.9d is 950,760 (shown in blue text on top of figures). However, the Figure 8.9c is closer to our expected result. In this case, the resultant skew angle is -17.12° , which deviates about 3° from the original de-skew angle.

APB method has a limitation, if text contents in a document could not be bounded in a rectangular form, then APB method fails to work for such text document. Figure 8.10 shows a representative of such images that could not be bounded in a rectangular shape. On another hand, our proposed method is robust to graphics contents and layout of the document. The main advantage of our proposed method is its invariant nature towards noise and borders. In this work, Arabic and Latin scripts are examined, and it is found that our proposed method is also invariant to different scripts.

More importantly, our proposed method is time efficient. This efficiency is achieved by implementing PHT and avoiding any intermediate rotation of the image. The image is only rotated for final de-skewing. It reduces much time in skew detection and correction.

Further, the images which are chosen from Pashto text, differ in contents, which ultimately proves the versatility of our proposed approach and provide reliability to handle Latin as well as Arabic like scripts. Further, the specs of our PC are lower than the PC

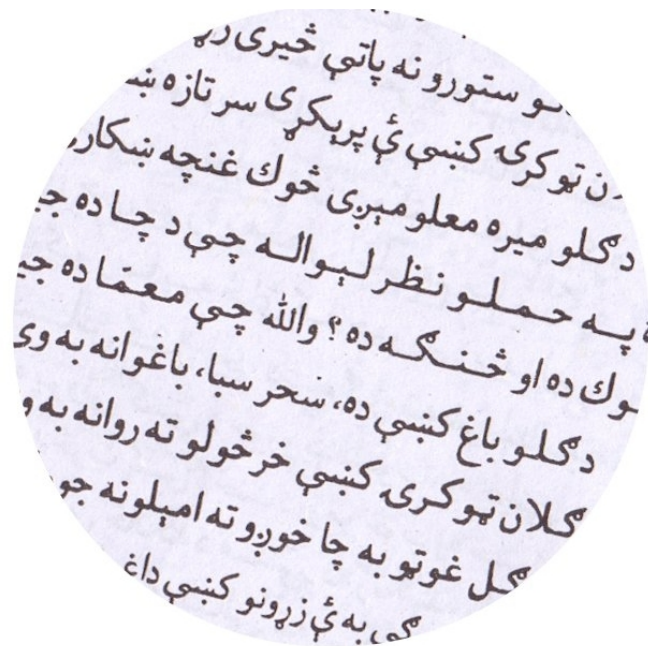


Figure 8.10: *The document which could not be bounded in a rectangular form cannot be de-skewed by APB [SSA15].*

used in [SSA15]. Therefore, we believe, that the average time for skew detection might be lesser than axis-Parallel Bounding Box method (APB) [SSA15]. Some typical examples of de-skewing images with our proposed method are shown in Figure 8.11.

8.7 Conclusion and Future Work

In this chapter, we have presented a novel approach for skew detection and correction in scanned documents. The proposed approach using a heuristic that the text-lines in a document are mostly parallel to each other. Thus, the cluster having a maximum number of parallel lines can give us skew angle. Our method is time efficient and gives us better results. It is also found, that our proposed method is robust to noise and borders. We have used PHT and avoid the use of some morphological operations, like rotation before the detection of the correct skew angle. It saves much time in computing skew angle of the document. It is empirically shown through real scanned documents, that our proposed method is suitable for Latin as well as Arabic like scripts.

Our method is used for de-skewing all images of KPTI dataset and has proven its effectiveness by achieving better accuracy. This work is also acknowledged and published in [ARA⁺16]. In future, the failure cases will be analyzed and will be focused to improve this method.

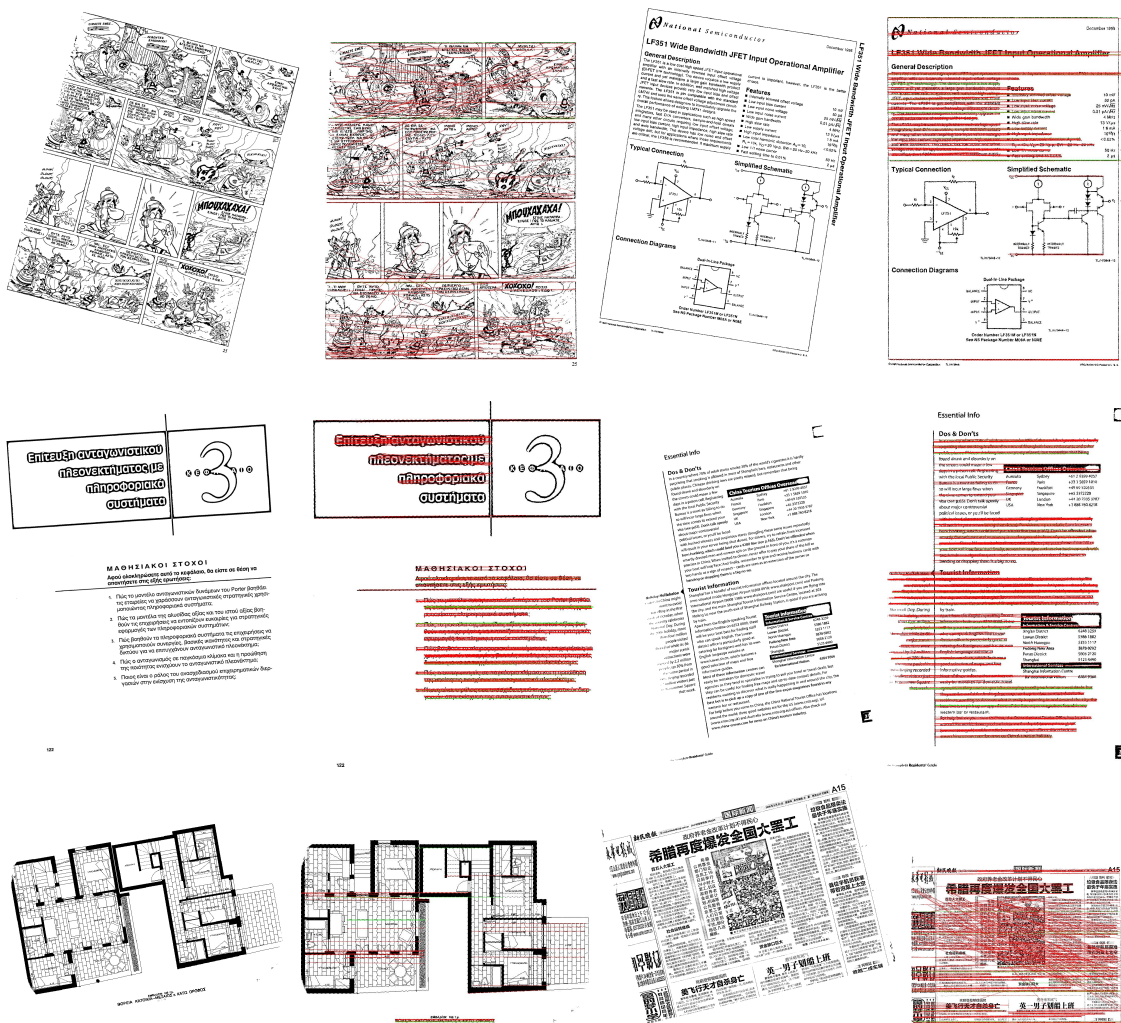


Figure 8.11: Successfully de-skewed images corresponding to skewed images at adjacent left. These images are taken from DISEC-13 dataset.

Textline Segmentation

This chapter provides an overview of the text-line extraction method. This method is one of the practical contributions of this thesis regarding pre-processing step. This chapter states the motivation and explains why we need a novel method that can segment text-lines in KPTI dataset. This chapter further describes the proposed method and reports its results on the KPTI dataset. The comparison has been mad with the state-of-the-art method. Finally, this chapter is comprehended with discussion section.

9.1 Motivation

The performance of OCR systems is mainly dependent on how clean and relevant data is fed as input. Existing research mainly used text-lines as input entities [Gra12b, GLF⁺09]. To facilitate OCR process, the extraction of clean text-lines caused a remarkable research in the field of text-line segmentation [RKC14, RRC15, AKB⁺12, KAB⁺12, ABK⁺12, AKB⁺13, PPALCB16]. However, this is not always easy, because one has to be careful about skew, curls, script variations, noise, and text-lines with different sizes.

In this chapter, the specific problem, which is being more focused is the line segmentation of large scale headings and titles. Such headings and titles abundantly exist in Arabic script. The current approaches for text-line extraction are specialized regarding script, contents, and layout. Therefore, their usage could not be generalized to such specific issue. The real text documents often contain images with large scale heading and titles. Such text lines could not be extracted precisely with state of the art methods. The state of the art approach, which is described in [RKC14, RRC15](winner of ICDAR 2013 handwriting text-line segmentation competition), shown not to be reliable for Arabic text documents with large scale headings and titles. The problem is pictorially shown in Figure 9.1, where a binary input image (a) is given to the state-of-the-art method [RKC14], and the result (b); (different colors show distinct text-lines); incorrect segmentation of title/heading



Figure 9.1: Input binary image (a) and the segmentation result (b) from the method [RKC14] (with default parameters). Note that, the page number is merged with a title, and the title is split into two lines.

line can be seen. This flaw motivated us to develop a new method, that can handle the Arabic text document and can effectively extract the text-lines.

In this work, the Pashto language is considered as a test case, where line segmentation approaches are evaluated against scribed/handwritten materials of KPTI dataset. More specifically, this work focuses on text-line extraction methods and proposes a new method for text-line extraction. The proposed method is equally applicable for Pashto as well as Arabic scripts.

Our method is based on convolving the input image by using a Hanning kernel. It smooths the accumulative response for Horizontal Projection Profile (HPP). The proposed method is tested on 180 real Pashto handwritten text images, which are acquired through a scanning process.

9.2 Related Work

We can categorize the related work based on the foundation concepts, such as; Projection Profile (PP), Hough Transform (HT), Connected Components (CCs) with bottom-up clustering approach, and use of HMM or artificial neural networks (ANN) on statistical



Figure 9.2: The segmentation results of Ryu's method [RKC14] on Pashto text documents. The titles are split into two or three text-lines. In these cases, the tuning of the parameters α and β does not work.

features. These techniques are evolved as they tackled the different scripts and layouts. Here, the most important text-line extraction methods are discussed regarding their pros and cons.

A piece-wise PP based approach is presented in [ASS07], which track/connect the valleys in PP. The tracking is done by considering different conditions of the problems. Once, the valleys are connected, then the two adjacent lines could be separated [ASS07]. This method is robust against many scripts and could work for handwritten documents as well. However, if a text-line generates more than one valleys, then it splits that particular line into two lines.

An approach based on partitioning the CCs is introduced by Louloudis [LGPH08], where the most likely CCs are partitioned into their text-lines. The approach exploits the average character height to decide the candidacy of a CC regarding a text-line. They used HT to detect general line tendency and improved the results in a post-processing

stage. This approach is robust against a limited skew and is better about isolating different lines. However, it is not checked on the documents, which contain Arabic like a script or which have text-lines with different height/size.

Another approach based on adaptive local connectivity map (ALCM) with steerable directional filter is presented by Zhixin Shi [SSG09]. The ALCM emphasizes the intensity of black pixels along the direction where maximum (CCs) are present. The approach is tested for Arabic documents and perform well in extracting text lines.

Bukhari et al.[BSB08, BSB09] presented a text-line segmentation approach, which is based on active contour (snake) over the ridges. The ridges are found by applying multi-oriented anisotropic Gaussian filter banks. This approach is robust against skew, curl, and noise. However, due to the use of Gaussian filter bank, it is computationally expensive.

Nicolaou et al. [NG09] presented a method based on shredding/stripping a document on the whitest regions. It is achieved by blurring the image first. In the blurred image, the white pixels are then recursively traced from left to right and from right to left. It results into strips of white pixels, where the positions of these white strips are used to segment/shred a document into different text-lines.

Bosch et al. [BTV12] presented a model based on statistical features for text line analysis and detection. The method used Hidden Markov Model (HMM) to learn the different states of the text-lines. The text lines are annotated about their different types (i.e. Normal text Line-region, Inner line-region, blank line-regions, and non-text line-regions). The model is trained and then evaluated on 20 text documents containing French manuscript.

A method presented by Koo et al. [KC12] is based on CCs with bottom-up clustering approach. They used energy function for clustering and introduced a cost function for the interaction of text-lines with the shape of the text-lines. This method is tested on images having Chinese script and outperforms other approaches. However, this method merges shorter text-lines with longer text-lines.

A state of the art approach is introduced by Ryu et al. [RKC14], which is the enhanced version of Koo's [KC12] method. The approach is based on connected components (CCs), this method extracts text-lines by converting the under-segmented CCs into normalized CCs using minimization of a cost function. The approach has won the ICDAR 2013 handwriting text-line segmentation competition. This approach is robust against curls and skew, and performs well in documents having a uniform size of text-lines. However, besides having many plus points, this approach has shown different results while handling Pashto/Arabic text images with large scale titles. In addition to the Figure 9.1, more examples of the main failure of the Ryu's [RKC14] method are shown in Figure 9.2.

We can conclude that PP based methods provide very good results for scanned docu-

ments. However, they are sensitive to skew, curls, and graphics. On another hand, the bottom-up clustering approaches can handle skew and curls lines, but they require script dependent heuristics for how to group CCs into one cluster or how to split CCs among different clusters. In short, they need tuning of parameters to handle different patterns and layout of the text documents.

9.3 Problem Definition

Text documents often contain headings and titles. The size of these titles is usually larger than other text in the document. This generic characteristic presents a challenge in the Pashto language regarding text-line extraction. See Figure 9.3, where the response of HPP as a plot along with a small piece of text are shown side by side. In this example, the title's text (one text-line) presents 5 peaks. It can be seen that the titles and headings with a large size are indistinct, and provide more peaks. There are two reasons, (1) usually, the number of characters or words are very limited in the heading/titles, and could not provide enough accumulative response to show the distinct peaks and valleys, and (2) the shapes of Pashto words/characters are irregular and provide many responses for peaks.

To tackle this problem, we evaluated state of the art method [RKC14] on Pashto scanned documents (It is referred as Ryu's method in the remaining text). However, the titles and headings are not correctly segmented by the Ryu's method with its default parameters.

The Ryu's method used two parameters α and β . The first parameter α adjusts the allowable size of a connected component (CC) (Default 30.0), while the parameter β allows adjusting the size of the partitioned CCs (Default 5.0). We tuned these parameters to find the optimal values. However, we could not find any optimal combination that could help in this particular case.

9.4 Methodology

Our proposed method is based on HPP. The HPP is simply the sum of pixels values along an x-axis. Let suppose horizontal PP (*HPP*) of a gray-scale image \mathbf{I} , with height \mathbf{h} and width \mathbf{w} , computed by equation 9.1.

$$HPP(v_{j=0}^h) = \sum_{i=0}^w (I(x_i, v_j)) \quad (9.1)$$

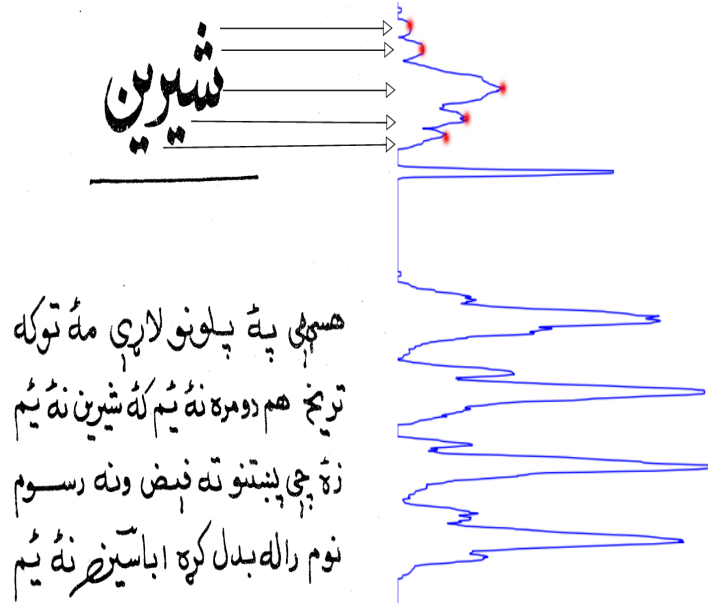


Figure 9.3: *The input image with a plot of horizontal projection profile at right. The title line presents 5 peaks.*

The input image is convolved with a filter based on a Hanning window [BT58]. This convolution makes the peaks and valleys more distinctive. The Hanning window \mathbf{H} with a certain length L with the weighted cosine values can be defined by equation 9.2.

$$H(n) = 0.5(1 - \cos(\frac{2\pi n}{L-1})), 0 \leq n \leq L-1 \quad (9.2)$$

Thus, the input image is convolved with a Hanning window of size 80 along the vertical axis. The size 80 is empirically found optimal and is fixed for the entire evaluation. Figure 2.9 illustrates the visual impact of the proposed convolution on the input image as well as on the HPP. Further, as shown in the Figure 2.9, the plot of the *HPP* is now more descriptive and smooth regarding text-lines extraction.

The rest of the process is done on a one-dimensional array of *HPP* having length h (i.e. the height of the input image). Processing one-dimensional array makes our method more efficient compared to the other text-line extraction methods. Now for the extraction of a text-line, we need to find the top row and the bottom row. Please refer to the plot of HPP in the Figure 9.4, if an index of a *peak* represents a center of a text-line, then the indexes at which *Up-Minima* and *Down-Minima* occur, are the top row and bottom row respectively. Logically, this could be achieved by traversing the HPP array for Up-Minima and Down-Minima providing a certain peak. Therefore, we introduce an algorithm which can take the index of a peak and return the index of up-Minima or down-Minima. Algorithm 2 describes the detail pseudo-code.

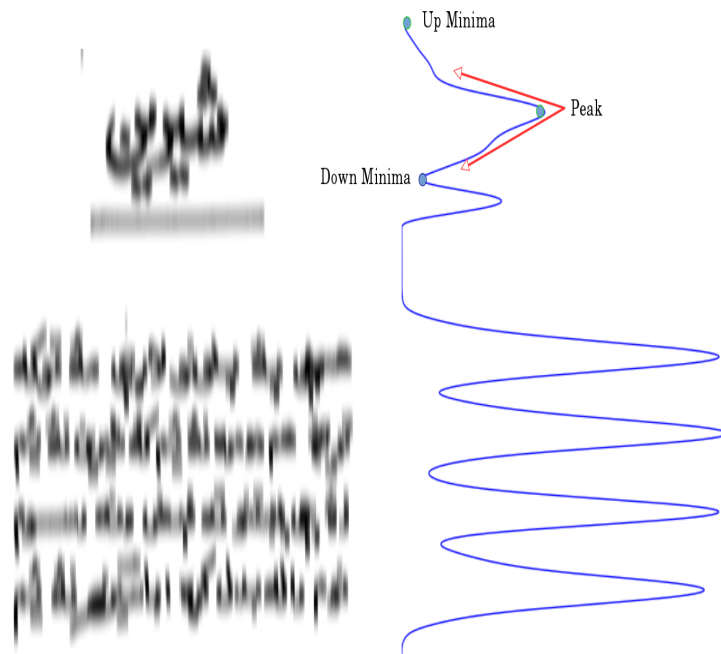


Figure 9.4: An input image after convolving with Hanning window of size 80 and a plot of its corresponding horizontal projection profile. The peaks are now smooth and clear and define the text-lines with more details. The green spots represent the up-Minima, down-Minima, and peak, while the red arrows represent the direction of the traversing.

Algorithm 2 Traverse the horizontal Projection Profile (HPP) array A for Minima, where other inputs are; I : peak's Index, D : directional flag (+1 for down-minima and -1 for up-minima).

```

1: function FINDMINIMA( $A, I, D$ )
2:    $i = I$ 
3:    $Minima = 0$ 
4:   while  $A[i] > A[i + D]$  do
5:      $i = i + D$ 
6:     if ( $i > 0$  and  $i < (len(A) - 1)$ ) then
7:       if  $A[i] \geq A[i + D]$  then
8:          $Minima = i + D$ 
9:       end if
10:    else
11:      return  $Minima$ 
12:    end if
13:  end while
14:  return  $Minima$ 
15: end function

```

Once we compute the up-Minima and down-Minima for each available peak, we can crop the text-lines from the original input image. To crop an image from another image, we need two coordinates, (i.e. (x_1, y_1) and (x_2, y_2)). In our case, y_1 is up-Minima and y_2 is

down-Minima, while x_1 is 0, and x_2 is equal to w (width of the input image). However, in general, x_1 and x_2 could be the first and last columns of a text block respectively.

9.5 Description of the Dataset

The dataset being used in this work is the subpart of KPTI dataset[AAR⁺16]. This subpart contains 180 images of Pashto real text documents. The selected images contain almost all the layout variations, for example, poetry, essay, religion, and reports. However, documents with larger headings and titles are of more interest. Therefore, their instances are more compared to other specific layouts. The total instances of the correct text-lines are 3,596. The images are first normalized to the width of 1250 pixels, while the aspect ratio is kept maintained. After normalization, the images are binarized with Otsu's method [Ots75]. It is noteworthy that we didn't correct any skew in these images and there exist normal skew in between angle $\pm 2^\circ$.

9.6 Evaluation

To evaluate our proposed method, we used two criteria. (1) The line-segmentation Accuracy in %. (2) The average time (AvTime) in seconds "s", that is taken for extracting the text-lines per document/page. In order to calculate Accuracy, we used the same scheme already reported in [ASS07], and is given below.

1. If a single connected component of a line is segmented into another line, such error is counted as 2 line errors.
2. If a line is split into 2 or more lines, then it is counted as 1 line error.
3. If n lines are merged together, then it is counted as n line errors.

