GHULAM ALI MIRZA
01-284151-002

Detection and Recognition of Artificial Urdu Text in Videos

A thesis submitted to the Department of Computer Science, Faculty of Engineering Sciences, Bahria University, Islamabad, in the partial fulfillment for the requirements of a Doctoral degree in Computer Science

Supervisor: Professor Dr. Imran Siddiqi

Department of Computer Science
Bahria University, Islamabad

August 2021

Abstract

Textual content appearing in videos represents an interesting index for semantic retrieval of videos (from archives), generation of alerts (live streams) as well as high level applications like opinion mining and content summarization. Key components of a textual content based retrieval system include detection (localization) of text regions and recognition of text through Video Optical Character Recognition (V-OCR) systems. While mature detection and recognition systems are available for text in non-cursive scripts, research on cursive scripts (like Urdu) is fairly limited and is marked by many challenges. These include complex and overlapping ligatures, context-dependent shape variations and presence of a large number of dots and diacritics.

This research aims at detection and recognition of artificial (caption) Urdu text appearing in video frames, primarily targeting the local News channels. Leveraging the recent advancements in deep neural networks (DNN), we propose robust techniques to detect and recognize Urdu caption text from frames with bilingual (English & Urdu) textual content, the most common scenario in majority of our News channels. Detection of textual content relies on adapting the deep convolutional neural networks(CNN) based object detectors for text localization. To cater multiple scripts, text detection and script identification are combined in a single end-to-end trainable system. For recognition, we employ an implicit segmentation based analytical technique that relies on a combination of a CNN and recurrent neural network (RNN) with a connectionist temporal classification (CTC) layer. Images of text lines extracted from video frames along with ground truth transcription are fed to the CNN for feature extraction. The extracted feature sequences are then employed by the recurrent part of the network to predict the likely sequence of characters. Finally, the CTC layer converts raw predictions into meaningful Urdu text.

As a part of this research, a comprehensive dataset named as 'UTiV' (Urdu Text in Videos), containing more than 11,000 video frames from various Urdu News channels was collected and labeled and has been made publicly available. The ground truth of each frame comprises information on location of text regions and the corresponding transcription along with other metadata. A comprehensive series of experiments is carried out on the collected dataset to study the impact of different object detectors, models in the convolutional base, pre-processing steps to segment text from background, the type of recurrent unit and so on. The detector reports an overall F-measure of 0.91 while a character recognition rate of 97.63% is realized by the recognition engine. A comparison with the state-of-the-art validates the effectiveness of the proposed detection and recognition techniques. In addition to the development of Urdu caption text detector and recognizer, we also integrated the two modules to develop an indexing and retrieval application. End-to-end retrieval experiments were also carried out and reported a high F-measure reading 0.89. The reported results not only validate the effectiveness and robustness of the proposed techniques but also demonstrate their potential usage in real world applications for end users.

Dedication

To, my beloved father

Mirza Masood Akhtar (Late)

Acknowledgments

I am ever so grateful to Almighty Allah, who has bestowed me with strength and perseverance to attain this milestone in my life.

I would like to extend my gratitude to my supervisor, Professor Dr. Imran Siddiqi, who has been a source of constant motivation and guidance throughout my PhD journey. His patience and dedication have helped me transform myself into a better researcher and fulfil my dream of attaining a PhD degree.

Finally, I would like to thank my family, especially my wife Mrs. Farah Ali and my sons M. Izaan Ali & M. Zohan Ali. Their continuous support, cooperation and love have always been a source of strength for me. I also owe my success to the prayers of my beloved parents and my parents in law. Their prayers and love have guided me throughout this journey.

GHULAM ALI MIRZA
Bahria University Islamabad, Pakistan

2021

"Once a new technology starts rolling, if you're not part of the steamroller, you're part of the road."

Stewart Brand

Contents

Abstract i

1 Introduction 1
- 1.1 Motivation . . . 3
- 1.2 Challenges in Cursive Scripts . . . 6
- 1.3 Problem Statement . . . 6
- 1.4 Research Objectives . . . 8
- 1.5 Research Questions . . . 8
- 1.6 Proposed Techniques . . . 9
- 1.7 Research Contributions . . . 9
- 1.8 Thesis Organization . . . 10

2 Literature Review 12
- 2.1 Introduction . . . 12
- 2.2 Text Detection Methods . . . 13
  - 2.2.1 Text Detection using Unsupervised Techniques . . . 14
  - 2.2.2 Text Detection using Supervised Techniques . . . 28
- 2.3 Text Recognition Methods . . . 36
  - 2.3.1 Document Text Recognition . . . 37
  - 2.3.2 Scene Text Recognition . . . 41
  - 2.3.3 Video Text Recognition . . . 43
  - 2.3.4 Discussion . . . 46
- 2.4 Challenges in Video Text Detection and Recognition . . . 47
- 2.5 Summary . . . 50

3 Data Collection and Labeling 51
- 3.1 Introduction . . . 51
- 3.2 Evaluation Metrics . . . 51
- 3.3 Ground Truth Labeling Tool . . . 54
  - 3.3.1 Labeling of Text Locations . . . 54
  - 3.3.2 Transcription of Text . . . 55
  - 3.3.3 Ground Truth Data Organization . . . 55
- 3.4 Statistics of Labeled Data . . . 56
- 3.5 Synthetic Data Generation . . . 57
- 3.6 Summary . . . 58

4 Detection of Textual Content **61**
4.1 Introduction . . . 61
4.2 Detection of Text using Image Analysis Techniques . . . 61
4.2.1 Gray Scale Filtering . . . 62
4.2.2 Detection of Edges . . . 62
4.2.3 Mean Gradient . . . 62
4.2.4 Binarization . . . 63
4.2.5 Morphological Processing . . . 63
4.2.6 Geometrical Constraints . . . 63
4.2.7 Validation of Text Regions . . . 64
4.3 Detection of Text using Deep Learning Techniques . . . 68
4.3.1 Deep Learning based Object Detectors . . . 68
4.3.2 Adapting Object Detectors for Text Detection . . . 72
4.3.3 Script Identification . . . 74
4.3.4 Hybrid Text Detector & Script Identifier . . . 74
4.4 Experiments and Results . . . 76
4.4.1 Experimental Settings . . . 77
4.4.2 Text Detection Results . . . 78
4.4.3 Script Identification Results . . . 78
4.4.4 Hybrid Text Detection & Script Identification Results . . . 80
4.4.5 Performance Comparison . . . 83
4.5 Summary . . . 84

5 Recognition of Textual Content **86**
5.1 Introduction . . . 86
5.2 Choice of Recognition Unit . . . 86
5.3 Recognition using Analytical Technique . . . 88
5.3.1 Pre-Processing . . . 88
5.3.2 Feature Extraction . . . 92
5.3.3 Sequence Prediction with Recurrent Nets . . . 93
5.3.4 Connectionist Temporal Classification (CTC) Layer . . . 96
5.4 Model Training and Recognition . . . 99
5.5 Experiments and Results . . . 100
5.5.1 Experimental Protocol . . . 100
5.5.2 Recognition Results . . . 101
5.5.3 Performance Comparison . . . 104
5.6 Summary . . . 105

6 Text Detection & Recognition: Application **107**
6.1 Introduction . . . 107
6.2 Video Indexing & Retrieval . . . 107
6.3 Summary . . . 111

