

---

# Robust Urdu Text Recognition from Video Stream using Machine Learning Techniques

---



Ph.D Thesis

*By*

Moeen Tayyab

119-FBAS/PHDCS/F15

*Supervisor*

Dr. Ayyaz Hussain

Professor (QAU)

*Co-Supervisor*

Dr. Zeeshan Hayder

Research Scientist (NESCOM)

**Department of Computer Science  
Faculty of Computing & Information Technology  
International Islamic University, Islamabad  
(2024)**

Accession # TH-26207

PhD  
ack. 42  
TAR

Image processing - computer

Recognition

Document Imaging

optical character recognition

---

*A dissertation submitted to the  
Department of Computer Science,  
International Islamic University, Islamabad  
as a partial fulfillment of the requirements  
for the award of the degree of  
Doctor of Philosophy in Computer Science*

---

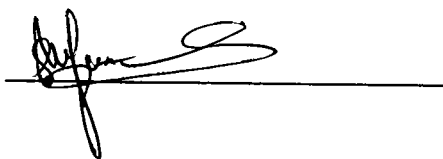
## Plagiarism Undertaking

I take full responsibility for the research work conducted during the Ph.D. The thesis is titled "Robust Urdu Text Recognition from Video Stream Using Machine Learning Techniques". I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not previously presented this thesis (or substantially similar research work) or any part of the thesis to any other degree awarding institution within Pakistan or abroad.

I understand that the management of International Islamic University Islamabad has a zero-tolerance policy toward plagiarism. Therefore, I as an author of the above-mentioned thesis, solemnly declare that no portion of my thesis has been plagiarized and any material used in the thesis from other sources is properly referenced. Moreover, the thesis does not contain any literal citing of more than 70 words (total) even by giving a reference unless I have the written permission of the publisher to do so. Furthermore, the work presented in the thesis is my original work. I have positively cited the related work of the other researchers by clearly differentiating my work from their relevant work.

I further understand that if I am found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my Ph.D. degree. Moreover, the University will also have the right to publish my name on its website which keeps a record of the students who plagiarized in their thesis work.

Moeen Tayyab:



Date: \_\_\_\_\_

**INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD  
FACULTY OF COMPUTING & INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SCIENCE**

Date: 04-03-2024

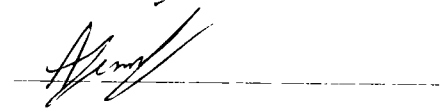
**Final Approval**

It is certified that we have read this thesis, entitled “Robust Urdu Text Recognition from Video Stream using Machine Learning Techniques” submitted by **Moeen Tayyab**, **Registration No. 119-FBAS/PHDCS/F15**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the award of the degree of PhD in Computer Science.

**Committee**

**External Examiner:**

Dr. Arif Jamal,  
Associate Professor,  
Foundation University, Islamabad



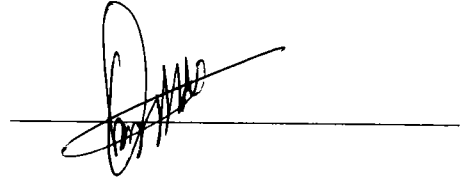
**External Examiner:**

Dr. Nadeem Anjum,  
Associate Professor,  
Capital University of Science & Technology, Islamabad



**Internal Examiner:**

Dr. Qamar Abbas,  
Assistant Professor,  
Department of Computer Science,  
International Islamic University, Islamabad



**Co-Supervisor:**

Dr. Zeeshan Hayder,  
Research Scientist,  
National Engineering & Scientific Commission, Islamabad



**Supervisor:**

Dr. Ayyaz Hussain,  
Professor,  
Department of Computer Science,  
Quaid-e-Azam University



---

## **Declaration**

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

**Moeen Tayyab**

---

## **Dedication**

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears.

I also dedicate this dissertation to my many teachers who have supported me throughout the process.

**Moeen Tayyab**

---

## Acknowledgments

This thesis would not have been possible without the inspiration and support of several wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible. Foremost, I would like to express my sincere gratitude to my supervisor, *Dr. Ayyaz Hussain*, for giving me guidance and counsel and having faith and confidence in me. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. He read and correct my early attempts at writing.

Besides my supervisor, I would like to thank the rest of my thesis committee: Dr. Mohammad Asmat Ullah Khan, Dr. Asim Munir, Dr. Muhammad Nadeem, Dr. Anwar Ghani, and Dr. Qamar Abbas, for their insightful comments and encouragement, but also for the hard question which unscented me to widen my research from various perspectives.

I sincerely acknowledge the contribution of different collaborators: Dr. Zeeshan Hayder, Australian National University, Australia, Dr. Abdul Rauf Baig, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. I am extremely grateful to these stalwart personalities who guided me.

This achievement would not have been possible without the pure love and support of my family and friends. They back me in every hurdle I faced during this travel. I am forever indebted to them for giving me the opportunities and experiences that have made me who I am.



---

## List of Publications From Thesis

1. **Moeen Tayyab**, Ayyaz Hussain, Mohammed Ali Alshara, Shakir Khan, Reemiah Muneer Alotaibi, and Abdul Rauf baig, "Visual News Ticker Surveillance Approach from Arabic Broadcast Stream" *Computers, Materials & Continua*, 2023, 74(3): 6177-6193. (*Impact factor 3.86*)
2. **Moeen Tayyab**, Ayyaz Hussain, Usama Mir, M. Aqeel Iqbal, and Muhammad Haneef, "Recognition of Visual Arabic Scripting News Ticker From Broadcast Stream", *IEEE Access*, 2022, 10: 59189-59204. (*Impact factor 3.37*)
3. **Moeen Tayyab**, Ayyaz Hussain "Convolutional Matching Technique of Urdu Text Recognition". 25th International Multitopic Conference (INMIC), 2023.

---

## Acronyms

<b>ANN</b>	Artificial Neural Network
<b>AL-ENT</b>	Al- Elrabia News Ticker
<b>BLSTM</b>	Bidirectional Long Short-Term Memory
<b>CNN</b>	Convolutional Neural Network
<b>CRF</b>	Conditional Random Field
<b>CRR</b>	Character Recognition Rate
<b>CTC</b>	Connectionist Temporal Classification
<b>CLE</b>	Center of Language Engineering
<b>DNN</b>	Deep Neural Network
<b>DCT</b>	Discrete Cosine Transform
<b>EAST</b>	Arabic images from the English-Arabic Scene Text
<b>FCN</b>	Fully Convolutional Network
<b>GBLSTM</b>	Gated Bi-directional Long Short-Term Memory
<b>GPU</b>	Graphical Processing Unit
<b>HMM</b>	Hidden Markov Model
<b>LSTM</b>	Long-Short Term Memory
<b>MDLSTM</b>	Multi-Dimensional Long-Short Term Memory
<b>MNIST</b>	Modified National Institute of Standards and Technology database
<b>OCR</b>	Optical Character Recognition
<b>PCA</b>	Principle Component Analysis
<b>ReLU</b>	Rectified Linear Unit
<b>RNN</b>	Residual Neural Network
<b>UPTI</b>	Urdu Printed Text Image
<b>UNT</b>	Urdu News Ticker
<b>WER</b>	Word Error Rate
<b>WRR</b>	Word Recognition Rate

---

## Abstract

Text that appears in videos serves as an intriguing index for high-level applications such as news ticker recognition due to different features such as information analysis, opinion mining, and language translation for media regulatory authorities, etc. Without automated systems, manual annotating is difficult. The recognition of text using Optical Character Recognition (OCR) is an essential part of a textual content-based retrieval system. While sophisticated recognition algorithms exist for text written in non-cursive scripts, research on cursive scripts (like Urdu) is inadequate and fraught with difficulties. These include complex and overlapping ligatures, context-dependent character shapes, and the presence of diacritics (dots, etc.). Urdu script can be written with several fonts including Nastaleeq, Rika, Naskh, Web Naskh, Pakistani Naskh, Tahreer Naskh, and the Pakistani Web Naskh. In Pakistan, Nastaleeq is the most commonly used Urdu font as it appears in Newspapers (Jang, Express, etc.), on TV channels (Geo, ARY, SAMMA, etc.), and in magazines. Despite its excessive use, automatic recognition of Urdu manuscripts has received little attention from the research community.

This research mainly focused on Urdu text recognition from the video stream(s) to improve the recognition rate. The study focuses on Pakistani news networks in order to recognize Urdu text that appears in new tickers from video frames. The Nastaleeq font is selected as the Urdu font for training and testing. The research proposes novel robust methodologies to recognize Urdu text from frames, the most typical case in the majority of the News channels, by leveraging recent advances in deep neural networks. An analytical approach is used for recognition. Characters are explicitly segmented using deep learning-based semantic segmentation techniques. Ground truth transcription and images of text lines taken from video frames are fed to the network. The last layer transforms raw predictions into meaningful Urdu character-segmented images. The syntactical model is used to conclude the recognized string of letters into a sentence of words.

In addition, new challenging datasets are created from the video stream(s) to evaluate the effectiveness of the suggested strategy. UPTI dataset is considered for evaluation purposes for fair performance comparison. The proposed method outperforms the current state-of-the-art method. Investigations show that the suggested technique enhances the performance of character classes with low symbol frequencies.

# Contents

<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Scope of the Research . . . . .	3
1.3 Challanges . . . . .	4
1.3.1 Image Distortion in Video Decoding . . . . .	4
1.3.2 Segmentation and Recognition . . . . .	4
1.4 Research Questions . . . . .	5
1.5 Problem Statement . . . . .	5
1.6 Research Contributions . . . . .	5
1.7 Thesis Organization . . . . .	6
<b>2 Literature Review</b>	<b>7</b>
2.1 Holistic Approach . . . . .	8
2.2 Analytical Approach . . . . .	9
2.2.1 Implicit Segmentation . . . . .	9
2.2.2 Explicit Segmentation . . . . .	10
2.3 Research Gaps . . . . .	11
2.4 Summary . . . . .	13
<b>3 Explicit Cursive Textual Segmentation</b>	<b>14</b>
3.1 Dataset Acquisition . . . . .	17
3.1.1 Dataset Preparation . . . . .	17

3.1.2	Character Families Hypothesis and Dataset Re-labeling . . . . .	21
3.2	Semantic Segmentation Process . . . . .	22
3.3	UrduSeg Segmentation Model . . . . .	26
3.4	Summary . . . . .	29
<b>4</b>	<b>Syntax Formation Model</b>	<b>30</b>
4.1	Karhunen-Loève Transformation based Classification Model . . . . .	30
4.2	LeNet-based Classification Model . . . . .	32
4.3	Words Code Generator . . . . .	34
4.4	Summary . . . . .	35
<b>5</b>	<b>Experimental Results</b>	<b>37</b>
5.1	Experimental Partitioning of Datasets . . . . .	37
5.1.1	UNT Dataset . . . . .	37
5.1.2	Al-ENT Dataset . . . . .	38
5.1.3	Al-Arabiya Dataset . . . . .	38
5.1.4	UPTI Dataset . . . . .	38
5.2	Performance Measurement . . . . .	38
5.2.1	Accuracy . . . . .	38
5.2.2	F-Score . . . . .	38
5.3	Segmentation Model Experiments . . . . .	39
5.3.1	Experimental Setups . . . . .	39
5.3.2	Segmentation Model Simulations Analysis . . . . .	39
5.3.3	Result Analysis . . . . .	46
5.4	KL-Transformation Classification Model Experiment . . . . .	50
5.5	LeNet-based Classification Model Experiment . . . . .	51
5.6	Experimental Study and its Analysis . . . . .	52
5.7	Summary . . . . .	54
<b>6</b>	<b>Conclusions and Future Work</b>	<b>57</b>
6.1	Conclusions . . . . .	57
6.2	Future Work . . . . .	58
	<b>Bibliography</b>	<b>60</b>

## List of Figures

1.1	<b>Top row:</b> Visualization of Urdu text and its corresponding ground truth. <b>Bottom row:</b> Concatenated segmented characters results using the proposed technique and textual results by word identification model. Best viewed in color. . . . .	3
3.1	Proposed model for news ticker recognition: SegNet model presenting downsample and upsampling architecture for characters semantic segmentation, syntax formation model for Urdu (Nastaleeq font) textual recognition. . . . .	15
3.2	Gray-scale image of Urdu news ticker in Nastaleeq font from the video stream. . .	18
3.3	Vertical gradient $G_x$ representation of ticker image . . . . .	18
3.4	Extracted text line image of Urdu news ticker in Nastaleeq font (1). . . . .	18
3.5	Extracted text line image of Urdu news ticker in Nastaleeq font (2). . . . .	18
3.6	Gray-scale image of an Arabic news ticker in Kufi font from the video stream. . . .	19
3.7	Extracted text line image of an Arabic news ticker in Kufi font. . . . .	19
3.8	Visualization of character-wise labeled Urdu news ticker lines in Nastaleeq font. . .	19
3.9	Visualization of character-wise labeled Arabic news ticker lines in Kufi font. . . .	19
4.1	Segmented and concatenated Nastaleeq font characters in [64x64] pixels dimensional buffer. . . . .	32
4.2	Extracted text line image of an Arabic news ticker in Kufi font. . . . .	33
5.1	Low-frequency characters segmentation issues. . . . .	41
5.2	Font size segmentation issues. . . . .	42
5.3	Comparison of graphical representation of training curves with Nastaleeq font. . .	42
5.4	Graphical comparison representation of SegNet training loss, validation loss, and validation accuracy curves with UNT. . . . .	45
5.5	Graphical representation of training curves with Kufi font Al-Arabiya dataset. . . .	46

5.6	Graphical representation of training curves with Kufi font Al-ENT dataset. . . . .	46
5.7	Graphical representation of KL-Transformation test on Urdu characters. . . . .	50
5.8	Graphical representation of LeNet using UNT dataset, training and validation accuracy, and training and validation loss curves. . . . .	53
5.9	Graphical representation of LeNet using Al-Arabiya dataset, training and validation accuracy, and training and validation loss curves. . . . .	53
5.10	Graphical representation of LeNet using Al-ENT dataset, training and validation accuracy, and training and validation loss curves. . . . .	54

# List of Tables

2.1	Related research work and research gaps . . . . .	12
3.1	Label and color assignment to isolated Nastaleeq font character shapes. Best viewed in color. . . . .	20
3.2	Label and color assignment to prime components of characters. Best viewed in color. . . . .	21
3.3	Label and color assignment to diacritic marks of characters incorporating homogeneous structural shapes. Best viewed in color. . . . .	21
3.4	Details of Urdu characters in UNT dataset . . . . .	23
3.5	Details of Arabic characters in Al-ANT dataset . . . . .	24
3.6	Details of Arabic characters in Al-ENT dataset . . . . .	25
4.1	Notations used in algorithm . . . . .	34
5.1	SegNet encoding layers with parameters and matrix shapes. . . . .	40
5.2	SegNet decoding layers with parameters and matrix shapes. . . . .	40
5.3	SegNet training parameters. . . . .	41
5.4	Urdu SegNet encoding layers with the number of filters and parameters. . . . .	43
5.5	Urdu SegNet decoding layers with the number of filters and parameters. . . . .	43
5.6	SegNet training comparison with proposed technique using UNT dataset. . . . .	44
5.7	SegNet model comparison: final validation, training loss, and validation accuracy of the proposed technique with UNT dataset. . . . .	47
5.8	Detailed Arabic characters segmentation performance of the proposed method with Al-Arabiya dataset . . . . .	47
5.9	Detailed Arabic characters segmentation performance of the proposed method with Al-ENT dataset . . . . .	48



5.10 Detailed Urdu characters segmentation performance of the proposed method with UNT dataset . . . . .	49
5.11 LeNet training using UNT dataset. . . . .	51
5.12 LeNet training using Al-Arabiya dataset. . . . .	52
5.13 LeNet training using Al-ENT dataset. . . . .	52
5.14 Customized Dataset Experimental Results Summary of Segmentation and Classi- fication Models. . . . .	54
5.15 Performance comparison with UPTI Nastaleeq font dataset. . . . .	55
5.16 Performance of the suggested approach using the UNT dataset for Urdu characters detection . . . . .	56

# Chapter 1

## Introduction

The quantity of digital multimedia data, including films, and photographs, has significantly increased during the past decade. Such vast video archives have created new challenges for the development of intelligent retrieval systems that allow viewers to quickly and easily access requested information.

Traditional video retrieval systems concentrate on matching the searched words with user-assigned annotations, ignoring the abundant information in videos. To enable smart information systems, content-based search algorithms can be utilized for visual information textual content. In addition to indexing and retrieval, a variety of innovative applications can be developed for the text that has been recognized using an OCR.

News Ticker Recognition is one of the applications. News ticker text analysis, generation of news alerts based on user-specified keywords, analysis of the same news across various news channels, opinion mining, and language translation are typical features of this application. Such applications are highly applicable to media regulatory authorities and law and enforcement agencies. Without automated systems, manual anatomizing is difficult.

The objective of the present research is to recognize Urdu cursive text that appears in News tickers from video streams using machine learning techniques. It mainly consists of ticker text segmentation and recognition to generate textual data for various online services. Deep convolutional neural network models are explored for text segmentation. The syntax formation model is employed as a post-segmentation process.

## 1.1 Motivation

Automated visual text reading is a vital research area on the basis of its applications. It has great significance due to different applications like scanning documents, signboard reading, mailbox scanning, license plate recognition, and other security and commercial applications. Big data analysis of news tickers is one of these applications from the daily broadcast streams. The display of headlines or other information in a news ticker has become a standard feature on many different news networks. News Ticker information can be applied for records management of political statements to provide services for political parties. Such applications are helpful for media regulatory authorities to maintain checks and balances on media channels etc. Systems recognizing text in non-cursive scripts are currently being utilized by regulatory agencies and media organizations in several developed nations. Recent developments are mostly focused on ticker recognition in the English language. However, recognition of tickers in other languages still requires considerable deliberation. In particular, recognition of news tickers in Arabic or Urdu languages is a challenging task due to the limited availability of labeled datasets and techniques specifically for cursive scripts. As of late, a few strategies [1] are presented to recognize the cursive text in an especially ad-hoc manner. These methods are utmost focusing on scanned printed text in a controlled way. Nonetheless, the data from real-world broadcast streams are noisy and comparatively troublesome to recognize by transfer learning approaches. Video distortions could be caused as a consequence of various factors like transmission, resolution, signal-to-noise ratio, and other compression artifacts. Contemporary datasets come up short on these antiquities and accordingly a deterrent to the improvement of a learning model. In fact, there is a need to research text recognition methods that could potentially be used to develop innovative news ticker retrieval systems at regional media organizations and regulatory authorities. Text segmentation and recognition approaches are investigated. The prime purpose of this thesis is to provide a thorough examination of Urdu cursive text recognition on news tickers from video streams. To be font-specific and standardization of news tickers, the Saudi TV channel Al-Arabiya (Kufi font) and a few Pakistani (Nastaleeq font) broadcast streams are selected.