We summed all the errors as E and computed the total Accuracy using the following formula 9.3.

$$Accuracy\% = 100 - \left(\frac{E}{TotalTextLines} \right) * 100 \quad (9.3)$$

Where Total Text-Lines are 3596 in this case.

Table 9.1: Results of text-line extraction methods

Method	Accuracy%	AvTime [s]
Ryu's Method[RKC14]	98.38	0.93
Our Method	99.30	0.23

9.7 Experimental Results

To evaluate the performance of our proposed method, we carried out two experiments on our dataset. In the first experiment, the Ryu's method is evaluated, where we used the application¹ publicly available by the authors [RKC14]. The application takes a binarized image as an input and returns a raw data file. The raw data file contains a labeled image. In this labeled image, all pixels of a correctly segmented text-line are labeled with the same value. In this experiment, the time taken by the application to return the labeled file is the required time for segmenting a particular input image into text-lines. The time taken for visualizing the segmentation results is excluded.

The second experiment is based on an evaluation of our proposed method. Our method can be used with two options. In option one, it returns the cropped text-lines without any visualization. In option two, our method not only returns the segmented text-lines but also returns the image with colored lines to visualize the segmentation of text-lines. The both methods (Ryu's and our proposed method) use the same technique for visualizing the text-line segmentation. In this visualization technique, each adjacent text line should have a distinct color, if a single line has two or more color, it means there is over segmentation. Similarly, if two different but adjacent lines have the same color, it means there is under segmentation. The time complexity of our method is measured without using visualization option. Table 9.1 reports the results regarding line segmentation Accuracy in % and average time (AvTime) in seconds [s].

Figure 9.5 shows the examples of successful extraction of text-lines using our method. These are the same instances where the Ryu's method fails. The results and visual examples of our method show that regarding the extraction of larger titles/headings as text-lines, our proposed method outperforms the Ryu's method.

¹<http://ispl.snu.ac.kr/youjw/Textline/>



Figure 9.5: The text-lines segmentation results of our proposed method on Pashto text documents.

9.8 Limitations

During the evaluation, it is observed that our proposed method is dependent on skew and graphics. The reason is the foundation concepts of the projection profile (PP). As in PP techniques, the accumulative response of black pixels is summed vertically or horizontally, but in the case of skew, the accumulative response for peaks and valleys is not that much descriptive for the isolation of text-lines boundaries. Thus, images having skew angle more than $\pm 2^\circ$, could not be segmented properly. Therefore, we recommend that first, the images should be de-skewed properly.

Another, the text documents of KPTI dataset are graphics free and are fed to the proposed method based on as they are. However, for text documents which contain figures, we suggest using text blocks segments instead of whole document.

9.9 Conclusion

In this chapter, we presented a text-line extraction method, which is based on Hanning window smoothing technique plus Horizontal Projection Profile (HPP). We have evaluated the proposed method on real Pashto text images. In this chapter, the problem that is being more focused is the text-line extraction of headings or titles with a larger scale in Arabic like scripts. The most sophisticated technique like Ryu's [RKC14] method fails to segment such text lines. In such cases, our proposed method is much better achieving both accuracy and efficiency compared to Ryu's method. As our proposed method based on HPP, therefore, the only limitation of our method is that it needs de-skewed images, while the Ryu's method is stable against curls and skew.

Part V

Pashto OCR

Scale and Rotation Invariant OCR for Pashto

This part of thesis overviews the OCR module of the entire DIA system and contains two chapters. The first chapter describes the scale and rotation invariant Pashto OCR using ligatures. It first states the motivation and then compares the state-of-the-art methods for scale and rotation invariant text recognition. It provides a detailed analysis regarding HMM, SIFT descriptor, and MDLTM based techniques, and reports on their performances for Pashto ligatures having scale and rotation variations.

The second chapter of this part proposes a benchmark based on deep learning paradigm using MDLSTM architecture. The proposed system evaluates and introduces the first ever baseline for Pashto real text-lines.

10.1 Motivation

Variation in text documents is inevitable. These variations mainly exist due to different shapes, fonts, scales, rotations, etc. The more we have these variations, the more complexities we get in the recognition of text documents. Particularly the scale and often the rotation variations exist in real text images. Figure 10.1 shows such variations in the Pashto text document. In these cases, usually the text segments are too short, and it is more appropriate to consider them as a whole, instead of recognizing them as a sequence under temporal classification problem. Therefore, ligatures instead of characters are appropriate choices to be used as recognizable units or target classes. Here, again we recall that ligatures are having the most stable connected shapes, and retain their most salient features all the time. The importance of ligature based classification particularly in this scenario is already explained in section 2.4.2, chapter 4. The concepts related to Pashto's words and their constituent ligatures are already explained in chapter 4. In addition to that, Figure 10.2 again recalls the concepts of Pashto words and ligatures.

The objective of this work is to develop a robust OCR system for Pashto text, that is

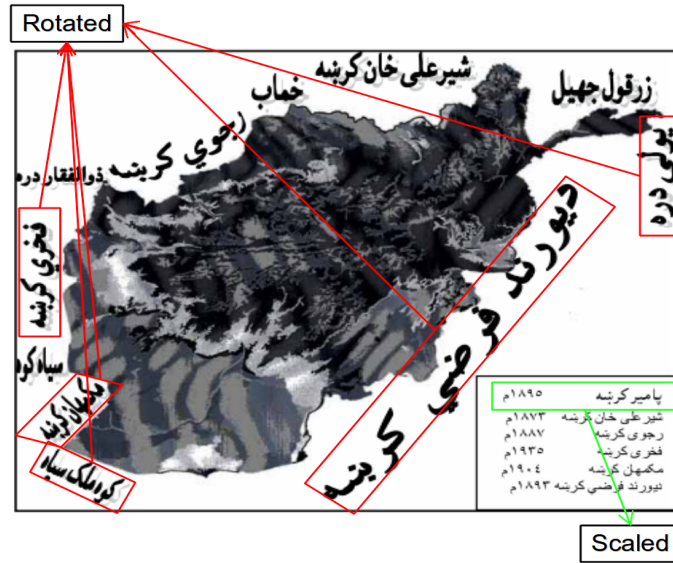


Figure 10.1: The scale and rotation variation in real world data. Most of the Pashto books; based on geographic and historical contents, contain Pashto text with scale and rotation variations [AAR⁺ 15a].

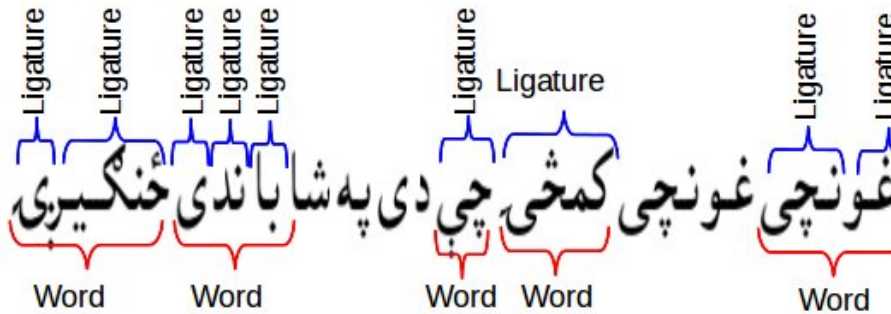


Figure 10.2: The concept of Ligatures/sub-words and words are shown. Sometimes, a ligature is also a word [AAR⁺ 15a].

invariant to scale and rotation variations. For this purpose, the existing state-of-the-art OCR techniques like LSTM, HMM, and SIFT are evaluated for the recognition of Pashto ligatures. The empirical analysis is based on assumptions. Such that, if we have already segmented segments, and each segment contains only one Pashto ligature, then we can develop an OCR system that can recognize segments as ligatures along with scale and rotation variations.

The database used in this work is a Pashto ligature database, we have already introduced it in section 6.3.2 and is referred as Ligature-Based-III. However, in this chapter, we will describe how we split this entire dataset into a train, test, and validation sets. The results of the experiments show that LSTM based model significantly outperform HMM and SIFT based techniques. The importance of this work is acknowledged and

subsequently published in ICDAR-2015¹. The next section briefly explained the most relevant literature.

10.2 Related Work

Existing research regarding OCR for cursive Arabic scripts can be categorized mainly into two types of approaches. (1) Analytical approaches and (2) holistic approaches [NHR⁺14]. Analytical approaches are based on features that define the typographical rules for the respective script. These features mainly include some language specific heuristics, principals, and geometric rules, that how characters combine and form a word. However, these approaches mostly need atomic segmentation for their efficient performance, while accurate segmentation is another complex issue particularly in Arabic cursive scripts. Furthermore, these approaches are specific and could not be generalized to other scripts. In contrast, the holistic approaches are based on the essence of recognizing a complete word or a sub-word as a whole. Holistic based approaches are well generalized and could be applied easily to other scripts. They differ significantly from traditional segment-and-classify approaches and do not need segmentation before OCR. They can be used as scale and rotation invariant OCR systems.

The most related work regarding the Pashto ligature is presented by Ahmad et al. [AAK10]. They have evaluated simple SIFT based key-points descriptor for the recognition of Pashto ligatures with scale and rotation variations. They provided the comparison of Principal Component Analyses (PCA) and SIFT with respect to scale and rotation cascading. Where SIFT based classifier significantly outperformed the PCA method. They have reported ligature recognition rate of 73%.

Furthermore, Hassan et al. [Hus02] presented a Feed Forward Neural Network (FFNN) as a classifier for Urdu script. They have used second-normal-moments as features. Which mainly contain solidity, axis-ratio, eccentricity, normalized-segment-length, and curvature. The recognition rate was reported as 100%, but the dataset contained only 200 ligatures, where the split for train and test sets was also not reported. Another, Ahmad et al. [AOS09] presented a recognition system based on FFNN, and have reported 70% ligature accuracy ligature.

Pal et al. [PS03] also presented a scale invariant approach for Urdu character recognition. They used the concept of water based reservoir plus topological contours. They were able to recognize basic characters and numerals with an accuracy of 97.8%. Similarly,

¹The work presented in this chapter is mainly taken from our already published work in ICDAR-2015, Ahmad, R., Afzal, M. Z., Rashid, S. F., Liwicki, M., & Breuel, T. (2015, August). Scale and rotation invariant OCR for Pashto cursive script using MDLSTM network. In Document Analysis and Recognition (ICDAR), 2015 13th International Conference on (pp. 1101-1105). IEEE.

Sabbour et al. [SS13] reported a classifier based on a k-nearest neighbor for Arabic and Urdu scripts. They have used shape context and contours as features and achieved about 91% accuracy.

There is a lot of work based on LSTM recurrent neural networks (RNNs) and have shown their dominance and significance in the field of Arabic script recognition [Gra12b, NUA⁺15, UHAR⁺13, RSRvdN13]. However, to the best of our knowledge, LSTM based techniques are not investigated so far for the recognition of ligatures having rotation variation in the domain of OCR systems.

10.3 Dataset

This work evaluates the dataset, that is being referred as Ligature-Based-III in section 6.3.2. It is a synthetic dataset and has 1,000 unique Pashto ligatures. Each ligature has 40 scales variations, and each scale has 12 rotation variations. Figure 10.3 shows 12 rotations variations of a Pashto ligature with a certain scale. These variations resulted in a total images of 480,000 ($1000 * 40 * 12 = 480,000$). The images are further split into train, test, and validation sets according to the following scheme.

- For a training set, we randomly selected 8 rotation instances out of 12 rotation variations, this process is repeated for each ligature and scale respectively.
- For a test and validation sets, we chose two rotation variation from the *remaining* four rotation variations. In which two instances are selected for the test set, and the remaining two are selected for the validation set.

Thus, the training set contains 320,000 images, and test and validation set each contains 80,000. Figure 10.4 depicts the split statistics of Ligature-Based-III dataset.

10.4 Methodology

Scale normalization and slant detection techniques have frequently been used to tackle the issue of scale and rotation variations in OCR systems. Then, why it is important to develop a scale and rotation invariant OCR? There are two reasons to this question, that are given below.

- A method based on holistic principals should learn the complete shape of an object regardless of its scale and rotation.
 - A classifier based on MDLSTM scans the input in all directions, thus establishes a temporal link to its learned features in all direction. It implicitly provides an
-

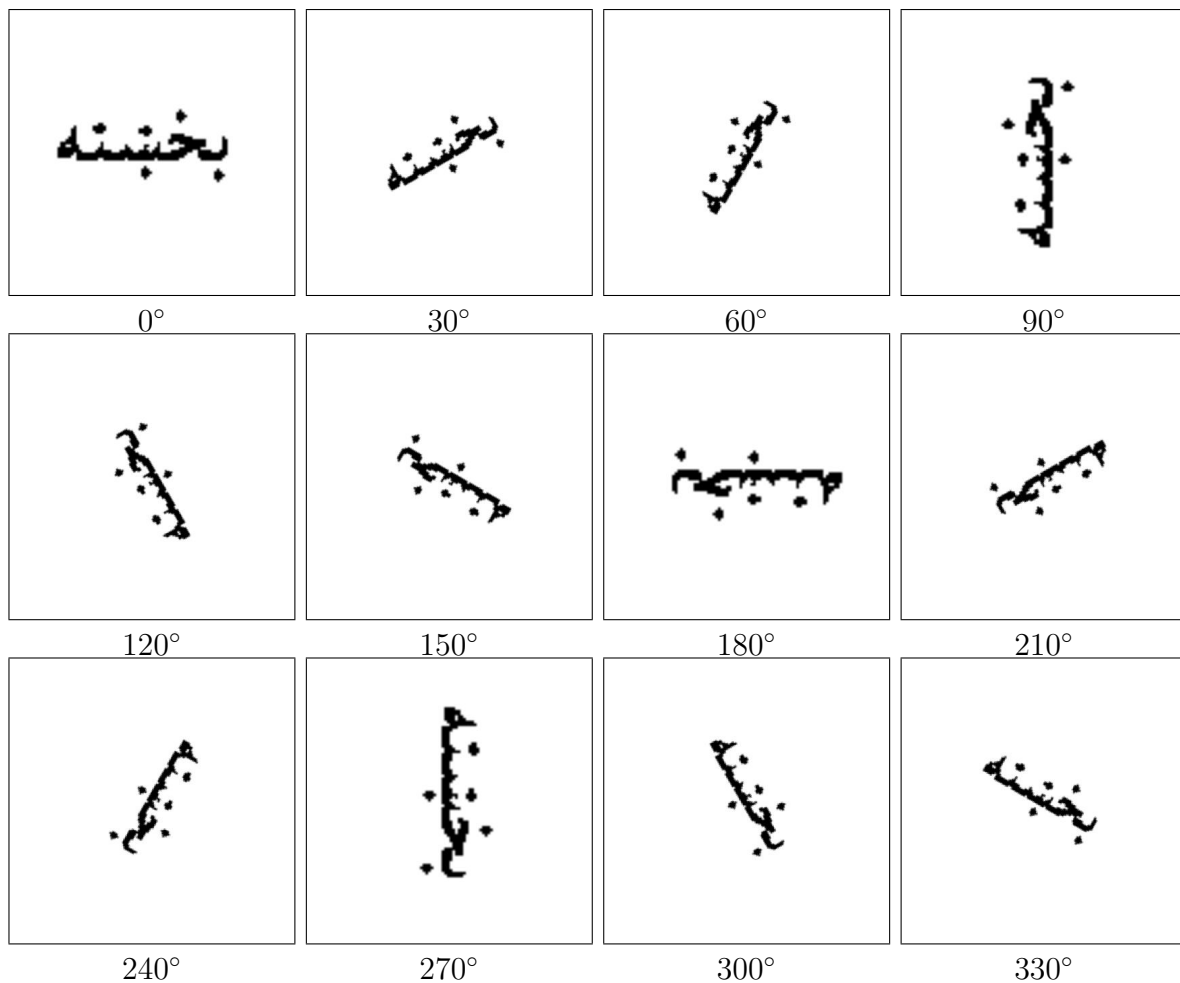


Figure 10.3: *A Pashto ligature and its 12 rotation variations, and their corresponding angles.*

indemnity to feed an input with any scale and rotation.

Furthermore, the text written in Pashto makes it less reliable to extract base-lines as it can be extracted in Latin or Chinese, Japanese, and Korean (CJK) scripts. Due to these reasons, normalization could not guarantee same recognition results not only for same scales but also for different scales [AH17].

Therefore, we present three different recognition systems to investigate the methods based on SIFT, LSTM, and HMM, such that to find which method could better handle the variance of scale and rotation. The detail description of each method is given in the following sub sections.

10.4.1 SIFT Based Ligature Matching

The SIFT features or key-points are one of the well-known features used in state-of-the-art methods for scale and rotation invariant recognition [Dav14]. Therefore, we selected

Split of Ligature-Based-III dataset

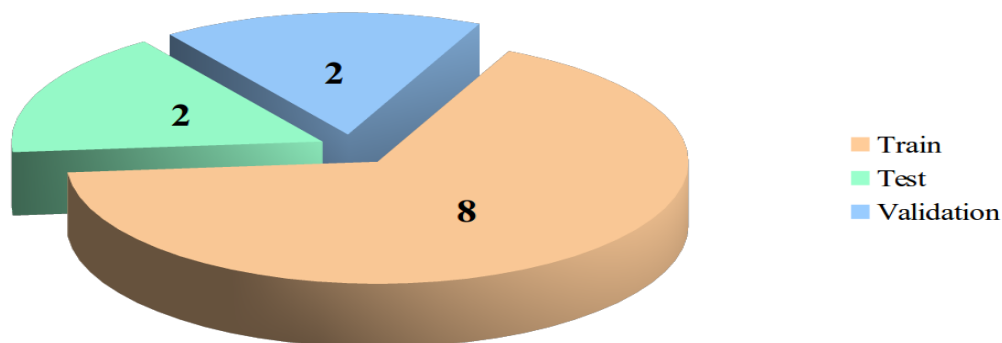


Figure 10.4: *The 12 rotation variations are split into a train, test, and validation sets. Such that 8/12 are randomly selected for a training set, and among the remaining 4, two each are selected for test and validation sets.*

this method as a baseline with the intent that it will maintain its dominance regarding scale and rotation variations in the domain of OCR as well. First, we have extracted the SIFT features from all the images of a train set. It resulted in a huge Train-SIFT-descriptor, in which each SIFT-descriptor is indexed with a label of a specific ligature. While in testing, the SIFT key points are first extracted from an image, and then the Train-SIFT-descriptor is searched for the best match. The two SIFT key-points will be similar if one key-point has an angle less than $distance-ratio \times 2^{nd}$ key-point. The value of the distance-ratio is 0.6.

It is worth mentioning that a single SIFT key-point has a length of 128 bytes [Dav14], and each ligature could easily generate an average of more than 70 key-points. It makes the Train-SIFT-descriptor very huge. In our case, the size of Train-SIFT-descriptor is 22.5GB, and searching such huge descriptor for a single test image takes more than 3 minutes. To speed up the matching process, we used 10 parallel units for the execution of matching algorithm. SIFT based ligature matching is developed using SIFT-Python library².

10.4.2 LSTM Based Recognition

The second system for the recognition of scale and rotation invariant OCR for Pashto ligature is based on MDLSTM (cf. Section 3.3.3). We chose MDLSTM because it can scan the image in all 4 directions, and could learn a complete shape irrespective of their rotation and scale variations.

²<http://www.janeriksolem.net/2009/02/sift-python-implementation.html>

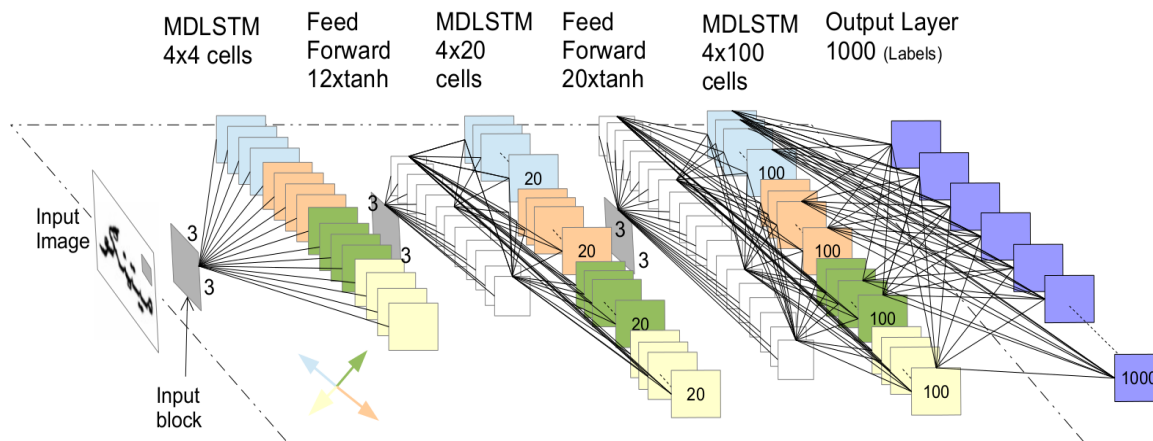


Figure 10.5: *The proposed 2D-LSTM architecture, with 4, 20 and 100 hidden layers. The 4 different colors in hidden layers; represent the direction in which pixel value has been read. Each cell is fully connected with all cells in the next layer. Image source [AAR⁺15a].*

The proposed system works in a hierarchical manner, and Figure 10.5 depicts that proposed system. The model consists of 3 hidden layers, which contain LSTM units, and two *tanh* layers, which are placed in between LSTM layers. The size of the hidden LSTM layers is 4, 20, and 100, and due to Multi-dimensional i.e. 4 directions, each hidden layer contains LSTM units equals to 4×4 , 4×20 , and 4×100 . The two *tanh* layers of size 12 and 20 are also known sub-sampling layers, which reduce the number of weights significantly. The size of the input block and intermediate blocks for hidden layers are kept the same, i.e. 3×3 .