7 Conclusion and Future Work **112**
7.1 Conclusion . . . 112
7.2 Future Work . . . 113

A Research Publications **114**

B Edit Distance Example **115**

C Ground Truth Labeling Tool **117**

D Sample Images of Hybrid Text Detector and Script Identifier **119**

E Preliminary Experiments–Recognition using Holistic Technique **122**

F List of Keywords used in Indexing Application **125**

G Recognizer Performance with Different CNN-LSTM Designs **127**

H Awards & Achievements **128**

List of Figures

1.1 A video frame with instances of scene and caption text . . . 2
1.2 Keyword supplied as query by user . . . 3
1.3 Videos retrieved against a query keyword . . . 3
1.4 Visual Search of BBC News (a): Textual content-based search for query keyword 'Pakistan' (b): Person search for 'Donald Trump' (c): Object search for 'JF17 Thunder' (d): Instance search for 'London Bridge' . . . 5
1.5 A sample text line in (a): Nastaliq (b): Naskh script . . . 7
1.6 Example of cursive text line highlighting recognition challenges . . . 7
1.7 Methodology of proposed technique for processing of caption text . . . 9

2.1 Text recognition modalities . . . 13
2.2 Typical steps in Video Optical Character Recognition . . . 14
2.3 Taxonomy of text detection methods . . . 15
2.4 An overview of key processing steps for detection of Urdu caption text by Jamil et al. [44] . . . 17
2.5 Label histogram Analysis for extraction of characters (Image Source [82]) . . . 17
2.6 Edge segmentation example (Image Source [84]) (a): Original image (b): Edge image, (c): Segmentation points (d): Edge segmentation with different colors . . . 18
2.7 Character localization technique proposed in [89] . . . 19
2.8 Application of Intensity and Shape filters for text detection in [92] . . . 20
2.9 CC-based text segmentation method in [94] . . . 21
2.10 Text detection system presented by Kim et al. [100] . . . 22
2.11 Combination of color and spatial information for text detection proposed in [113] 24
2.12 An overview of key steps for text detection in [114] . . . 25
2.13 Color reduction and edge-preserving filtering for text detection in [117] . . . 25
2.14 Clustering based text detection reported in [119] . . . 26
2.15 Structure of polynomial neural network (PNN) employed for text detection in [123] 29
2.16 Illustration of 2D wavelet-decomposition on a video frame (Image Source [71]) . 31
2.17 The 'SegLink' network architecture proposed by Shi et al. [149] . . . 32
2.18 Connectionist Text Proposal Network (CTPN) (Image Source [150]) . . . 33
2.19 Character proposal network presented in [151] . . . 34
2.20 CNN architecture of TextBoxes [153] . . . 34
2.21 Application of stroke width transform (SWT) (Image Source [33]) . . . 35
2.22 Taxonomy of text detection methods . . . 37
2.23 Binary Convolutional Encoder-Decoder network (B-CEDNet) [252] . . . 42
2.24 Architecture of Char-net (Image Source [253]) . . . 42
2.25 Overview of CNN Ensemble employed in [154] . . . 45
2.26 Sample images in 'ALIF' dataset [273] . . . 45

2.27 Low resolution caption text examples 49

3.1 Text regions in an image and the corresponding ground truth image 52
3.2 (a) A text line in a video frame (b) Ground truth transcription of text 53
3.3 Comparison of ground truth and system produced transcriptions to quantify recognition performance 53
3.4 Screen shot of ground truth labeling tool for text data 54
3.5 Interface to enter transcription of text 55
3.6 Screen shot of an XML file containing ground truth information of a frame 56
3.7 (a): Distribution of number of Urdu characters per line. (b): Distribution of Urdu lines per frame 57
3.8 Frequency of Top-30 Urdu characters in the collected data 59
3.9 Sample text lines extracted from video frames 60
3.10 Generation of synthetic text lines 60
3.11 Synthetically generated text lines to resemble caption text 60

4.1 An image with occurrences of Urdu and English text and the corresponding vertical edges in the image 63
4.2 Steps I-III in unsupervised text detection 65
4.3 Steps IV-VI in unsupervised text detection 66
4.4 Sample blocks of (a) Non-text and (b) text regions which are employed to train a classifier 66
4.5 Overview of detection system using hand-crafted features 67
4.6 Bank of Gabor filters with 4 scales and 6 orientations 67
4.7 Summary of R-CNN Family Models 70
4.8 Detection method of YOLO, (Image Source [306]) 71
4.9 Architecture of Single Shot Detector, (Image Source [307]) 71
4.10 Region-based Fully Convolutional Networks (R-FCN) for Object Detection (Image Source [310]) 72
4.11 Overview of adapting object detectors for text detection 73
4.12 Anchor boxes (base size 256×256) at three scales $(1.0, 2.0, 5.0)$ and five aspect ratios $(0.125, 0.1875, 0.25, 0.375, 0.50)$ 73
4.13 Example of script identification 74
4.14 Training loss of various detectors 76
4.15 Computation of precision and recall (a):Ground Truth Bounding Box (b): Detected region is larger than ground truth (c):Detected region is smaller than ground truth (d):Detected region overlaps perfectly with the ground truth 79
4.16 Text detection results on sample images (Faster R-CNN with Inception) 79
4.17 Imperfect Localization of Text Regions 80
4.18 Detection output of hybrid text detection and script identification for different detectors (a): SSD (b): R-FCN (c): Faster RCNN (d): Yolo 82
4.19 Impact of size of training data on text detection performance (Faster R-CNN with Inception) 83
4.20 Impact of video resolution on text detection performance (Faster R-CNN with Inception) 84

5.1 (a):A complete Urdu word (b):Ligatures (c):Main body (primary ligature) (d):Dots and diacritics (secondary ligatures) 87
5.2 An overview of the key processing steps 89

5.3 Identification of polarity of text (a):Original image (b):Gray scale image (c):Text blobs (d):Filled text blobs serving as a mask to extract corresponding blobs from the gray image (e): Final image (Image on the right is inverted while the one on left remains unchanged) . . . 90
5.4 Binarization results on a sample text line (a): Grayscale Image (b):Niblack (c):Otsu's Global Thresholding (d):Feng's Algorithm (e):Sauvola's Algorithm (f):Wolf's Algorithm . . . 92
5.5 Architecture of the convolutional neural network employed for feature extraction 93
5.6 Architectures of (a): Simple RNN (b): GRU (c): LSTM . . . 95
5.7 Architecture of the bidirectional (left-to-right & right-to-left) RNN model with CTC output layer . . . 97
5.8 CTC Decoding Example . . . 98
5.9 RNN output (probabilities) with three time-steps and two characters (given in (a)) along with CTC blank ('-') . . . 99
5.10 (a): All possible alignments of character sequences producing the ground truth text in Figure 5.9 (b): Summary of CTC loss calculation . . . 99
5.11 Training loss for video and synthetically generated text lines . . . 103
5.12 Recognition rates as a function of size of training data . . . 103
5.13 Screen shot of the recognition application developed in C#.NET and Python . . . 105
5.14 Examples of recognition errors . . . 106

6.1 Keyword supplied as query by user . . . 108
6.2 Search Screen of the Retrieval Application . . . 109
6.3 Retrieval Results for a Query Keyword . . . 109