Fully Connected Network (FCN) and SegNet segmentation architectures are explored followed by a proposed syntactical model. Deep learning models, especially Convolutional Neural Networks (CNNs), are specifically used for two-dimensional image data analytic problems. However, recently single dimensional CNN has also demonstrated tremendous performance, mainly for time series data analysis [2]. In the area of video/image processing, CNN has set up opportunities for numerous tasks including person re-identification, tracking, video analysis, text recognition,

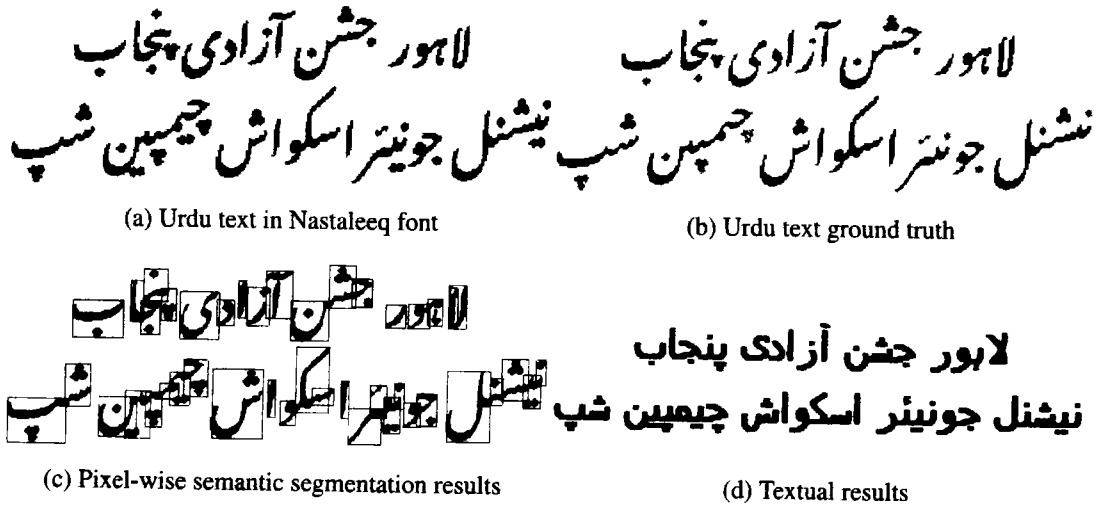


Figure 1.1: **Top row:** Visualization of Urdu text and its corresponding ground truth. **Bottom row:** Concatenated segmented characters results using the proposed technique and textual results by word identification model. Best viewed in color.

object detection etc. [3]. The latest methods extricate deep representation from images through different CNN models that have incredible success in computer vision domains among others. In the current study, CNN is employed for the extraction of pixel-wise features. These features are upsampled to form characters or character components. The syntactical model is used to conclude the recognized string into a sentence of words. The effectiveness of the models is analyzed by conducting experiments using novel Arabic and Urdu news ticker datasets with character-level as well as component-level labeling. Fig. 1.1 illustrates a visualization of cursive Urdu text and its corresponding ground truth.

## 1.2 Scope of the Research

The main focus is on visual Urdu text recognition. The research includes the investigation of machine learning-based techniques. The study primarily focuses on the Nastaleeq font style for Urdu writing printed text. To this extent, news ticker text data is extracted from video streams. Maximum and minimum limits of the font size and limitations of video distortions are the parameters.

## 1.3 Challenges

There are certain challenges in cursive text recognition which are as follows.

### 1.3.1 Image Distortion in Video Decoding

Compression artifacts are the primary problem for any video processing application. Distortion in video images because of compression causes loss of some media's data. The compressed version could not replicate enough data to reproduce the original. This results in a decline in the quality of the video. Compression algorithms are not intelligent enough to discriminate between distortions and those objectionable to the viewer. Such artifact causes complications for OCR towards high accuracy.

Also, the resolution is proportional to computation and accuracy. Low resolution corresponds to low computation but is troublesome for OCR because of image distortion.

### 1.3.2 Segmentation and Recognition

There are several performance issues associated with Urdu character recognition due to complications in segmentation and recognition. S Naz et. al [4] also highlight some of the primary problems in their survey. As stated above, a word may have one or more ligatures, every letter can have two three, or four different shapes depending upon its positioning in ligature. It can be at the start, in the middle, in the end, or standalone (isolated). Some letters have four shapes like "Hay", and some have two shapes like "Alif". "Hamza" is always isolated (has only one shape). Shape adaptation of letters depends on the context. Such language rules are harder for Urdu text recognition from a character segmentation point of view. The Urdu language is one of the bidirectional languages. As stated, it is read and written from right to left direction. However, numerics are read and written opposite in directions. This distracts the reading sequence. Heuristic methods can work on clean data but are infeasible for applications like Urdu News Ticker Recognition. Selected Nastaleeq font style is written all of the ligatures or words are tilted, bidirectionality, nonmonotonic writing, stretching, complex dots placement, positioning of words, a word may have one or more ligatures. All these cause complications in isolating multiple shaped characters with font size variations, which is a major challenge for Urdu cursive text recognition from the video broadcast streams using machine learning techniques.

## 1.4 Research Questions

The thesis will focus on the following research questions.

1. How to accurately segment the Urdu text character from news tickers in the presence of font size and shape variation and other image distortions?
2. How can we modify the learning-based Urdu text segmentation model to achieve high accuracy with optimum computation?
3. How to overcome the challenges of imbalanced data and Inter-class variability in shapes?

## 1.5 Problem Statement

Urdu text recognition is a challenging task. It is the language that is written in a cursive way from right to left. The letters are generally joined with each other within a word. However, a word may not have its entire letters joined to form ligatures or sub-words of a word. This involves significant problems at each stage.

Heuristic techniques for character segmentation with contemporary methods may work on clean data but are infeasible for applications like Urdu News Ticker Recognition. The complications in character recognition for Urdu cursive text from video broadcast streams include distortions, bidirectional, nonmonotonic writing, stretching, complex dots placement, positioning of words, words may have one or more ligatures, words are tilted, and font size invariance. All these cause big challenges to isolate multiple-shaped and imbalanced distributed data of characters for recognition using machine learning techniques.

## 1.6 Research Contributions

This dissertation covers the challenges of cursive text recognition using machine learning techniques. We proposed two-step frameworks for Urdu and Arabic character segmentation and recognition. The design, implementation, and technical details of innovative Urdu text recognition for news ticker information systems are presented in this dissertation. This thesis's research contributions are described below.

- Introduction of novel Arabic and Urdu news ticker datasets, consisting of character-level as well as character component-level labeling with Kufi and Nastaleeq fonts respectively.

- Addresses the challenge of a huge number of ligature classes and proposes an explicit approach to character-wise segmentation.
- Extends FCN and SegNet architectures for effective character-wise cursive text recognition that provides significant improvements over traditional feature learning techniques.
- Proposes model by the grouping of similarly shaped classes for more effective segmentation.
- Experimental evaluation of the proposed technique is performed by applying it to the benchmark UPTI dataset and custom-built dataset with cursive Arabic and Urdu script.
- Momentous improvement in the overall performance of the proposed technique has been noted as compared to state-of-the-art existing methodologies.

## 1.7 Thesis Organization

The thesis is organized as follows:

**Chapter 2** provides the literature and provides background knowledge of techniques relevant to the thesis. The associated methods and literature are classified into different categories. This section helps us to identify research gaps, challenges, and directions toward the cursive text recognition process.

**Chapter 3** introduces the in-depth analysis of the proposed Urdu text segmentation. This presents an explicit cursive textual segmentation method and the dataset acquisition process.

**Chapter 4** proposes a novel syntax generation model for word identification by introducing spaces between words, which leads segmented characters to a linguistically recognized format.

**Chapter 5** discusses the results of the proposed methodology. Experiments with a novel cursive scripting language segmentation model and the detailed result analysis of the proposed technique are presented. Classification models and the Words Code Generator technique are analyzed with experimental results.

**Chapter 6** describes the main conclusions of our investigation and summarises the key findings. The chapter reviews major contributions and recommends possible future research opportunities in this area.

## Chapter 2

### Literature Review

Video content includes sometimes low-quality channels as well as distorted and blurred news tickers, making the data an appropriate test case for an automatic Urdu News Recognition system. For the new ticker text recognition use case, accurate readability of the recognized text is important. From a recognition point of view, each character's shape varies according to its place in the ligature. The presence of a large number of ligatures in the training set is critical for getting higher accuracy rates on test sets. The cursive style of the script makes accurate character-level text segmentation almost impossible [5]. The blurring impact of video channel distortion hinders recognition.

Very limited research has been done on news ticker-based Urdu/Arabic script recognition. The techniques can be primarily categorized into two categories i.e. 1) analytical and 2) holistic approaches [4]. An analytical approach relies on the segmentation of text into characters, explicitly or implicitly. Holistic approaches employ the segmentation of ligatures, as units of recognition. Structural and statistical features are the main studies of attention in numerous kinds of research. Structural features rely on the structure of the patterns, under study characters or ligatures i.e. information of loops, curves, joining or ending points, distribution of horizontal and vertical lines, etc. Statistical features consider statistics calculated from the arrangement of pixels in the pattern of images. These features normally comprise moments, profiling, projections, and pixel densities, etc.

Most of the focus has been on different strategies for cursive text recognition with scanned images. Hussain et al. [6] have done adequate work on Urdu OCR with the Hidden Markov Model (HMM) and manually designed rule-based post-processing methods. They mostly used the Nastaleeq font.



They investigate both holistic [7] and analytical [8] techniques using the Center of Language Engineering (CLE) dataset. Hussain et al. [8] propose a hybrid approach of the Bag-of-Feature framework and HMMs for sequence recognition. In the evolutionary contributions of Arabic and Urdu OCR tasks, advanced machine learning techniques have been introduced. Variants of Recurrent Neural Networks (RNNs) are extensively used in text recognition research for multiple languages. They compare performances of RNN-based text recognizers. Rahal et al. [9] used a very similar concept by using improved Long Short-Term Memory (LSTM) for Arabic text recognition in videos. References [10] and [11] proposed Multidimensional Long Short Term Memory (MDLSTM) with Connectionist Temporal Classification (CTC) output layer. Ji Gan et al. [12] introduce an effective architecture for Chinese text recognition, which concurrently takes advantage of CNN and LSTM networks. Mirza et al. [13] highlighted text image problems from videos with complex backgrounds and proposed a deep learning model with the composition of CNN and LSTM. Experiments are carried out on 12,000 text lines extracted from 4,000 video frames from Pakistani news channels. Mirza et al. [14] proposed a similar UrduNet model with the composition of CNN and LSTM. A comprehensive series of experiments are carried out on a custom-built dataset from more than 13,000 video frames.

In this study, CNN networks are explored as well but in a different manner. The proposed scheme is applied to news tickers of the popular Saudi TV channel Al-Arabiya, Al Ekhbaria, and a few Pakistani broadcast streams. From the method evaluation perspective, an extensively used UPTI dataset is selected. It is a publicly available dataset developed by Sabbour et al. [15] for the research community. This dataset includes various degraded variants of text lines with ligature-based annotations. It includes ten thousand text images having printed Urdu Nastaleeq font scripts. Further details of holistic and analytical approaches are categorized as follows.

## 2.1 Holistic Approach

The holistic approach deals with whole ligatures. The models are trained to recognize ligatures directly. Akram et al. [16] investigated and modified the Tesseract OCR to support the Urdu Nastaleeq style with a holistic approach. For this, they used a custom-built dataset extracted from 17,453 unique words. They target a large set of ligature problems and reduce them to 1475 main body types without diacritics. Results on font sizes 14 and 16 are declared as 97.87% and 97.71% respectively. Images used were cleaned and segmented. They also developed Tesseract-based Urdu OCR with four different recognizers. By this, they support multi-font sizes from 14 to 44

and accuracy up to 86.15% per ligature on the dataset of 224 document images. Farhan et al. [17] indicate the challenge of computation complexity and introduce a computationally efficient holistic approach for Arabic text recognition. Their technique is based on similarly shaped word clustering. They suggested Discrete Cosine Transform (DCT) for Arabic OCR, with a word recognition rate of 84.8% on printed text. Based on the performance of deep learning in computer vision tasks, Rehman et al. [18] introduce a holistic approach with CNN based classification model. They claim accuracy up to 84.2% on custom-built Urdu dataset. Ahmad et al. [19] present a dissimilar technique from the previous one by using robust gated contextual information among ligatures. They developed a model incorporating raw pixel value as a feature. They used a Gated Bi-directional Long Short Term Memory (GB LSTM) learning model on the UPTI dataset having 43 classes and aligned input, with a declared 96.71% recognition rate. Architecture is trained on undegraded and tested on unseen image data. Javed et al. [20] also proposed the CNN model, evaluated 18,000 Urdu ligatures and 98 different classes, and realized a recognition rate of 95%. They rely on fine-tuning and the use of pre-trained CNN to avoid local optima problems. Moreover, images for the network are dynamically resized rather than fixed-resized by maintaining an aspect ratio without significant distortion and placed at the top left corner of fixed-size images.

In this approach, each ligature is examined as a separate class for models. Lehal et al. [21] specify more than 25,000 ligatures in Urdu. This indicates a large number of ligature classes with a substantial quantity of samples per class. These enormous numbers of classes cause strain in training models. Such an approach is inappropriate for real-time applications such as news ticker recognition.

## 2.2 Analytical Approach

The analytical method relies on the segmentation of text into characters. Implicit and explicit segmentation of analytical approaches is discussed below.

### 2.2.1 Implicit Segmentation

An implicit approach is used by Ahmed et al. [22] using Bidirectional Long Short Term Memory (BLSTM) and CTC output layer. They highlighted complexities related to the bidirectional writing of Urdu script, words are written from right to left whereas numbers are from left to right. Associating dots with base characters is a more challenging task. They use UPTI dataset scripts, claiming 88.94% accuracy. Hasan et al. [23] also used BLSTM architecture with a CTC output layer to

recognize the UPTI dataset. They analyze that normalizing input images to some height is necessary for uniform information. For this, they use textual content baseline information and claim 94.85% accuracy. Naz et al. [24] also used MDLSTM. They extract features from more relevant data from the character sequences for the recognition engine. They achieved higher accuracy rates with statistical features and MDLSTM. They presented feature extraction using right-to-left sliding windows. Their methodology on the UPTI database reports a recognition rate of 94.97%. Naz et al. [25] observed that the MDLSTM technique using raw pixels has not been explored before. They investigate MDLSTM using raw pixels for Urdu Nastaleeq font recognition. Experiments show that MDLSTM attained a recognition accuracy of 98% using the UPTI dataset, significantly outperforming the state-of-the-art techniques. Naz et al. [26] proposed an approach to zoning features because of its efficiency, low complexity, and high speed with significant information. They used zoning features with a combination of Two Dimensional Long Short Term Memory networks (2DLSTM) as learning classifiers. Their methodology on the UPTI database reports a recognition rate of 93.39%. Akram et al. [27] presented an implicit Urdu character recognition technique with Nastaleeq font, based on the recognition of characters and joiners. They identify that the connected stroke of a ligature image has sequential pairs of characters and their joiners. The joiner maintains the connecting stroke shape of the character with the next. A detailed analysis is carried out to extract the artistic features of characters and their joiners. The system is tested on 1600 text lines of the UPTI dataset with the HMMs classifier, resulting in a 98.37% recognition rate.

The analytical approach has been used efficiently in numerous studies. References [24–26] uses a sliding window feature extraction technique, followed by the recognizer. This is also referred to as recognition-based segmentation as both processes are carried out in parallel. The approach has a deciding factor of the aggregated number of segments. Less segments lead to less computation but underperform on widely written words. Increasing the segments leads to more computation with more junk segments, which needs to be handled by the model [1]. That may cause under and over-segmentation.

### 2.2.2 Explicit Segmentation

An explicit scheme of the hybrid approach is introduced by Naz et al. [28] and Ahmed et al. [29]. They integrate convolutional RNN for effective features learning of Urdu Nastaleeq cursive scripts with a large number of classes. The CNN extracts low-level translational invariant features followed by MDLSTM for contextual feature extraction. Naz et al. [28] address the challenge of a large number of ligature classes by proposing novel learning mechanisms to learn from a small set

of classes. Evaluation on the UPTI dataset attains a recognition rate of 98.12% with a number of 44 classes. Ahmed et al. [29] point out segment challenges due to the variant shape of characters and space between words. They claimed that CNN is suitable for visual image pattern learning. In the view of cursive scripted language, such explicit segmentation and feature extraction approaches are appropriate where extracting features is challenging towards desirable accuracy. They used Arabic images from the English-Arabic Scene Text (EAST) dataset. The top performance reported by using a filter size of 3 x 3 and 0.005 learning rate is 14.57% error rate.

The explicit segmentation approach divides the text into characters. References [28, 29, 40] work on explicit approach for character segmentation. However, isolating characters needs extensive and complicated knowledge of the character's shapes and their starting and ending points within ligature or words.

Feature extraction method introduced by Naz et al. [28] with CNN by using the Modified National Institute of Standards and Technology (MNIST) dataset does not exactly contemplate Urdu text features. In addition, as mentioned prior [14] proposes an implicit method by introducing the UrduNet model in composition with CNN and LSTM for new tickers recognition from videos. It can be analyzed that the UrduNet model is troublesome in training with low accuracy even on its self-training dataset.

Few recent kinds of research on holistic, implicit, and explicit methods are summarized in Table. 2.1. Accuracies, methodologies, and limitations are shown in the columns.

## 2.3 Research Gaps

The research gaps identified after a comprehensive literature review are given below:

- Methods for cursive script recognition work effectively in a controlled environment. Images used were cleaned and segmented in most research. The total number of unique classes in holistic technique is very high even if dots and diacritics are removed ([32] reports 3645 ligature classes). The high dimensional feature space and higher number of labels make existing approaches intractable in terms of data scarcity, training, and prediction cost. Video data is noisy with font size variations, and traditional techniques cannot perform.
- The approach has a deciding factor of the aggregated number of segments. This leads to over-segmentation and under-segmentation. Fewer segments lead to less computation but underperform on widely written words.