The processing of input image is carried out by first dividing it into small patches of size 3×3 . These patches are further fed to LSTM layer, where they are processed in 4 directions. Subsequently, these patches are collected in a *tanh* layer, and then they are passed to LSTM layer of size 20, until the last hidden layer of size 100 is reached. The final hidden layer provides maximum size for learning as many features as possible. The final output layer is of size 1,000. The training is done on 320,000 images, while testing and validation are also carried out in parallel to training. The size of the images in test and validations sets is 80,000 each per sets. The network has taken a total of 153 epoch to complete the training process. Each epoch is completed in 7 hours and 44 minutes approximately.

Other parameters of the MDLSTM model are; learning rate is set to $1e - 5$, and the momentum is set to 0.9. The *task* is set to classification, as each image contains only one class. We used the open source library known as RNNLib³ for this experiment.

³<http://sourceforge.net/projects/rnml>

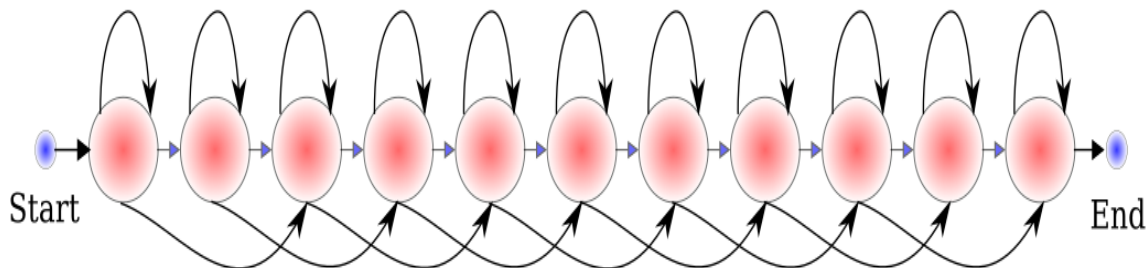


Figure 10.6: *The 10 state HMM topology having self-loops and one state skip. The 10 state topology provides the best recognition accuracy so far regarding Pashto ligatures.*

10.4.3 HMM Based Recognition of Pashto Ligatures

The third system is developed using the classical statistical model i.e. Hidden Markov Model (HMM) known for its sequence learning. There is immense work regarding HMM based applications [BC01]. In addition to that, it has been effectively used in OCR domain as well [PSK⁺08, CK94, KA94, RSB11]. The purpose of the HMM based model is to investigate the problem of rotation invariant OCR system. It is a counter experiment and helps to understand that the recognition of rotated and scaled ligatures itself is beyond the simple recognition of ligatures. It is observed that HMM based models even could not learn from rotation variations. Therefore, we have to drop the rotation variations from the training as well as test and validation sets. The only instances remained in these sets are scaled variation. Although the proposed HMM model is comparatively tested on an easy task, it gives us the worst results.

The proposed topology is based on a single right to left multi-state HMM model. The model is trained on raw pixels as features. Each ligature is first cropped and then resized to the height of 30 pixels. The optimal results are achieved with 10 states HMM and with 512 Gaussian mixture. The proposed topology is shown in Figure 10.6. This experiment is carried out using the HTK⁴ toolkit.

10.5 Evaluation

Based on experimental results, the LSTM based method achieved the best results of 99% ligature recognition rate. The runner up system, that is SIFT based descriptor matching, gives 94.3% ligature recognition accuracy. The worst results are given by HMM based model exploiting only scale invariant recognition for Pashto ligature. HMM achieved

⁴<http://htk.eng.cam.ac.uk/>

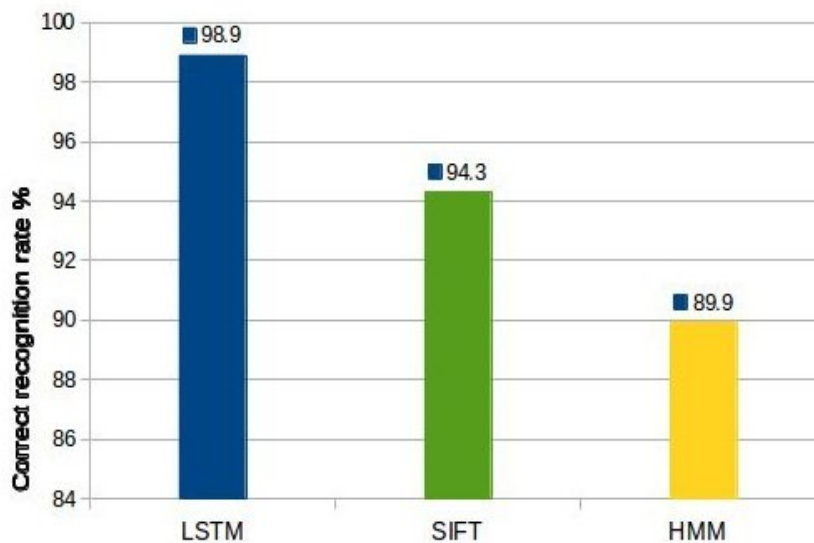


Figure 10.7: *Recognition performance of LSTM, HMM, and classification based on SIFT. LSTM not only learns the script dependent patterns but also learns the scales and rotation variations very well. Image source[AAR⁺15a].*

89.9% ligature recognition accuracy.

A careful interpretation of these results is required. In principle, all three methods could likely be improved by different choices of parameters, different training schedules, and different model structures. However, in all three cases, we chose reasonable and common defaults that have been used previously in other OCR applications. The three systems also do not solve quite the same problem: the HMM-based recognizer was given inputs that were normalized for scale and rotation, and therefore, did not have to generalize across such variation in the input, meaning that the HMM-based recognizer had an easier problem to solve than either the LSTM or SIFT recognizer. The lower performance of HMM-based methods compared to LSTM-based methods observed here is consistent with similar observations for other scripts like in; [GFS05, Zen15, WME⁺10].

Based on analysis of different error cases, we reached the following tentative conclusions. SIFT has been found good in cases, where classification is required among the ligatures having different primary ligatures. However, SIFT has miss classified those ligatures, where the primary ligatures are nearly the same. An example of such miss classification is shown in Figure 10.8. It can be observed that SIFT makes miss classifications, where base sub-word/primary ligature is same, and other additional parts of the characters are not the same. It is also observed that SIFT is more sensitive to image resolution rather than the shape. It means that SIFT can generate a different number of features for the same shape with different resolutions. Further, when dealing a huge dataset, the size of the SIFT descriptor becomes another issue. In our case, it is not practical to wait more

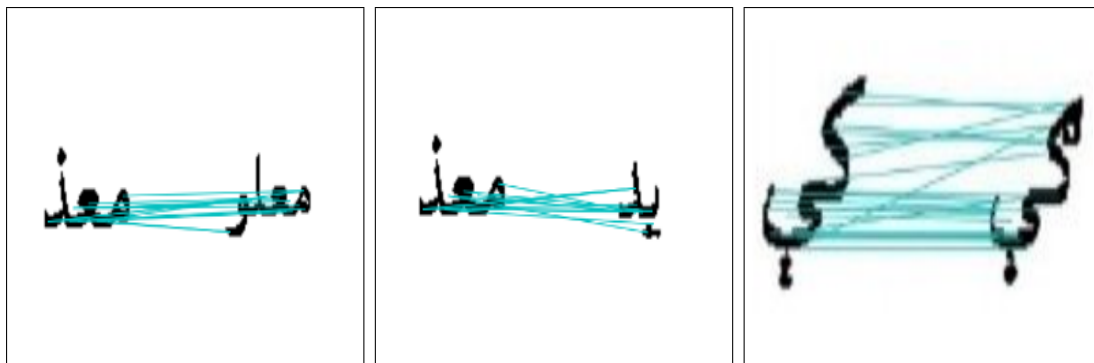


Figure 10.8: *Miss-classification due to shape similarity. Here some of the sub-words are same, but additional parts of the characters are not the same. SIFT has misclassified certain number of such images.*

than 3 minutes for just one shape (ligature) to be classified. Though there are other ways to speed up this process by implementing SURF features or visual Bag of Words (BoG), we have to compromise in some part of recognition rate

A very important aspect about the use of SIFT features is the selection of problem domain. For example, the problem domain in this research is to recognize Pashto cursive script. Our intention was to check different state of the art approaches for the scaling and rotation variations in the Pashto cursive script. SIFT features are considered to be robust against scale and rotation. However, it fails to give good results compare to LSTM. Alternatively, LSTM is a good choice to handle these issues, though the training will take longer time than SIFT, but it is only once.

10.6 Conclusion and Future Work

The work presented in this chapter, suggest that LSTM based model is the best methods for recognizing Pashto ligatures in term of scale and rotation variations. Even simple implementations yield comparatively low error rates with little parameter tuning or effort. Future work with LSTM-based Pashto recognition includes exploring more complex LSTM architectures, as well as feature extraction. SIFT-based methods have higher error rates. One potential advantage of SIFT-based methods is that it is fairly easy to understand how they work and to explain why they fail when they fail. Furthermore, transformations and invariance under transformation are explicitly engineered into SIFT-based methods, giving potentially better control over the degree of invariance and integration with other computer vision techniques for such approaches. We believe that SIFT-based methods can potentially be improved by match refinement strategies. HMM-based methods perform worst of the three methods, even though they were actually given already normalized data. The benchmarks in this paper, as well as previous

results from the literature, underlined that HMM-based methods are not very promising for high-performance OCR compared to LSTM. Nevertheless, we plan on understanding the behavior of HMM-based recognizers and subclasses of inputs where they outperform LSTM methods.

Since our overarching goal is to make Pashto writing more generally available, our next steps are now to integrate the LSTM-based recognizers to scan-based OCR system for real Pashto script.

Pashto OCR and Deep Learning Benchmark

This chapter presents one of the important contributions regarding practical achievements of this thesis. It presents for the first time a deep learning benchmark applied to the real Pashto text. Further, this chapter contains a related work focusing mainly on very closed languages like Arabic, Urdu, Persians, etc. This chapter further describes the proposed architecture for OCR system and discusses the overall experimental setups. Finally, it evaluates the results and discusses the top 20 confusions.

11.1 Motivation

Until now, we presented the work regarding the Pashto OCR considering synthetic data as test cases. However, the real depth of the OCR for the Pashto language could only be explored with the help of real-world data. The real world data gives us more clues and indicators regarding the complexities and reveals the characters or patterns that pose more complex behavior in OCR. To analyze all these things, we present a deep learning benchmark on the real Pashto imagebase. This deep learning benchmark is the first ever work that deals Pashto language and basis on empirical analysis regarding optimization of the power of LSTM.

In this work, the real Pashto imagebase KPTI (cf. Chapter 7) is evaluated for the real world challenges which may exist in the Pashto language in OCR. The main contributions of this work include (1) Evaluating BLSTM (cf. Section 3.3.2) on KPTI data, (2) evaluating MDLSTM (cf. Section 3.3.3) on KPTI data, (3) empirical analysis about finding the optimal parameters for deep learning models (i.e. BLSTM & MDLSTM), and (4) the impact of scale normalization on MDLSTM architecture. It is subsequently followed by a discussion regarding top 20 confusions in classification. There are also some important patterns which exist in KPTI dataset and pose extra complexities toward mature OCR system. The work presented in this chapter is already acknowledged and published

in ICFHR-2016¹. Thus the majority of the parts in this chapter are taken from that work [AAR⁺16].

As this work reports the baseline on KPTI dataset, therefore it is not comparable with the existing research related to the Pashto language. However, it is important to briefly explain the latest research that referred LSTM models for OCR systems. Next section contains related work.

11.2 Related Work

Arabic, Persian, and Urdu are very similar to Pashto. However, Pashto contains extra alphabets, which cannot be recognized by the OCRs designed for Arabic, Persian and Urdu languages. Although, we have discussed a lot of related work in Chapter 5, following are the more closely related work, in which LSTM based architectures are used in the field of OCR.

Ahmed et al. [ANR⁺16b] presented the BLSTM architecture on a limited dataset containing 300 text-lines from Balinese² language. They also provided an empirical analysis regarding hidden layer size, learning rate, etc. The maximum character recognition rate was 98.75% when they chose the size of a hidden layer as 100.

Al Azawi et al. [AALB15, AAHLB14] used LSTM based approach as a language model for the correction of OCR results in the post-processing stage. They have made a comparison of BLSTM approach and Weighted Finite State Transducers (WFSTs) with context-dependent confusion rules. In this work, the performance of LSTM based methods was significant, and in the case of Urdu Nastaliq, they achieved a character error rate (CER) of 1.58%, while on the same test set, the baseline results of the OCR was 6.9% as CER.

Maalej et al. [MK16] used MDLSTM along with dropout for the recognition of Arabic text using IFN/ENIT dataset. They used dropout layers in between LSTM units and *tanh* layers with 50% dropout ratio, and reduced label error rate (LER) from 16.97% to 12.09%. Further, in another work [MK16], they used the dropout technique inside the MDLSTM units and before the MDLSTM units, and they have found that using dropout before MDLSTM units is useful. They further reduced the label error to 11.62%.

Yousefi et al. [YSBS15] suggested a normalization of baseline position in Arabic text-lines as a pre-processing step, and have succeeded to achieve even good results by 1D-LSTM compared to 2D-LSTM. They used IFN/ENIT dataset for this work.

¹Ahmad, Riaz, M. Zeshan Afzal, S. Faisal Rashid, Marcus Liwicki, Thomas Breuel, and Andreas Dengel. "KPTI: Katib's Pashto Text Imagebase and Deep Learning Benchmark." In *Frontiers in Handwriting Recognition (ICFHR)*, 2016 15th International Conference on, pp. 453-458. IEEE, 2016.

²https://en.wikipedia.org/wiki/Balinese_language

Ul-Hassan et al. [UHBD16] used the BLSTM model available in the OCRopus³, for regenerating ground-truth and improve the transcription after each iteration. The work has an importance, particularly for ancient scripts. First, they used the segmented form of each unique symbol/character and cluster them according to shape similarity. Second, they used OCRopus to learn and incorporate the correction in the ground-truth after each iteration. They were able to achieve a reduction in CER from 24% to 7% in few iteration.

Ul-Hassan et al. [UHAS⁺15] also presented an LSTM based model for multiple script identification in text-line level. They have synthetically created English and Greek text-line datasets and achieved a character recognition rate of 98%.

Another breakthrough is made by Bluche et al. [BLM16], where they presented an end-to-end OCR system. Their system took input as paragraph level instead of isolated text-lines. They used MDLSTM along with *attention* strategy to scan each line and find the next line in the paragraph. The system was not only very slow, but the results were also bad compared to the results obtained by text-lines.

We can conclude from the related work, that the LSTM based models are used in many modules to improve the overall pipeline of DIA system. The performance of LSTM based approaches is acknowledged from image binarization [APPS⁺15] to post processing. Still, there is a need for an OCR system regarding the cursive script especially the Pashto language.

11.3 Dataset

In this work, we used the KPTI dataset. The detail about the KPTI dataset is already given in Chapter 7, there we explained and shared its protocol, contents, and statistics. Recall that KPTI contains 17,015 Pashto text-lines. These lines are extracted from images already acquired from real Pashto scribed books. Table 11.1 shows and recalls some important statistics of KPTI dataset. Similarly, Figure 11.1 also shares some of the text lines from KPTI dataset.

11.4 Methodology

Our proposed models are based on the adaptation of classic and sophisticated variant of Recurrent Neural Networks (RNN)s, known as LSTM. The simple RNN are famous for

³Open source Python library for LSTM/BLSTM usage; <https://github.com/tmbdev/ocropy>

Table 11.1: *KPTI statistics in terms of text-lines per set, and total characters contained by each set.*

Katib's Pashto Text Imagebase (KPTI)		
Split	Text Lines	Total Character
Train Set	11,910	485,300
Test Set	2,553	103,992
Validation Set	2,552	103,952
Total unique classes/ labels		96

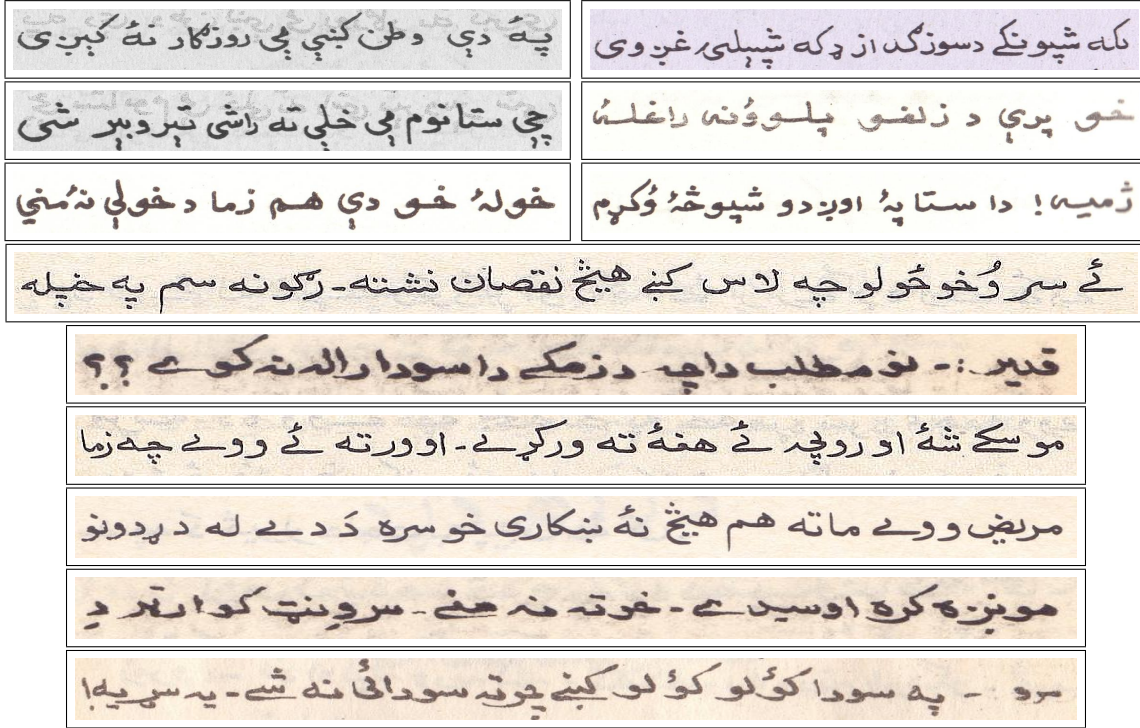


Figure 11.1: *Random selection of text-lines from KPTI dataset, the variation in terms of human involvement can be seen clearly.*

their context learning capabilities. However, they suffered due to long and short term dependency, known as vanishing gradient and weights exploding problems respectively.

On another hand, LSTM based networks are capable of handling vanishing gradient and weight exploding problems. As mentioned in Section 3.3.1, LSTM cells are just like artificial neuron cell, which can retain the most likely weights for optimal outputs, and thus avoiding the vanishing gradient problems. In this work, we are proposing two different models. The first proposed model is based on one-dimensional Bidirectional BLSTM (cf. Section 3.3.2) and the second proposed model is based on Multi-Dimensional MDLSTM (cf. Section 3.3.3). As BLSTM can only scan the input sequence in two directions (horizontally; left, right). Therefore, such architectures can explore the vicinity and keep learning the context information in one direction. However, MDLSTM provides an extra feature to LSTM models, by scanning the input sequence in four dimensions

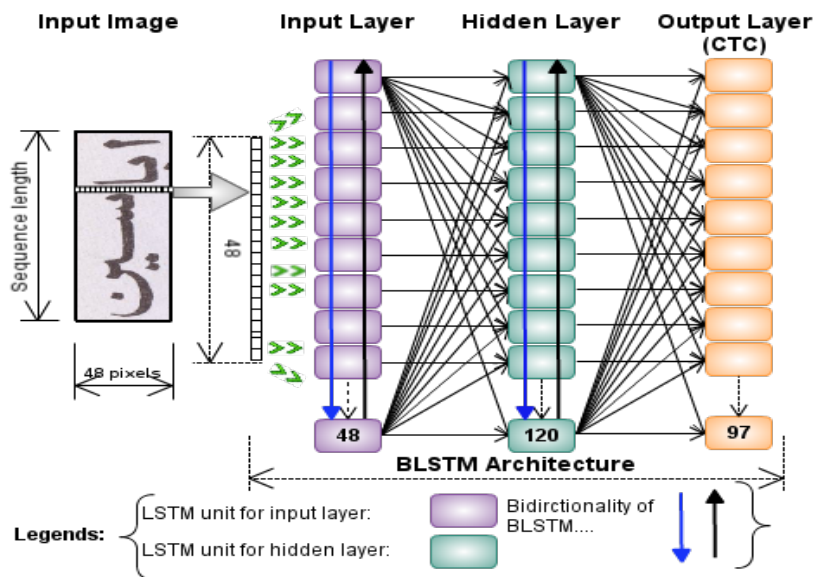


Figure 11.2: *Proposed BLSTM architecture [AAR⁺16].*

(left, right, top, bottom). The detailed topology of the proposed models is given as below.

11.4.1 Topology of Proposed BLSTM Model

The parameters are selected empirically for the architecture of our proposed BLSTM model. The size of the hidden layer is important because it not only depends on the problem domain but also on the abstraction of the data. Thus, based on empirical analysis, we found that 120 LSTM cells give optimal results in Pashto OCR and are a better choice for a hidden layer size. Table 11.2 shows the results of such analysis. The overall architecture of BLSTM network consists of three layers, such as input layer, hidden layer, and an output layer. The size of the input layer is 48 which is equal to the height of the input image (input images are normalized to the height of 48 pixels). The normalization of images to a fixed height is one of the limitations of BLSTM. The input layer is then followed by a hidden layer with 120 LSTM units which is subsequently followed the by output layer. Here, the output layer is also known as Transcription Layer/CTC layer and represent the transcriptions/labels in a 1D array of size 97, (96 unique Pashto characters and punctuation) including a null character as well. This null character is required for the CTC [GFGS06] layer. It is the CTC layer that aligns the most probable predicted labels with the input sequences without explicit segmentation. Learning rate is set to 10^{-4} . Figure 11.2 shows the architecture of the proposed model.