B.1 . . . 116

C.1 A single frame loaded in the labeling Software and key components of the tool . 117
C.2 (a): Ground truth information on bounding box of a text region and the frame (b): Transcription of text with information on text type and text script . . . 118

D.1 Hybrid text detector and script identifier output: Express News . . . 119
D.2 Hybrid text detector and script identifier output: Samaa News . . . 120
D.3 Hybrid text detector and script identifier output: Dunya News . . . 120
D.4 Hybrid text detector and script identifier output: Ary News . . . 121

F.1 List of 100 Urdu keywords for indexing and retrieval application . . . 125
F.2 List of 100 English keywords for indexing and retrieval application . . . 126

G.1 Recognition rates as a function of number of convolutional layers . . . 127
G.2 Recognition rates as a function of number of LSTM stacks and hidden units . . . 127

List of Tables

2.1 An overview of unsupervised text detection methods . . . 27
2.2 Summary of supervised text detection methods . . . 36
2.3 Summary of Text Recognition Methods . . . 47

3.1 Summary of attributes stored for each text line . . . 54
3.2 Statistics of labeled video frames . . . 56

4.1 Training parameters of hybrid text detector & script identifier network . . . 77
4.2 Tuned Parameters of adapted and standard Faster R-CNN models . . . 77
4.3 Data distribution for text detection experiments . . . 77
4.4 Text Detection Results . . . 78
4.5 Script identification confusion matrix . . . 80
4.6 Performance of Script Identification . . . 80
4.7 Performance of hybrid text detector and script identifier . . . 81
4.8 Performance comparison with other techniques . . . 84

5.1 Architectural details of the proposed CNN . . . 93
5.2 Architectural details of the recurrent network . . . 96
5.3 Training parameters of recognition network . . . 100
5.4 Distribution of dataset including synthetically generated text lines . . . 101
5.5 Summary of character recognition rates in different experimental settings . . . 102
5.6 Recognition rates as a function of training data . . . 102
5.7 Results comparison with other recognition techniques . . . 104

6.1 Character Recognition Rate . . . 110
6.2 Results . . . 110

E.1 Summary of pre-trained models employed in our study . . . 123
E.2 Recognition Rates of Analytical and Holistic Techniques . . . 124

Acronyms and Abbreviations

ANN	Artificial Neural Network
B-CEDNet	Binary Convolutional Encoder-Decoder Network
BLSTM	Bidirectional Long Short-Term Memory
CAE	Convolutional Auto Encoder
CBIR	Content Based Image Retrieval
CBIVR	Content Based Image and Video Retrieval
CBVR	Content Based Video Retrieval
CER	Character Error Rate
CERB	Candidate Edge Recombination
CLE	Center of Language Engineering
CNN	Convolutional Neural Network
CPN	Character Proposal Network
CRF	Conditional Random Field
CRR	Character Recognition Rate
CTC	Connectionist Temporal Classification
CTPN	Connectionist Text Proposal Network
DAE	Deep Auto-Encoder
DBN	Dynamic Bayesian Network
DCT	Discrete Cosine Transformation
DNN	Deep Neural Network
DSAN	Double Supervised Network with Attention Mechanism
FCN	Fully Convolutional Network
FCNN	Fuzzy-Clustering Neural Network
FCRN	Fully-Convolutional Regression Network
FPN	Forward Pass Network
FSF	Fourier-statistical features
FT	Fourier transformation
GLCM	Grey Level Co-occurrence Matrix
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HLS	Hue Lightness Saturation
HMM	Hidden Markov Model
HoG	Histogram of Gradient
IAPR	International Association of Pattern Recognition
ICDAR	International Conference on Document Analysis and Recognition
ISPR	Inter Services Public Relations Pakistan
LBP	Local Binary Pattern
LC	Ligature Classes

LER	Line Error Rate
LRR	Line Recognition Rate
LSTM	Long-Short Term Memory
MDLSTM	Multi-Dimensional Long-Short Term Memory
MSER	Maximally Stable Extremal Regions
NB	Naïve Bayes
NMS	Non-maximum Suppression
OCR	Optical Character Recognition
PEMRA	Pakistan Electronic Media Regulatory Authority
ResNet	Residual Neural Network
RLSA	Run Length Smoothing Algorithm
RNN	Recurrent Neural Network
RPN	Region Proposal Network
SSD	Single Shot Multi Box Detector
SWT	Stroke Width Transform
SVM	Support Vector Machine
SVT	The Street View Text Dataset
TSDNN	Two Stream Deep Neural Network
UNHD	Urdu-Nasta'liq Handwritten Dataset
UPTI	Urdu Printed Text Images
V-OCR	Video Optical Character Recognition
VGG	Visual Geometry Group
WER	Word Error Rate
WRR	Word Recognition Rate
YOLO	You Only Look Once

Chapter 1

Introduction

The last decade has witnessed a tremendous increase in the digital multimedia data including videos and images. This growth can be primarily attributed to the large number of low cost optical sensors as well as the enhanced connectivity with increased bandwidth allowing capture and sharing of multimedia data. Hundreds of hours of video is being uploaded every minute on video sharing portals [1] and the proportion of video in World's Internet traffic has grown from 66% in 2014 to 80% in 2019 [2]. Such enormous collections of videos have opened up a whole new world of challenges to develop smart retrieval systems allowing users an efficient and effective retrieval of desired content.

The conventional video retrieval systems rely on matching the queried words with user-assigned annotations and, ignore the rich information in videos that can be exploited for effective indexing and subsequent retrieval. Content-based search systems, on the other hand, may exploit the visual information (objects, buildings, persons etc.), audio (spoken words), textual content (News tickers, anchor names, subtitles, etc) or a combination of these to support smart retrieval. Examples of typical queries to such intelligent systems include retrieving all videos where a particular individual has appeared or all instances where a particular keyword (for example 'Breaking News') has been flashed [3]. Among various search modalities, the focus of our present study lies on the textual content appearing in videos.

Textual content in videos can be categorized into two different classes, scene text and caption text (also known as artificial/graphics text). Scene text is captured through the camera during video recording and may not always be co-related with the content. Examples of scene text include signboards, advertisement banners, building names, text on T-shirts and play cards etc. It may also include handwritten text. Scene text is useful for applications like robot navigation [4, 5] and systems to assist the visually impaired [6, 7, 8]. Caption text or artificial text, on the other hand, is superimposed on videos and, in most cases, is more related with the content. Typical examples of artificial text include News tickers, scorecards and movie credits etc. The correlation of caption text with the actual content makes it more appropriate for indexing and retrieval applications.

An example video frame containing occurrences of scene as well as caption text is illustrated in Figure 1.1.

Figure 1.1: A video frame with instances of scene and caption text

The initial research on smart video retrieval primarily exploited simple image analysis techniques supporting retrieval using attributes like color, texture and shape etc. Subsequent theoretical as well as technological advancements led to more sophisticated systems supporting shot boundary detection, video summarization, semantic video search and video captioning [9]. Typically, features extracted from key frames or regions corresponding to objects in frames are employed for indexing purposes. The recent paradigm shift from hand-engineered to machine learned features and the possibility of end-to-end trainable deep neural networks (DNN) have served to significantly enhance the robustness of smart retrieval systems [10, 11, 12]. Such systems exploit different retrieval modalities including caption text [13], faces [14], spoken keywords [15] objects [16] and other visual cues [17].