Table 2.1: Related research work and research gaps

Study	Approach	Methodology	Results	Limitation
Israr et al. 2017, [30]	Holistic	Segmentation-free Nastaleeq Urdu recognition with Hidden Markov Models	92% on UPTI dataset with Nastaleeq font, trained on 3000 ligature and tested 6000 ligatures	Images used were cleaned and segmented. The total number of unique classes in such a technique would be very high even if dots and diacritics are removed (Habib et al report 3645 ligature classes). The high dimensional feature space and higher number of labels make existing approaches intractable in terms of data scarcity, training, and prediction cost.
Javed et al. 2018, [31]		Classification of Urdu Ligatures Using Convolutional Neural Networks	95% on Custom dataset with 38000 ligatures for training and 17000 ligatures for testing with Nastaleeq font	
Habib et al. 2021, [32]		The genetic algorithm is utilized for Urdu Ligature Recognition	96.72% on UPTI dataset with Nastaleeq font, Trained on 189003 ligature training and 30755 for testing	
Ahmed et al. 2019, [33]	Implicit	They presented an implicit approach by using Bidirectional Long Short Term Memory (BLSTM)	88.94% on UPTI dataset, 10,000 text lines with Nastaleeq font 46% train 10% test and 44% validation	The approach has a deciding factor of the aggregated number of segments. This leads to over-segmentation and under-segmentation.
Naz et al. 2017, [34]		They achieved higher accuracy rates with statistical features and MDLSTM.	98.12% on UPTI dataset, 10,000 text lines with Nastaleeq font 68% train 16% test and 16% validation	
Mirza et al. 2020, [35]		The combination of CNN and LSTM networks is proposed	A recognition rate of 87% on a custom-build dataset containing cur-sive text from 13,000 video frames is reported.	
Mohammad et al. 2019, [36]	Explicit	Contour-based segmentation technique gives a clear description of the word character's shape.	98.1% on Arabic Printed Text Image (APTII) with 38,000 words	Connected components method applied at multi-stages of segmentation, aiming to segment the unconnected and vertically overlapped characters. Ligatures are segmented by identified vertical projection. The method works well on clean data. However robust segmentation can lead to the highest accuracy on difficult data.
Osman et al. 2020, [37]		Projection profile method with statistical and topological features	97.51% on Arabic Printed Text Image (APTII): around 3300 words	
Aziz et al. 2022, [38]		Projection profile method with statistical and topological features	97.51% on Arabic Printed Text Image (APTII), around 2400 words	
Aziz et al. 2020, [39]		The hybrid approach of projection profile and connected components	95% on Arabic Printed Text Image (APTII), with 20% of 14,000 used for character/units recognition	

- For character-wise segmentation, the connected components method is applied at multi-stages, aiming to segment the unconnected and vertically overlapped characters. Ligatures are segmented by identified vertical projection. The method works well on clean data. However, isolating characters from noisy video data with font size variation is a complicated challenge.

This study focuses on challenging explicit approaches for Arabic and Urdu news ticker recognition. A robust semantic segmentation model is proposed to trounce complications in character/component-wise segmentation for the focused language scripts. In the context of semantic segmentation, pixel-level segmentation refers to the segmentation of characters pixels from the background whereas character-level segmentation refers to segmenting characters within a ligature or word.

## 2.4 Summary

This chapter presented a comprehensive survey for cursive text recognition. The techniques are mainly categorized into analytical and holistic approaches. In the evolutionary contributions of Arabic and Urdu OCR tasks, advanced machine-learning techniques have been used. This research focuses on challenging explicit approaches for Arabic and Urdu news ticker recognition. A robust semantic segmentation model and text formation method are proposed.

## Chapter 3

### Explicit Cursive Textual Segmentation

This chapter presents an efficient cursive news ticker text segmentation technique. The proposed technique follows an explicit method for text segmentation into characters. The chapter provides the background of the explicit method, proposed technique, and dataset preparation procedure. As explained earlier, the study targets Urdu and Arabic News Tickers. To this rationale, detected tickers are segmented by the proposed robust and advanced deep learning-based architecture in a unified manner for high accuracy. The ticker text image is semantically segmented into the character's components. These components are concatenated to form characters followed by a classification method. The study also addresses text segmentation challenges in relation to training data distributions. The text format generation method arranges the recognized string of letters into sentence formation of words and will be explained in the next chapter.

As discussed, to extract characters, it is necessary to have a complex understanding of letter patterns, as well as their beginning and ending positions inside ligatures or words. Reference at. el. [41] recently presents an explicit method of Arabic character segmentation. The segmentation is performed in three steps namely segmentation by vertical projection, connected components, and baseline removal. In the first step, dots and diacritics are removed on a binary text image using the connected component method. Practically the video data is difficult and contains noisy artifacts as described earlier, causing deformed, merging of dots, diacritic marks, and letters. The suggested method malfunctioned towards accuracy. The connected components method is also applied in the second stage of segmentation, aiming to segment the unconnected and vertically overlapped characters. In the final segmentation stage, the ligatures are segmented by identified vertical projection. Accurate identification of the baseline with no minor errors is crucial. Most of

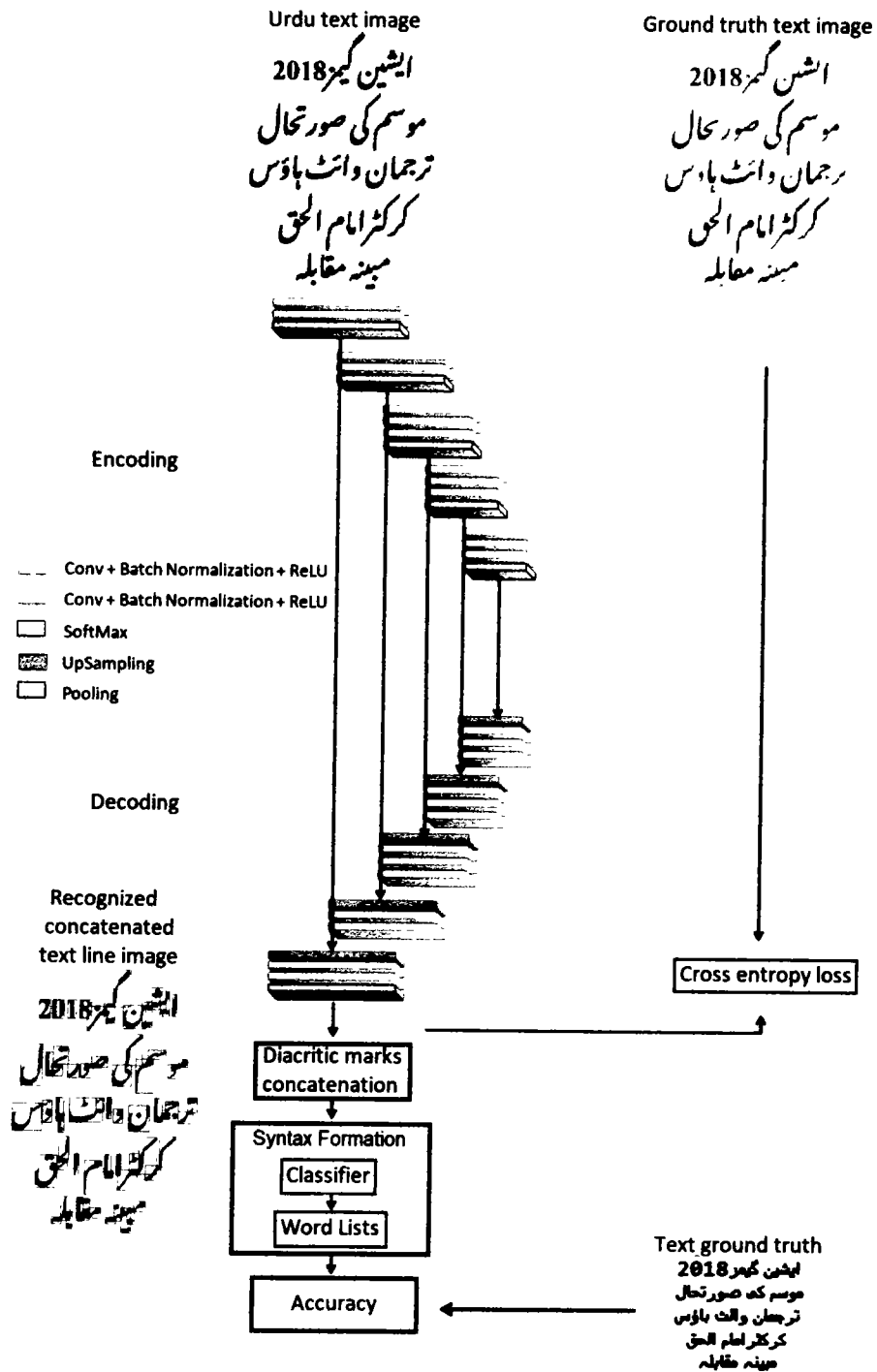


Figure 3.1: Proposed model for news ticker recognition: SegNet model presenting downsample and upsampling architecture for characters semantic segmentation, syntax formation model for Urdu (Nastaleeq font) textual recognition.



the assumptions for segmentation are based on the baseline. It could underperform on less squarish fonts as fonts must fulfill the assumptions as per the methodology. Robust binarization, especially in small fonts, is very essential, as assumption sometimes goes wrong even on squarish fonts as well. Unfair class data distribution for feature extraction using LeNet-5 is also a factor towards accuracy. Less number of classes with de-shaped samples up to certain limits (per class) is a way towards model robustness and simplicity. [42] also used a heuristic method for segmentation with previously mentioned limitations. These contemporary methods may work on clean data that is infeasible for applications like Arabic or Urdu news ticker recognition.

The effectiveness of a two-step deep learning approach is demonstrated by He Ming Yao et al. [43] for solving electromagnetic inverse scattering problems. This study employs a two-step deep learning strategy based on the explicit method to solve cursive text recognition problems with high accuracy. It proves the effectiveness compared to the other conventional methods, model simplicity not only in architectural terms but also in parametric reduction, and computational improvements compared with the highly complex conventional methods. The proposed model takes advantage of labeled data at two steps of semantic segmentation and word identification for more robust and concentrated learning. Mirza et al. [14] propose an implicit method by introducing the UrduNet model in combination with CNN and LSTM for new tickers. It can be observed that the suggested model seems difficult to train data and accuracy even on its own training dataset is very low. In the current investigation, the ticker images are first segmented by the segmentation model. The model segments the text image line into characters or character components. The segmented data is then followed by the syntax formation model. By learning in two steps, labels for the segmentation step help in robust and concentrated fine cursive text features learning from the image, while the specific purpose of syntax formation is to introduce spaces between words, which leads to a linguistically recognized form. It also helps to identify consecutive characters. The proposed segmentation process is further discussed in the coming sections, while the syntax formation or word identification model is discussed in the next syntax formation chapter. Figure. 3.1 presents an advanced Urdu ticker model suggested in current research. Ticker images are fed to the segmentation model with corresponding character-wise/component-wise labeled image data followed by the concatenation process.

### 3.1 Dataset Acquisition

For the proposed scheme of two-step learning, we need to label Urdu news ticker data at two levels or steps. No such dataset is available for Urdu text recognition experimentation. In order to formulate the current research idea, a novel Urdu News Ticker (UNT) dataset is acquired. It provides accurate character-level and component-level labeling. UPTI is currently the most famous dataset in the Urdu recognition research field that only provides ligature-based annotations. Additionally, datasets from Al-Arabiya and Al-Ekhbariya news channels (Al-ANT) and (Al-ENT) are acquired in order to carry out the ongoing research hypothesis for Arabic ticker segmentation. Data collection and processing are discussed in dataset preparation section 3.1.1.

#### 3.1.1 Dataset Preparation

To formulate the proposed idea of research, text image data with character-wise labeling at the pixel level is essential. For dataset preparation, broadcast streams are captured by the World Call Digital TV box, GK7601E-HDCA. Video data are captured from popular Saudi news channels Al-Arabiya and Al-Ekhbariya for the Arabic news tickers collection. Pakistani news channels ARY News, Geo News, Express News, GNN, 24 News, Samaa News, and Dunya News are selected for the Urdu news tickers collection. All videos are in 1080p resolution. One-third of the bottom portion of video frames are considered as ticker regions as shown in Figure. 3.2 from Pakistani (Urdu Nastaleeq) news streams. Based on the assumption, this bottom portion is good enough as particularly news tickers lie in this region. To be more focused, it is also assumed that textual content is bright on a dark background. Selected region image  $I$  is clamped with threshold  $t$ , obtaining result  $T$ , where  $t$  is 75-pixel value as shown by Equation. 3.1. Using the sobel operator  $f_x$ , vertical gradient  $G_x$  is calculated, which at each point contains the vertical derivative approximations and clamped with  $t$  as given in Equation. 3.2 and Equation. 3.3.  $G_x$  image is shown in Figure. 3.3. Vertical representation of the normalized moving average of  $M$  matrix locates the text area along with approximate font size. Substantial font size ticker lines are extracted. In this study text image data with font sizes from 25pt to 42pt are selected. The region and objects aside from ticker text are excluded with the help of distance and parallel lines. Extracted tickers are shown in Figure. 3.4 and Figure. 3.5. Similarly, Arabic news streams are processed as shown in Figure. 3.6 and Figure. 3.7. For Al-Arabiya and Al-Ekhbariya news channels, it is assumed that the text is dark on a bright background.

$$T = \begin{cases} 0 & I \leq t \\ I & I > t \end{cases} \quad (3.1)$$

$$G_x = f_x * T \quad (3.2)$$

$$M = \begin{cases} 0 & G_x \leq t \\ G_x & G_x > t \end{cases} \quad (3.3)$$

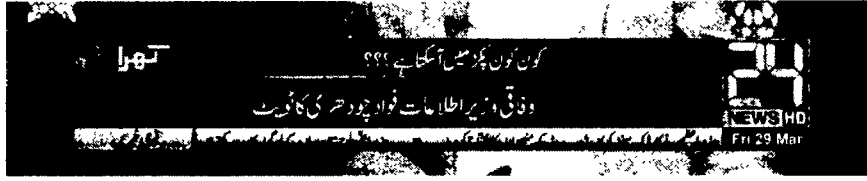


Figure 3.2: Gray-scale image of Urdu news ticker in Nastaleeq font from the video stream.

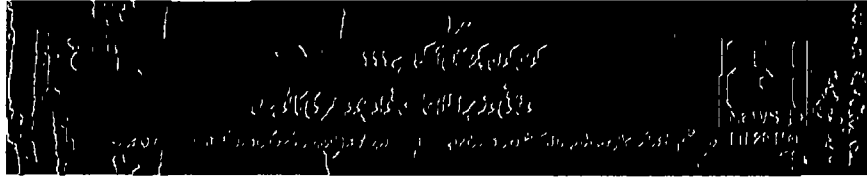


Figure 3.3: Vertical gradient  $G_x$  representation of ticker image

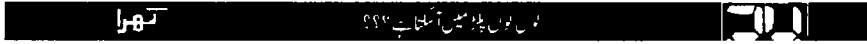


Figure 3.4: Extracted text line image of Urdu news ticker in Nastaleeq font (1).

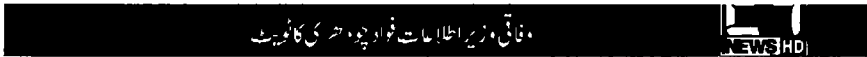


Figure 3.5: Extracted text line image of Urdu news ticker in Nastaleeq font (2).

The two Arabic datasets contain 1189 news tickers from Al-Arabiya with Kufi font and 2100 news tickers from Al-Ekhbariya with Al-Ekhbariya specific Kufi font. They have 42 classes. The Urdu dataset contains 5945 news tickers with Nastaleeq font containing 48 classes. A few text lines from the datasets with their character-wise label representations are shown in Figure. 3.8 and

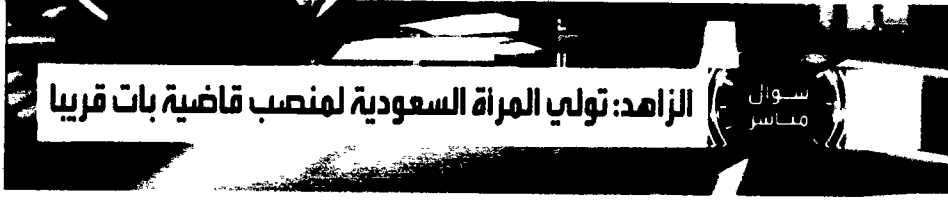


Figure 3.6: Gray-scale image of an Arabic news ticker in Kufi font from the video stream.



Figure 3.7: Extracted text line image of an Arabic news ticker in Kufi font.

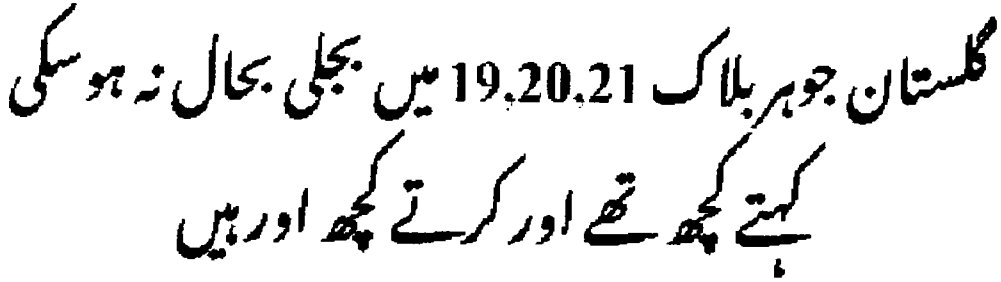


Figure 3.8: Visualization of character-wise labeled Urdu news ticker lines in Nastaleeq font.



Figure 3.9: Visualization of character-wise labeled Arabic news ticker lines in Kufi font.

Figure. 3.9. Label values assigned to character classes for Nastaleeq font are shown in Table 1. Numeric i-e. 0 to 9 are also part of tickers and are included in the dataset with label values 38 to 47. Miscellaneous characters like brackets, slash, semicolons, etc. are also included and assigned 48 label values. Characters with label values 3, 5, 8, 12, 15, 17, and 29 are not part of Arabic letters. However, the letter HUMZA is included in the Arabic data as it is part of the tickers dataset.

Table 3.1: Label and color assignment to isolated Nastaleeq font character shapes. Best viewed in color.