11.4.2 Topology of Proposed MDLSTM Model

In general, MDLSTM based networks are capable of scanning an image in 2^d directions, where d is the dimension of input data. In our case, the input image is of 2 dimensions i.e. $d = 2$, and $2^d = 4$. Such architectural design of LSTM makes it robust against many variations. These variations include scale (by taking any dimension of sequence), rotation, registration, etc.

Figure 11.3 illustrates the architecture of our proposed MDLSTM model. There are some basic terms, which need more attention before understanding the architecture of MDLSTM and it's working. These terms are given below.

- *inputBlock*
- *hiddenSize*
- *subsampleSize*
- *hiddenBlock*

The *inputBlock* refers the block size ("*width*" \times "*height*"), which is used to initially divide the input image into small patches. In our proposed model, we have set the *inputBlock* size as 1×4 . The *hiddenSize* describes the number of hidden layers or LSTM layers and their constituent LSTM units. In our case 4, 20, 100 is the optimal parameters for *hiddenSize*. Table 11.3 shows the detail about the parameter selection for MDLSTM. The proposed model has a pool of five hidden layers, in which three hidden layers are LSTM based, and two are of sub-sampling layers. These layers are hierarchically arranged in the network.

Note that, for each LSTM layer, the number of LSTM units will be equal to the size of that layer multiplied by the number of directions in which the input image is scanned. In our proposed case, the image is scanned in 4 directions. Thus, the LSTM units in our proposed model become 4×4 , 20×4 , 100×4 . Figure 11.3 shows the LSTM units using four different colors for four different directions. Another term is *subsampleSize*, it describes the size of feed-forward *tanh* layers. In our case, the size of (sub-sampling) feed-forward *tanh* layers is 16 and 80. These *tanh* layers are placed in between each pair of LSTM layer. More specifically, the first sub-sample *tanh* layer of size 16 is placed immediately after first hidden layer. Similarly, the second sub-sample *tanh* layer of size 80 is placed in between second and third hidden layers. The purpose of these feed-forward *tanh* layers is to reduce the weights significantly.

The final and important term is the *hiddenBlock* size. It refers the sub-sample sizes and is required in two different stages. First, it is required to hold the activation of the first hidden layer and feed them to first *tanh* layer. Second, it is required to hold the

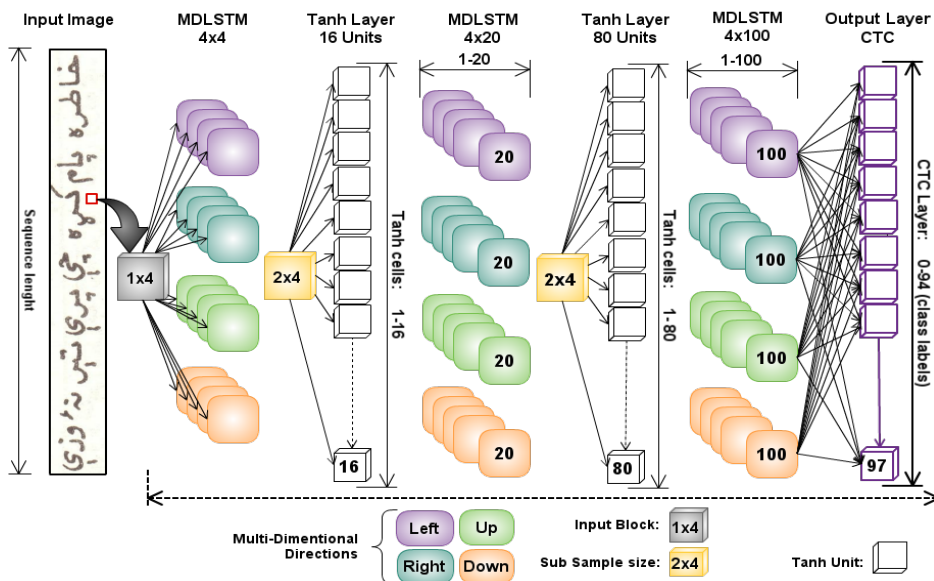


Figure 11.3: *Proposed MDLSTM architecture. Image source [AAR⁺16].*

activations of the second hidden layer and to feed it to second *tanh* layer. In our case, we kept *hiddenBlock* size the same as 2×4 for each sub-sampling stage. Finally, the CTC layer received the activations from the last hidden layer and collapsed all the values in a one-dimensional array of size equal to the number of classes plus one extra label for blank.

11.4.3 Evaluation Metric

The normalized edit distance is used as the evaluation protocol for our experiments. The overall error is calculated from the ratio of insertions (I), substitution (S), and deletion (D) with a total number of characters in transcriptions. The Character Error Rate (CER) is given in Equation 11.1.

$$CER\% = \left(\frac{I + S + D}{Total\ Characters\ in\ test\ set} \right) * 100 \quad (11.1)$$

11.4.4 Selection of Optimal Parameters

We have carried out two empirical analysis to find the optimal parameters for our proposed models. First one is for the selection of parameters for BLSTM, and the second one is for finding the optimal parameters for MDLSTM. Tables 11.2 and 11.3 show both empirical analysis for BLSTM and MDLSTM respectively. Note that, we picked a small set of 500 images from our main dataset to conduct the empirical analysis, we choose 390 images for a training set and 110 images for a validation set. Therefore, such results are

Table 11.2: Empirical analysis for BLSTM; Different sizes of hidden layers verses validation error rate in %.

Hidden Layer Size (BLSTM)	Train Set CER %	Valid Set CER %	Total Epochs	Time elapsed per epoch (mm:ss)
20	42%	58%	145	00:11
50	27%	57%	101	00:27
80	23%	53%	83	00:47
100	22%	51%	72	01:03
120	19%	50%	71	01:21
150	11%	50%	77	01:52

Table 11.3: Empirical analysis for MDLSTM; The impact of size of hidden layers on the fixed size of tanh layer i.e. 6,12.

Hidden Layer Size (MDLSTM)	Train Set CER %	Valid Set CER %	Total Epochs	Time elapsed per epoch (mm:ss)
2,10,50	28%	53%	265	00:27
4,20,100	14%	51%	202	01:02
8,30,150	27%	51%	151	02:03
12,40,180	29%	52%	140	03:30

Table 11.4: Empirical analysis for MDLSTM; The impact of size of tanh layers on the optimal size of hidden layer i.e. 4,20,100.

Tanh Layer Size (MDLSTM)	Train Set CER %	Valid Set CER %	Total Epochs	Time elapsed per epoch (mm:ss)
6,12	14%	51%	202	01:02
8,20	18%	48%	165	01:07
12,30	14%	49%	164	01:10
20,40	11%	46%	145	01:14
16,80	08%	44%	141	01:21
40,80	20%	45%	113	01:30

not comparable with the results produced in other experiments. Another, a small set of images are sufficient to indicate a clue to predict the behavior of learning of the models. Further, during the empirical analysis, we have observed the indicators like CER and average time taken by an epoch during training on the validation set, and have tuned network's parameters to get optimal results for indicators. Table 11.2 presents data for BLSTM, thereby we can conclude that a hidden size of 120 LSTM units gives us better results. The learning rate is set to 10^{-4} , which gives us optimal performance.

Similarly, to find the optimal parameters for MDLSTM, we need to find exactly which size of the hidden layer can give us optimal results. For this purpose, initially, we fixed

the *tanh* layer size to 6,12 and tuned different sizes for the hidden layer. Table 11.3 shows the different parameters that we checked for MDLSTM, a size of 4,20,100 for hidden layer gives us optimal performance with learning rate 10^{-4} . Further, we checked the influence of different sizes for *tanh* layers using the optimal hidden layer size (i.e. 4,20,100) and found a size of 16,80 as optimal for *tanh* layers, as it gives us better results in our problem domain. Table 11.4 presents the impact of different sizes for *tanh* layers. The experiments are performed using RNNLib platform⁴. The actual experiments are discussed in detail in the following section.

11.5 Experiments

The experimental setup is divided mainly into two tasks. Table 11.6 shows the detail of these tasks regarding their target strategies.

In Task-I, we have evaluated the Pashto text-lines based on as they are. More specifically we did nothing in pre-processing stage regarding normalization or noise removal etc. By default, the heights of the images are in between the range of 50-244 pixels. The only operation that is performed in the pre-processing stage is the conversion of images into greyscale. These greyscale images are directly fed to the proposed MDLSTM for training, and subsequently, the best network model is tested on the test set. Further, in Task-I, we have excluded the testing of BLSTM, as it is incapable of handling an input image/sequence with different heights.

In Task-II, we have introduced height normalization as a pre-processing step. Each image is normalized to the height of 48 pixels. We have investigated the normalized data both on BLSTM and MDLSTM. There are two experiments in Task-II. In the first experiment, the BLSTM model is checked against the normalized data. Table 11.6 shows the results.

In the second experiment of Task-II, the same normalized images are fed to MDLSTM, and almost the same result is achieved as in Task-I. Recall, that we used the normalized height of 48 pixels for each text line image. Table 11.6 shows the results of normalized images in a row/s referred by Task-II.

11.6 Results and Discussion

We have achieved a CER of 9.22% as baseline results for the newly introduced KPTI dataset. It is empirically proved that MDLSTM outperforms the BLSTM model for the recognition of Pashto text-lines. Further, the normalization does not affect the results in

⁴<http://sourceforge.net/projects/rnnl>



Figure 11.4: The color circles (*Red= deletion, Green= Substitution, Blue= Insertion*), are showing failure points in the predicted text. Each predicted text has it's original input image as above.

Table 11.5: Top 20 confusion related to Pashto alphabets in KPTI dataset.

Gt	Pred	Count	Gt	Pred	Count
S	--	878	---	ه	55
--	S	866	---	ا	53
ه	ه	99	ت	ن	50
---	س	81	---	'\x87'	49
---	م	70	م	--	49
---	ا	68	--	ئ	49
---	ن	67	--	ى	49
و	د	66	ى	--	47
ه	ه	65	ن	--	46
ه	---	56	---	پے	46

Table 11.6: Results of BLSTM and MDLSTM on KPTI dataset.

	Task description	Model	CER% (Test set)
Task-I	Non-normalized Data	MDLSTM	9.22%
Task-II	Normalized Data	BLSTM	16.16%
	Normalized Data	MDLSTM	9.33%

the perspective of reduction in CER. The results are keenly observed for a case to case issues, particularly for top 20 confusions (see Table 11.5). Top two confusions indicate the misclassification of *spaces*. More precisely, 878 times true spaces are classified as background pixels, and 866 times false spaces (background pixels) are classified as true spaces.

In addition to space confusions, the confusions at number 3, 8, 9, and 10 are related to **چ**, **ځ**, **ه**, and **د**. We already discussed these characters in Section 2.5.2 under Pashto specific challenges. Here, we empirically found that these characters present challenges in their classification. They are not only similar but are also occurred in almost similar context. Furthermore, the confusion related to Pashto **ی** [Yey]s are also in the top 20s.

Another, it is also found that due to the involvement of human's nature, variations occur in writing styles. Specifically overlapping, writing in a condensed manner (due to lack of space), and ornamental style for calligraphic beauty. Figure 11.4 depicts the worst results from Task-I. The 1st image with its predicted text (Figure 11.4) is an example of bleed-through factor. Similarly, 2nd image shows the condensed writing behavior of Katib/calligrapher, and the 3rd image in the same Figure 11.4 presents some extended portions in characters for calligraphic beauty. Excepts the issue of bleed-through (generic), all the other issues are specifically associated with the Pashto text written by Katibs, and therefore, pose specific challenges towards Pashto OCR.

11.7 Conclusion and Future Work

In this chapter, we have introduced a baseline for the recognition of real Pashto text images. We have investigated the state-of-the-art deep learning approach based on LSTM and its most reputable variants like BLSTM and MDLSTM on our newly KPTI dataset. The results show that MDLSTM is far better than BLSTM regarding Pashto text recognition. Furthermore, in our case, CER is independent of height normalization, which indicates the robustness of MDLSTM against the registration/scale variations. In this work, we have also done an empirical analysis regarding finding the optimal parameters for the proposed BLSTM and MDLSTM models. Based on results, it is evident that some Pashto characters relatively cause more challenging behaviors compared to the other ones. We have found that the KPTI dataset also has Pashto specific hurdles like condense human writings, ornamental calligraphic beauty, and variations due to human nature.

Post-Processing, Space Recognition Anomaly

This chapter comprehensively reveals the space-anomaly exists in all those languages that use Arabic script. Further, it also presents a combined solution that not only improves the accuracy but also enhances the rendering phase after OCR. This work evaluates the real Pashto data (KPTI) for space-anomaly and validates the generalization of the proposed approach by a counter experiment using Urdu Nastaliq text (UPTI). The results signify the importance of this work and are more beneficial for languages using Arabic script.

12.1 Motivation

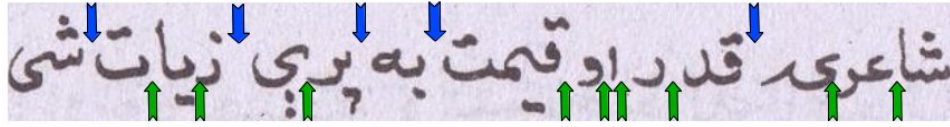
The work that is being focused in this chapter is based on facts and observations that we have revealed in the previous chapter (cf. Chapter 11). As mentioned in Section 11.6, we shared the top 20 confusions, where the top 2 confusions are only related to the miss classification of *spaces*. Therefore, in this chapter, we focus these top two confusions, and comprehensively investigate the conceptual reasons that cause these confusions. We have referred this as *space anomaly* in different parts of this thesis. Space anomaly mainly causes difficulties in recognition of languages which use Arabic script. Existing work regarding Arabic OCR systems primarily focused on achieving best accuracy, and there is a minimal effort regarding the conceptual foundations behind such anomalies.

To understand this anomaly, one has to be familiar with two basic concepts regarding OCR for Arabic scripts. First, the minimum requirements for the acceptable rendering of Arabic texts after OCR. Second, in case we ignore it then what will be the impact of space anomaly on an OCR. This chapter reveals these concepts in a more empirical way, thus provides rendering related concepts in Section 12.2.1, and the results in Section 12.6 shows the difference if we neglect this issue.

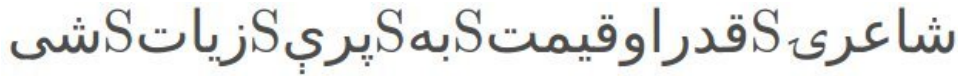
The primary cause of this anomaly is the presence of *breaker characters* (cf. Section 2.4.1),

آ ا ذ ر ز ږ ږ څ و ے ی

Figure 12.1: *Breaker characters of Pashto language, which are the super set of all breaker characters used in Arabic and Persian languages.*



(a)



(b)

Figure 12.2: *An input image (a) and its corresponding ground truth (b). While the regular spaces are indicated by blue arrows, the spaces caused by breaker characters are indicated by green arrows.*

especially when they constitute a sentence or a paragraph in Arabic like scripts. The breaker characters break the cursiveness and leave an effect like space. That space is not modeled in the optimal ground-truth, thus causing the classifier not to generalize the learning towards the regular spaces. As a result, the trained model misclassify the real spaces with the spaces implicitly created by breaker characters. Furthermore, the misclassification of regular spaces subsequently affects the rendering process after OCR. As if the regular spaces are not recognized, then two adjacent characters will join and form another shape. The detailed introduction and the concept of rendering associated with the space-anomaly are further explained in section 12.2.

The scope of this anomaly exists in all those scripts having either *breaker character* as a sub-set of their character sets, or using Arabic script for their written communication. According to precise estimation, more than 70 languages are using Arabic script¹, where this anomaly affects all these languages in the domain of OCR.

In addition to that, we present a joint solution to address this issue. The solution is not only valid for enhancing the accuracy of OCR, but it is also capable of handling the space anomaly. We have tested the proposed method on the same KPTI dataset, the reason is that we had a baseline, thereby we could easily compare the impact of our proposed approach, and could analyze the space recognition problem.

¹https://en.wikipedia.org/wiki/List_of_languages_by_writing_system

12.2 Problem Definition

To understand this problem, please refer to Figure 12.2 with an input image (a) and its corresponding ground truth (b). In the ground truth, the label **S** is used to visualize the presence of regular spaces. Note, it is the optimal ground truth for rendering the text (available in the image) correctly. Blue arrows point the original locations of the regular spaces, while green arrows indicate the spaces of the breaker characters. Now during OCR, an accuracy of regular spaces should be normal. But in our experiments, the case is entirely different. We have achieved a regular space as the top most confusing label (top two confusions cf. Table 12.1). As shown in the ground truth (please refer to Figure 12.2), that only regular spaces (blue spots) marked with a letter **S** but spaces caused by breaker characters (shown in green) do not have any corresponding labels. Thus, the visual spaces resulting from the breaker characters turn the classifier ambiguous and make it hard to differentiate between two types of spaces.

Table 12.1: *Top two confusions, that are taken from our previous work presented in Chapter 11.*

Ground Truth	Prediction	Counts	Notation
S	–	878	1 st Confusion
–	S	866	2 nd Confusion

Moreover, handwritten data differs from printed text. For example, if datasets contain synthetic or printed text (like Arabic Printed Text Images (APT_I) [SIK⁺09], and Urdu Printed text Images (UPT_I) [SS13]), then a uniform distance is found between these two types of *spaces*. Thus, a model obtained by supervised learning could learn these distances/variations in printed text. However, documents containing handwritten text do not have a uniform gap for spaces due to the involvement of humans. Therefore, the learning models fail to distinguish between regular spaces and spaces caused by breaker characters. Due to this problem, not only the overall accuracy is decreased but regular spaces are also confused.

In addition to that, this confusion ultimately affects the rendering phase after OCR. Solving this issue in Arabic like scripts will be beneficial in OCR systems for both accuracies and rendering the output text in a standard form. The following section explains the impact of this problem on the rendering phase.

12.2.1 Impact on Rendering

The effect of misclassification of spaces on rendering Arabic like text requires some basic knowledge regarding these languages. Please refer to Table 12.1, taken from the work of

Figure 12.3: *The impact of miss-classification of regular spaces on rendering.*

our previous chapter ?? and focus on the top two confusions that are related to regular space \mathbf{S} . We described these confusions below.

- *1st confusion* : Regular spaces are identified as background pixels or nothing.
- *2nd confusion* : Regular spaces are introduced where they are not expected according to the ground truth.

In Table 12.1, 878 times *regular spaces are classified as nothing* (1st confusion) and 866 times spaces are introduced unnecessary (2nd confusion). The 2nd confusion can somehow be ignored because background pixels are classified as regular spaces. It ultimately does not affect the recognition rate regarding over all character recognition. In contrast, 1st confusion needs extra attention. In cursive scripts like Arabic, Urdu, Persian, and Pashto, spaces are added in between two adjacent words. But this rule is withdrawn when any word ends with *breaker* character. However, if a word is ending with the *non-breaker* character, a regular space shall be used. Now, intuitively it is clear that if a classifier is unable to classify the normal spaces (as in 1st confusion), then the characters in rendering phase will connect to each other and will form odd shapes. Let suppose an OCR system predicted a text for the same input image shown in Figure 12.2, and where it classified the regular spaces as nothing (1st confusion), then the rendered text will look like an image shown in Figure 12.3. We can see that the only spaces exist which are caused by the breaker characters. In short, as much as the (1st confusion) exists with larger values, more invalid shapes could be encountered after the rendering phase. Thus, minimizing the (1st confusion) will lead to facilitate the rendering process and subsequently will result in standards shapes of a target language.

12.3 Related Work

The writing style of Pashto, Urdu, and Persian is inherited from Arabic script [Rah04]. Therefore, these languages are the candidates and present the problem of space anomaly [AAR⁺15b, DH10]. Before discussing the related work, it is important to know that the impact of space anomaly is marginally high in text-line recognition instead of words or ligatures. In other words, the holistic or segmentation-free (cf. Section 5.3) approaches

are mainly affected by this anomaly. However, by reviewing the existing research work, we could not find any work that specifically addressed the space anomaly. The majority of works focus on reporting CER and do not address top confusions and the reasons behind these confusions. Although we have already presented the related work regarding holistic approaches in Section 5.3, here we also report some more similar work.

Hamdani et al. [HDN14] presented a comprehensive recognition system for OpenHaRT Arabic handwritten text. The system is based on CART trees along with a hybrid classifier based on HMM and RNN. The system achieved a CER of 8.3%. However, they did not report the top confusions, and therefore could not provide any clue regarding space anomaly.

Dreuw et al. [DJN08] presented a very similar work regarding space-anomaly. In this work, they have explicitly modeled the white-spaces in IFN/ENIT dataset. Note that, IFN/ENIT dataset only contains the images having Tunisian town names. They have added white-space labels in three different positions.

- No spaces (ns): The available ground truth annotation is used as it is.
- Between word white-spaces (bws): A white-space is added only between two sub-words².
- Between word and within word white-spaces (bwws): White-space labels are added in between the models of isolated-, beginning-, and end-shaped characters.

They have achieved better results using the *between word and within word white-spaces (bwws)* scheme. The bwws scheme is partially similar to our proposed approach. However, they have imposed the addition of space label in between all characters which come as individual, isolated, or end-shaped. In contrast, our proposed system only adds space labels in the ground truth where exactly a visual space appears in the corresponding image. In other words, we have targeted only the breaker characters, which conceptually cause these spaces.