From the view point of text-based video indexing and retrieval, textual content in video frames needs to be detected (localized) and recognized. Keywords in the transcription of a video frame can then be extracted and employed for indexing and subsequent retrieval. Keywords refer to the words that are provided as query by the user (Figure 1.2). Once a keyword is provided, the system queries the database that contains videos indexed on keywords. Videos containing instances of the relevant keyword are then provided to the user in the retrieval phase (Figure 1.3).

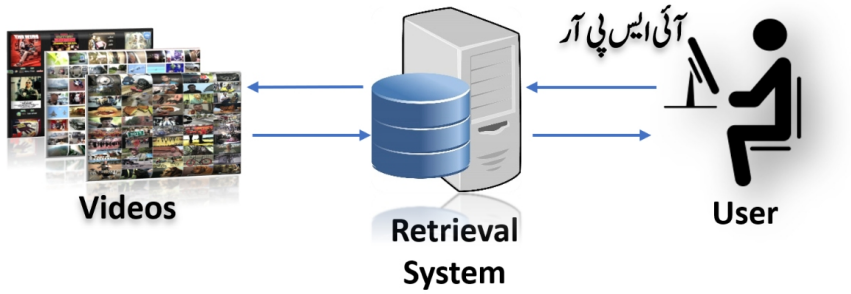


Figure 1.2: Keyword supplied as query by user

Figure 1.3: Videos retrieved against a query keyword

The present study focuses on detection and recognition of Urdu caption text appearing in videos. Once the text is extracted and recognized using a video-OCR, in addition to indexing and retrieval, a number of interesting applications can be developed using the recognized text. Typical examples include summarization of News tickers, generation of alerts on user specified keywords and comparative analysis of same News across multiple News channels etc.

1.1 Motivation

Content based retrieval of images and videos (CBVIR) has been explored by researchers since many years [18]. Many different applications for indexing and retrieval of images and videos have been proposed. In case of image databases, user may provide the query in the form of an image or some attributes of image like color, shape or texture to retrieval all relevant images. For videos, as discussed earlier, the visual content, audio or textual information in the video could be used as an index.

QBIC [19] developed by IBM, is known to be the first commercial Content Based Image Retrieval (CBIR) system. In this system, user may retrieve images based on one or more features like color, shape or texture. A similar image retrieval system has been developed at the Massachusetts Institute of Technology called the Photobook [20]. Other preliminary retrieval systems include IMatch [21] and VisualSEEK/WebSEEK [22] developed at Columbia University, FIRE (Flexible Image Retrieval Engine) developed at RWTH Aachen University, MUVIS developed at Tampere University of Technology and ALIPR developed by researchers of Penn State University. Similarly, an interesting retrieval system has been developed at University of Maryland (UMD) which is based on text and speech recognition for indexing videos [23]. Another significant contribution is the Informedia project at the Carnegie Mellon University (CMU). The Informedia-I supports indexing and retrieval using speech, image and natural language processing while Informedia-II incorporates other interesting features like video summarization and production of collages with other features.

As compared to CBIR, a wider spectrum of applications are offered for content based intelligent retrieval of videos. A number of government agencies in the United States (US) and the National Institute of Standards & Technology (NIST) have been able to reflect the importance of CBVIR since 2003 by regularly sponsoring the Text Retrieval Conference Video Retrieval Evaluation (TRECVID) [24]. TRECVID offers a huge collection of videos and various algorithms for video retrieval are submitted to the company for evaluation and comparisons.

Among relatively recent works on this problem [25, 26], a discussion on the latest techniques and challenges in CBVR systems with respect to text, audio and visual content is presented in [18]. Different types of features, classifiers, their combinations and various querying methods are discussed and critically analyzed. CBVR systems have paved way for a large number of real world applications. Few examples include smart retrieval of video lectures using speech recognition and V-OCR [27], video search using spoken words and visual cues like objects and places [28] and the Ontological-PENN system [29] that recognizes semantic concepts in videos etc. A very effective retrieval system has been developed by the Visual Geometry Group at University of Oxford to search BBC content using objects, people and text as queries. The system has been developed on videos corresponding to more than five years of News broadcasts from six different BBC channels. A screen shot of a retrieval session on BBC archives using different types of queries is illustrated in Figure 1.4.

It can be observed from the above discussion that while CBVIR is a mature area of research, retrieval systems targeting local needs of our country have not been extensively researched. From the view point of textual content-based retrieval, systems detecting and recognizing text in non-cursive scripts are already in use by regulatory bodies, agencies and media houses in many developed countries. Among cursive scripts, recognition of Arabic text also attracted significant research attention in the recent years both for caption [30, 31, 32, 33, 34, 31] and scene text [35, 36]. From the perspective of Urdu text, a major proportion of research endeavors target printed text in document