Class label	Shapes of characters with colormap	Class label	Shapes of characters with colormap	Class label	Shapes of Characters with colormap
1	آاا ...	17	ژژژ ...	33	ووو ...
2	ببب ...	18	سسس ...	34	ههه ...
3	پپپ ...	19	ششش ...	35	ههه ...
4	تتت ...	20	صصص ...	36	یییی ...
5	ٹٹٹ ...	21	ضضض ...	37	ےےے ...
6	ثثث ...	22	ططط ...	38	0 ...
7	ججج ...	23	ظظظ ...	39	1 ...
8	چچچ ...	24	ععع ...	40	2 ...
9	ححح ...	25	غغغ ...	41	3 ...
10	خخخ ...	26	فففف ...	42	4 ...
11	د ...	27	قققق ...	43	5 ...
12	ڈڈ ...	28	ککک ...	44	6 ...
13	ذذ ...	29	گگگ ...	45	7 ...
14	ررر ...	30	للال ...	46	8 ...
15	ڑڑڑ ...	31	ممم ...	47	9 ...
16	ززز ...	32	ننن ...	48	!-()/? ...

### 3.1.2 Character Families Hypothesis and Dataset Re-labeling

The dataset is re-labeled by examining families of Urdu alphabets [17]. In the view of families of characters, characters having homogeneous structural shapes (prime component) with non-identical diacritic marks are considered separate class groups. Family-wise prime components of characters with class labels are tabulated in Table. 3.2. Characters with labels 2 to 6, 7 to 10, 11 to 13, 14 to 17, 18 to 19, 20 to 21, 22 to 23, and 24 to 25 are grouped together. Diacritic marks are considered separate classes to identify characters within the family. These marks with their class labels are tabulated in Table. 3.3. Common diacritic marks mentioned in the table and diacritic marks of character groups containing single labels are also considered separately. Diacritic marks with label values 64 and 65 are not part of Arabic letters.

Table 3.2: Label and color assignment to prime components of characters. Best viewed in color.

Grouping of class labels	New assign label	Shapes of prime component of characters with new colormap	Grouping of class labels	New assign label	Shapes of prime component of characters with new colormap
2 to 6	51	ب ر ز ر ر ر ...	18 to 19	55	س س س س س ...
7 to 10	52	ح ح ح ح ح ...	20 to 21	56	ص ص ص ص ص ...
11 to 13	53	د د د ...	22 to 23	57	ط ط ط ط ط ...
14 to 17	54	ز ز ز ز ز ...	24 to 25	58	ع ع ع ع ع ...

Table 3.3: Label and color assignment to diacritic marks of characters incorporating homogeneous structural shapes. Best viewed in color.

Class label	Shapes of diacritical notations of characters with colormap	Class label	Shapes of diacritical notations of characters with colormap
61	• ...	64	• ...
62	...	65	ط ...
63	• ...	-	-

This method leads to a significant reduction in the number of classes (42 to 35 for Arabic and 48 to 37 for Urdu) and model parameters ( $k$  of softmax classifier as given in Equation (3.5)). At the

recognition stage, these diacritic marks are concatenated with the detected shapes by a restrained distance formula within the width of the prime component of the characters. For font size diversity, limitations in font sizes with a fair representation of each size are tried to ensure. Details of the Urdu characters in the UNT dataset for the proposed model are tabulated in the Table. 3.4, whereas for Arabic datasets Al-ANT and Al-ENT are tabulated in Table. 3.4 and Table. 3.4 respectively.

### 3.2 Semantic Segmentation Process

This study along the way uses FCN and SegNet deep learning architectures for Urdu text recognition. The network learns to assign each pixel a class label determined by the object's or character's surroundings, spatial location, and orientation it belongs to. When using these models for semantic segmentation, the recognized output is also an image of the same size as of label image rather than a vector. The encoder layer performs convolution with filters. The outputs are batch normalized after which an element-wise Rectified Linear Unit ReLU operation is performed as given in Equation. (3.4). Where  $x$  is the batch normalization output value.

$$\phi(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \quad (3.4)$$

In FCN architectures, the input image is downsampled by moving through the convolution and fully connected layers. Upsampled output is the predicted label map of the same size as the input image. Finally, the softmax classifier predicts the label map on a probabilistic basis given in Equation (3.5). Where  $\phi_i$  represents the values at the output layer and  $k$  is the total number of outputs.

$$\text{softmax}(\phi_i) = \frac{\exp(\phi_i)}{\sum_j^k \exp(\phi_j)} \quad (3.5)$$

Table 3.4: Details of Urdu characters in UNT dataset

Character class	Frequency	Frequency Percent	Group Frequency	Character class	Frequency	Frequency Percent	Group Frequency	Frequency Percent
ا	14828	13.59	14828	ب	141	0.129	141	0.129
ب	3971	3.638	9807	پ	1771	1.623	1771	1.623
پ	2446	2.242		ت	1097	1.005	1097	1.005
ت	868	0.795		ث	6141	5.628	6141	5.628
ث	1904	1.745		ج	75	0.069	75	0.069
ج	618	0.567		چ	4481	4.107	4481	4.107
چ	1582	1.450	4942	ح	5772	5.291	5772	5.291
ح	1223	1.121		خ	7344	6.732	7344	6.732
خ	1011	0.927		د	6199	5.682	6199	5.682
د	1125	1.031		ڈ	3961	3.630	3961	3.630
ڈ	3136	2.874	3809	ڈ	1037	0.950	1037	0.950
ڈ	552	0.506		ن	13397	12.280	13397	12.280
ن	121	0.111		ہ	3499	3.207	3499	3.207
ہ	8323	7.629	10019	و	100	0.092	100	0.092
و	196	0.180		ز	176	0.161	176	0.161
ز	1488	1.364		ذ	128	0.117	128	0.117
ذ	10	0.009		ر	70	0.064	70	0.064
ر	3794	3.478	5389	س	75	0.069	75	0.069
س	1594	1.461		ش	88	0.080	88	0.080
ش	623	0.571	825	ص	50	0.046	50	0.046
ص	201	0.185		ض	70	0.064	70	0.064
ض	317	0.291	797	ط	40	0.037	40	0.037
ط	479	0.439		ظ	25	0.023	25	0.023
ظ	1698	1.556	1839	ع	1236	1.133	1236	1.133
				Total	109079	100	109079	100



Table 3.5: Details of Arabic characters in AI-ANT dataset

Character class	Frequency	Frequency Percent	Group Frequency	Group Frequency Percent	Character class	Frequency	Frequency Percent	Group Frequency	Group Frequency Percent
	1757	13.231	1757	13.231	ـ	163	1.2275	163	1.2275
ا	320	2.4098	850	6.4011	و	1587	11.951	1587	11.951
ب	332	2.5002			ی	828	6.2354	828	6.2354
پ	198	1.4911			ز	572	4.3076	572	4.3076
ت	293	2.2065	568	4.2774	ح	173	1.3028	173	1.3028
ث	183	1.3781			خ	583	4.3904	583	4.3904
ج	92	0.6928			ط	490	3.6900	490	3.6900
د	554	4.1720	708	4.7519	ق	87	0.6552	87	0.6552
ذ	77	0.5799			ک	1248	9.3983	1248	9.3983
ر	481	3.6223	559	4.2097	گ	83	0.6250	83	0.6250
ز	78	0.5874			ن	98	0.7380	98	0.7380
س	495	3.7277	593	4.4657	ی	43	0.3238	43	0.3238
ص	98	0.7380			ه	54	0.4067	54	0.4067
ض	174	1.3103	275	2.0709	و	46	0.3464	46	0.3464
ط	101	0.7606			ز	45	0.3389	45	0.3389
ظ	156	1.1748	248	1.8676	ح	52	0.3916	52	0.3916
ع	92	0.6928			خ	42	0.3163	42	0.3163
ف	417	3.1403	509	3.9084	ط	34	0.2560	34	0.2560
ق	102	0.7681			ک	36	0.2711	36	0.2711
ک	338	2.5454	338	2.5454	ن	35	0.2636	35	0.2636
گ	175	1.3179	175	1.3179	ی	467	3.5168	467	3.5168
Total					Total				
					13279				
					100				
					100				

Table 3.6: Details of Arabic characters in AI-ENT dataset

Character class	Frequency	Frequency Percent	Group Frequency	Character class	Frequency	Frequency Percent	Group Frequency	Group Frequency Percent
ا	3162	13.5965	3162	ح	294	1.2642	294	1.2642
ب	576	2.4768	1551	خ	2856	12.2807	2856	12.2807
ت	597	2.5671		د	1491	6.4112	1491	6.4112
ث	378	1.6254		ذ	1029	4.4247	1029	4.4247
ج	491	2.1113	932	ر	311	1.3373	311	1.3373
چ	294	1.2642		س	1050	4.5150	1050	4.5150
ح	147	0.6321		ش	882	3.7926	882	3.7926
د	997	4.2871	1100	ص	127	0.5461	127	0.5461
د	103	0.4429		ض	2247	9.6620	2247	9.6620
ر	848	3.6464	953	ط	104	0.4472	104	0.4472
ز	105	0.4515		ظ	111	0.4773	111	0.4773
س	891	3.8313	1038	ع	78	0.3354	78	0.3354
ص	147	0.6321		ف	98	0.4214	98	0.4214
ش	260	1.1180	324	ق	83	0.3569	83	0.3569
ط	74	0.2752		ك	81	0.3483	81	0.3483
ظ	281	1.2083	393	گ	93	0.3999	93	0.3999
ع	112	0.4816		ل	76	0.3268	76	0.3268
ف	714	3.0702	840	م	61	0.2623	61	0.2623
ق	126	0.5418		ن	64	0.2752	64	0.2752
ك	609	2.6187	609	هـ	63	0.2709	63	0.2709
گ	315	1.3545	315	و	840	3.6120	840	3.6120
			Total			23256	100	100

We use the FCN-8s standard network due to its best performance in FCN variants to avoid spatial location information loss when moving deeper by fusing the process of three pooling layers. FCN-8s sums the 2x upsampled convolution7 (with a stride 2 transposed convolution) with pooling4 outputs, upsamples them with a stride 2 transposed convolution and sums them with pooling3, and applies a transposed convolution layer with stride 8 on the resulting feature maps to obtain the segmentation map [44].

Like FCN, SegNet architectures are also broadly in two parts, operating encoder, and decoder. The input text image is first downsampled (pooling) in an encoding process like the CNN architecture of ResNet or FCN. Decoding is a reversal process of encoding, processed with upsampling layers rather than downsampling layers. At the deepest encoding output, SegNet discards the fully connected layer. This reduces the number of parameters compared to other recently proposed architectures [45, 46]. The decoding output is fed to a softmax classifier to form a predicted label map of the same size as the input image on a probabilistic basis. In this investigation, eighteen layers of encoder architecture are used. Like the VGG16 network, the encoder layers have correspondence with the decoder layers, and hence decoder also has eighteen layers as shown in Figure. 3.1.

All network parameters are randomly initialized. Pooling window dimensions and stride values are also factors of the network. The output of the softmax classifier is a  $k$  channel image of probabilities where  $k$  is the number of classes. The predicted segmentation corresponds to the class with maximum probability at each pixel.

As compared to SegNet, the training performance of FCN-8s is comparatively much lower. The study focuses on SegNet architecture for further investigation. While testing with Nastaleeq font, the SegNet characters segmentation model leads to class correlation drawbacks. This is because of inter-class correlations between character classes that have homogeneous structures (prime components) with identical diacritic marks, causing learning complexity and underperforming particularly on characters with low frequencies. Further details will be discussed in the Experimental Results Chapter 5. Following subsection 3.3 describes the proposed technique for the above-mentioned limitations.

### 3.3 UrduSeg Segmentation Model

The proposed learning architecture incorporates the grouping of homogeneously shaped prime character component classes with their diacritic marks separately identified in a unified manner. The dataset contains correlated classes with similar character shapes as shown in Table. 3.1. Con-

sequences can be found while testing results on trained learning models. In addition to this, many characters such as labeled values 4, 6, 13, 17, etc. exist with low frequencies comparatively in Urdu writing. This causes certain segmentation issues. Low-frequency representation of data samples leads to unfair distribution of data. This imbalance in data distribution inherits biases. Figure. 5.1 in Experimental Results chapter shows segmented Urdu text lines with two low-occurring characters (label values 4 and 6). The segmentation model gets confused with correlated characters at the pixel level. Similarly as shown in Figure. 5.2, improper distribution of font sizes in the dataset also causes biased effects in the learning model. The training model will be further described in the Experimental Results chapter.

To overcome the mentioned issues, this work proposed a novel approach for data curation and augmentation in order to increase fairness by incorporating the concept of Urdu/Arabic character families for curation with more diversity. This aims to increase fairness by maximizing the balance among sensitive features. We relabeled the dataset by contemplating the character families for learning-based segmentation to encounter inter-class correlations between the homogeneous shapes with similar character structures (prime component) and non-identical diacritic marks.

To this end, Arabic and Urdu ticker datasets are re-labeled by grouping character classes on the basis of character families, having homogeneous structural shaped prime character components with their diacritic marks are considered as separate classes. The grouping processes of characters comfort low symbolic representations and reduce the unfair distribution of data.

This method leads to a significant reduction in the number of classes (42 to 35 for Arabic and 48 to 37 for Urdu) and model parameters ( $k$  of softmax classifier). We used encoding and decoding architecture with collectively thirty-six layers for deep feature learning as shown in Figure. 3.1. Diacritic marks are considered separate classes to identify characters within the family. Common diacritic marks mentioned in the table and diacritic marks of character groups containing single labels are also considered separately.

At the recognition stage, these diacritic marks are concatenated with corresponding prime components by a heuristic mechanism. Prior to the concatenation procedure, the cleaning/filtering process is carried out. Statistical modulus operation with window size  $3 \times 3$  is performed to remove minor semantic segmentation errors on recognized ticker images. Labeled components in the recognized ticker image are examined. The components with area  $A$ , lower than certain limits are removed (less than equal to window size  $3 \times 3$ ). Separate copies of recognized ticker images are maintained for prime components and others for diacritic marks where characters and diacritic

**Algorithm 1:** Character components concatenation algorithm

---

**Input:** *Segmented character components image*  
**Output:** *Concatenated characters image*

**for** each of recognized prime component **do**  
    **if** check if component conditioned to grouping **then**  
        Calculate width of prime component  
        **if** prime component contains single diacritics within width **then**  
            Group the component and assigned concatenated character label  
        **else**  
            Calculate minimum horizontal distance between prime component and diacritics  
            Group the prime component diacritic with minimum distance  
        **end**  
    **end**  
**end**

---

marks are relabeled in sequence order. The same labels are split under the condition of the component area in parallel with area-based cleaning and distances between the same labeled components (window size 7x5 pixels). The splitting condition depends upon the area of the component greater than certain limits depending on font size  $A_{max}$  and multiple components of the same recognized label (where  $A_{max} = \text{round}(\text{fontSizeHeight} \times 0.1)$  and window size 7x5 pixels difference in case of diacritic). The sequence labeled assignment is on the principle of the left boundary of the component. Diacritic marks are concatenated with the detected shapes by a restrained distance formula within the width of the prime component of the characters. The Algorithm 1 describes the concatenation process as it is applied in current research. The prime component, conditioned members of a particular family with their predefined possible diacritic marks within the width of the prime component are grouped. In the case of multiple diacritic marks, the center points of the horizontal width of prime components and possible diacritic marks are calculated under the condition that the diacritic marks are not already grouped with some other characters. The lowest horizontal distance from the calculated center point of diacritic marks and the prime component are grouped. The new label is assigned to the diacritic and the prime component of the recognized character and coordinate are stored. The character coordinates from the ground truth are also stored for evaluation purposes, where the same characters are considered identically. On an observation basis, no multiple diacritic marks within the width of the prime component with the same labels are assumed. Considering Nastaleeq is a difficult Urdu font, heuristics can fail somewhere depending on the misclassification of pixels. The result and accuracy will be discussed in the coming chapter.

### 3.4 Summary

This chapter presented the technical details of the explicit ticker text segmentation model for Urdu and Arabic. The technique can be adapted to other cursive scripts as well. SegNet deep learning model is applied for current examinations. Novel news ticker datasets for two-step deep learning with character-wise, character components-wise image labeling, and in textual form are acquired to formulate the proposed model. The proposed strategy of segmentation considers the Urdu/Arabic character's families to reduce data unfairness and to increase the data diversity of letters for better training convergence. The diacritic marks are concatenated with their prime character components at the post-segmentation stage and are placed in buffers for the next step of the syntax formation process.

TH-26207

## Chapter 4

### Syntax Formation Model

Words identification from the recognized string of letters is essential for the complete text recognition process. The characters obtained from the previous step need to be arranged into the sentence of words. The textual formation process is performed in two steps. Words codes are the outputs of this process to be stored for the retrieval operation.

In the context of cursive text recognition, the segmented letters or recognized strings of letters from the proposed segmentation model are compiled into a complete sentence of words by introducing spaces between words. The objective of this step is not only to determine word ending characters but also the consecutive characters within words. The potential inter-connected consecutive characters with the same class label like seen, sheen, swad and noon (label values 18, 19, 20, and 32) occasionally lie in the words of the dataset. Arabic or Urdu characters generally have multiple shapes determined by their appearance and position within the word [47]. Mostly the ending letters in Arabic and Urdu writing have a particular shape that is nearly similar to the individual character.

The Syntax generation model is shown in Figure. 4.2. Karhunen-Loève transformation and LeNet-based classification models with the Words Code Generator technique are proposed and are explained in the following sections.

#### 4.1 Karhunen-Loève Transformation based Classification Model

The current research develops an efficient method of classification. The Karhunen-Loève Transformation (KL) transformation is employed for this objective. KL transformation is also called

Principle Component Analysis (PCA). It is an orthonormal decomposition process by the projection of relevant eigenvectors of the covariance matrix  $C$  for 3-dimensional ( $x$ ,  $y$ , and  $z$ ) data as given in Equation. 4.1. It is a  $N \times N$  symmetric matrix, where  $N$  is the number of dimensions.  $var$  and  $cov$  are variance and covariance respectively.

$$C = \begin{bmatrix} var(x) & cov(x, y) & cov(x, z) \\ cov(y, x) & var(y) & cov(y, z) \\ cov(z, x) & cov(z, y) & var(z) \end{bmatrix} \quad (4.1)$$

The standardization prior to the KL-transformation, the normalization process is performed to avoid sensitivity regarding the variances of the initial characters variables by using the following Equation. 4.2.

$$v = \frac{p - \mu}{\sigma} \quad (4.2)$$

Where  $p$  is pixel value,  $\mu$  is mean and  $\sigma$  is the standard deviation. The eigenvector i.e. the diagonal vector of the variance matrix as given in Equation. 4.1, directed to maximum variation of the letters prototype data. A precise calculation of the finest wide variety of factors may be received via way of means of the Fisher criterion [48]. In this investigation, the best and minimum number of appropriate eigenvectors in the transformed matrix is selected considering experimental results and will be further discussed in the Experimental Results chapter. As described, diacritic marks are concatenated with their prime components at the post-segmentation stage. Segmented character images from the segmentation model are arranged right-to-left accordingly. These letters are sorted on a left boundary basis and are placed in 64x64 pixels dimension buffers, as a post-concatenation process of diacritic marks with prime components as shown in Figure. 4.1. The letters larger than 64x64 pixels in dimension, height-wise, width-wise, or both are scaled down to a maximum of 64 pixels height-wise, width-wise, or both by maintaining the aspect ratios. The buffer is flattened to a vector for KL-transform. This model ignores the classification of letters, which have a similar shape at the end of the words as well as ligatures. 45 and 56 classes are used for Arabic and Urdu classification models respectively. This includes possible characters, characters with space ahead, and consecutive characters. Consecutive characters with space ahead are not included as no example lies in the considered dataset.