We can conclude from the related work that Pashto, Urdu, and Arabic languages are having enough research work regarding OCR systems, but minimal effort is made on solving the issue of space recognition in these languages. Secondly, datasets associated with such languages do not contain line-segments. These line-segments are essential for realizing the problem of space recognition. The dataset like IFN/ENIT has only town names instead of continuous text-lines or sentences, so having less significance regarding space-anomaly. However, The KPTI dataset for the Pashto language is an excellent platform for exploring space anomalies because it contains real world text lines of Pashto text. Secondly, it is the superset of many cursive script languages (Arabic, Urdu, etc.). Other candi-

²The default transcription of the KPTI dataset has already a white-space label between words.

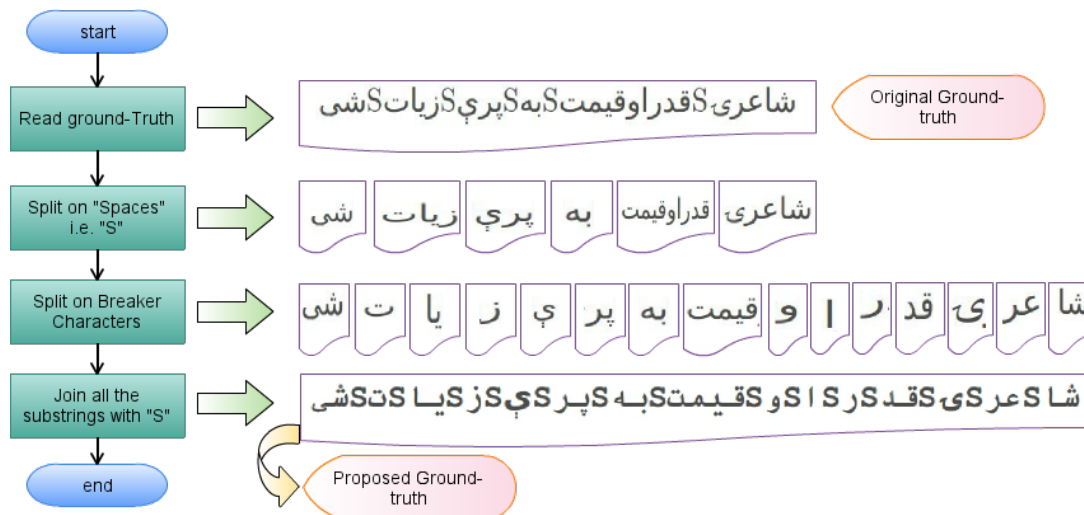


Figure 12.4: The proposed solution for transforming default ground truth to required ground truth.

dates include UPTI (Urdu Printed Text Imagebase for Nasta'liq script) and KHATT³ (for Arabic handwritten text) datasets because they also contain text-line segments.

12.4 Proposed Solution

As discussed above, there are two leading causes of space-anomalies. (1) Breaker-characters, where there is a breaker character there will be space. (2) These spaces are not modeled in the optimal ground truth. Therefore, we present a solution by proposing a modification in the default transcription/ground-truth. The same label (i.e., S) is added immediately after each breaker-character. To do this, we require language-specific knowledge for identifying the cases where breaker characters introduce space. In this work, we have considered the Pashto language as a test case and have used the KPTI dataset for evaluation. In the Pashto language, the breaker characters are the superset of other breaker characters exist in Arabic and Persian languages. However, in the case of Urdu, two breaker characters i.e. ڈ [da:l] and ڑ [are:] are not present in Pashto. By including ڈ , and ڑ into breaker characters, we cover all the breaker characters that exist in Arabic, Persian, Urdu and Pashto languages. Another requirement is to identify the labels that are used to represent such breaker characters. The identification of labels is dependent on how a certain dataset is transcribed. For example, UPTI dataset uses label conventions like *Meem* for م and *Bey* for ب . However, in KPTI dataset utf-8 codecs are used to label the characters of Pashto language.

Figure 12.4 visually illustrates the complete process of redefining the default ground-

³KHATT (KFUPM Handwritten Arabic Text) database, <http://khatt.ideas2serve.net/>

Algorithm 3 This function modifies the existing ground truth by inserting space label "S" after breaker characters. Taking existing ground truth "Trans" as input argument and return proposed ground truth as "newGt".

```

1: function GENERATEGT(Trans)
2:   breakerChars[]
3:   newGt = []
4:   for chr in Trans do
5:     if (chr in breakerChars) then
6:       chr = chr + "S"
7:     end if
8:     newGt.append(chr)
9:   end for
10:  return newGt = "".join(newGT)
11: end function

```

truth. Further, Algorithm 3 also describes the whole process. The conversion is done, by traversing the actual ground truth for breaker characters, and by inserting label "S" *immediately after* breaker character ⁴. We can observe the change between the original and the proposed ground-truth after the application of our proposed solution.

It is worth mentioning that the insertion of these extra spaces neither affects the rendering nor the meaning of the text. Further, they enhance the readability of the text. Therefore, it is recommended to leave these spaces after classification phase. However, if one is still interested in discarding these extra "spaces", then it could be easily removed in a post-processing step. The removal can be done by just searching the predicted text for breaker characters and if the preceding label predicted as *space*, discard it.

12.5 Datasets and Evaluation Metric

We investigated the effectiveness of the proposed solution via a Multi-Dimensional Long Short Term Memory (MDLSTM) network architecture. It is the same neural network model which is used in [AAR⁺16] for achieving the baseline results on the KPTI dataset. The reason for selecting the same system is the empirical analysis provided in that work. Furthermore, it helps in showing the direct consequence of applying our approach, i.e., the network remains the same while the transcription changes. Therefore, any improvement in connection with our proposed solution would be easily comparable. Figure 12.5 shows the architecture of the adapted MDLSTM model.

The main parameters of the proposed MDLSTM models include three MDLSTM hidden layers of size 4, 20, 100 and two *tanh* layers of size 16 and 80, which are placed in between

⁴Arabic script is written from right to left, therefore, "immediately after" should be considered in that context.

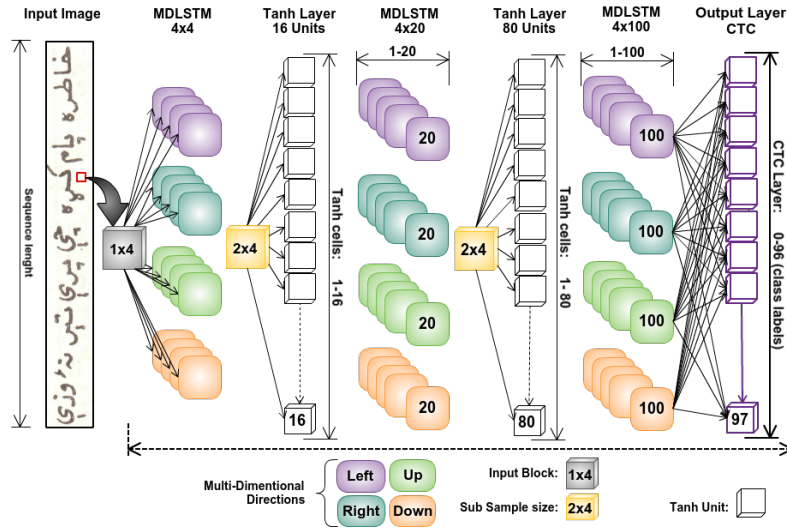


Figure 12.5: *MDLSTM architecture, taken from the work of Ahmad et al. [AAR⁺16].*

hidden layers. The final layer is the Connectionist Temporal Classification (CTC) layer. The CTC layer aligns the predicted labels corresponds to the target ground truth by finding the most probable path. For more detail, please read Section 3.3.3.

We used KPTI dataset that contains 17,015 handwritten text lines of Pashto language. The complete description of the KPTI dataset is already given in Chapter 7. Here, we use the same dataset with the same split ratio regarding train set, validation set, and test set (i.e., 70%, 15%, and 15% each).

12.5.1 Evaluation Metric

The evaluation protocol for our experiments is the error based on Levenshtein's [Lev66] distance between the ground-truth and the predicted text. This error is computed from the ratio of deletion (D), insertion (I), and substitution (S) concerning a total number of characters in transcriptions. The Character Error Rate (CER) is given in Equation 12.1.

$$CER\% = \left(\frac{D + I + S}{Total\ Characters\ in\ test\ set} \right) * 100 \quad (12.1)$$

Similarly, the reduction in 1st confusion and 2nd confusion is the required metric, that describes the importance of our proposed method. Note that, minimizing the error in 1st confusion will subsequently improve the rendering phase.

Table 12.2: *The reduction in overall CER, 1st and 2nd confusion, with respect to our proposed method on KPTI and UPTI dataset.*

Dataset	Method	1 st confusion	2 nd confusion	Overall CER%
KPTI	Baseline [AAR ⁺ 16]	878	866	9.22
	Proposed	176 (80%) ↓	352 (59%) ↓	6.33
UPTI	Baseline [NUA ⁺ 15]	293	422	3.60
	Proposed	20 (93.3%) ↓	26 (93.8%) ↓	1.01

12.6 Results and Discussion

We checked the adapted MDLSTM for KPTI dataset with proposed transcription. The results of the baseline are compared with the new one, and a reduction of 3% is found in overall CER. Similarly, the particular error regarding space recognition is also computed. For this purpose, we checked the original regular spaces and achieved about more than 80% error reduction. The results signify the contribution of our proposed work and provide an achievement toward a mature OCR system in cursive script languages.

To validate the generalization of our proposed solution, we carried out a counter experiment on the UPTI dataset. The split of Naz et al. [NUA⁺15] is used to make the results comparable. The overall CER obtained on UPTI dataset compared to the reported CER in the main work [NUA⁺15], validates that our proposed solution not only works better for Pashto language but also benefits other cursive languages as well. The Table 12.2 shows the top two confusions and overall CER on the respective test sets. The results representing the baseline are taken from the work [NUA⁺15].

Based on our results on UPTI dataset, we have achieved about 93% reduction in space recognition (1st confusion), which subsequently links the rendering phase thereby assures the formation of standard shapes in target languages. In addition to the space recognition issue, the overall CER is reduced by 2.59%, which is a remarkable improvement in a character recognition rate.

12.7 Conclusion

We have comprehensively addressed the issue of space anomaly in this chapter. The chapter described the conceptual basis and the impact of this problem in more detailed manner. Further, we have also explained the scope of this issue and have stated that more than 70 languages suffered due to this anomaly. In other words, languages either derive from Arabic script or contain breaker characters present this issue. More specifically, the problem exclusively exists due to the presence of breaker characters, because breaker characters introduce spaces in images while the ground truth does not reflect any spaces.

We have also visualized this issue with real images in KPTI dataset. The impact of space misclassification is discussed, and its implications are explained on overall accuracy and ultimately on rendering phase. Further, we have provided a solution by reusing existing space label in the ground truth for the spaces caused by breaker character at the corresponding image location. The proposed solution is verified on KPTI dataset, thus reducing the CER by 3%, compared to the baseline recognition system. There is also 80% reduction in misclassification of spaces. Similarly, to generalize the use of our proposed method, a more complicated script like Urdu Nastaliq is also examined for space recognition issue. An error reduction of 2.59% is achieved for overall CER compared to the baseline. The benefits of this work are twofold, first is the improvement in the recognition rate, and second is the quality rendering after OCR.

This chapter contains introductory material regarding end-to-end Pashto OCR system based on Graphical User Interface (GUI). The end-to-end system is one of the main practical contributions of this thesis. Initially, this chapter states motivations and then describes the required tools. Further, it also explains the primary functions of the end-to-end system and visually illustrates the functionality of each function. These functions are data preparation, training, and testing.

13.1 Motivation

We have observed that majority of the research work regarding DIA system focus on partial achievements, and very less effort has been made regarding an end-to-end solution. Another, integration of partial achievements needs inner compatibility and intuitive operability towards a combined effect. As a result, integration becomes very difficult for various sources/researchers to make partial modules functional in one unit. An end-to-end system is important not only for research community but also for a user because they provide a synergy effect and solve a target problem via a complete pipeline under an end-to-end operation. Therefore, the primary motivation of this chapter is to present an end-to-end system regarding Pashto OCR. It further highlights a contribution towards an end-to-end OCR system, which is one of the major practical contributions of this thesis.

In addition to that, for Arabic script, no such platform could provide an end-to-end system in the field of DIA. We have developed this end-to-end system with an intention to keep it more generic and flexible regarding integrability towards the future developments. We have integrated all the major contributions of this thesis in the current version of the end-to-end system. It is flexible and provides an efficient graphical user interface (GUI). Currently, the proposed end-to-end system has RNNLIB¹ as back-end engine and

¹<https://sourceforge.net/p/rnnl/wiki/Home/>

provides the power of Multi and Bi-directional LSTM via GUI environment.

The proposed end-to-end system comprised of mainly three modes, which are (1) data preparation, (2) training, and (3) testing modes. The data preparation mode provides functionality regarding dataset creation, annotation correction, skew correction, line segmentation, etc. Similarly, the training mode provides utilities regarding the model selection, parameter selection, and easy access to available trained models. Finally, the testing mode makes it easy to evaluate the existing trained models on different test sets.

Next Section describes the tools used in the development of the end-to-end system.

13.2 Tools and Requirements

Regarding structure, our end-to-end system comprises of two parts, i.e. (I) back-end, and (II) front-end. Therefore, we used different tools and developmental technologies in connection with back-end and front-end. For better understanding, we explain and describe these tools and requirements under their concern categories. Next section describes the tools used as back-end.

13.2.1 Back-end Tool

We used RNNLIB as a back-end engine. It is a recurrent neural network library specifically designed for sequence labeling problems such as handwriting and speech recognition. It implements LSTM architecture along with more traditional neural network structures, such as standard recurrent networks, etc. [Gra12c]. It's more important characteristics are:

- Bidirectional LSTM [GS05], which provides an exact implementation of BLSTM architecture presented in Section 3.3.2. It can access the long range contextual information in both backward and forward direction.
- Connectionist Temporal Classification (CTC) [GFGS06], which enables the system to align the transcription of unsegmented sequence data.
- Multidimensional Recurrent Neural Networks (MDRNN) [Gra12b], it extends the capability of the system to scan data with more than one spatiotemporal dimension (images, videos, fMRI scans etc.).

The training and testing modes of our end-to-end system mainly use the RNNLIB. However, we made modifications in some modules for transferring data/information between front-end and back-end (i.e., between front-end/GUI and RNNLIB library).

13.2.2 Front-end Tool, Qt Designer

We use Qt to develop the GUI of our end-to-end system. Qt is a framework for the development of cross-platform applications². Qt supports platforms like OS X, Windows, Linux, Android, Blackberry, and much more. It is not a programming language but is a framework written in C++. A compiler named MOC (Meta-Object Compiler) is used to extend the C++ language with features like signals and slots. The code written in Qt is Qt-extended C++, the MOC parses it and generate the standard compliant C++ sources. Thus, any standard C++ compiler like GCC, ICC, MinGW and MSVC can be used to compile the output of MOC.

In addition to that, Qt is also known for its quick developmental process. It provides a lot of flexibility over many controls. One can drag and drop a control/object at any position on a window form. Another advantage of Qt is, it supports bindings for other languages like Java, Python, etc. This binding-mechanism makes a project integrable with other modules written in other languages. As we used Python language mainly to implement work like skew detection and line segmentation, therefore, Qt is a good choice to be used for the development of an end-to-end system.

Qt provides a special designing environment known as Qt Designer. The version of Qt Designer that we used in this thesis is 4.8.6. Developers use this tool and quickly build various GUI based applications. It is Qt Designer that stores all the meta information related to a GUI designing in a *.ui* File, the *.ui* file contains a script of QML (Qt Modelling/Markup Language). However, to understand *.ui* files and make them interactive, we need language-specific bindings. As we mentioned that we use Python as a programming language, therefore, we used Python bindings. In this thesis, we used PyQt4³ bindings to make the GUI components interact-able with the back-end. A detail documentation about Qt Designer can be found here⁴.

13.3 Description of GUI

The GUI environment is designed in such way to provide a convenient and flexible look. At first launch, it starts the main window, which is a pure MDI (Multiple Document Interface) based window form. The MDI links the three moods namely data preparation, training, and testing. It provides a base for many other window-forms and works like a bridge between all other modules. Figure 13.1 shows the main MDI form of our end-to-end OCR system. Further, we explain each mode in detail and pictorially illustrate their

²http://wiki.qt.io/About_Qt

³<https://wiki.python.org/moin/PyQt>

⁴<http://doc.qt.io/qt-5/qt designer-manual.html>



Figure 13.1: *Multiple Document Interface (MDI), bridge all the main components of end-to-end OCR system.*

functionalities.

13.3.1 Data Preparation Mode

Our proposed end-to-end system provides varieties of features regarding data preparation. It makes our end-to-end system more attractive regarding data preparation and its subsequent use. Under data preparation mode, there are four different sub-tasks. These sub-tasks are scan images, pre-process data, data split, and create NetCDF files. The functionalities of these modules are given below in more detail.

Scan Images

Scanning module provides interaction with the default scanner, attached to the computer. This module could easily be invoked by clicking the push-button captioned as *Scan Images*. In fact, it launches the scanner's software and provides access to many features of the scanner. These functions include resolution settings, color settings/grey-level, and file format (.PDF, .JPG, etc.). In case, if there is no scanner installed then it gives a message that *Scanner is not installed!*. Figure 13.2 shows the form that provides scanning facility.



Figure 13.2: Acquisition of text-images via scanner, this module provides the scanning facility of text-images.

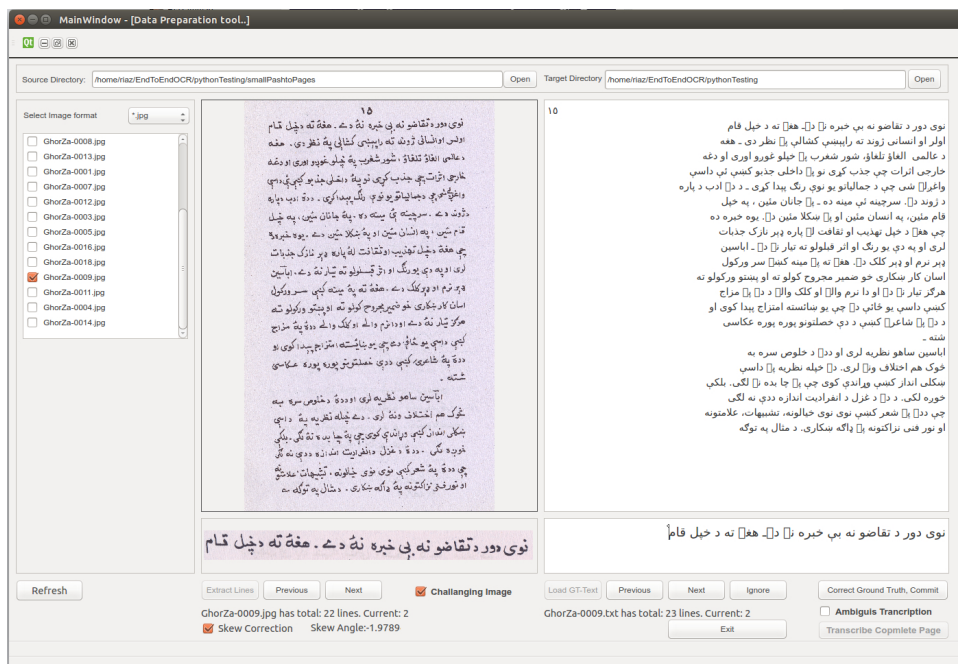


Figure 13.3: This module provides data pre-processing, in this step we can de-skew document, extract text-lines, and make annotation correction if needed.

Pre-process Images

Scanned images often contain skew, which subsequently creates issues for accurate line segmentation. To address these matters, we included an essential pre-process module,

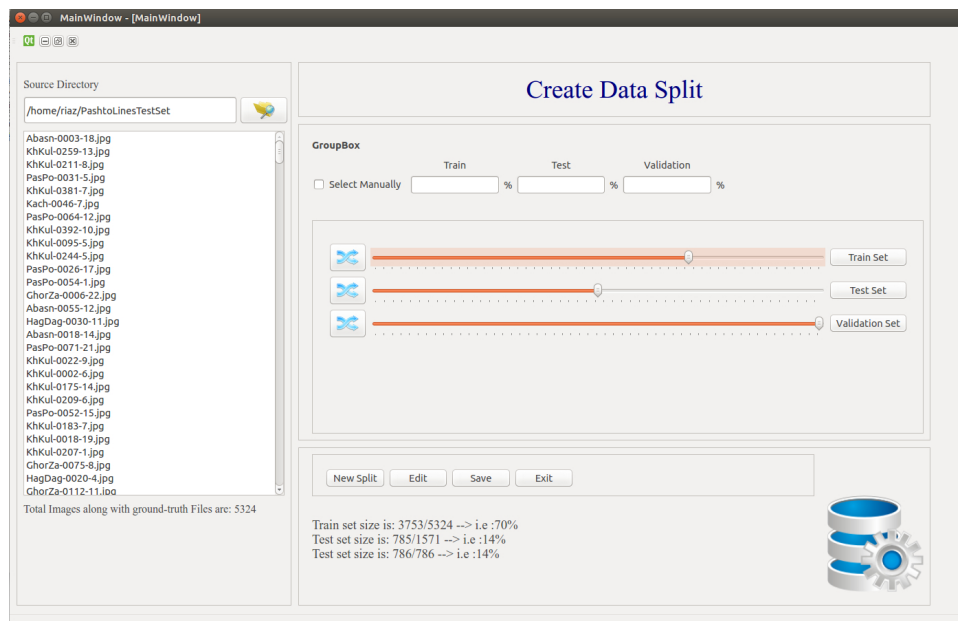


Figure 13.4: *This form provides a convenient way to split data. The lower display area shows the detail information about each train, test, and validation sets.*

that provides three types of functionalities, (1) skew correction, (2) text-line extraction, and (3) annotation correction if needed. This module integrates the contributions of this thesis made for skew correction and text-line segmentation and provides a testbed for the proposed methods. In addition to that, we designed the module in such manner that it could be easily configured for other methods as well. Figure 13.3 illustrates the window-form that provides pre-processing functionality. The module helps the user in preparing the data for further processing, in particular for an appropriate split among train, test and validation sets.