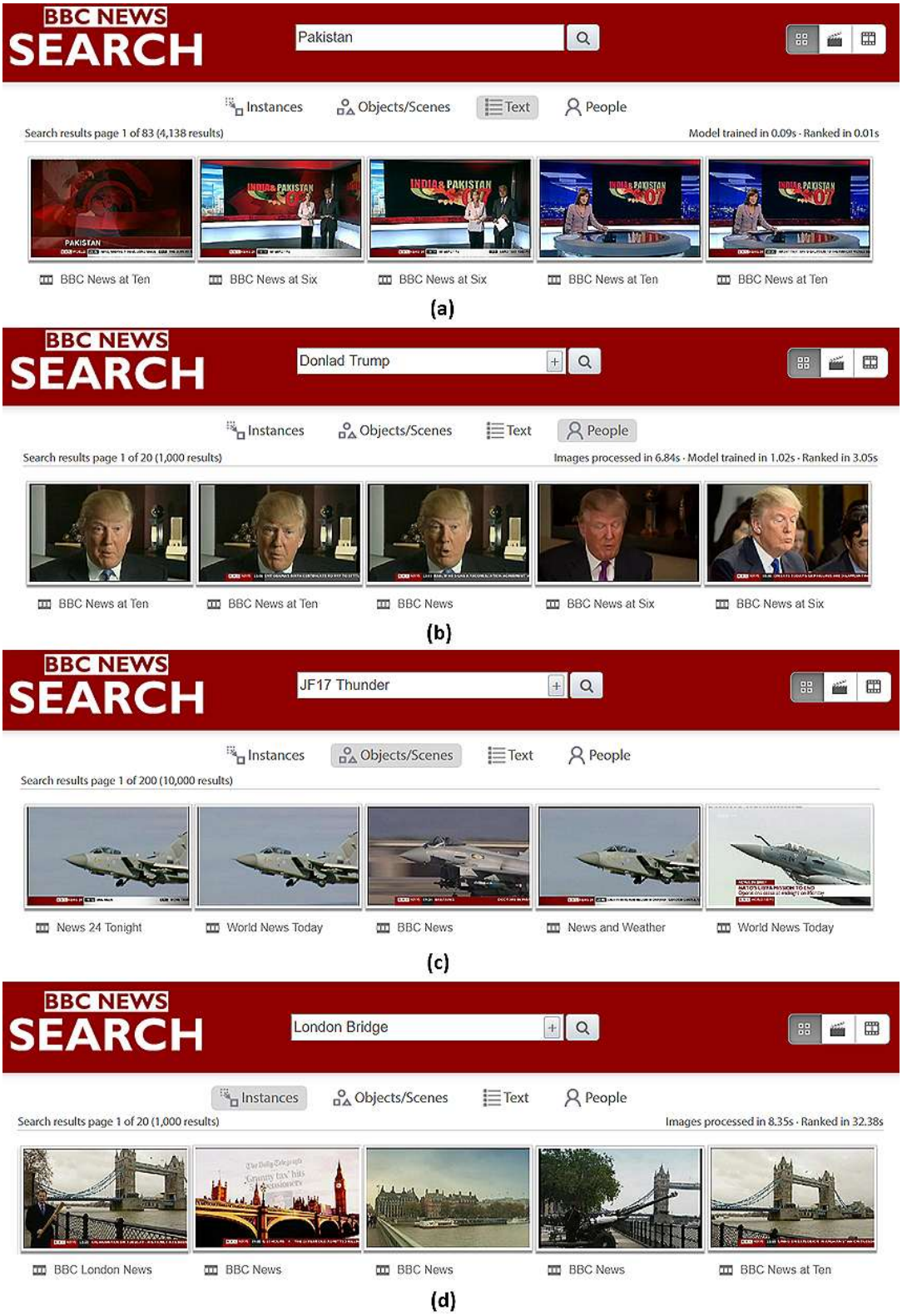


Figure 1.4: Visual Search of BBC News (a): Textual content-based search for query keyword 'Pakistan' (b): Person search for 'Donald Trump' (c): Object search for 'JF17 Thunder' (d): Instance search for 'London Bridge'

images [37, 38, 39] with few preliminary studies on Urdu handwriting as well [40, 41, 42, 43]. However, the literature is fairly limited once it comes to detection and recognition of Urdu caption text. Though few pilot studies on detection [44, 45, 46, 47] and recognition [48, 49] have been carried out, they are far from expectations of a practical retrieval system. A major limiting factor has been the non-availability of large labeled datasets supporting the development and subsequent evaluation of such systems. Indeed there is a need to investigate and propose robust text detection and recognition techniques which could eventually be employed to develop smart retrieval (and other associated) applications, targeting our local regulatory bodies (like PEMRA) and media houses.

1.2 Challenges in Cursive Scripts

To highlight the challenges of cursive scripts, it is important to mention the complex word formation in such scripts. Typically, a word in a cursive language like Arabic or Urdu is a combination of one or more ligatures where a ligature represents one or more characters joined together through joiner rules. These joiner rules determine which characters are joined and which appear in isolated form. The shape of characters within a ligature is a function of its position (initial, middle, end, isolated etc.). Ligatures can therefore be considered as partial words [50]. Ligatures are further categorized into primary and secondary components, the primary component being the main body of the ligature while the secondary components represent dots and diacritics [38]. It is also worth mentioning that many ligatures share the same primary component and differ only in number and/or position of dots (leading to high inter class similarity). Furthermore, the non-uniform intra and inter word spacing in such scripts makes segmentation of lines into words highly complex hence ligatures or characters are mostly employed as units of recognition.

Cursive text is printed (or rendered) in one of the standard scripts, Naskh and Nastaliq being two popular scripts (for Arabic and Urdu respectively). Naskh script follows a horizontal baseline, i.e. characters are joined along a horizontal line. In Nastaliq, on the other hand, characters are joined diagonally making it highly cursive. This diagonal style also results in overlapping of neighboring characters both horizontally as well as vertically making segmentation of characters much more challenging as compared to Naskh. Figure 1.5 illustrates an Urdu text line printed in both Naskh and Nastaliq scripts. Among these, Nastaliq being the more common script for Urdu text, makes the subject of our current study. An example cursive (Urdu) text line illustrating various recognition challenges is presented in Figure 1.6.

1.3 Problem Statement

This study is aimed at research and development of techniques to detect and recognize Urdu caption text appearing in video frames. More specifically, we target data from local News channels which

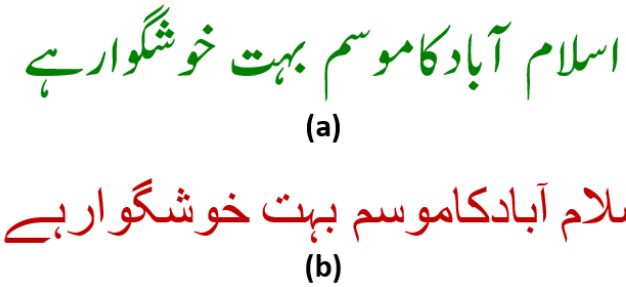


Figure 1.5: A sample text line in (a): Nastaliq (b): Naskh script

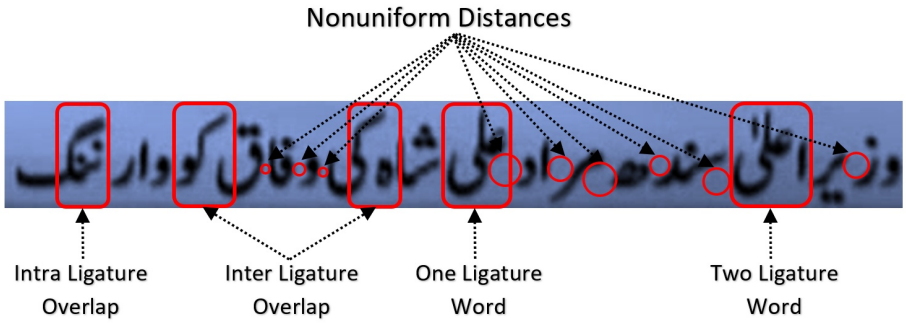


Figure 1.6: Example of cursive text line highlighting recognition challenges

typically contain bilingual (Urdu & English) textual content. Localizing the textual regions and converting them from image to text using a V-OCR can then be exploited to develop smart retrieval systems and other related applications.

In contrast to printed text, detection and recognition of text appearing in video frames is marked by many challenges. Typical problems include low resolution of text, complex and non-homogeneous backgrounds and, different font sizes and colors etc. Another set of challenges is also introduced by the complexity of the script to be recognized. Thanks to more than five decades of extensive research [51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61], mature detection and recognition systems have been developed targeting text in non-cursive scripts (languages based on Roman script for instance). Research on cursive scripts (like Arabic, Persian, Urdu etc.) is much more challenging and the research attention of the pattern recognition community in this problem is relatively recent (especially for caption text). Furthermore, development of a text detector that could work in multi-script environments also remains an open problem.

In our study, we target detection and recognition of cursive Nastaliq text appearing in video frames using Urdu as a case study, though the findings can be generalized to other cursive scripts as well. As mentioned earlier, research on detection and recognition of Urdu caption text is in its infancy and requires significant research endeavors to propose robust solutions which can eventually be employed in real world applications. The current research is a step in this direction with the aim

to develop caption text detection and recognition system primarily focusing on the content of our local News channels.

1.4 Research Objectives

The key objectives of this research study are listed in the following.