The presented approach is adequate for Kufi and Nastaleeq cursive fonts within the mentioned



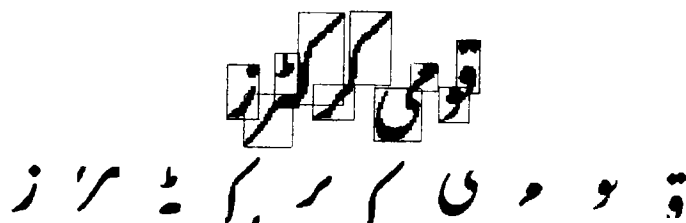


Figure 4.1: Segmented and concatenated Nastaleeq font characters in [64x64] pixels dimensional buffer.

range of font size. This step is only for ending and consecutive character identification. The classification process determines ending characters with certain accuracy and limitations for space insertion. In a few cases, space between the word does not affect the meaning of a word or sentence like  $\text{ہے}$  or  $\text{ہے}$  means the same in presented Urdu cursive writing. Similarly, sometimes writing two words without space in between does not affect the reading. The study suggests a words searching method based on a searching procedure to identify complete words within the recognized string. Lists of words are organized in ascending order for searching. The process will be further discussed in the Words Code Generator section.

## 4.2 LeNet-based Classification Model

As explained, inserting space between words from the recognized character representations is compiled into a sentence of words. For this objective, another approach is employed using the LeNet-5 model with minor modifications in the final layer to classify more than 10 classes. The network only contributes to the recognition of consecutive characters and the word's terminal character in order to add space after each word. Depending on where it appears in the word, Urdu/Arabic letters contain multiple shapes. As discussed, the shape of the last character of the word is almost similar to the independent letter shape and can be differentiated from the other shapes.

The segmented image's recognized letters are ordered from right to left. After the concatenation of prime components with diacritic markings, these characters are sorted based on the left borders and placed in 32x32 pixels dimension buffers, as illustrated in Figure. 4.2. The character that is greater than 32x32 pixels in height, width or both are scaled down to a maximum of 32 pixels in height, width, or both while maintaining the aspect ratios. The classification model utilizes these character candidates as input. This process only determines the ending and successive characters. The letters, which have a similar shape to the letter at the end of the words and ligatures are not

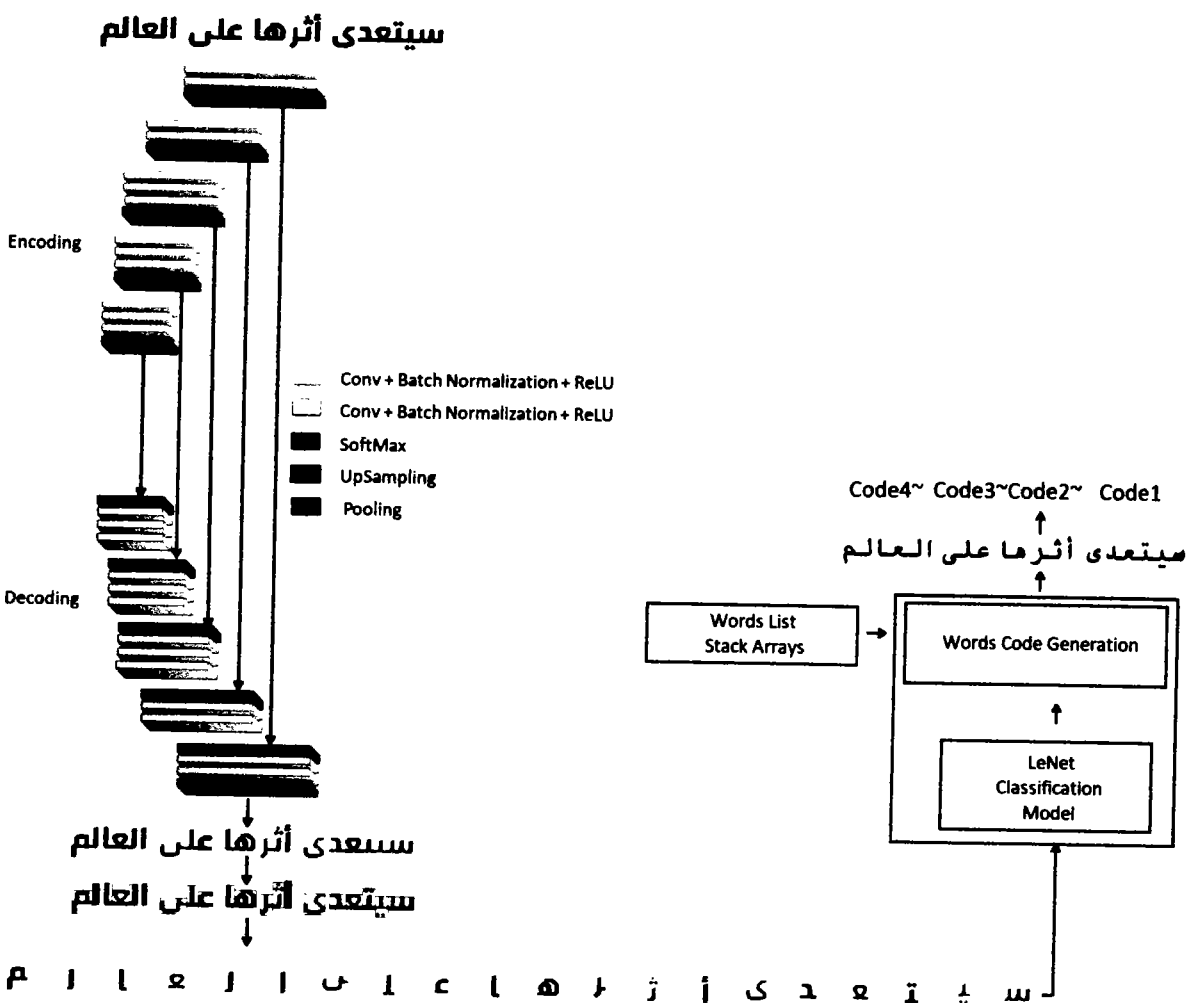


Figure 4.2: Extracted text line image of an Arabic news ticker in Kufi font.

addressed in this model. 45 and 56 classes are used for Arabic and Urdu classification models respectively.

Most of the time the shape of the ending character of the word can be similar to the last character of the ligature. This method does not assure that identified symbols will be appropriately arranged into word-formation sentences. For further corrections, a searching-based syntax rectification technique is developed, which will be addressed in the next section of Words Code Generator.

4.3 Words Code Generator

The suggested syntax correction approach entails a search for entire words inside the identified string. Words lists are organized in ascending order. A track of these key lists is maintained in a separate array of pointers. This pointer array is characterized as a sparse array because it contains information about word lists. Each list has a fixed number of words,  $n$ , ranging from 2 to  $m$ , where  $m$  is the maximum word length in data. These lists are arranged in order to perform an efficient search of words [49]. Using a search procedure, we attempted to find the whole word starting with the longest word. Unless a word is not found in the list, continue searching in a list of words of length  $(n-1)$  until the word list length reaches to two characters. Failing in every step, which is finally an isolated character. The Algorithm 2 for word detection/identification is given below. Table. 4.1 shows notations used in the algorithm.

For sentence organization, the proposed classification and word-searching methods reinforce each other. In a searching methodology, without a classification method, a sequence of incorrect textual recognition followed by a single miss-classified character is possible.

Table 4.1: Notations used in algorithm

Notations used in algorithm	Notation description
Maximum length of word	$m$
Recognized string of letters	<i>recognizedString</i>
An array indicating spaces within recognized string of letters	<i>spaceLocationsArray</i>

The unique code of the word is the combination of the list number and the index number with the list. This code is used to save data for later retrieval. The unsigned short data type of 16-bit can be used to store codes. The storage technique is space efficient because most words have more than two letters. The approach decreases the use of storage resources while also allowing for effective searches using these key codes. This efficient scheme of storage and searching is suggested in this study for future implementation. Figure. 4.2 describes the layout of the proposed Arabic New ticker architecture.

**Algorithm 2:** Urdu word detection algorithm

---

**Input:**  $m$ ,  $recognizedString$   
**Output:**  $spaceLocationsArray$   
**Initialized:**  $indx \leftarrow 1$   
**while**  $indx \leq \text{length}(recognizedString)$  **do**  
    **for**  $i \leftarrow 1$  **to**  $m$  **do**  
         $uWord[i] \leftarrow recognizedString[indx]$ ;  
         $indx \leftarrow indx + 1$ ;  
        **if**  $indx > \text{length}(recognizedString)$  **or**  $(recognizedString(indx) =$   
             $'specialcharacter')$  **then**  
            |  $break$ ;  
        **end**  
    **end**  
     $n \leftarrow \text{length}(uWord)$ ;  
    **if**  $n > 0$  **then**  
         $found \leftarrow 0$ ;  
        **if**  $found = 0$  **and**  $n = m$  **then**  
             $found \leftarrow \text{findwordMCharacterList}(uWord[1 \text{ to } n])$ ;  
            **if**  $found = 0$  **then**  
                |  $n \leftarrow n - 1$ ;  
            **end**  
        **end**  
        ...  
        **if**  $found = 0$  **and**  $n = 2$  **then**  
             $found \leftarrow \text{findwordTwoCharacterList}(uWord[1 \text{ to } n])$ ;  
            **if**  $found = 0$  **then**  
                |  $n \leftarrow n - 1$ ;  
            **end**  
        **end**  
         $indx \leftarrow indx - \text{length}(uWord) + n$ ;  
         $spaceLocationsArray \leftarrow \text{saveSpaceLocations}(n, indx - 1)$ ;  
    **end**  
**end**

---

## 4.4 Summary

The suggested syntax formation approach involves classification and word-searching techniques to identify words inside the identified string. The process of textual formation is divided into two parts of the classification model and the word identification model results in a complete sentence of words by introducing spaces between words. For this, the identification of the ending letter of

the word is performed with the KL transformation and LeNet-based classification model, and a searching-based technique.

## Chapter 5

# Experimental Results

Semantic segmentation learning-based networks are explored in this study. SegNet architecture is composed of encoder and decoder layers. The encoder and decoder layers are symmetrical with each other. The upsampling operation of the decoder uses the max-pooling indices of the corresponding encoder. Unlike FCNs, no learnable parameters are used for upsampling.

Experiments are performed on Intel® Core™ i3-8100 with CPU@3.60GHzx4, 8GB DDR-5 RAM, and SSD drive computer system. A high-end GeForce RTX™ 2070/PCIe/SSE2 NVIDIA Graphics Processing Unit (GPU) is used to accelerate the training process. Architectures are implemented in C/C++. The following sections will discuss experiments carried out in the research.

### 5.1 Experimental Partitioning of Datasets

Experiments are performed using the following datasets. UNT, Al-ENT, and Al-Arabiya datasets contain news tickers that are single channels, placed in 205x1365 dimensions with corresponding 205x1365 dimensions character-wise labeled images (ground truths). For evaluation purposes, the UPTI benchmark dataset is used.

#### 5.1.1 UNT Dataset

It is an Urdu dataset containing 5945 news tickers with Nastaleeq font containing 48 classes. 3000 images are used for training, 1000 images are used for validation, and 1945 images are used for testing the Urdu segmentation model.

### 5.1.2 Al-ENT Dataset

A total of 2100 news tickers are present in the Al-ENT dataset, collected from the Al-Ekhbariya news channel with Kufi font and 42 classes. 1170 for training, 130 for validation, and 800 for testing the Arabic segmentation model.

### 5.1.3 Al-Arabiya Dataset

This dataset contains 1189 Arabic news tickers collected from the Al-Arabiya news channel with 42 classes. 715 for training, 200 for validation, and 274 for testing are used in the Arabic segmentation model.

### 5.1.4 UPTI Dataset

It is a reputed benchmark Urdu dataset used for evaluation purposes. The dataset contains 10000 text line images with Nastaleeq font. UPTI may be less deformed but its reputed data can be used. UPTI dataset is used in numerous studies

## 5.2 Performance Measurement

The following are the performance measures used to evaluate the current research.

### 5.2.1 Accuracy

Accuracy is illustrated below in Equation. 5.1. It is the total number of pixels correctly recognized by a total number of pixels. Only the pixels in the text area are taken into account while calculating pixel accuracy.

$$Accuracy = \frac{Total\ Correct\ Pixels}{Total\ Number\ of\ Pixels} \quad (5.1)$$

### 5.2.2 F-Score

Segmented letter pixels are bounded as prediction masks in the post-filtration process that removes misclassified pixels based on the area within recognized components.

Precision, Recall, and F-Score are illustrated below in Equation. 5.2, 5.3, and 5.4 to evaluate predicted masks with the available target masks of provided input. The scores are calculated on

overlapping masks with a minimum of 50% overlap.

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5.4)$$

### 5.3 Segmentation Model Experiments

This section discusses the detailed experimental analysis of the proposed segmentation model. The following subsections describe the experimental setups, segmentation model simulations analysis, and result analysis.

#### 5.3.1 Experimental Setups

The encoder network consists of eighteen convolutional layers and each encoder layer has a corresponding decoder layer. ReLU as an activation function is used to further speed up the training. To produce class probabilities for each pixel independently, the final decoder output is connected to a multiclass softmax classifier. Table. 5.1 and Table. 5.2 illustrate encoding and decoding layers used in the network with parameters and matrix shapes. The number of bias parameters are also included in the total number of parameters. Sigmoid decay is used for the learning rate.

#### 5.3.2 Segmentation Model Simulations Analysis

Multiple experiments are performed for segmentation. In the first experiment, training is performed with Nastaleeq font by ignoring the proposed technique of character families strategy. The training details are summarized in Table. 5.3 (second column). This took 109 hours to train. The blue line in graph Figure. 5.3 shows SegNet learning curves for training loss, validation accuracy, and validation loss.

In the second experiment, training is performed with Nastaleeq font by considering the proposed learning architecture that incorporates the grouping of homogeneous shaped prime component



Table 5.1: SegNet encoding layers with parameters and matrix shapes.

Encoding layers	Kernel size	Number of filters	Convolution output	Batch normalization	Mask pooling	BN + Bias parameters	Total number of parameters
Convolution	3	64	64x205x1365	Yes	No	128	768
Convolution	3	64	64x205x1365	Yes	No	128	37,056
Max Pooling	2	-	-	No	Yes	-	-
Convolution	3	64	64x103x683	Yes	No	128	37,056
Convolution	3	64	64x103x683	Yes	No	128	37,056
Max Pooling	2	-	-	No	Yes	-	-
Convolution	3	64	64x52x342	Yes	No	128	37,056
Convolution	3	64	64x52x342	Yes	No	128	37,056
Convolution	3	64	64x52x342	Yes	No	128	37,056
Max Pooling	2	-	-	No	Yes	-	-
Convolution	3	64	64x26x171	Yes	No	128	37,056
Convolution	3	64	64x26x171	Yes	No	128	37,056
Convolution	3	64	64x26x171	Yes	No	128	37,056
Max Pooling	2	-	-	No	Yes	-	-
Convolution	3	64	64x13x86	Yes	No	128	37,056
Convolution	3	64	64x13x86	Yes	No	128	37,056
Convolution	3	64	64x13x86	Yes	No	128	37,056
Max Pooling	2	-	-	No	Yes	-	-

Table 5.2: SegNet decoding layers with parameters and matrix shapes.

Decoding layers	Kernel size	Number of filters	Deconvolution output	Batch normalization	Mask upsampling	BN + Bias parameters	Total number of parameters
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	64x13x86	Yes	No	128	37,056
Deconvolution	3	64	64x13x86	Yes	No	128	37,056
Deconvolution	3	64	64x13x86	Yes	No	128	37,056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	64x26x171	Yes	No	128	37,056
Deconvolution	3	64	64x26x171	Yes	No	128	37,056
Deconvolution	3	64	64x26x171	Yes	No	128	37,056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	64x52x342	Yes	No	128	37,056
Deconvolution	3	64	64x52x342	Yes	No	128	37,056
Deconvolution	3	64	64x52x342	Yes	No	128	37,056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	64x103x683	Yes	No	128	37,056
Deconvolution	3	64	64x103x683	Yes	No	128	37,056
Upsample	2	-	-	No	Yes	-	-
Deconvolution	3	64	64x205x1365	Yes	No	128	37,056
Deconvolution	3	49	49x205x1365	Yes	No	-	28,401

classes. Their diacritic marks are segmented and are considered separate classes. Training details are summarized in Table. 5.3 (in the third column). It took 104 hours to train. The green line in graph Figure. 5.3 shows SegNet learning curves of training loss, validation accuracy, and validation loss.

Table 5.3: SegNet training parameters.

Training parameters	Urdu ticker training without character grouping	Urdu ticker training with characters grouping	Arabic ticker training with characters grouping	Arabic ticker training with characters grouping
Symbol classes	48	37	35	35
Training epochs	250	250	100	100
Batch size	1	1	1	1
Learning rate	0.01	0.01	0.01	0.01
Training time	109 hrs	104 hrs	34 hrs	58 hrs

In view of the graphical representation of the training of models (blue and green curves of training and validation loss), the overall convergence of proposed methods for semantic segmentation per epoch is head and shoulders above. The network gets confused in the semantical classification of low-frequency characters. It proves that the proposed strategy of segmentation considering the families of Urdu letters is more effective.