Create Split

The next step after preparing data is to split the data into a proper train, test, and validation sets. The module *Split Data* provides a convenient way for users to split data into parts with custom sizes. It offers two options to split data, in the first option, the user can specify the size of the train, test, and validations sets by setting integer values in percentage (%). In the second option, the user can do the same by moving a slider to a particular value. Splitting the data using the sliders is very convenient and easy way for a user. Figure 13.4 shows the window-form that provides the facility of split data. Each slider has a push-button at the left side, that can be used to shuffle the entire data before committing the partition. Similarly, on the right side, there is also a push-button, it can be used to save and determine the desired partition as train, test or validation sets.

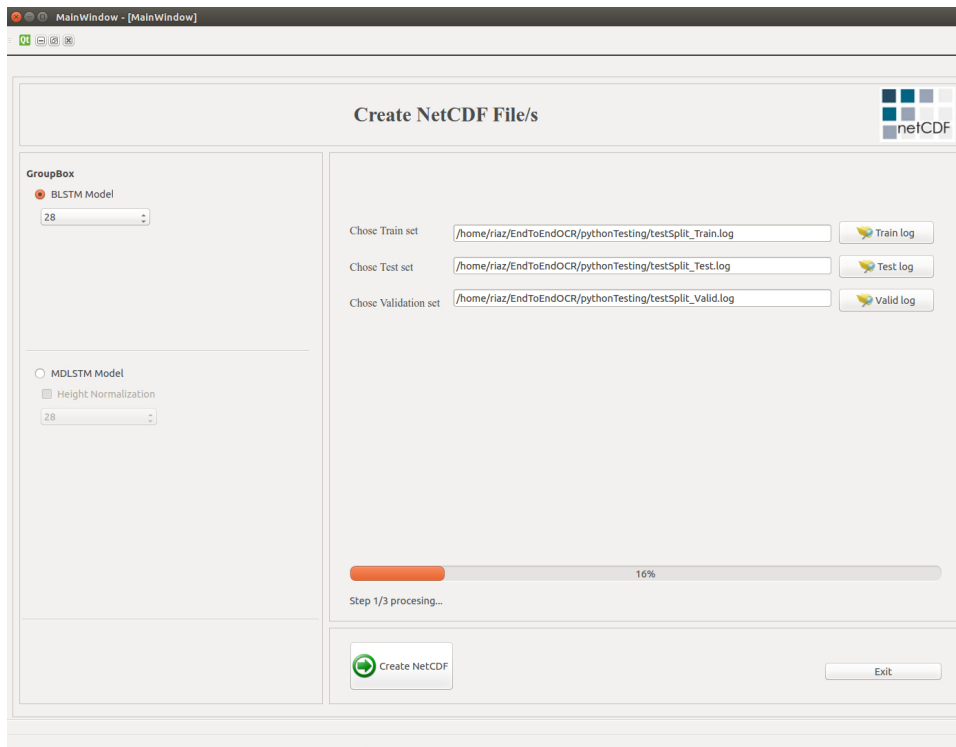


Figure 13.5: *This provides a simple process to create NetCDF files for train, test, and validation sets.*

Prepare NC Files (NetCDF)

In the data preparation mode, the final step is about to transform the data into such form that is convenient for processing by Neural Network models. In fact, in machine learning, the data should be array-oriented and machine-independent such that it could easily be shared and process for efficient scientific computations. We provide this because the RNNLIB needs the data in Network Common Data Form/format (NetCDF)⁵. It is worth mentioning, that the creation of NetCDF data is dependent on models like BLSTM and MDLSTM, and will be different for both models. Therefore, we introduce an easy and flexible interface that provides easy steps for creation of such data files. Figure 13.5 shows the window-form where we can easily select the target architectures and can create their corresponding NetCDF files.

13.3.2 Training Mode

Once we have proper data in NetCDF format, we can train deep based learning models. For training purpose, we introduce training-mode, that is one of the core components of our end-to-end system. This mode mainly provides four modules, which include model-

⁵This project took advantage of netCDF software developed by UCAR/Unidata (<http://doi.org/10.5065/D6H70CW6>).

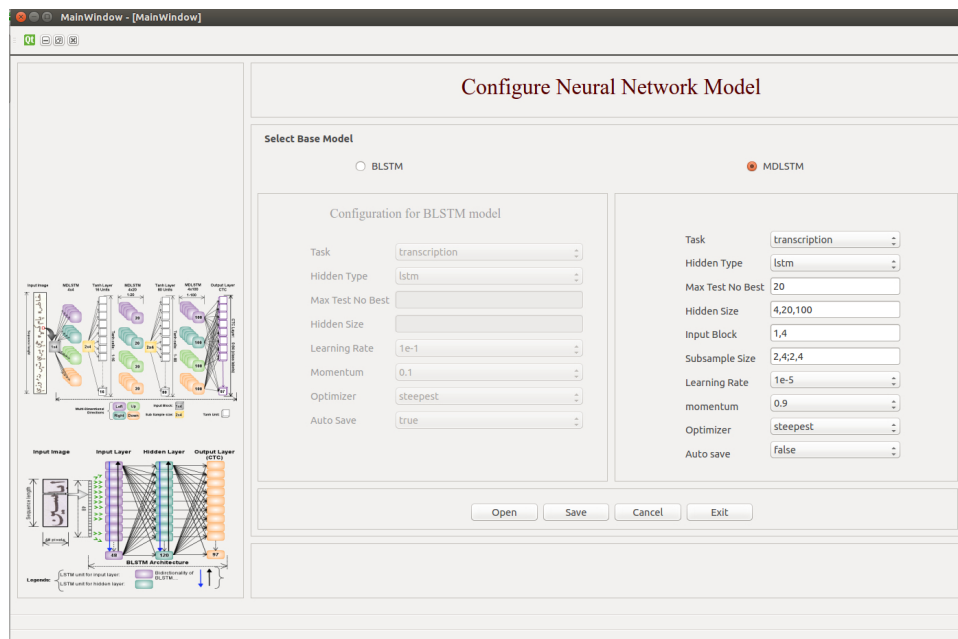


Figure 13.6: *This module provides facility to configure a network model by choosing either BLSTM or MDLSTM architectures.*

configuration, selection of an existing-model, initiate a training process, and show the status of currently running training. In the next sections, we briefly discuss these modules with the help of figures.

Configure New Model

This module provides an access and helps the user to define and configure new training model for an experiment task. The current version of this module provides configuration facilities for both BLSTM and MDLSTM architectures. In addition to that, it also provides an easy way to choose the different parameters (i.e., hidden-layer size, *tanh*-layer size, description of classification, etc.) for the intended architecture. Figure 13.6 shows the window-form that illustrates the functionality of how to configure new RNN based network model. Once, a user is ready to save the configuration. The *Save* button could do the rest of the work. It is important to note that, clicking this button will thoroughly check the parameters, and will make sure that the parameters provided are in the required format. After this, the module asks for a suitable name for the network. We recommend that also note down this name, as it is critical and used in many other places to refer the training as well as the network model for reusing scenarios. Further, the name should be more descriptive, so that a user can get the idea that for what particular task the model was used.

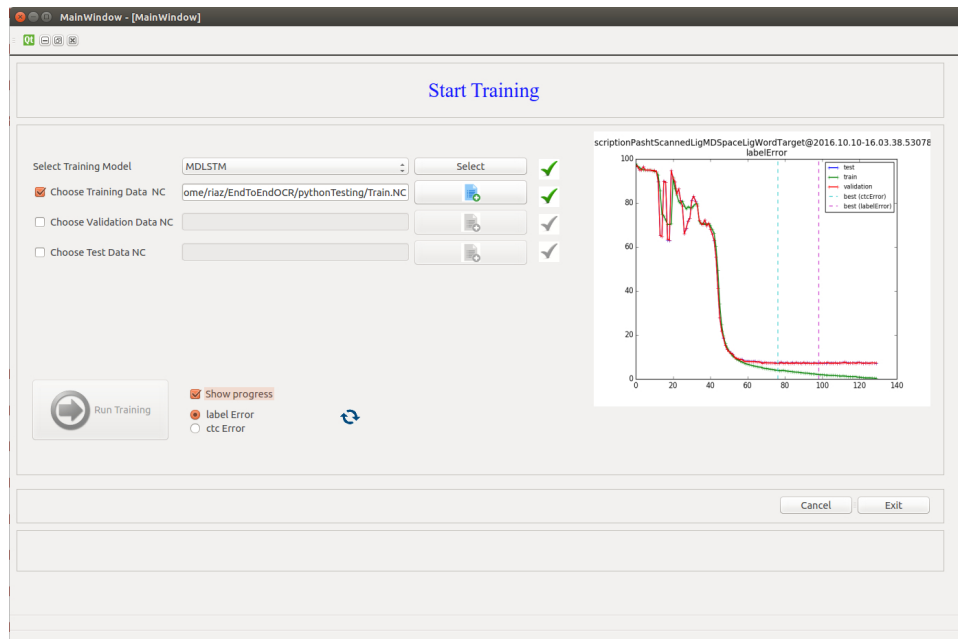


Figure 13.7: *This module provides training facility; the user should select the NetCDF files corresponding to each train, test, and validation sets. To start training, just click the Start Train button.*

Existing Model

Another, if a user wants to evaluate some already existing model, then this can be done by just clicking the button control captioned as *Existing Model*. This module provides easy access to the already available model, and one can rename as well as change its other parameters.

Start Training

Once the configuration of a network model completes, the initiation of training becomes easy on a configured model. This module in the training mode provides the training facility. Training could be starting by just clicking a push-button named as *Start Training*. First, this module asks the user to select the configured model and then requires the data files created in NetCDF format. Figure 13.7 shows the overall training environment of our end-to-end system. Finally, by clicking the *Start Train* will launch the training process and its done!

Here it is worth mentioning that the minimum requirement to start training is the selection of NetCDF data of a train set. However, a user can choose the NetCDF files for validation and test sets as well. In the latter case, the network not only validates its learning but it will give accuracy for the test set as well. However, if one skips choosing the test set, later on, the test mode/function provides the facility to test a trained model.

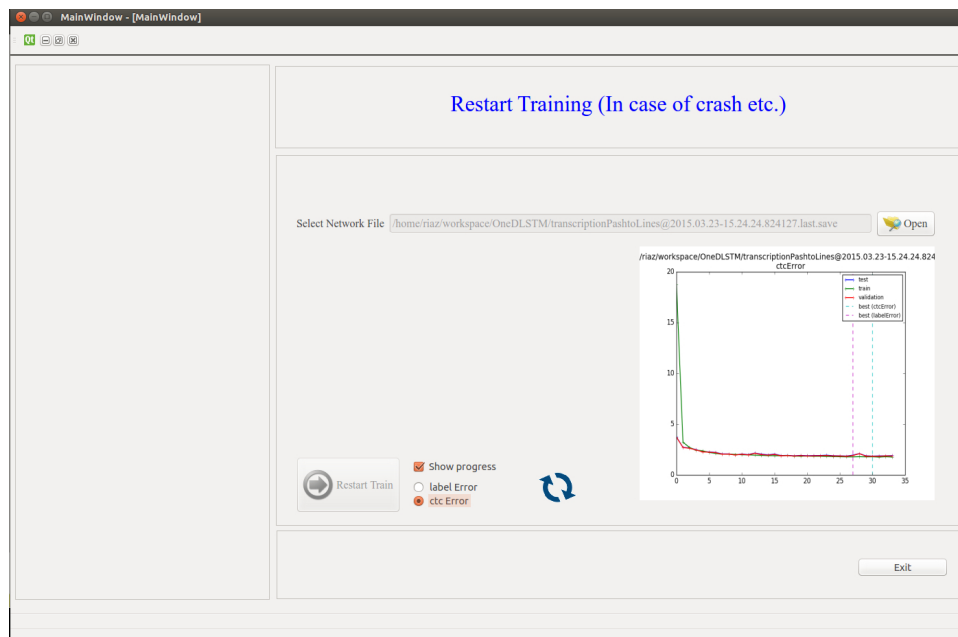


Figure 13.8: *This module provides restarting facility of a training process. Usually, the training may take longer time, and thus some external factors can cause to interrupt the training process. In this case, this module helps the user to restart/resume the training process once again.*

On this form, there is a progress indication, if it shows running status, then it means the training already started. However, the checkbox captioned as *Show progress* remains to disable until the training completes the first epoch successfully. Once this checkbox becomes enable, it provides two options, either to show label-error plot or ctc-error⁶ plot.

As RNNLIB mainly runs at the back-end and performs the training process, therefore, the user can close the GUI of the end-to-end system. We have designed the system such that its closure does not affect the running processes. When a user launches the *Start Train* module some next time, the module itself could implicitly be linked to the running training instance. However, in case of multiple training processes, a user can choose a training process among the running training procedures.

Restart Training

The training process usually runs for a longer duration, as it depends on data as well as the deepness of the network model. Therefore, it is inevitable some time to stop the training process, or the system can crash, or may be a failure of power can interrupt the training process, etc. In this case, a user must want to restart the training process from where it stopped. We have provided a module that provides a facility to the user,

⁶CTC is the Connectionist Temporal Classification error, and is equal to the negative log probability of correctly labelling the entire input sequence.

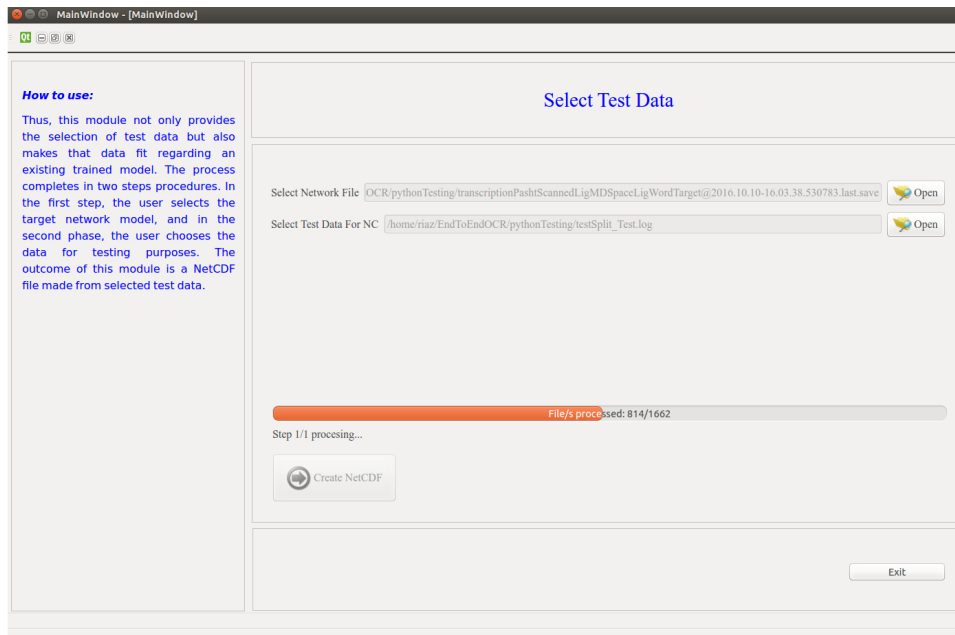


Figure 13.9: (*Testing Mode, Select Test Data*), This module selects test data and provides NetCDF file for that data. The progress-bar shows the number of images processed so far. The module gets essential information from the already trained model and makes the NetCDF files of test data compatible with the trained model.

such that one can resume the training process. In fact, the RNNLIB stores the network files after an epoch, where the network has the best errors. The module provides access to these network-files and relaunches the training process by assessing a user via GUI. Figure 13.8 shows the window-form that describes the functionality regarding reviving training process.

13.3.3 Testing Mode

Our end-to-end system provides another essential function known as a Testing mode. Testing of existing models or benchmarking the different dataset via current models are the most demanding research task in DIA. Thus, we provide a *test module* that provides such facility for investigating the existing models on different datasets. The testing module comprises of three submodules, including *selection of test data*, *start testing*, and *plot errors*. The next sections briefly describe each submodule.

Select Test Data

In this module, the user can select the test data for evaluating an existing already trained model. Another significant use of this module lies in a fact that usually the classes/labels

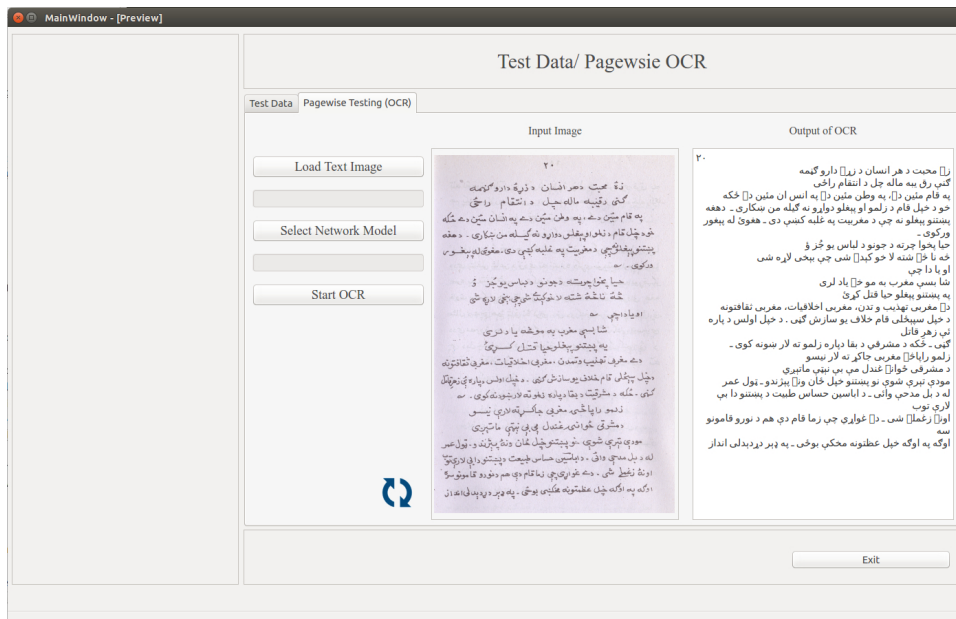


Figure 13.10: *This form provides pagewise OCR service. It loads the image first. Then the trained model is selected by clicking Select Network Model button. Then finally by clicking Start OCR system, the user can get OCR results on the text editor at the right side of the input image.*

used in training set should also present in a test set. In other words, the class labels of the test set should be same or should be a subset of class names used in a train set. Thus, this module not only provides the selection of test data but also makes that data fit regarding an existing trained model. The process completes in two steps procedures. In the first step, the user selects the target network model, and in the second phase, the user chooses the data for testing purposes. The outcome of this module is a NetCDF file made from selected test data. Figure 13.9 shows the window-form that provides the facility of choosing the testing data for further benchmarking.

Start Testing

To start only testing on an already trained model, we have introduced *Start Testing* module which is one of the essential components of our end-to-end OCR system. This module provides two-fold benefits to end user. One, it provides testing on data obtained from a chunk of image documents. The NetCDF file created in *Select Test Data* module is given along with trained network model, and the results are obtained. If the network file and the NetCDF are not compatible, it pops ups an error. Second, it also provides the testing facility on a single page. The latter service is more attractive for a user, as one can realize the essence of an OCR system. This service loads a text image, then requires the network file (already trained), and finally, by clicking "Start OCR" button, we can get the results. The input image and the OCR results are displayed along sides.

Figure 13.10 shows the windows-form that provides page wise OCR results.

13.3.4 Closure of End-to-End OCR System

Finally, to close the proposed end-to-end system, a user can just click a push-button captioned as *Exit Application*. A message appears and confirms the closure of the application. In addition to that, before the application closed, a background process collects all the information regarding the currently running processes initiated by the end-to-end system. After gathering information, the front-end/GUI closes successfully.

13.4 Conclusion

In this chapter, we have presented the practical implementation of our proposed methods in the form of an end-to-end system. As already mentioned, that there is very less work regarding the complete pipeline of DIA system that focuses on Arabic like cursive scripts. Therefore, we have introduced an entire pipeline covering the most important features of a typical DIA system. This chapter describes the complete structure of our proposed end-to-end system. We have also illustrated the functionalities of different modules with the help of snap shots and figures. We used RNNLIB as a back-end engine, which equips with the deep-learning paradigm and provides the implementation of LSTM with scanning capability in bidirectional as well as in multi directional. For front-end/GUI, we used Qt Designer, which makes our GUI more flexible and convenient for end users. Furthermore, we have integrated our proposed methods into this end-to-end system, in particular, skew detection and line-segmentation techniques.

Conclusion and Future Work

This chapter concludes the thesis regarding its contributions and the objectives achieved during this thesis. Furthermore, it also shows the limitations and elaborates factors that may limit the efficiency and accuracy of the presented work. In addition to that, this chapter also shares the future work that could strengthen this study in particular and will facilitate the research regarding entire DIA system in general.