- To develop a comprehensive benchmark dataset of video frames with ground truth information to allow algorithmic development, as well as, evaluation of (Urdu) caption text detection and recognition systems.
- To propose a robust caption text detector specifically targeting the local News channels with bilingual (Urdu & English) textual content.
- To investigate different pre-processing techniques which effectively segment detected text from background for subsequent recognition.
- To develop an effective recognition method for Urdu caption text in an attempt to advance the current state-of-the-art on this problem.
- To evaluate the proposed methods on the developed dataset and assess the effectiveness of these methods in the context of the current state-of-the-art techniques.

1.5 Research Questions

In pursuit of the aforementioned objectives, the following research questions were identified for this study.

- How can we solve the caption text detection problem for our local News channels using conventional image analysis based techniques?
- How the current advancements in deep learning-based object detection can be exploited for this problem and do they outperform image analysis based solution?
- What are the effective pre-processing methods that can segment caption text from the background? Is the pre-processing required? How the text recognition performance evolves as a function of pre-processing methods?
- Which recognition unit (ligature or character) is more appropriate for development of recognition systems for cursive scripts like Urdu and what are the effective techniques for each that can be investigated for recognition purposes?
- How sequence modeling can be applied to Urdu text lines to develop a video OCR and how the choice of model impacts the recognizer performance?

- What is performance of the developed detection and recognition techniques with respect to current state-of-the-art using standard evaluation metrics?

1.6 Proposed Techniques

The key hypothesis of our study is that the recent advancements in different areas of deep (machine) learning can be adapted to effectively solve the problem of Urdu caption text detection and recognition. Leveraging these advancements, a system for detection and recognition of Urdu text appearing in video frames is presented. Object detectors based on deep convolutional neural networks (CNN) are adapted to text detection problem by modifying the anchor boxes and training the detectors on examples of text lines. Since it is common to have videos with caption text in multiple-scripts, cursive text is distinguished from Latin text using a script-identification module. Finally, detection and script identification are combined in a single end-to-end trainable system. For recognition, we present an analytical technique that relies on a combination of CNN, recurrent neural networks (RNN) and connectionist temporal classification (CTC) trained in an end-to-end framework. Text lines extracted from video frames are pre-processed to segment the background and are fed to a CNN for feature extraction. The extracted feature sequences are fed to different variants of Bi-Directional RNNs along with the ground truth transcription to learn sequence-to-sequence mapping. Finally, a CTC layer is employed to produce the final transcription. Experimental study of the system is carried out on a comprehensive dataset of more than 11,000 video frames; the detector reports an F-measure of 0.91 while the recognition engine realizes a character recognition rate of 97.63%. The overall methodology including data labeling, text detection, pre-processing and finally recognition of caption text is illustrated in Figure 1.7.

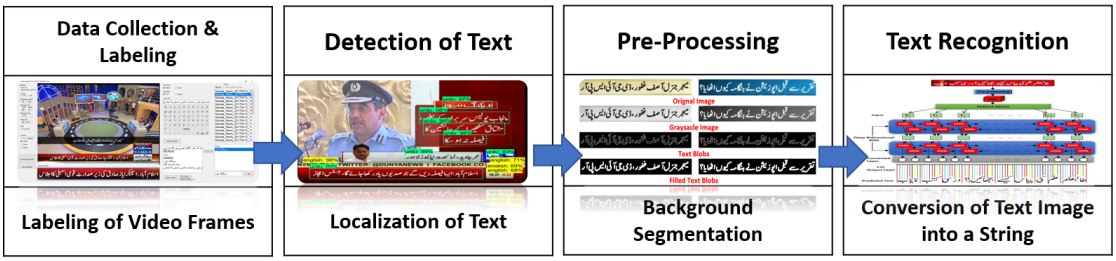


Figure 1.7: Methodology of proposed technique for processing of caption text

1.7 Research Contributions

The significant contributions of this study are listed as follows.

- Development of a comprehensive dataset of more than 11,000 video frames collected from local News channels along with ground truth information on location as well as transcription of text. The dataset has been named as 'UTiV' (Urdu Text in Videos).

- Public availability [1] of the developed dataset to contribute to enhance research on Urdu caption text detection and recognition – problems that are tailored to our local needs.
- Leveraging the recent advancements in deep learning based object detection and adaptation of these detectors for localization of Urdu caption text.
- Combining text detection and script identification in a single system accustoming the solution to the bilingual textual content in local News channels.
- Through investigation of pre-processing techniques to convert detected text regions into binary images for effective recognition.
- Development of a comprehensive recognition framework for Urdu caption text that exploits a convolutional neural network (CNN) with different variants of recurrent neural networks (RNN, GRU, LSTM).
- Analytical experimental studies to validate the proposed techniques and identify the optimal configurations for detection and recognition.

1.8 Thesis Organization

This thesis is organized as follows.

Chapter 2 (Literature Review) provides an overview of related work on text detection as well as text recognition from images and videos. Though the primary focus of our research is Urdu caption text, for completeness, well-known contributions to detection and recognition of text in other scripts are also discussed. The chapter aims to analyze the current state-of-the-art on problem at hand and identifies the research gaps that call for further investigations.

Chapter 3 (Data Collection and Labeling) presents the details of dataset collected and labeled as a part of this research. Evaluation metrics, details of ground truth labeling tool and statistics of labeled data are also presented in this chapter. Furthermore, synthetic data generation (to enrich the training set) is also discussed.

Chapter 4 (Detection of Textual Content) presents the technical details on the first aspect of our research i.e. detection of textual content. We discuss text detection using image analysis-based techniques, identify the demerits and provide a rationale for choosing the deep learning-based techniques. Details on adaptation of CNN-based object detectors for localization of Urdu caption text are then presented. The chapter next introduces a hybrid detector that combines text detection and script identification in a single system. Details of experiments, corresponding results and the

[1] https://drive.google.com/drive/folders/1U3M6WTReCu4PYxk88aXlTQDqsSn4gHAq

accompanying discussion finally conclude the chapter.

Chapter 5 (Recognition of Textual Content) introduces the technical details of the proposed V-OCR architecture for Urdu text. The discussion includes choice of recognition unit (ligature vs. character), pre-processing of text lines and the architectural details of the feature extractor (CNN), the sequence predictor (different variants of RNN) and the final transcription generation (CTC layer). The chapter also presents the key experiments, their findings and a comparison with current state-of-the-art.

Chapter 6 (Text Detection & Recognition: Applications) summarizes real world applications that can be developed on top of the detection and recognition modules. Details on textual content based indexing and retrieval, the key target application of our study, are also presented. The chapter also presents end-to-end results from the perspective of a retrieval application.

Chapter 7 (Conclusion and Future Work) summarizes the key findings of our study and presents the concluding remarks. The chapter also recalls the key contributions of this research and identifies future research directions on this subject.

Chapter 2

Literature Review

2.1 Introduction

Detection and recognition of textual content have been studied for many decades by the pattern recognition and computer vision communities [51, 54, 55]. Most of the earlier studies on these problems are reported on scanned documents only [52, 62, 63]. However, over the years, as new applications were envisaged, text recognition in other modalities such as historical documents, scene images, handwritten documents and videos (Figure 2.1), was also investigated. The ultimate goal, in all these diverse modalities, is to detect the textual content and convert it into machine readable string. From the perspective of caption text, textual content appearing in videos carries useful semantic information that can be exploited for indexing and retrieval applications. Key components of a Video Optical Character Recognition System (V-OCR) include detection (localization) of textual content, extraction of text segmenting it from the background, identification of script and recognition of text. Figure 2.2 presents an overview of these steps along with examples.