The dataset contains correlated classes with similar character shapes as shown in Table. 3.1. Consequences can be found while testing results on trained learning models. In addition to this, many characters such as labeled values 4, 6, 13, 17, etc. exist with low frequencies comparatively in Urdu writing. This causes certain segmentation issues. Low-frequency representation of data samples leads to unfair distribution of data. This imbalance in data distribution inherits biases. Figure. 5.1 shows segmented Urdu text lines with two low occurring characters (label values 4 and 6). The segmentation model gets confused with correlated characters at the pixel level. Similarly as shown in Figure. 5.2, improper distribution of font sizes in the dataset also causes biased effects for the learning model.

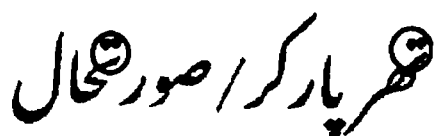


Figure 5.1: Low-frequency characters segmentation issues.

راولپنڈی / فائرنگ

Figure 5.2: Font size segmentation issues.

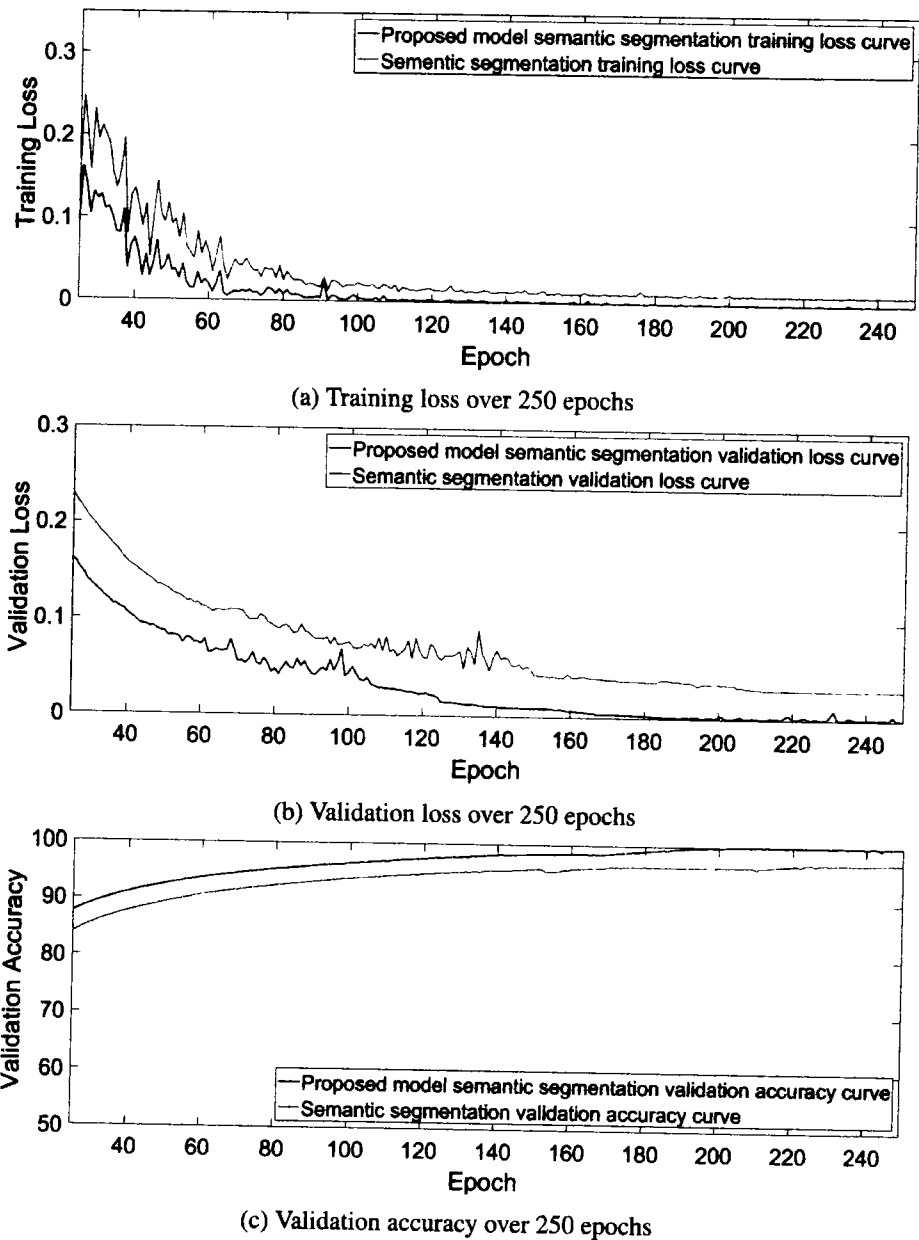


Figure 5.3: Comparison of graphical representation of training curves with Nastaleeq font.

Table 5.4: Urdu SegNet encoding layers with the number of filters and parameters.

Encoding layers	Kernel size	Model 3		Model 4		Model 5		Model 6	
		No of filters	No of parameters	No of filters	No of parameters	No of filters	No of parameters	No of filters	No of parameters
Convolution	3	48	576	32	384	16	192	8	96
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Max Pooling	2	-	-	-	-	-	-	-	-
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Max Pooling	2	-	-	-	-	-	-	-	-
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Max Pooling	2	-	-	-	-	-	-	-	-
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Max Pooling	2	-	-	-	-	-	-	-	-
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Max Pooling	2	-	-	-	-	-	-	-	-
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Convolution	3	48	20,880	32	9,312	16	2,352	8	600
Max Pooling	2	-	-	-	-	-	-	-	-

Table 5.5: Urdu SegNet decoding layers with the number of filters and parameters.

Decoding layers	Kernel size	Model 3		Model 4		Model 5		Model 6	
		No of filters	No of parameters	No of filters	No of parameters	No of filters	No of parameters	No of filters	No of parameters
Upsample	2	-	-	-	-	-	-	-	-
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Upsample	2	-	-	-	-	-	-	-	-
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Upsample	2	-	-	-	-	-	-	-	-
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Upsample	2	-	-	-	-	-	-	-	-
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Upsample	2	-	-	-	-	-	-	-	-
Deconvolution	3	48	20,880	32	9,312	16	2,352	8	600
Deconvolution	3	33	14,385	33	9,601	33	4,817	33	2,425

To overcome the mentioned issues, this work proposed a novel approach for data curation and augmentation in order to increase fairness by integrating the concept of Urdu character families for curation with more diversity. This aims to increase fairness by maximizing the balance among sensitive features. We relabeled the dataset by contemplating the character families for learning-based segmentation to encounter inter-class correlations between the homogeneous shapes with similar character structures (prime component) and non-identical diacritic marks. For validation, data is selected randomly from the dataset. Validation accuracy is important because it indicates how well a neural network can generalize to new data. A high validation accuracy means that the network is able to identify patterns in the data and not just memorize the training set.

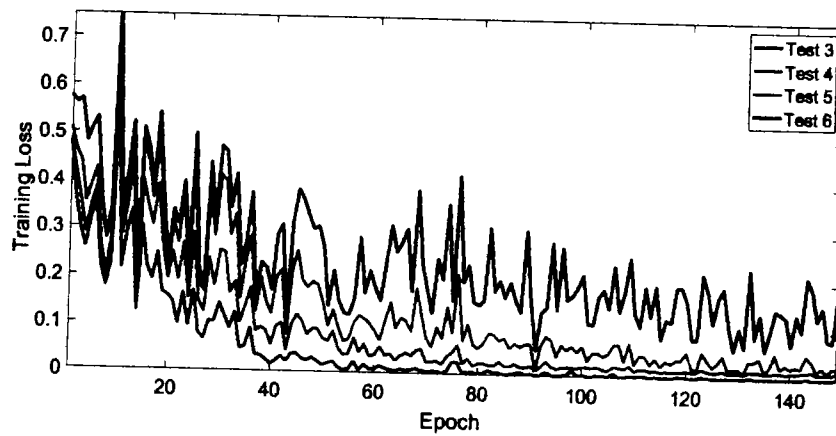
We additionally performed four experiments of the proposed methods by varying encoding and decoding parameters. Parametric details for encoding and decoding are tabulated in Table. 5.4 and Table. 5.5 respectively. The number of filters are reduced from Models 3 to 6. A number of bias parameters are also included in the total number of parameters. Training details are summarized in Table. 5.6. Training curves of experiments are shown in Figure. 5.4.

Table 5.6: SegNet training comparison with proposed technique using UNT dataset.

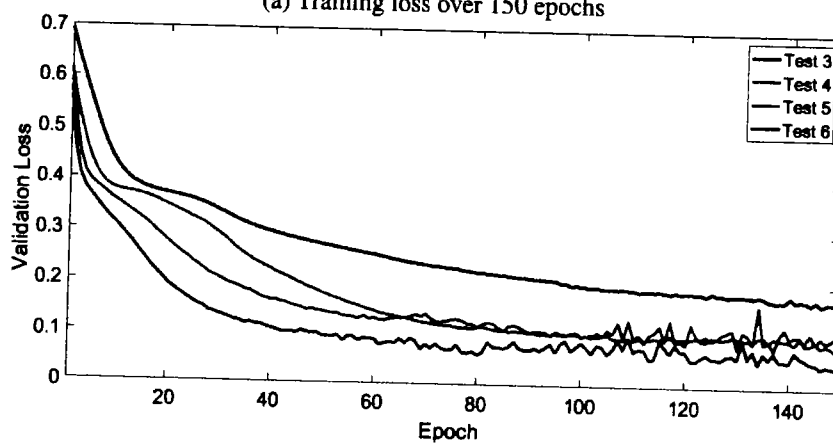
SegNet training details	Model 3	Model 4	Model 5	Model 6
Training epochs	150	150	150	150
Batch size	1	1	1	1
Gamma	0.3	0.3	0.3	0.3
Learning rate	0.01	0.01	0.01	0.01
Normalization	Pixel	Pixel	Pixel	Pixel
Avg. GPU memory utilization	96.3%	71.0%	42.7%	28.9%
Avg. GPU compute utilization	93%-94%	89%-90%	81%-82%	78%-79%
Avg. GPU temperature	65%-70%	60%-65%	60%-64%	60%-65%
Training time on UNT dataset	35 hrs	24 hrs	17 hrs	14 hrs

In a further experiment, training is performed with Arabic Kufi font from Al-Arabiya with the proposed learning architecture and with the best-performing model on the UNT dataset. Training details are summarized in Table. 5.3 (in the fourth column). It took 34 hours to train. Figure. 5.5 shows SegNet learning curves of training loss, validation accuracy, and validation loss.

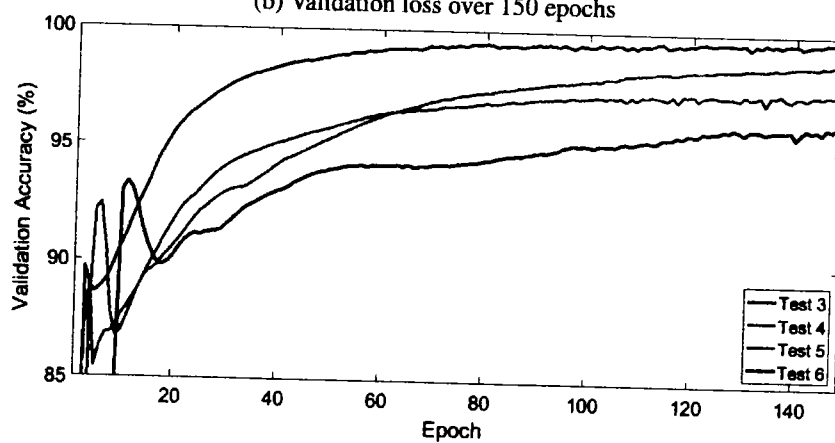
Training is also performed with Arabic Kufi font from AL-ENT with the proposed learning architecture. Training details are summarized in Table. 5.3 (in the fifth column). It took 58 hours to train. Figure. 5.6 shows SegNet learning curves of training loss, validation accuracy, and validation loss.



(a) Training loss over 150 epochs



(b) Validation loss over 150 epochs



(c) Validation accuracy over 150 epochs

Figure 5.4: Graphical comparison representation of SegNet training loss, validation loss, and validation accuracy curves with UNT.

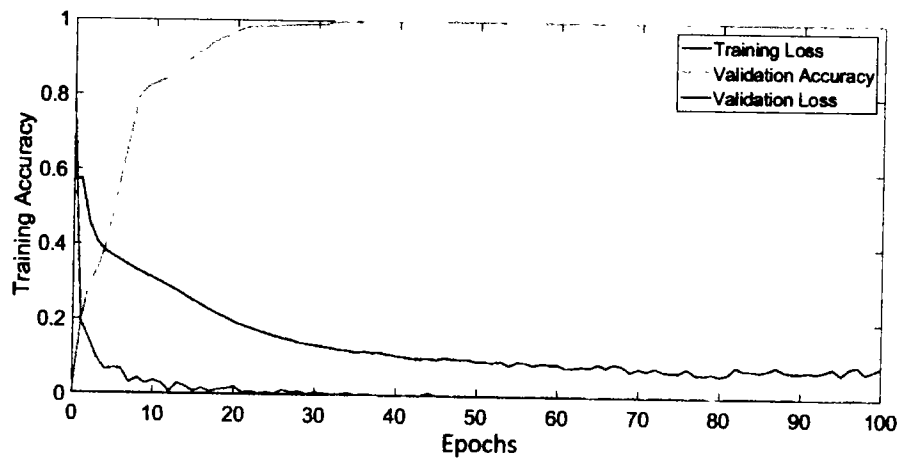


Figure 5.5: Graphical representation of training curves with Kufi font Al-Arabiya dataset.

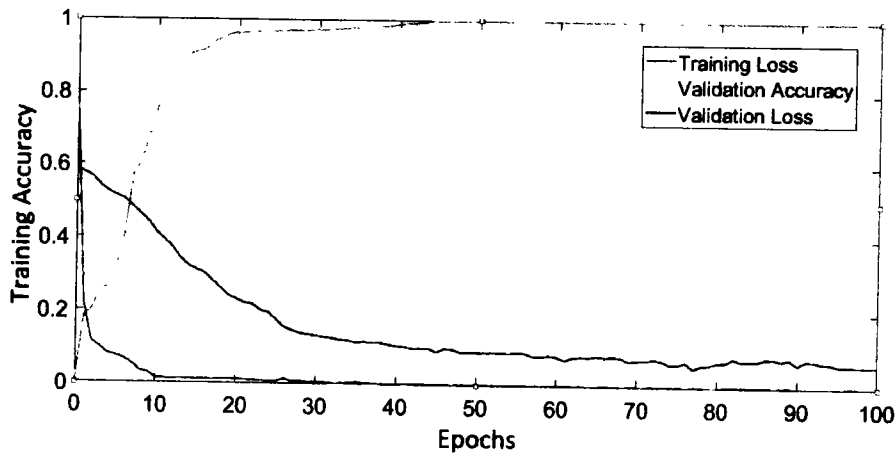


Figure 5.6: Graphical representation of training curves with Kufi font Al-ENT dataset.

### 5.3.3 Result Analysis

The final validation accuracy of the Urdu SegNet training model without the characters grouping concept is 97.06% with a validation loss of 0.0203 and a training loss is 0.0106. The validation accuracy of the proposed training model by considering the characters grouping concept is 99.98% with a validation loss of 0.0013 and a training loss of 0.00058. The graphical representation of training models (blue and green curves of validation accuracy, training, and validation loss) are shown in Figure. 5.3.

Final validation accuracies, training, and validation losses of further training experiments of the proposed model and a variable number of filters (Models 3 to 6) are tabulated in Table. 5.7.

Table 5.7: SegNet model comparison: final validation, training loss, and validation accuracy of the proposed technique with UNT dataset.

SegNet results	Validation loss	Training loss	Validation accuracy
SegNet Model 3	0.0424	0.00414	99.68%
SegNet Model 4	0.0738	0.01732	97.50%
SegNet Model 5	0.1084	0.02448	98.67%
SegNet Model 6	0.1672	0.17910	95.90%

Table 5.8: Detailed Arabic characters segmentation performance of the proposed method with Al-Arabiya dataset

Class label	F-Score	Precision	Recall	Class label	F-Score	Precision	Recall
ا	0.92469	0.91381	0.93585	ي	0.97506	0.96506	0.98528
ب	0.97155	0.96765	0.97549	ج	0.97931	0.97699	0.98165
ت	0.93892	0.92795	0.95017	د	0.97745	0.97890	0.97602
ث	0.93223	0.92186	0.94285	ذ	0.97609	0.97959	0.97263
ح	0.92981	0.89489	0.96757	ر	0.97075	0.95547	0.98654
خ	0.96070	0.94445	0.97753	ز	0.96359	0.94138	0.98689
د	0.94993	0.93646	0.96380	س	0.94742	0.91413	0.98324
ذ	0.94598	0.92709	0.96567	ش	0.95941	0.95149	0.96748
ر	0.93384	0.91754	0.95075	ص	0.93306	0.91257	0.95450
ز	0.95644	0.94276	0.97054	ض	0.95948	0.94840	0.97083
س	0.93564	0.91679	0.95530	ط	0.94072	0.90254	0.98229
ش	0.95191	0.93655	0.96779	ظ	0.96273	0.92814	1
ص	0.92486	0.90162	0.94934	ف	0.97563	0.95243	1
ض	0.94462	0.92119	0.96929	ق	0.95794	0.91929	1
ط	0.93279	0.91098	0.95568	ك	0.96022	0.92350	1
ظ	0.94625	0.91959	0.97452	ل	0.93727	0.88196	1
ف	0.93674	0.93340	0.94011	م	0.96507	0.93251	1
ق	0.93300	0.89585	0.97337	ن	0.93964	0.88616	1
ك	0.88355	0.83223	0.94162	هـ	0.99230	0.98473	1
ل	0.94205	0.91751	0.96794	و	0.98657	0.97351	1
م	0.97272	0.96255	0.98311	ز	0.86173	0.81830	0.91004
Average					0.9493	0.9270	0.9723

Training improvement in the proposed segmentation model is due to minimizing inter-character correlation at the pixel level by the grouping process. This restrains the unfair distribution of data



Table 5.9: Detailed Arabic characters segmentation performance of the proposed method with Al-ENT dataset

Class label	F-Score	Precision	Recall	Class label	F-Score	Precision	Recall
ا	0.95645	0.95325	0.95968	ي	0.99186	0.98982	0.99391
ب	0.98084	0.98479	0.97693	ك	0.98213	0.97634	0.98799
ت	0.95466	0.94598	0.96351	ل	0.97331	0.97245	0.97418
ث	0.93248	0.91037	0.95571	م	0.97699	0.97483	0.97916
ج	0.96612	0.94043	0.99327	ن	0.96489	0.95002	0.98024
ح	0.98626	0.98983	0.98272	هـ	0.97941	0.97343	0.98546
خ	0.97686	0.97253	0.98124	و	0.98333	0.97112	0.99587
د	0.93409	0.91037	0.95909	ز	0.98180	0.98276	0.99385
ذ	0.91471	0.90614	0.92345	س	0.97621	0.96254	0.99028
ر	0.96152	0.94309	0.98070	ش	0.98981	0.98952	0.99011
ز	0.94988	0.94922	0.95056	ص	0.94295	0.90037	0.98976
س	0.98136	0.97317	0.98969	ط	0.94869	0.90239	1
ش	0.96315	0.96856	0.95781	ظ	0.95961	0.92237	1
ص	0.98003	0.97681	0.98328	ع	0.97088	0.94342	1
ط	0.96998	0.96473	0.97529	ف	0.95813	0.91964	1
ظ	0.97976	0.96034	1	ق	0.94310	0.89234	1
ع	0.94008	0.93776	0.94242	ك	0.98807	0.97644	1
ف	0.95678	0.92563	0.99012	خ	0.94691	0.89918	1
ق	0.91065	0.83674	0.99890	ج	0.99212	0.98438	1
ك	0.97700	0.96466	0.98967	ح	0.99337	0.98684	1
خ	0.99093	0.98634	0.99558	د	0.93050	0.88025	0.98685
Average					0.96534	0.94883	0.98326

having low-frequency representations and low data diversity. The imbalance in data distribution inherits biases in individual datasets. On that basis, we extend the Arabic models by considering the character’s family. The final validation accuracy of the proposed Arabic SegNet training model using the Al-Arabiya dataset with 35 numbers classes is 99.93% with a validation loss of 0.085 and training loss of 0.00017. The final validation accuracy of the proposed Arabic SegNet training model using Al-ENT data with 35 numbers classes is 99.98% with a validation loss of 0.0011 and training loss of 0.00030.