14.1 Conclusion

This thesis addresses the problem of recognition of cursive scripts considering the Pashto language as a test case. In general, cursive scripts that derived from Arabic script present complex behavior towards sophisticated OCR system. The major complexities include dependence on shape-context, shape variations per class, the complication in segmentation, varieties in fonts, calligraphic beauty, and much more. In addition to that, each language establishes its distinct challenges that could make it hard for an established OCR to generalize well. In this regard, this thesis has a distinction and takes the Pashto language, which is the superset of Arabic, Persian, and Urdu languages. Further, research regarding the Pashto language is quite limited, and the available OCR systems could not provide satisfactory results on the Pashto text. It presents a research gap and might need exploration towards sophisticated OCR system. Furthermore, the Pashto language is famous for its rich literary heritage, and results in a lot of written material. This material needs digitization and thus signifies the importance of this work. Therefore, this research is the pioneering study regarding the OCR research for the Pashto language. This thesis also covers nearly all the aspects that could cause problems as well as could contribute towards the solutions of OCR for Arabic like scripts. This research has achieved contributions in three different levels that are data level, conceptual level, and practical level.

This thesis contributes in two major datasets for the Pashto OCR system. The first dataset is based on synthetic data and contains 1000 unique Pashto ligatures with 40 scales and 12 rotation variations per scale. The dataset is suitable for investigating the impact of scale and rotation variations on the recognition of cursive script languages. This synthetic dataset has two different versions namely Ligature-Based-II and Ligature-Based-III.

The second significant contribution regarding data level is the creation of real Pashto text image-base named KPTI. The KPTI dataset is created keeping the major points in mind that which written materials are more important for digitization concerning the Pashto language. As the Pashto text mainly penned by scribes *Katibs*, and consequently retain the most complex patterns due to human involvement. Therefore, in this thesis, we focused on such scribed data and created a comprehensive dataset. The KPTI dataset thus provides a useful benchmark for investigating not only the Pashto language but also the other cursive scripts.

The second major contribution of this thesis is the conceptual contribution. The thesis provides depth analysis of Arabic like languages and presents fundamental factors which are mainly causing complexities in OCR systems. Therefore, provides the comprehensive materials regarding shape similarity among characters, interchangeably used characters, insertion and omission of the space character, ligature formation rules, primary-ligatures, and their shape-codes. In addition to that, we found that the MDRNN based methods learn the additional parts (i.e., dots, diacritics, etc.) of the Pashto by more than 6% compared to Bidirection RNN. This difference is found very less in other scripts, particularly in Latin writings.

Another conceptual contribution is the identification of breaker-characters. Though other researchers have already discussed these characters, they have not reported the association of these breaker-characters with space-anomaly, post processing, and ultimately with rendering after OCR. Therefore, this thesis provided comprehensive discussions regarding space-anomaly in languages derived from Arabic scripts.

The practical contributions of this thesis are of manifolds. It presents skew detection and correction method by exploiting the concept of parallelism of text lines in a text document. The datasets like Tobacco800, KPTI, and DISEC are tested, and we have achieved better results compared to the state-of-the-art methods. Another practical contribution is the introduction of a novel text-line segmentation technique for scanned documents. The primary motivation was the poor results of state-of-the-art methods in the segmentation of large headings and titles. Our line line-segmentation method is based on Hanning-window filter plus horizontal projection profile. The method has shown better results compared to state-of-the-art methods, i.e., Ryu's method [RKC14].

Another practical contribution is the benchmarking of deep learning approach regarding scale and rotation invariant recognition of Pashto text. In this regard, we have evaluated SIFT descriptor based matching technique and compared to HMM and MDLSTM classifiers. We found MDLSTM gives better results compared to SIFT and HMM based approaches. The MDLSTM based approach gives us nearly 99% ligature recognition accuracy on Ligature-BasedIII dataset. It signifies the skill of LSTM and its hallmark regarding scanning capability in multi-dimensional. However, this work relies on the assumption that the text should be segmented in ligatures. In fact, in cursive scripts like Pashto and especially in handwritten text, extraction of the ligature is a very complex task. So, this approach could work in printed text rather than handwritten text. Another, the number of ligatures in a language itself causing scalability issue, and strategies that rely on temporal or context learning, could face bottlenecks in succeeding good results.

Another practical contribution is the improvisation of a first-ever baseline for the recognition of real Pashto text. In this contribution, we have investigated the state-of-the-art deep learning approach based on LSTM architectures. We proposed two recognition systems based on the most popular variants of LSTM, i.e., BLSTM and MDLSTM. Both systems are evaluated on our newly presented KPTI dataset. The results show that MDLSTM is far better than BLSTM concerning Pashto text recognition. The proposed MDLSTM model achieves 9.22% character error rate (CER), while BLSTM model achieves 16.16% CER. Furthermore, MDLSTM model has proved that height normalization in our case does not influence the accuracy, which validates the effectiveness of MDLSTM respecting registration/scale variations. The results of this work have also confirmed the Pashto-specific challenges regarding shape similarities and different characters (i.e., Pashto ﻱ [Yey]s, and ﻭ ﻭ ﻩ , ﻩ). These characters have shown their association in top-10 confusions.

Furthermore, the first two confusions in confusion-matrix are related to space misclassification. This thesis dealt this issue on conceptual plus empirical way. We have provided a joint approach based on theoretical findings related to breaker-characters, and have proposed a modification in the default ground-truth. The proposed approach not only improves the overall accuracy by 3% but also reduce the space misclassification by 80%. Thereby 80% gain in space classification means 80% improvement in rendering phase after OCR. As these spaces are used for break-up purposes and cause calligraphic beauty in text rendering, therefore, their misclassification leads to odd look of relevant text.

Another, a significant practical contribution is the integration of all the outcomes of this thesis into one platform, i.e., an end-to-end OCR system. Nearly, all individual research in Ph.D. studies could not contribute towards a complete end-to-end system. This thesis has a distinction and provides a complete OCR system with flexible and easy graphical user interface (GUI). The system covers almost all the stages from scanning to

post-processing/rendering. The system is designed to integrate the future enhancements easily. The system presents a feeling of a synergy effect of all the different module for the accomplishment of an OCR tasks. The end-to-end system uses the power of LSTM based library, i.e., RNNLIB as a back-end engine, while the front-end is designed in Qt Designer with full adherence of object oriented programming (OOP).

14.2 Future Work

Two major future directions appeared during the progression of this thesis including (1) the recognition of cursive scripts, and (2) enhancement in end-to-end OCR system. Though our proposed MDLSTM based system has succeeded in getting 6.33% CER (Section 12.6) on real KPTI dataset, there is a need for further research work to improve these results. A proper direction might be the one based on ligature based classification. As benchmarking the Pashto Ligature-BasedIII dataset, we got clues that if achieving 99% ligature recognition rate for the scale and rotation variation using MDLSTM model is possible then we could get better results considering ligatures as recognizable units instead of characters in a regular text. However, the issue of scalability and its handling under temporal classification is an open research question and needs further explorations. For example, in case of Pashto language, we achieved 19,268 unique ligatures (classes) in 2.3 million Pashto words. Thus handling 19,268 entities under context-learning scenario needs a comprehensive research to beat the performance of recognition systems using character based classification.

In addition to that, ligature based classification requires a uniform distribution of instances per class in a training set, while real data does not provide correspondent distribution of ligatures due to irregularity in frequencies of ligatures. Further, we could achieve some initial results by offering uniform instances of each class in a training set. However, it is only possible via synthetic dataset. I believe that exploiting the power of modern GPUs¹ along with parallel programming principal we could achieve this milestone.

The second future direction is related to equipping the end-to-end system with more enhanced features. As the current version of our end-to-end system is designed mainly for cursive scripts, its functionality can be extended to other scripts as well. Another, we used RNNLIB as a back-end OCR engine, there might be a good future work to integrate OCROpus² and Tesseract so that the researchers could find analysis regarding comparison, etc. In addition to that, a future work relating to the integration of *Tensorflow*³ and

¹Graphics Processing Units

²<https://de.wikipedia.org/wiki/OCROpus>

³<https://www.tensorflow.org/>

*caffe*⁴ libraries/frameworks would be another significant contribution to enhance the capability of our end-to-end system. This combination will make the system more efficient regarding abstraction towards designing and utilization of GPU's power.

⁴<http://caffe.berkeleyvision.org/>

Bibliography

- [AAF⁺06] G Agam, S Argamon, O Frieder, D Grossman, and D Lewis. The complex document image processing (cdip) test collection. *Illinois Institute of Technology*, 2006.
- [AAF10] AF Al-Azawi and Moayad A Fadhil. Arabic text steganography using kashida extensions with huffman code. j. *Applied Sci*, 10:436–439, 2010.
- [AAHLB14] Mayce Al Azawi, Adnan Ul Hasan, Marcus Liwicki, and Thomas M Breuel. Character-level alignment using wfst and lstm for post-processing in multi-script recognition systems-a comparative study. In *International Conference Image Analysis and Recognition*, pages 379–386. Springer, 2014.
- [AAK10] Riaz Ahmad, Syed Hassan Amin, and Mohammad AU Khan. Scale and rotation invariant recognition of cursive pashto script using sift features. In *Emerging Technologies (ICET), 2010 6th International Conference on*, pages 299–303. IEEE, 2010.
- [AALB15] Mayce Al Azawi, Marcus Liwicki, and Thomas M Breuel. Combination of multiple aligned recognition outputs using wfst and lstm. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 31–35. IEEE, 2015.
- [AAR⁺15a] Riaz Ahmad, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, and Thomas Breuel. Scale and rotation invariant ocr for pashto cursive script using mdlstm network. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1101–1105. IEEE, 2015.

- [AAR⁺15b] Riaz Ahmad, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, Andreas Dengel, and Thomas Breuel. Recognizable units in pashto language for ocr. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1246–1250. IEEE, 2015.
- [AAR⁺16] Riaz Ahmad, M Zeshan Afzal, S Faisal Rashid, Marcus Liwicki, Thomas Breuel, and Andreas Dengel. Kpti: Katib’s pashto text imagebase and deep learning benchmark. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 453–458. IEEE, 2016.
- [ABK⁺12] Muhammad Zeshan Afzal, Syed Saqib Bukhari, Martin Krämer, Faisal Shafait, and Thomas M Breuel. Robust stereo matching for document images using parameter selection of text-line extraction. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 331–334. IEEE, 2012.
- [ABM95] Badr Al-Badr and Sabri A Mahmoud. Survey and bibliography of arabic optical text recognition. *Signal processing*, 41(1):49–77, 1995.
- [ACM⁺15] Muhammad Zeshan Afzal, Samuele Capobianco, Muhammad Imran Malik, Simone Marinai, Thomas M Breuel, Andreas Dengel, and Marcus Liwicki. Deepdocclassifier: Document classification with deep convolutional neural network. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1111–1115. IEEE, 2015.
- [AH17] Q UIAin Akram and Andreas Hussain, Sarmad. Ligature-based font size independent ocr for noori nastalique writing style. In *2017 IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR)*, 2017.
- [AHK08] Ashraf AbdelRaouf, Colin A Higgins, and Mahmoud Khalil. A database for arabic printed character recognition. In *International Conference Image Analysis and Recognition*, pages 567–578. Springer, 2008.
- [AK17] Markus Ebbecke Marcus Liwicki Andreas Klsch, Muhammad Zeshan Afzal. Real-time document image classification using deep cnn and extreme learning machines. 2017.
- [AKAL17] Muhammad Zeshan Afzal, Andreas Kölsch, Sheraz Ahmed, and Marcus Liwicki. Cutting the error by half: Investigation of very deep cnn and advanced training strategies for document image classification. 2017.
-

- [AKB⁺12] Muhammad Zeshan Afzal, Martin Kramer, Syed Saqib Bukhari, Faisal Shafait, and Thomas M Breuel. Improvements to uncalibrated feature-based stereo matching for document images by using text-line segmentation. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 394–398. IEEE, 2012.
- [AKB⁺13] Muhammad Zeshan Afzal, Martin Krämer, Syed Saqib Bukhari, Mohammad Reza Yousefi, Faisal Shafait, and Thomas M Breuel. Robust binarization of stereo and monocular document images using percentile filter. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 139–149. Springer, 2013.
- [AMAHA12] Somaya Al Máadeed, Wael Ayouby, Abdelâali Hassaïne, and Jihad Mohamad Aljaam. Quwi: an arabic and english handwriting dataset for offline writer identification. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 746–751. IEEE, 2012.
- [ANA⁺15] Riaz Ahmad, Saeeda Naz, Muhammad Zeshan Afzal, Sayed Hassan Amin, and Thomas Breuel. Robust optical recognition of cursive pashto script using scale, rotation and location invariant approach. *PloS one*, 10(9):e0133648, 2015.
- [ANA⁺17a] Riaz Ahmad, Saeeda Naz, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, and Andreas Dengel. Deepkhatt: A deep learning benchmark on arabic script. In *Accepted, Document Analysis and Recognition (ICDAR), 2017 14th International Conference on*. IEEE, 2017.
- [ANA⁺17b] Riaz Ahmad, Saeeda Naz, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, and Andreas Dengel. Deepkhatt: A deep learning benchmark on arabic script. In *Accepted, Document Analysis and Recognition (ICDAR), 2017 14th International Conference on*. IEEE, 2017.
- [ANA⁺17c] Riaz Ahmad, Saeeda Naz, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, and Andreas Dengel. The impact of visual similarities of arabic-like scripts in terms of learning in an ocr system. In *Accepted, Document Analysis and Recognition (ICDAR), 2017 14th International Conference on*. IEEE, 2017.
- [ANA⁺17d] Riaz Ahmad, Saeeda Naz, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, and Andreas Dengel. The impact of visual similarities of arabic-like scripts in terms of learning in an ocr system. In
-

- Accepted, Document Analysis and Recognition (ICDAR), 2017 14th International Conference on.* IEEE, 2017.
- [ANR⁺16a] Saad Bin Ahmed, Saeeda Naz, Muhammad Imran Razzak, Shiekh Faisal Rashid, Muhammad Zeeshan Afzal, and Thomas M Breuel. Evaluation of cursive and non-cursive scripts using recurrent neural networks. *Neural Computing and Applications*, 27(3):603–613, 2016.
- [ANR⁺16b] Saad Bin Ahmed, Saeeda Naz, Muhammad Imran Razzak, Rubiyah Yusuf, and Thomas M Breuel. Balinese character recognition using bidirectional lstm classifier. In *Advances in Machine Learning and Signal Processing*, pages 201–211. Springer, 2016.
- [Ant01] Martin Anthony. *Discrete mathematics of neural networks: selected topics*. SIAM, 2001.
- [AOS09] Zaheer Ahmad, Jehanzeb Khan Orakzai, and Inam Shamsher. Urdu compound character recognition using feed forward neural networks. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 457–462. IEEE, 2009.
- [APPS⁺15] Muhammad Zeshan Afzal, Joan Pastor-Pellicer, Faisal Shafait, Thomas M Breuel, Andreas Dengel, and Marcus Liwicki. Document image binarization using lstm: A sequence learning approach. In *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*, pages 79–84. ACM, 2015.
- [ARA⁺16] Riaz Ahmad, S Faisal Rashid, M Zeshan Afzal, Marcus Liwicki, Andreas Dengel, and Thomas Breuel. A novel skew detection and correction approach for scanned documents. In *DAS 2016, 12th Intl IAPR Workshop on Document Analysis Systems, At Santorini, Greece*, 2016.
- [ARA⁺17] Riaz Ahmad, S Faisal Rashid, M Zeshan Afzal, Marcus Liwicki, and Andreas Dengel. Text-line segmentation of large titles and headings in arabic like script. In *2017 IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR)*, 2017.
- [ASAOZ14] Atallah M Al-Shatnawi, Bader M Alfawwaz, Khairuddin Omar, and Ahmed M Zeki. Skeleton extraction: Comparison of five methods on the arabic ifn/enit database. In *Computer Science and Information Technology (CSIT), 2014 6th International Conference on*, pages 50–59. IEEE, 2014.
-

- [ASS07] Manivannan Arivazhagan, Harish Srinivasan, and Sargur Srihari. A statistical approach to line segmentation in handwritten documents. In *Electronic Imaging 2007*. International Society for Optics and Photonics, 2007.
- [BAB14] Saurav Biswas, Muhammad Zeshan Afzal, and Thomas Breuel. Using recurrent networks for non-temporal classification tasks. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 135–140. IEEE, 2014.
- [Bai95] Henry S Baird. The skew angle of printed documents. In *Document image analysis*, pages 204–208. IEEE Computer Society Press, 1995.
- [Ban03] Erinn Banting. *Afghanistan: The Land*. Crabtree Publishing Company, 2003.
- [BC01] Horst Bunke and Terry Caelli. *Hidden Markov models: applications in computer vision*, volume 45. World Scientific, 2001.
- [BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [BKD95] Dan S Bloomberg, Gary E Kopec, and Lakshmi Dasari. Measuring document image skew and orientation. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, pages 302–316. International Society for Optics and Photonics, 1995.
- [BLM16] Théodore Bluche, Jérôme Louradour, and Ronaldo Messina. Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. *arXiv preprint arXiv:1604.03286*, 2016.
- [BLT09] Shuyong Bai, Linlin Li, and Chew Lim Tan. Keyword spotting in document images through word shape coding. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 331–335. IEEE, 2009.
- [BSB08] Syed Saqib Bukhari, Faisal Shafait, and Thomas M Breuel. Segmentation of curled textlines using active contours. In *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*, pages 270–277. IEEE, 2008.
- [BSB09] Syed Saqib Bukhari, Faisal Shafait, and Thomas M Breuel. Script-independent handwritten textlines segmentation using active contours.
-

- In *2009 10th International Conference on Document Analysis and Recognition*, pages 446–450. IEEE, 2009.
- [BT58] Ralph Beebe Blackman and John Wilder Tukey. The measurement of power spectra from the point of view of communications engineering part i. *Bell System Technical Journal*, 37(1):185–282, 1958.
- [BTV12] Vicente Bosch, Alejandro Hector Toselli, and Enrique Vidal. Statistical text line analysis in handwritten documents. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 201–206. IEEE, 2012.
- [Car15] James Caron. The lives of amir hamza shinwari: on personal histories against an imperial border. *Tanqeed; a magazine of politics and culture*, 10:43–53, 2015.
- [CDM03] Peter J Claus, Sarah Diamond, and Margaret Ann Mills. *South Asian Folklore: An Encyclopedia: Afghanistan, Bangladesh, India, Nepal, Pakistan, Sri Lanka*. Taylor & Francis, 2003.
- [CK94] Jia-Lin Chen and Amlan Kundu. Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):208–214, 1994.
- [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.
- [CP97] BB Chaudhuri and U Pal. An ocr system to read two indian language scripts: Bangla and devnagari (hindi). In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 2, pages 1011–1015. IEEE, 1997.
- [CP98] BB Chaudhuri and U Pal. A complete printed bangla ocr system. *Pattern recognition*, 31(5):531–549, 1998.
- [CRC13] Youssouf Chherawala, Partha Pratim Roy, and Mohamed Cheriet. Feature design for offline arabic handwriting recognition: handcrafted vs automated? In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 290–294. IEEE, 2013.
- [CSD⁺88] G Ciardiello, G Scafuro, MT Degrandi, MR Spada, and MP Roccotelli. An experimental system for office document handling and text recog-
-

- dition. In *Proc. 9th Int. Conf. on Pattern Recognition*, pages 739–743, 1988.
- [DAHG⁺15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.
- [Dav14] Anne David. *Descriptive grammar of Pashto and its dialects*, volume 1. Walter de Gruyter, 2014.
- [DEKM98] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.
- [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.
- [DJN08] Philippe Dreuw, Stephan Jonas, and Hermann Ney. White-space models for offline arabic handwriting recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
- [DKN14] Patrick Doetsch, Michal Kozielski, and Hermann Ney. Fast and robust training of recurrent neural networks for offline handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 279–284. IEEE, 2014.
- [DMN04] Michael Decerbo, Ehry MacRostie, and Premkumar Natarajan. The bbn byblos pashto ocr system. In *Proceedings of the 1st ACM workshop on Hardcopy document processing*, pages 29–32. ACM, 2004.
- [Doe98] David Doermann. The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998.
- [Edd96] Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.
- [EHL5M05] Ramy El-Hajj, Laurence Likforman-Sulem, and Chafic Mokbel. Arabic handwriting recognition using baseline dependant features and hidden markov modeling. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 893–897. IEEE, 2005.
-

- [Fab14] Jonathan Fabrizio. A precise skew estimation algorithm for document images using knn clustering and fourier transform. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2585–2588. IEEE, 2014.
- [FAZQA⁺16] Ali Farhat, Ali Al-Zawqari, Abdulhadi Al-Qahtani, Omar Hommos, Faycal Bensaali, Abbas Amira, and Xiaojun Zhai. Ocr based feature extraction and template matching algorithms for qatari number plate. In *Industrial Informatics and Computer Systems (CIICS), 2016 International Conference on*, pages 1–5. IEEE, 2016.
- [FGS08] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. Phoneme recognition in timit with blstm-ctc. *arXiv preprint arXiv:0804.3269*, 2008.
- [FQXS14] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. In *Interspeech*, pages 1964–1968, 2014.
- [Fri94] Bernd Fritzke. Growing cell structures a self-organizing network for unsupervised and supervised learning. *Neural networks*, 7(9):1441–1460, 1994.
- [GEBS04] Alex Graves, Douglas Eck, Nicole Beringer, and Juergen Schmidhuber. Biologically plausible speech recognition with lstm neural nets. In *International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pages 127–136. Springer, 2004.
- [GETS99] Andrew Gillies, Erik Erlandson, John Trenkle, and Steve Schlosser. Arabic text recognition system. In *Proceedings of the Symposium on Document Image Understanding Technology*, pages 253–260, 1999.
- [GFGS06] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.
- [GFS05] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pages 753–753, 2005.
- [GLF⁺09] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for
-

- unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2009.
- [GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 iee international conference on*, pages 6645–6649. IEEE, 2013.
- [Gra12a] Alex Graves. Connectionist temporal classification. *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 61–93, 2012.
- [Gra12b] Alex Graves. Offline arabic handwriting recognition with multidimensional recurrent neural networks. In *Guide to OCR for Arabic scripts*, pages 297–313. Springer, 2012.
- [Gra12c] Alex Graves. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 5–13. Springer, 2012.
- [GS05] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.
- [GSK⁺16] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 2016.
- [Hab67] AH Habibi. The cultural, social and intellectual state of the people of afghanistan in the era just before the advent of islam. *Afghanistan. Historical and Cultural Quaterly*, Jg. XX, H, 3:1–7, 1967.
- [Han62] WJ Hannan. The rca multi-font reading machine. *Optical Character Recognition*. Spartan Books, 1962.
- [HDK⁺14] Mahdi Hamdani, Patrick Doetsch, Michal Kozielski, Amr El-Desoky Mousa, and Hermann Ney. The rwth large vocabulary arabic handwriting recognition system. In *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*, pages 111–115. IEEE, 2014.
- [HDN14] Mahdi Hamdani, Patrick Doetsch, and Hermann Ney. Improvement of context dependent modeling for arabic handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 494–499. IEEE, 2014.
- [Hen83] Michael MT Henderson. Four varieties of pashto. *Journal of the American Oriental Society*, pages 595–597, 1983.
-

-
- [Her82] HF Herbert. The history of ocr, optical character recognition. *Manchester Center, VT: Recognition Technologies Users Association*, 1982.
- [HFD90] Stuart C Hinds, James L Fisher, and Donald P D'Amato. A document skew detection method using run-length encoding and the hough transform. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, volume 1, pages 464–468. IEEE, 1990.
- [Hoc98] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [Hus02] Syed Afaq Husain. A multi-tier holistic approach for urdu nastaliq recognition. In *Multi Topic Conference, 2002. Abstracts. INMIC 2002. International*, pages 84–84. IEEE, 2002.
- [HYR86] Akihide Hashizume, Pen-Shu Yeh, and Azriel Rosenfeld. A method of detecting the orientation of aligned components. *Pattern Recognition Letters*, 4(2):125–132, 1986.
- [IK88] John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.
- [IOO91] S Impedovo, L Ottaviano, and S Occhinegro. Optical character recognitiona survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(01n02):1–24, 1991.
- [Ish93] Yasuto Ishitani. Document skew detection based on local region complexity. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 49–52. IEEE, 1993.
- [JKJ04] Keechul Jung, Kwang In Kim, and Anil K Jain. Text information extraction in images and video: a survey. *Pattern recognition*, 37(5):977–997, 2004.
- [KA94] Shyh-Shiaw Kuo and Oscar E. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):842–848, 1994.
- [KAB⁺12] Martin Krämer, Muhammad Zeshan Afzal, Syed Saqib Bukhari, Faisal Shafait, and Thomas M Breuel. Robust stereo correspondence for doc-
-

- uments by matching connected components of text-lines with dynamic programming. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 734–737. IEEE, 2012.
- [Kad02] Mohammed Waleed Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, The University of New South Wales, 2002.
- [Kai81] Ellen M Kaisse. Separating phonology from syntax: A reanalysis of pashto cliticization. *Journal of Linguistics*, 17(02):197–208, 1981.
- [KB13] Er Kavneet Kaur and Vijay Kumar Banga. Number plate recognition using ocr technique. *International Journal of Research in Engineering and Technology*, 2(09), 2013.
- [KC99] Mohammad S Khorsheed and William F Clocksin. Structural features of cursive arabic script. In *BMVC*, pages 1–10. Citeseer, 1999.
- [KC12] Hyung Il Koo and Nam Ik Cho. Text-line extraction in handwritten chinese documents based on an energy minimization framework. *IEEE Transactions on Image Processing*, 21(3):1169–1175, 2012.
- [KFK02] Ergina Kavallieratou, Nikos Fakotakis, and G Kokkinakis. Skew angle estimation for printed and handwritten documents using the wigner–ville distribution. *Image and Vision Computing*, 20(11):813–824, 2002.
- [KGJAF13] Ihab Khoury, Adrià Giménez, Alfons Juan, and Jesús Andrés-Ferrer. Arabic printed word recognition using windowed bernoulli hmms. In *International Conference on Image Analysis and Processing*, pages 330–339. Springer, 2013.
- [Kha47] Ghani Khan. *The Pathans: A Sketch*. National Information and Publications, 1947.
- [Kha13] Muhammad Sher Ali Khan. Ghani khan: The poet-painter (1914-1996). *The Journal of Humanities and Social Sciences*, 21(2):63, 2013.
- [Kho02] Mohammad S Khorsheed. Off-line arabic character recognition—a review. *Pattern analysis & applications*, 5(1):31–45, 2002.
- [KJ13] Mandip Kaur and Simpel Jindal. An integrated skew detection and correction using fast fourier transform and dct. *International Journal of Scientific & Technology Res*, 2, 2013.
-

- [KS05] Mohammed Waleed Kadous and Claude Sammut. Classification of multivariate time series and structured data using constructive induction. *Machine learning*, 58(2):179–216, 2005.
- [LAA⁺06] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666. ACM, 2006.
- [LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [LDL05] Jian Liang, David Doermann, and Huiping Li. Camera-based analysis of text and documents: a survey. *International journal on document analysis and recognition*, 7(2):84–104, 2005.
- [Leh12] Gurpreet Singh Lehal. Choice of recognizable units for urdu ocr. In *Proceeding of the workshop on document analysis and recognition*, pages 79–85. ACM, 2012.
- [Lev66] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.
- [LFA15] Marcus Liwicki¹², Volkmar Frinken, and Muhammad Zeshan Afzal. Latest developments of lstm neural networks with applications of document image analysis. *Handbook of Pattern Recognition and Computer Vision*, page 293, 2015.
- [LG06] Liana M Lorigo and Venugopal Govindaraju. Offline arabic handwriting recognition: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 28(5):712–724, 2006.
- [LGBS07] Marcus Liwicki, Alex Graves, Horst Bunke, and Jürgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proc. 9th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 367–371, 2007.
- [LGPH08] Georgios Louloudis, Basilios Gatos, Ioannis Pratikakis, and Constantin Halatsis. Text line detection in handwritten documents. *Pattern Recognition*, 41(12):3758–3772, 2008.
-

- [LLT08] Shijian Lu, Linlin Li, and Chew Lim Tan. Document image retrieval through word shape coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1913–1918, 2008.
- [Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [LR13] Gurpreet Singh Lehal and Ankur Rana. Recognition of nastalique urdu ligatures. In *Proceedings of the 4th International Workshop on Multilingual OCR*, page 7. ACM, 2013.
- [LS05] Ming Li and Ronan Sleep. A robust approach to sequence classification. In *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pages 5–pp. IEEE, 2005.
- [LSL⁺14] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*, 2014.
- [LSST⁺02] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444, 2002.
- [Lu95] Yi Lu. Machine printed character segmentation; an overview. *Pattern recognition*, 28(1):67–80, 1995.
- [MAA⁺12] Sabri A Mahmoud, Irfan Ahmad, Mohammad Alshayeb, Wasfi G Al-Khatib, Mohammad Tanvir Parvez, Gernot A Fink, Volker Märgner, and Haikal El Abed. Khatt: Arabic offline handwritten text database. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 449–454. IEEE, 2012.
- [MAAK⁺14] Sabri A Mahmoud, Irfan Ahmad, Wasfi G Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, and Gernot A Fink. Khatt: An open arabic offline handwritten text database. *Pattern Recognition*, 47(3):1096–1112, 2014.
- [MFE⁺14] Erik Marchi, Giacomo Ferroni, Florian Eyben, Leonardo Gabrielli, Stefano Squartini, and Bjorn Schuller. Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 2164–2168. IEEE, 2014.
- [MK16] Rania Maalej and Monji Kherallah. Improving mdlstm for offline arabic handwriting recognition using dropout at different positions. In *Interna-*
-

- tional Conference on Artificial Neural Networks*, pages 431–438. Springer, 2016.
- [MOLS⁺13] Olivier Morillot, Cristina Oprean, Laurence Likforman-Sulem, Chafic Mokbel, Edgar Chammas, and Emmanuèle Grosicki. The uob-telecom paristech arabic handwriting recognition and translation systems for the openhart 2013 competition. In *12th International Conference on Document Analysis and Recognition (ICDAR), 2013*, page NIST, 2013.
- [Mor60] Georg Morgenstierne. Khushhal khande national poet of the afghans. *Journal of the Royal Central Asian Society*, 47(1):49–57, 1960.
- [MTK16] Rania Maalej, Najiba Tagougui, and Monji Kherallah. Recognition of handwritten arabic words with dropout applied in mdlstm. In *International Conference Image Analysis and Recognition*, pages 746–752. Springer, 2016.
- [NAAR16] Saeeda Naz, Saad Bin Ahmed, Riaz Ahmad, and Muhammad Imran Razzak. Zoning features and 2dlstm for urdu text-line recognition. *Procedia Computer Science*, 96:16–22, 2016.
- [Nag92] George Nagy. At the frontiers of ocr. *Proceedings of the IEEE*, 80(7):1093–1100, 1992.
- [NBK01] Atul Negi, Chakravarthy Bhagvati, and B Krishna. An ocr system for telugu. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 1110–1114. IEEE, 2001.
- [NG09] Anguelos Nicolaou and Basilis Gatos. Handwritten text line segmentation by shredding text into its lines. In *2009 10th International Conference on Document Analysis and Recognition*, pages 626–630. IEEE, 2009.
- [NHR⁺14] Saeeda Naz, Khizar Hayat, Muhammad Imran Razzak, Muhammad Waqas Anwar, Sajjad A Madani, and Samee U Khan. The optical character recognition of urdu-like cursive scripts. *Pattern Recognition*, 47(3):1229–1248, 2014.
- [Now06] Richard S Nowakowski. Stable neuron numbers from cradle to grave. *Proceedings of the National Academy of Sciences*, 103(33):12219–12220, 2006.
- [NSBP09] Prem Natarajan, Krishna Subramanian, Anurag Bhardwaj, and Rohit Prasad. Stochastic segment modeling for offline handwriting recognition. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 971–975. IEEE, 2009.
-

- [NUA⁺15] Saeeda Naz, Arif I Umar, Riaz Ahmad, Saad B Ahmed, Syed H Shirazi, and Muhammad I Razzak. Urdu nastaliq text recognition system based on multi-dimensional recurrent neural network and statistical features. *Neural Computing and Applications*, pages 1–13, 2015.
- [NUA⁺16] Saeeda Naz, Arif Iqbal Umar, Riaz Ahmed, Muhammad Imran Razzak, Sheikh Faisal Rashid, and Faisal Shafait. Urdu nastaliq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks. *SpringerPlus*, 5(1):2010, 2016.
- [NUA⁺17] Saeeda Naz, Arif I Umar, Riaz Ahmad, Imran Siddiqi, Saad B Ahmed, Muhammad I Razzak, and Faisal Shafait. Urdu nastaliq recognition using convolutional–recursive deep learning. *Neurocomputing*, 2017.
- [Ots75] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [PBKL14] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE, 2014.
- [PC96] U Pal and BB Chaudhuri. An improved document skew angle estimation technique. *Pattern Recognition Letters*, 17(8):899–904, 1996.
- [Pea95] Barak A Pearlmutter. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural networks*, 6(5):1212–1228, 1995.
- [Pen55] Herbert Penzl. *A grammar of Pashto: A descriptive study of the dialect of Kandahar, Afghanistan*, volume 2. American Council of Learned Societies, 1955.
- [PGLS13] Alexandros Papandreou, Basilis Gatos, Georgios Louloudis, and Nikolaos Stamatopoulos. Icdar 2013 document image skew estimation contest (disec 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1444–1448. IEEE, 2013.
- [PM13] Mohammad Tanvir Parvez and Sabri A Mahmoud. Arabic handwriting recognition using structural and syntactic pattern attributes. *Pattern Recognition*, 46(1):141–154, 2013.
- [Pos86] Wolfgang Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proceedings of the 8th International Conference on Pattern Recognition*, pages 687–689, 1986.
-

- [PPALCB16] Joan Pastor-Pellicer, Muhammad Zeshan Afzal, Marcus Liwicki, and María José Castro-Bleda. Complete system for text line extraction using convolutional neural networks and watershed transform. In *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*, pages 30–35. IEEE, 2016.
- [PPEBZM⁺15] Joan Pastor-Pellicer, S España-Boquera, Francisco Zamora-Martínez, M Zeshan Afzal, and Maria Jose Castro-Bleda. Insights on the use of convolutional neural networks for document image binarization. In *International Work-Conference on Artificial Neural Networks*, pages 115–126. Springer, 2015.
- [PRM15] R Prajna, VR Ramya, and HR Mamatha. A study of different text line extraction techniques for multi-font and multi-size printed kannada documents. *International Journal of Computer Applications*, 119(11), 2015.
- [PS03] U Pal and Anirban Sarkar. Recognition of printed urdu script. In *ICDAR*, volume 2003, pages 1183–1187, 2003.
- [PSK⁺08] Rohit Prasad, Shirin Saleem, Matin Kamali, Ralf Meermeier, and Prem Natarajan. Improvements in hidden markov model based arabic ocr. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
- [PT97] GS Peake and TN Tan. A general algorithm for document skew angle estimation. In *Image Processing, 1997. Proceedings., International Conference on*, volume 2, pages 230–233. IEEE, 1997.
- [Rah04] Tariq Rahman. Language policy and localization in pakistan: proposal for a paradigmatic shift. In *SCALLA Conference on Computational Linguistics*, volume 99, page 100, 2004.
- [RH15] Michael Ryan and Novita Hanafiah. An examination of character recognition on id card using template matching approach. *Procedia Computer Science*, 59:520–529, 2015.
- [RKC14] Jewoong Ryu, Hyung Il Koo, and Nam Ik Cho. Language-independent text-line extraction algorithm for handwritten documents. *IEEE Signal processing letters*, 21(9):1115–1119, 2014.
- [RMM⁺94] John C Russ, James R Matey, A John Mallinckrodt, Susan McKay, et al. The image processing handbook. *Computers in Physics*, 8(2):177–178, 1994.
-

- [RRC15] Anupama Ray, Sai Rajeswar, and Santanu Chaudhury. Text recognition using deep blstm networks. In *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*, pages 1–6. IEEE, 2015.
- [RSB11] Sheikh Faisal Rashid, Faisal Shafait, and Thomas M Breuel. An evaluation of hmm-based techniques for the recognition of screen rendered text. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1260–1264. IEEE, 2011.
- [RSRvdN13] Sheikh Faisal Rashid, Marc-Peter Schambach, Jörg Rottland, and Stephan von der Nüll. Low resolution arabic recognition with multidimensional recurrent neural networks. In *Proceedings of the 4th International Workshop on Multilingual OCR*, page 6. ACM, 2013.
- [RVF12] Leonard Rothacker, Szilard Vajda, and Gernot A Fink. Bag-of-features representations for offline handwriting recognition applied to arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 149–154. IEEE, 2012.
- [Sam03] Robert Sampson. The poetry of abdu’l rahman baba: The gentle side of pushtun consciousness. *Central Asia:[journal of Area Study Centre].*, 52:213, 2003.
- [Sat09] Sohail Abdul Sattar. *A Technique for the Design and Implementation of an OCR for Printed Nastaliq Text*. PhD thesis, NED University of Engineering & Technology, Karachi, 2009.
- [SBK08] Chandan Singh, Nitin Bhatia, and Amandeep Kaur. Hough transform based fast skew detection and accurate skew correction methods. *Pattern Recognition*, 41(12):3528–3546, 2008.
- [Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
- [SCS⁺09] Shirin Saleem, Huaigu Cao, Krishna Subramanian, Matin Kamali, Rohit Prasad, and Prem Natarajan. Improvements in bbn’s hmm-based offline arabic handwriting recognition system. In *Document Analysis and Recognition, 2009. ICDAR’09. 10th International Conference on*, pages 773–777. IEEE, 2009.
- [Seb02] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
-

-
- [SG89] Sargur N Srihari and Venugopal Govindaraju. Analysis of textual images using the hough transform. *Machine vision and Applications*, 2(3):141–153, 1989.
- [Sha02] Zahra A Shah. Ligature based optical character recognition of urdu-nastaleeq font. In *Multi Topic Conference, 2002. Abstracts. INMIC 2002. International*, pages 25–25. IEEE, 2002.
- [SHP08] Sohail A. Sattar, Shamsul Haque, and Mahmood K. Pathan. Nastaliq optical character recognition. In *ACM Southeast Regional Conference*, pages 329–331, 2008.
- [SIK⁺09] Fouad Slimane, Rolf Ingold, Slim Kanoun, Adel M Alimi, and Jean Hennebert. A new arabic printed text image database and evaluation protocols. In *2009 10th International Conference on Document Analysis and Recognition*, pages 946–950. IEEE, 2009.
- [SKB⁺06] Faisal Shafait, Daniel Keysers, Thomas M Breuel, et al. Layout analysis of urdu document images. In *2006 IEEE International Multitopic Conference*, pages 293–298. IEEE, 2006.
- [SKEA⁺13] Fouad Slimane, Slim Kanoun, Haikal El Abed, Adel M Alimi, Rolf Ingold, and Jean Hennebert. Icdar2013 competition on multi-font and multi-size digitally represented arabic text. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1433–1437. IEEE, 2013.
- [SKH⁺13] Fouad Slimane, Slim Kanoun, Jean Hennebert, Adel M Alimi, and Rolf Ingold. A study on font-family and font-size recognition applied to arabic word images at ultra-low resolution. *Pattern Recognition Letters*, 34(2):209–218, 2013.
- [SM99] Larry Spitz and Arman Maghbouleh. Text categorization using character shape codes. In *Electronic Imaging*, pages 174–181. International Society for Optics and Photonics, 1999.
- [Smi07] Ray Smith. An overview of the tesseract ocr engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007.
- [Spi95] A Lawrence Spitz. An ocr based on character shape codes and lexical information. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 2, pages 723–728. IEEE, 1995.
-

- [SS97] Alan F Smeaton and A Lawrence Spitz. Using character shape coding for information retrieval. In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 2, pages 974–978. IEEE, 1997.
- [SS13] Nazly Sabbour and Faisal Shafait. A segmentation-free approach to arabic and urdu ocr. In *DRR*, 2013.
- [SSA15] Mahnaz Shafii and Maher Sid-Ahmed. Skew detection and correction based on an axes-parallel bounding box. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(1):59–71, 2015.
- [SSB14] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pages 338–342, 2014.
- [SSG09] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *2009 10th International Conference on Document Analysis and Recognition*, pages 176–180. IEEE, 2009.
- [SSR10] Tanzila Saba, Ghazali Sulong, and Amjad Rehman. A survey on methods and strategies on touched characters segmentation. *International Journal of Research and Reviews in Computer Science*, 1(2):103–114, 2010.
- [Ste91] Richard S Stephens. Probabilistic approach to the hough transform. *Image and vision computing*, 9(1):66–71, 1991.
- [SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [Tau35] Gustav Tauschek. Reading machine, December 31 1935. US Patent 2,026,330.
- [TJT96] Øivind Due Trier, Anil K Jain, and Torfinn Taxt. Feature extraction methods for character recognition-a survey. *Pattern recognition*, 29(4):641–662, 1996.
- [TR96] Habibullah Tegey and Barbara Robson. A reference grammar of pashto. *EDRS*, 1996.
- [UHAR⁺13] Adnan Ul-Hasan, Saad Bin Ahmed, Faisal Rashid, Faisal Shafait, and Thomas M Breuel. Offline printed urdu nastaleeq script recognition with bidirectional lstm networks. In *Document Analysis and Recognition (IC-*
-

- DAR*), 2013 12th International Conference on, pages 1061–1065. IEEE, 2013.
- [UHAS⁺15] Adnan Ul-Hasan, Muhammad Zeshan Afzal, Faisal Shafait, Marcus Liwicki, and Thomas M Breuel. A sequence learning approach for multiple script identification. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1046–1050. IEEE, 2015.
- [UHBD16] Adnan Ul-Hasan, Syed Saqib Bukhari, and Andreas Dengel. Ocroract: A sequence learning ocr system trained on isolated characters. In *DAS*, pages 174–179, 2016.
- [UHSL15] Adnan Ul-Hasan, Faisal Shafaity, and Marcus Liwicki. Curriculum learning for printed text line recognition of ligature-based scripts. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1001–1005. IEEE, 2015.
- [VR77] Gordon J. Vanderbrug and Azriel Rosenfeld. Two-stage template matching. *IEEE Transactions on Computers*, 26(4):384–393, 1977.
- [WAA09] Mehreen Wahab, Hassan Amin, and Farooq Ahmed. Shape analysis of pashto script and creation of image database for ocr. In *Emerging Technologies, 2009. ICET 2009. International Conference on*, pages 287–290. IEEE, 2009.
- [WME⁺10] Martin Wöllmer, Angeliki Metallinou, Florian Eyben, Björn W Schuller, and Shrikanth S Narayanan. Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In *Interspeech*, pages 2362–2365, 2010.
- [WR14] Neha Watts and Jyoti Rani. Performance evaluation of improved skew detection and correction using fft and median filtering. *Performance Evaluation*, 100(15), 2014.
- [XPK10] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter*, 12(1):40–48, 2010.
- [YSBS15] Mohammad Reza Yousefi, Mohammad Reza Soheili, Thomas M Breuel, and Didier Stricker. A comparison of 1d and 2d lstm architectures for the recognition of handwritten arabic. In *SPIE/IS&T Electronic Imaging*, pages 94020H–94020H. International Society for Optics and Photonics, 2015.
-

-
- [Zen15] Heiga Zen. Acoustic modeling in statistical parametric speech synthesis—from hmm to lstm-rnn. *Proc. MLSLP*, 2015.
- [ZSV14] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
-