Text in videos may contain complex background, multiple foreground colors, different fonts styles, sizes and orientations. All these properties make detection and recognition a challenging problem. The literature is very rich and comprehensive when it comes to detection and recognition of textual content both from still images and videos. In the following sections, we will give an overview of notable contributions to text detection (Section 2.2) and text recognition (Section 2.3) along with a discussion on challenges in each of these tasks. Since we primarily target detection and recognition of Urdu text, a special focus is given to discussion of methods which employ cursive (Arabic, Urdu etc.) text. Nevertheless, for the sake of completeness, techniques targeting text in western languages also make a part of our discussion.

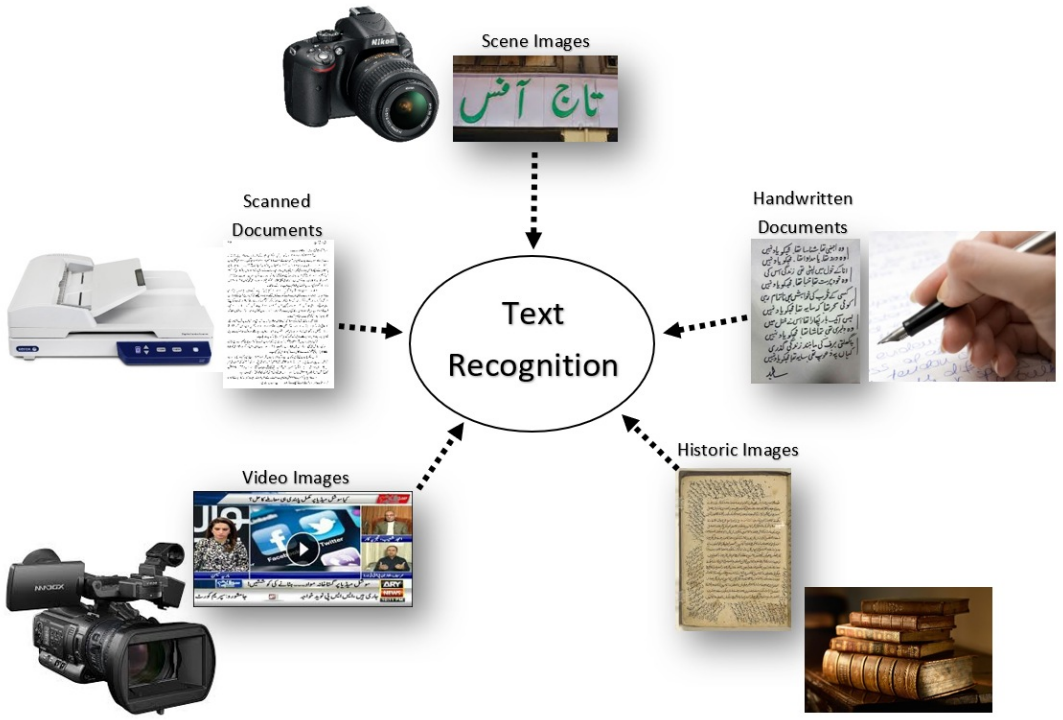


Figure 2.1: Text recognition modalities

2.2 Text Detection Methods

Detection of textual content in videos, images, documents and natural scenes has remained an attractive research problem. Over the recent years, a wide variety of approaches have been proposed for text detection, localization and extraction both in videos and still images. The domain has matured progressively over the years starting with trivial image analysis based systems to complex end-to-end learning based systems. In the following, we discuss significant contributions to detection of text while detailed surveys on the problem (and related problems) can be found in [55, 64, 13, 65, 66, 67, 68, 69, 69].

Text detection refers to localization of textual content in images. Techniques proposed for detection of text are typically categorized into unsupervised and supervised approaches (Figure 2.3). While unsupervised approaches primarily rely on image analysis techniques to segment text from background, supervised methods involve training a learning algorithm to discriminate between text and non-text regions. Supervised approaches for detection of textual content typically employ state-of-the-art learning algorithms which are trained on examples of text and non-text blocks either using pixel values or by first extracting relevant features. Classifiers like Naïve Bayes [70], Support Vector Machine [71], Artificial Neural Network [72] and Deep Neural Networks [73] have been investigated for this problem over the years.

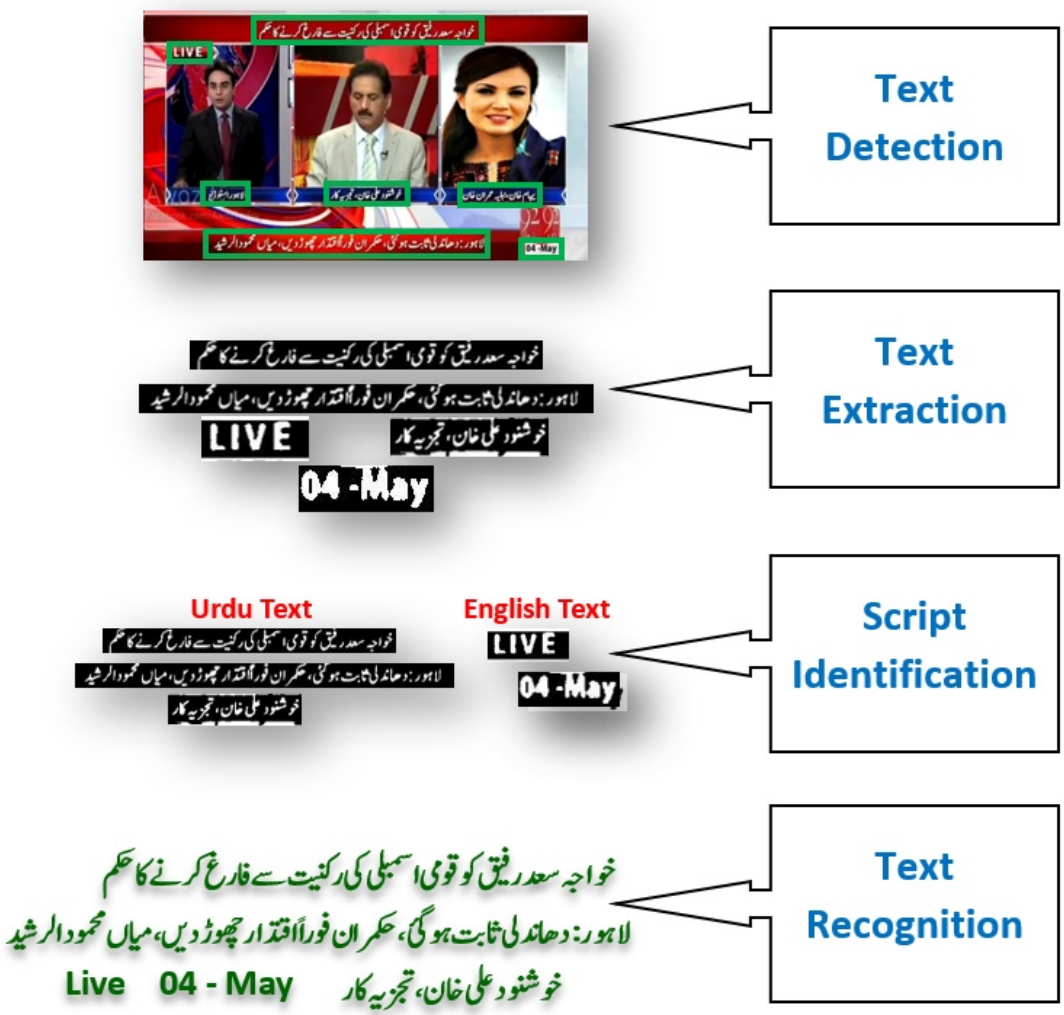


Figure 2.2: Typical steps in Video Optical Character Recognition

The subsequent sections discuss the well-known text detection methods proposed in the literature.