Testing on the Arabic dataset with the proposed Arabic model results in a weighted average f-score of 94.93% on character-level and 91.01% on pixel-level segmentation within the text using the Al-Arabiya dataset. Testing on the Arabic dataset with the proposed Arabic model results in a

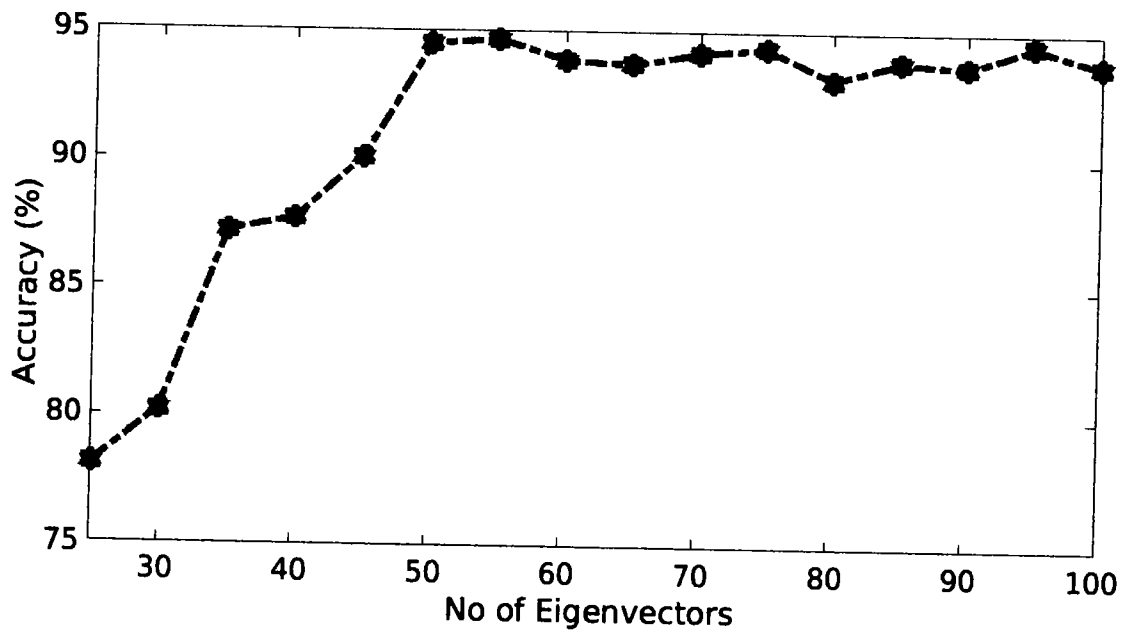
Table 5.10: Detailed Urdu characters segmentation performance of the proposed method with UNT dataset

Class label	F-Score	Precision	Recall	Class label	F-Score	Precision	Recall
ا	0.93062	0.92774	0.93353	ح	0.90633	0.85123	0.96906
ب	0.95808	0.94817	0.96821	ج	0.94704	0.91183	0.98508
پ	0.93929	0.92868	0.95015	چ	0.94846	0.93240	0.96510
ت	0.92561	0.92084	0.93043	ٹ	0.93601	0.92417	0.94817
ث	0.91763	0.90399	0.93169	ڈ	0.91359	0.89049	0.93794
د	0.91948	0.91259	0.92649	ڈ	0.96271	0.95902	0.96644
ذ	0.98263	0.97800	0.98731	ڊ	0.92144	0.90944	0.93378
ڊ	0.98688	0.98431	0.98947	ڙ	0.93636	0.92490	0.94811
ڙ	0.97179	0.96910	0.97450	ں	0.97499	0.96489	0.98532
ں	0.97859	0.97181	0.98547	۰	0.91299	0.89337	0.93350
۰	0.92078	0.89963	0.94296	۱	0.98919	0.98900	0.98939
۱	0.91592	0.91145	0.92044	۲	0.93089	0.91369	0.94875
۲	0.90659	0.90136	0.91189	۳	0.95992	0.94111	0.97950
۳	0.96739	0.94869	0.98686	۴	0.92922	0.86780	1
۴	0.94863	0.93579	0.96183	۵	0.90699	0.84389	0.98031
۵	0.94419	0.92549	0.96368	۶	0.93753	0.88241	1
۶	0.96369	0.95145	0.97625	۷	0.93278	0.87403	1
۷	0.97294	0.95853	0.98780	۸	0.94233	0.89096	1
۸	0.95345	0.94622	0.96081	۹	0.96443	0.93132	1
۹	0.97133	0.96351	0.97929	۱۰	0.94147	0.88942	1
۱۰	0.96139	0.95513	0.96775	۱۱	0.93592	0.87956	1
۱۱	0.95346	0.92401	0.98486	۱۲	0.95054	0.90575	1
۱۲	0.94148	0.91076	0.97435	۱۳	0.94212	0.89059	1
۱۳	0.93704	0.90239	0.97446	۱۴	0.84897	0.80241	0.90127
Average					0.9425	0.9197	0.9675

weighted average f-score of 96.53% on the character level and 89.94% on pixel-level segmentation within the text using the ENT dataset. Testing on the Urdu dataset with the proposed model results weighted average f-score of 94.25% on character level and 89.10% on pixel-level segmentation within the text. Table. 5.8, Table. 5.9, and Table. 5.10 show character-wise detailed performance for Arabic and Urdu letters. Numerics are single-shaped, unjoined independent characters without diacritics. These binarized characters' shapes are more uncorrelated so the model can recognize them with high accuracy.

## 5.4 KL-Transformation Classification Model Experiment

Almost nine thousand Urdu letter candidates are processed by KL-Transformation (PCA) with a variety of shapes collected from the Urdu segmentation output for Urdu classification. The 50 most independent vectors indicated by high eigenvalues are selected. Improvement in test data by increasing vectors from 50 to 100 does not generate much impact on the result with increasing cost to multiplications and distance calculations. Similarly decreasing the vector to 40 produces decreasing results in matching up to 6.81%. Figure. 5.7 shows a graphical representation of the hypothesis. We have selected 50 appropriate number vectors. The minimum Euclidean distance with each of the candidates is the recognized outcome. By considering the same Urdu classification strategy, around twenty-five hundred characters of Arabic font are processed by PCA to construct Arabic classification models. 45 and 56 classes are used for Arabic and Urdu classification models respectively. For textual ground truths, the number of labels is double the number of letters (excluding ا، ب، ت، ث، ج، ح، خ، د، ذ، ر، ز، س، ش، ص، ط، ظ، و، ے، numerics, and extra characters like a full stop, etc.), four possible consecutive characters for Urdu and one consecutive character for Arabic are included.



**Figure 5.7: Graphical representation of KL-Transformation test on Urdu characters.**

5.5 LeNet-based Classification Model Experiment

The LeNet model is used and implemented on Matlab. Following the concatenation procedure, all candidates are stored in a 32x32 pixel buffer. As illustrated in the design Figure 3.1, these candidates are trained with their corresponding ground truth. The LeNet-5 model used with minor modifications in the final layer consists of more numbers of classes. As for KL transform, 45 and 56 classes are used for Arabic and Urdu classification models respectively. For textual ground truths, the number of labels is double the number of letters (excluding ا ب ج د ه و ز ح ط ق ك خ غ ف ي, numerics, and extra characters like a full stop, etc.), four possible consecutive characters for Urdu and one consecutive character for Arabic are included. The Urdu network is trained for 2736 iterations using learning rates of 0.01 and a batch size of 128. We select 25000 images (32x32) collected from the segmented outputs. Training details for the LeNet models are tabulated in Table. 5.11. The training curve for the model is shown in Figure. 5.8. The black dotted curves show the validation accuracy and loss along with the blue and orange lines for training accuracy and loss respectively. The X-axis shows the number of epochs whereas the Y-axis shows accuracy and Loss. The training process is stopped when the improvement in accuracy becomes stable. The final validation accuracy is 94.60%.

Similarly, for the Arabic classification model, all the candidates are placed in a 32x32 pixels buffer dimension followed by the concatenation procedure. For this 45 classes are used. The network is trained for 3920 iterations using learning rates of 0.01 and batch size of 32 using the Al-Arabiya dataset. We select 13700 images (32x32) collected from the segmented outputs. Training details for the LeNet models are tabulated in Table. 5.12. The training curve for the model using the Al-Arabiya dataset is shown in Figure. 5.9. The final validation accuracy is 95.70%. Similarly, we select 12615 images (32x32) for the Al-ENT classification model. Training details for the LeNet models are tabulated in Table. 5.13. The training curve for the model using the Al-ENT dataset is shown in Figure. 5.10. The final validation accuracy is 94.71%.

Table 5.11: LeNet training using UNT dataset.

LeNet parameters	Values
Number of Training Characters	25000
Number of Validation Characters	3000
Batch Size	128
Learning Rate	0.01
Training Time	1m 30sec

Table 5.12: LeNet training using Al-Arabiya dataset.

LeNet Parameters	Values
Number of Training Characters	13700
Number of Validation Characters	1000
Batch Size	32
Learning Rate	0.01
Training Time	1m 24sec

Table 5.13: LeNet training using Al-ENT dataset.

LeNet Parameters	Values
Number of Training Characters	12615
Number of Validation Characters	1000
Batch Size	32
Learning Rate	0.01
Training Time	1m 19sec

The KL-transform-based Urdu classification model results in 94.51% on test data whereas the Arabic classification model results in 96.74% and 95.12% on testing data of Al-Arabiya and Al-ENT respectively.

LeNet is tested on testing part of segmented data from segmentation models. The final validation accuracy is 94.77% and the validation loss is 0.18%. for the Urdu classification model. The final validation accuracy is 96.89% and the validation loss is 0.14% for the Arabic classification model using the Al-Arabiya dataset. The final validation accuracy is 95.70% and the validation loss is 0.14% for the Arabic classification model using the Al-ENT dataset.

5.6 Experimental Study and its Analysis

The comprehensive summary in relation to datasets character and pixel level accuracy, KL-transform, and LeNet results are detailed in Table. 5.14. Model evaluation performed on Urdu UPTI data with 1600 selected text images, declares 98.36% and 98.47% at the textual level with KL-transformation and LeNet models respectively. The confidence matrix is tabulated in Table. 5.16 for character detection. Accumulative character level accuracy is 98.57%. Results comparison on the UPTI dataset

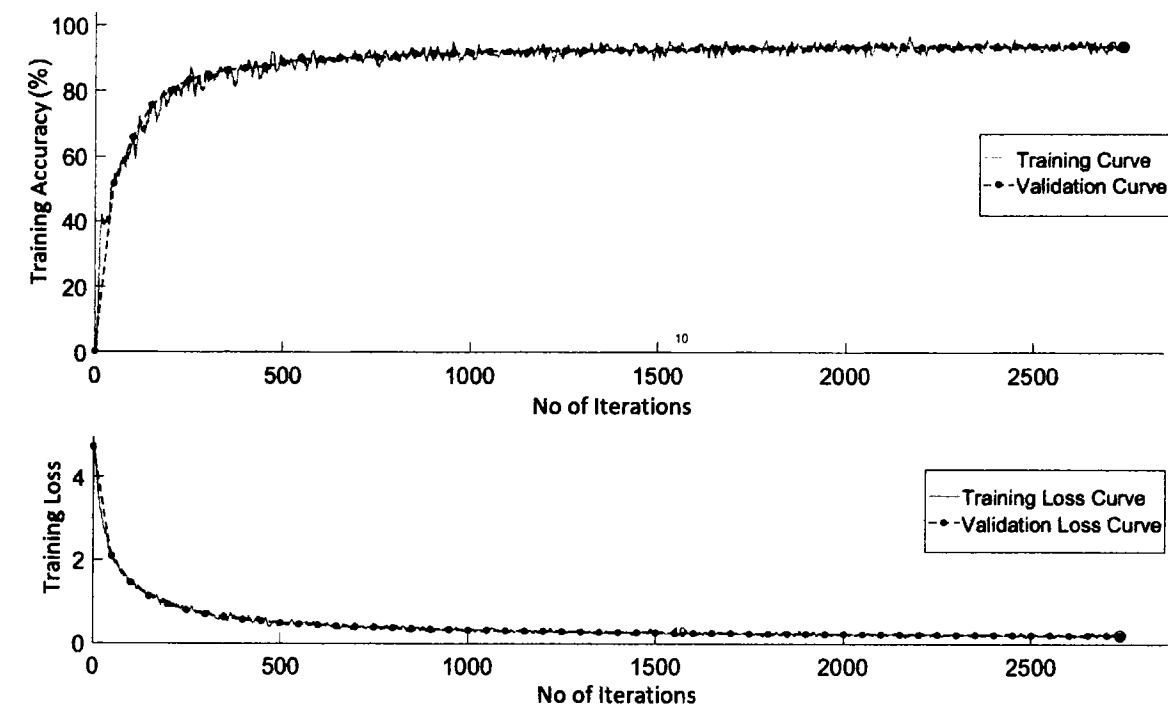


Figure 5.8: Graphical representation of LeNet using UNT dataset, training and validation accuracy, and training and validation loss curves.

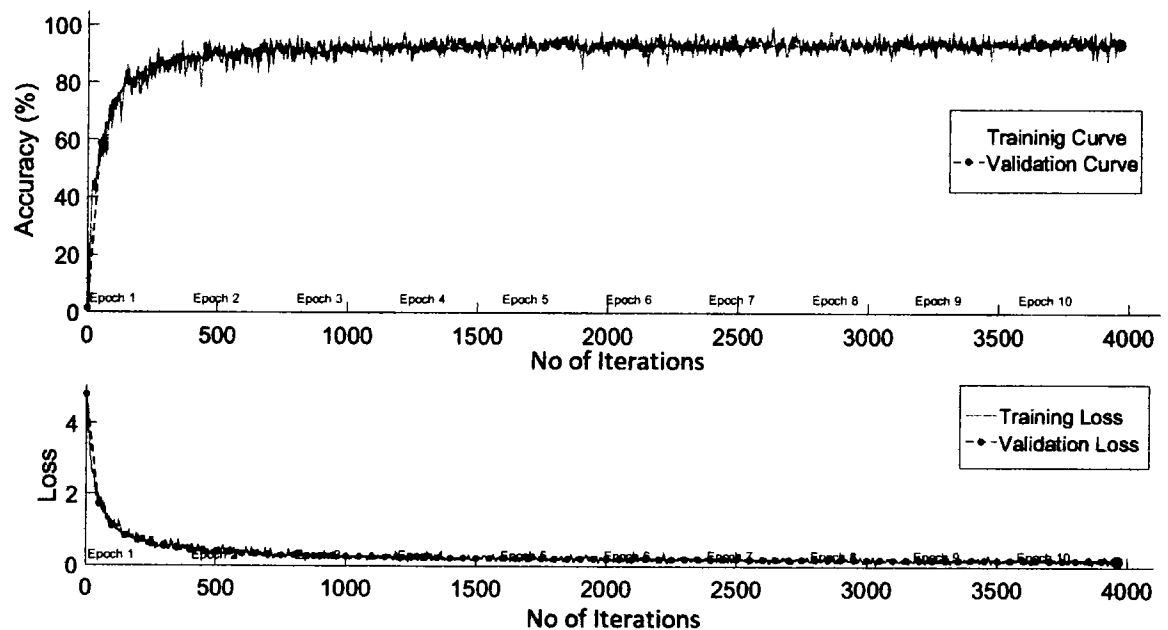


Figure 5.9: Graphical representation of LeNet using Al-Arabiya dataset, training and validation accuracy, and training and validation loss curves.

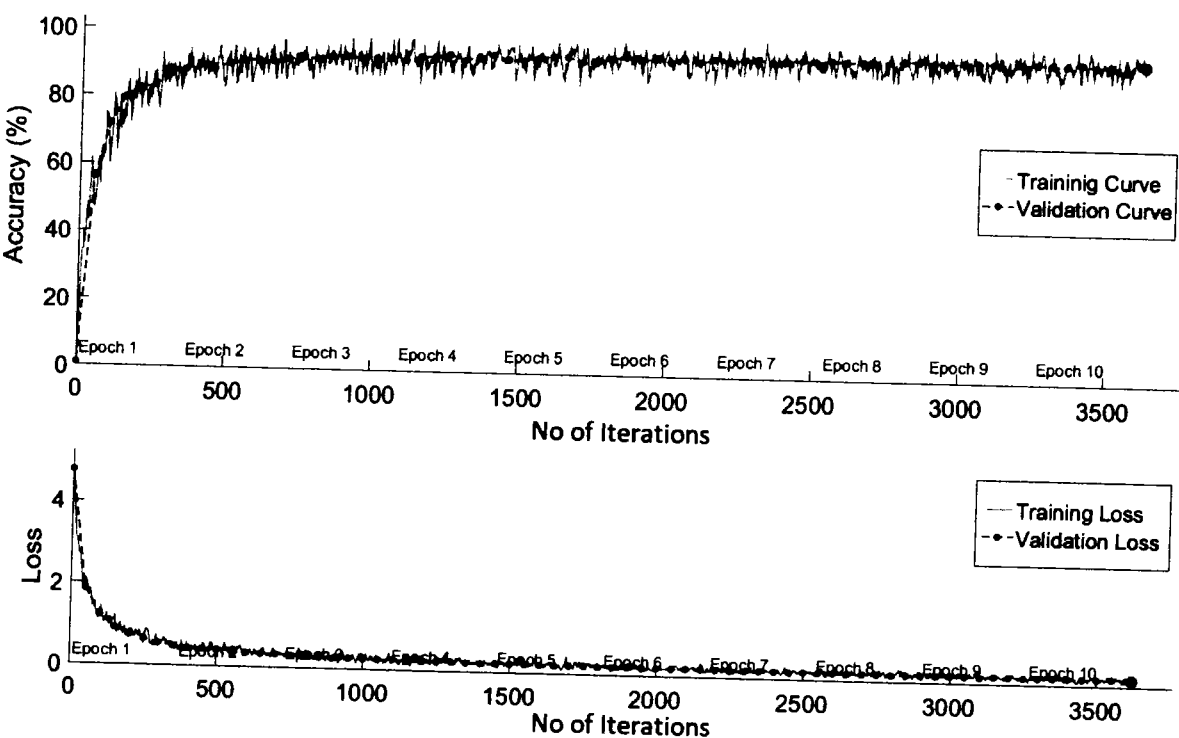


Figure 5.10: Graphical representation of LeNet using AI-ENT dataset, training and validation accuracy, and training and validation loss curves.

is summarized in Table. 5.15.

Table 5.14: Customized Dataset Experimental Results Summary of Segmentation and Classification Models.

Custom Datasets	Character Level Accuracy	Training with LeNet	Training with KL-Transform	Text recognition result with LeNet	Text recognition result with KL-Transform
Al-Arabiya News Ticker	94.93%	96.89%	96.74%	94.40%	94.34%
Al-Ekhbariya News Ticker	96.53%	95.70%	95.12%	96.26%	96.10%
Urdu News Ticker	94.25%	94.77%	94.51%	94.08%	94.00%

5.7 Summary

In this chapter, experiments with a novel cursive scripting language segmentation model and the detailed result analysis of the proposed technique are presented for effective text recognition. Karhunen-Loève transformation and LeNet-based Classification models and Words Code Generator technique are analyzed with experimental results. Various experiments are conducted on

Table 5.15: Performance comparison with UPTI Nastaleeq font dataset.

Recent works with UPTI dataset	Accuracy in percentage
Hassan et al. [23]	86.40%
Ahmed et al. [22]	89.00%
Hassan et al. [23]	94.85%
Naz et al. [24]	94.97%
Naz et al. [10]	96.40%
Naseer et al. [50]	97.07%
Naz et al. [25]	98.00%
Naz et al. [28]	98.12%
This Study with KL-Tran	98.36%
This Study with LeNet	98.47%

ticker images to prove this fact. The human visual assessment shows that the proposed method in two steps for recognition is effective for concentrated learning and high-performance results. Evaluations are provided to compare the proposed technique with the benchmarked dataset. The quantitative analysis of the experimental results verifies that the proposed recognition model exhibits the best performance in terms of explicit character recognition.



Table 5.16: Performance of the suggested approach using the UNT dataset for Urdu characters detection

[illegible]

## Chapter 6

# Conclusions and Future Work

### 6.1 Conclusions

Text in videos contains rich semantic information that can be used to create a variety of informative applications. The robust recognition of textual information from video frames is one of the core features of such systems, and it is the primary focus of this study. Our research was focused on the development of algorithms for recognizing Urdu caption text in video frames. We explored the most advanced deep learning-based algorithms for caption text recognition and created techniques with high performance using conventional evaluation metrics.

The study proposes highly accurate Urdu and Arabic scripting ticker segmentation and text recognition models. Arabic tickers are collected from the Al-Arabiya and Al-Ekhbaria news channels whereas Urdu tickers are gathered from several popular Urdu Pakistani news channels. We investigate the explicit method of character-wise text segmentation technique with Kufi and Nastaleeq fonts. The proposed learning architecture incorporates the grouping of homogeneous shaped prime component character classes to improve the performance of character classes that have low symbol frequencies. This is because the character's prime components and diacritic marks are considered separately for the learning model.

The investigated techniques included FCN and SegNet models trained on custom-built datasets. Text line images extracted from video frames are fed into the network, along with the ground truth transcription, to predict the character pixels semantically. Novel news ticker datasets, which provide accurate character-level labeling are used for training purposes.

The experimental study of the segmentation model was carried out on 5945 Urdu news tickers (3000 for training, 1000 for validation, and 945 for testing), 1189 Arabic news tickers from the Al-Arabiya dataset (715 for training, 200 for validation, and 274 for testing), and 2100 news tickers from the Al-Ekhbaria dataset (1170 for training, 130 for validation, and 800 for testing for another Arabic segmentation model). Overall testing results on the Urdu dataset with the proposed model results weighted average f-score of 94.25% on character level and 89.10% on pixel-level segmentation within the text. Results on the Al-Arabiya dataset with the proposed Arabic model in a weighted average f-score is 94.93% on character-level and 91.01% on pixel-level segmentation within the text. Testing on the Al-Ekhbaria dataset with the proposed Arabic model results in a weighted average f-score of 96.53% on the character level and 89.94% on pixel-level segmentation within the text. The textual formation process is performed in two steps. Words codes are the outputs of this process to be stored for the search operation. Compared to other techniques, we trained our model on the custom-built dataset and test on the UPTI benchmark dataset. The proposed model achieves a textual recognition rate of 98.47% on the UPTI dataset, outperforming state-of-the-art methods.

We recall the key aspects of this research in the following.

- Introduction of novel Arabic and Urdu news ticker datasets consisting of character-level as well as character component-level labeled images with Kufi and Nastaleeq fonts respectively.
- Addresses the challenge of a huge number of ligature classes and proposes an explicit approach to character-wise segmentation.
- Extends FCN and SegNet architectures for effective character-wise cursive text recognition that provides significant improvements over traditional feature learning methods.
- Proposes model by the grouping of similarly shaped character classes for more effective segmentation.
- Experimental evaluation of the proposed technique on the UPTI benchmark dataset and custom-built dataset with cursive Arabic and Urdu script.

## 6.2 Future Work

The presented research proposed Urdu cursive text recognition techniques as a case study, The findings can be applied to various cursive scripts as well. In future research on this topic, the work

can be extended to the improved detector and recognizer to process more difficult data in terms of image resolution, noise, and other distortions, etc.

Recognition performance can be improved further by including a language model as post-recognition processing. The language model can efficiently utilize contextual information in a word sequence and anticipate the most likely word, hence correcting recognition problems. The concept of the dictionary at the syntax correction stage can improve performance levels. In the current approach, the textual image font normalization process can help to further increase robustness. Other learning models like UNet etc. can be used for performance evaluation. In terms of application development, the optimized system can work in real time, allowing users to establish keyword-based alerts in live video broadcasts. Similarly, transcriptions of textual data in videos can be further processed to create automatic analysis and content-mining features. With visual content, the use of audio can also be applied to strengthen text-based applications.

Moreover, it is hard to segment text with smaller font sizes and more de-shaped symbols. Datasets consisting of more balanced data distribution with different learning architectural parameters can be useful for more robust models. The suggested concept of the storage and retrieval process can be investigated for real-time news ticker recognition systems.

The findings of this work are expected to be valuable for the pattern classification community for researchers focusing on cursive text recognition in particular.

## Bibliography

- [1] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: Present contributions and future directions," *IEEE Access* 6, pp. 46 019–46 046, 2018.
- [2] S. U. Khan, I. U. Haq, Z. A. Khan, N. Khan, M. Y. Lee, and S. W. Baik, "Atrous convolutions and residual gru based architecture for matching power demand with supply," *Sensors*, pp. 1–18, 2021.
- [3] S. U. Khan, T. Hussain, A. Ullah, and S. W. Baik, "Deep-reid: deep features and autoencoder assisted image patching strategy for person re-identification in smart cities surveillance," *Multimedia Tools and Applications*, pp. 1–22, 2021.
- [4] S. Naz, K. Hayat, M. I. Razzak, M. W. Anwar, S. A. Madani, and S. U. Khan, "The optical character recognition of Urdu-like cursive scripts," *Pattern Recognition*, vol. 47, pp. 1229–1248, 2014.
- [5] S. U. Rehman, B. U. Tayyab, M. F. Naeem, A. U. Hasan, and F. Shafait, "A multi-faceted ocr framework for artificial urdu news ticker text recognition," *13th IAPR International Workshop on Document Analysis Systems*, pp. 211–216, 2018.
- [6] Q. U. A. Akram, S. Hussain, F. Adeeba, S. U. Rehman, and M. Saeed, "Framework of urdu nastalique optical character recognition system," *Proceedings of Conference on Language and Technology*, 2014.
- [7] S. T. Javed and S. Hussain, "Segmentation based urdu nastalique ocr," *Iberoamerican Congress on Pattern Recognition, Springer, Cuba*, pp. 41–49,, 2013.
- [8] S. Hussain, S. Ali, and Q. U. A. Akram, "Nastalique segmentation-based approach for urdu ocr," *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 357–374,, 2015.

- [9] N. Rahal, M. Tounsi, A. Hussain, and A. M. Alimi, "Deep sparse auto-encoder features learning for arabic text recognition," *IEEE Access*, pp. 18 569–18 584, 2021.
- [10] S. Naz, A. I. Umar, R. Ahmed, S. B. Ahmed, S. H. Shirazi, I. Siddiqi, and M. I. Razzak, "Of-line cursive nastaliq script recognition using multidimensional recurrent neural networks," *Neurocomputing*, pp. 228–241, 2016.
- [11] S. B. Ahmed, I. A. Hameed, S. Naz, M. Razzak, and R. Yusof, "Evaluation of handwritten urdu text by integration of mnist dataset learning experience," *IEEE Access*, vol. 7, pp. 153 566–153 578, 2019.
- [12] J. Gan, W. Wang, and K. Lu, "In-air handwritten chinese text recognition with temporal convolutional recurrent network," *Pattern Recognition, Elsevier*, 2020.
- [13] A. Mirza and I. Siddiqi, "Impact of psyed saqib raza rizvire-processing on recognition of cursive video text," *Pattern Recognition and Image Analysis*, pp. 565–576, 2019.
- [14] A. Mirza, O. Zeshan, M. Atif, and I. Siddiqi, "Detection and recognition of cursive text from video frames," *Journal on Image and Video Processing (EURASIP)*, pp. 1–19, 2020.
- [15] N. Sabbour and F. Shafait, "A segmentation free approach to arabic and urdu ocr," *Proceedings of The International Society for Optical Engineering (SPIE)*, 2013.
- [16] Q. U. A. Akram, S. Hussain, A. Niazi, U. Anjum, and F. Irfan, "Adapting tesseract for complex scripts: An example for urdu nastalique," *Workshop on Document Analysis Systems (DAS)*, pp. 191–195, 2014.
- [17] S. S. R. Rizvi, A. Sagheer, K. Adnan, and A. Muhammad, "Optical character recognition system for nastalique urdu-like script languages using supervised learning," *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 1 953 004(1)–1 953 004(35), 2019.
- [18] A. U. Rehman and S. Hussain, "Large scale font independent urdu text recognition system," *Pattern Recognition, Elsevier*, 2020.
- [19] I. Ahmad, X. Wang, Y. Mao, G. Liu, H. Ahmad, and R. Ullah, "Ligature based urdu nastaleeq sentence recognition using gated bidirectional long short term memory," *Cluster Computing*, pp. 703–714, 2017.
- [20] N. Javed, S. Shabbir, I. Siddiqi, and K. Khurshid, "Classification of urdu ligatures using

- convolutional neural networks - a novel approach," *International Conference on Frontiers of Information Technology (FIT)*, pp. 93–97, 2017.
- [21] G. S. Lehal and A. Rana, "Recognition of nastalique urdu ligatures," *Conference: Proceedings of the 4th International Workshop on Multilingual OCR*, pp. 1–5, 2013.
- [22] S. B. Ahmed, S. Naz, M. I. Razzak, S. F. Rashid, M. Z. Afzal, and T. M. Breuel, "Evaluation of cursive and non-cursive scripts using recurrent neural networks," *Journal Neural Computing and Applications*, pp. 603–613, 2016.
- [23] A. U. Hasan, S. B. Ahmed, S. F. Rashid, F. Shafait, and T. M. Breuel, "Offline printed urdu nastaleeq script recognition with bidirectional lstm networks," *Proceedings of the Twelfth International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1061–1065, 2013.
- [24] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, and S. H. Shirazi, "Urdu nastaliq text recognition system based on multi-dimensional recurrent neural network and statistical features," *Neural Computing and Applications*, pp. 219–231, 2017.
- [25] S. Naz, A. I. Umar, R. Ahmad, M. I. Razzak, S. F. Rashid, and F. Shafiat, "Urdu nastaliq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks," *SpringerPlus*, vol. 5 (1), pp. 1–16, 2016.
- [26] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, "Zoning features and 2dlstm for urdu text-line recognition," *International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, pp. 16–22, 2017.
- [27] Q. U. A. Akram and S. Hussain, "Improving urdu recognition using character-based artistic features of nastalique calligraphy," *IEEE Access*, vol. 7, pp. 8495–8507, 2019.
- [28] S. Naz, A. I. Umar, R. Ahmad, I. Siddiqi, and S. B. Ahmed, "Urdu nastaliq recognition using convolutional recursive deep learning," *Neurocomputing*, pp. 80–87, 2017.
- [29] S. B. Ahmed, S. Naz, M. I. Razzak, and R. Yousaf, "Deep learning based isolated arabic scene character recognition," *Accepted and published in Arabic Script Analysis and Recognition (ASAR)*, pp. 46–51, 2017.
- [30] I. U. Din, I. Siddiqi, S. Khalid, and T. Azam, "Segmentation-free optical character recognition for printed urdu text," *Journal of Image Video Processing*, vol. 62, pp. 1–22, 2017.
- [31] N. Javed, S. Shabbir, I. Siddiqi, and K. Khurshid, "Classification of urdu ligatures using

- convolutional neural networks - a novel approach," *International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan*, pp. 93–97, 2017.
- [32] N. H. Khan, A. Adnan, A. Waheed, M. Zareei, A. Aldosary, and E. M. Mohamed, "Urdu ligature recognition system: an evolutionary approach," *Computers, Materials & Continua*, vol. 66(2), p. 1347–1367, 2021.
- [33] S. B. Ahmed, I. A. Hameed, S. Naz, M. I. Razzak, and R. Yusof, "Evaluation of handwritten urdu text by integration of mnist dataset learning experience," *IEEE Access*, vol. 7, p. 153566–153578, 2019.
- [34] S. Naz, A. I. Umar, R. Ahmad, I. Siddiqi, S. B. Ahmed, M. I. Razzak, and F. Shafait, "Urdu nastaliq recognition using convolutional–recursive deep learning," *Neurocomputing*, vol. 243, p. 80–87, 2017.
- [35] A. Mirza, O. Zeshan, M. Atif, and I. Siddiqi, "Detection and recognition of cursive text from video frames," *Journal on Image and Video Processing (EURASIP)*, vol. 1, p. 1–19, 2020.
- [36] K. Mohammad, A. Qaroush, M. Ayyesh, M. Washha, and A. Alsadeh, "Contour-based character segmentation for printed arabic text with diacritics," *Journal of Electronic Imaging*, vol. 28, p. 43030–43030, 2019.
- [37] H. Osman, K. Zaghw, M. Hazem, and S. Elsehely, "An efficient language-independent multi-font ocr for arabic script," *10th International Conference on Advances in Computing and Information Technology, London, United Kingdom, November*, p. 57–71, 2020.
- [38] A. Qaroush, B. Jaber, K. Mohammad, M. Washaha, E. Maali, and N. Nayef, "An efficient, font independent word and character segmentation algorithm for printed arabic text," *Journal of King Saud University Computer and Information Sciences*, vol. 34(1), pp. 1330–1344, 2022.
- [39] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, "Segmentation-based, omnifont printed arabic character recognition without font identification," *Journal of King Saud University Computer and Information Sciences*, vol. 34(6), p. 3025–3039, 2020.
- [40] S. A. Hussain, S. Zaman, and M. S. Ayub, "A self organizing map based urdu nasakh character recognition," *International Conference Emerging Technology (ICET)*, pp. 267–273, 2009.
- [41] A. Qaroush, A. Awad, M. Modallal, and M. Ziq, "Segmentation-based, omnifont printed



- arabic character recognition without font identification,” *Journal of King Saud University Computer and Information Sciences*, 2020.
- [42] O. H. Z. K. H. M. and E. S. “An efficient language-independent multi-font ocr for arabic script,” *10th International Conference on Advances in Computing and Information Technology, London, United Kingdom*, pp. 57–71, 2020.
- [43] H. Yao, W. E. I. Sha, and L. Jiang, “Two-step enhanced deep learning approach for electromagnetic inverse scattering problem,” *IEEE Antennas and Wireless Propagation Letters*, pp. 2254–2258, 2019.
- [44] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.
- [45] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv*, pp. 1409–1556, 2014.
- [46] V. Badrinarayanana, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *arXiv*, pp. 1–14, 2016.
- [47] S. Naz, A. I. Umar, S. H. Shirazi, S. B. Ahmed, M. I. Razzak, and I. Siddiqi, “Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey,” *Education and Information Technologies (EIT)*, pp. 1225–1241, 2015.
- [48] N. A. Khan, R. J. D. I. Haye, and H. A. Hegt, “A license plate recognition system,” *Proceedings of SPIE: Applications of Digital Image Processing*, pp. 14–24, 1998.
- [49] S. A. Hassan, I. U. Haq, M. Asif, M. B. Ahmad, and M. Tayyab, “An efficient scheme for real-time information storage and retrieval systems: A hybrid approach,” *International Journal of Advanced Computer Science and Applications*, pp. 427–431, 2017.
- [50] A. Naseer and K. Zafar, “Comparative analysis of raw images and meta feature based urdu ocr using cnn and lstm,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, pp. 419–424, 2018.