2.2.1 Text Detection using Unsupervised Techniques

Unsupervised approaches for detection of text are typically based on image analysis techniques and use segmentation methods (edges, spatial grouping etc.) to differentiate text from rest of the image. This section describes some of the well-known existing unsupervised approaches for text detection from video frames and images. Generally, unsupervised methods are classified into edge-based (gradient-based), connected component-based (region based), texture-based, and color-based methods as discussed in the following.

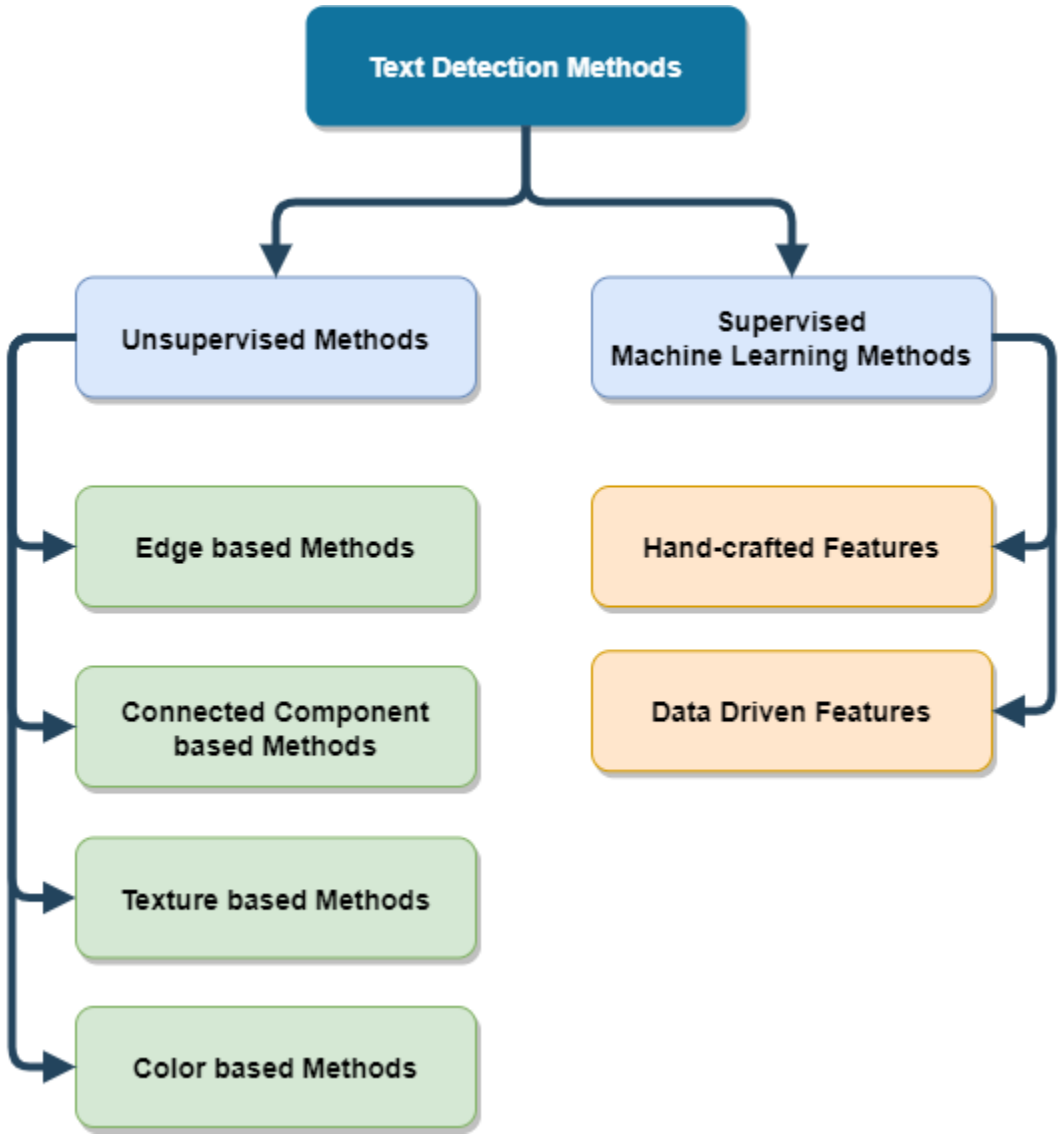


Figure 2.3: Taxonomy of text detection methods

2.2.1.1 Edge-based Methods

Edge-based methods have been employed for many years for text detection in images [74]. These methods exploit the high contrast between text and its background by finding the edges in an image. Regions of high edge density are then merged under some heuristics to filter out non-text regions. Typically, an edge detector (e.g. Sobel or Canny operator) is applied on the image to find the edges, which is followed by smoothing and morphological operations. Edge based methods work reasonably good in case of sharp images but generate a large number of false alarms if the input image is noisy.

Among known edge-based methods, Cai et al. [75] exploit edge strength and density to detect textual regions in images. As a first step, non-text regions are eliminated by applying edge detection

and then local thresholding is used to highlight low contrast text. The method is evaluated on Hong Kong Jade TV station videos containing Chinese text and CNN news for English text and an overall detection rate of 93.6% on 14,685 text lines is reported. Likewise, edge density and morphological operations are also employed by Ye et al. [76]. As a second step, wavelet features are computed to capture the textural information in the text regions. The authors report a detection accuracy of 93.4% on Chinese text and 93.9% on English text. The technique is shown to be robust against variations in text font, color and size.

In other similar methods, Shivakumara et al. [77] employed edge based features to detect text from the video images. The authors also proposed to exploit straightness as a novel edge feature to eliminate the unwanted edges from detected regions hence improving the localization performance. A detection rate of 82% is realized on a custom dataset of images. A subsequent work by the same authors [78] exploit the combination of low and high components of Sobel and Canny edge detectors to handle the varying contrast in images; improving the detection rates up to 85.6%. The work was later extended [79] to combine Sobel edge filter along with color differences and boundary growing technique. Experimental study of the proposed technique on Hua's Dataset [80] reported a detection rate of 89.67%.

Guru et al. [81] also make use of an edge detector for text detection problem. Candidate text regions are first determined by applying block-wise eigen value computation on image gradients. Furthermore, k-means is used to identify the text regions among the candidate text blocks. Once candidate text blocks are identified, edges of text regions are extracted using Sobel filter and, bounding boxes are generated using horizontal and vertical projection profiles. The identified text regions are validated using geometric properties. A dataset of 800 video frames is used to carry out the experimental study of the system and 84.5% detection rate is reported. In [44], Jamil et al. studied Urdu text detection in videos using a combination of edge-based features and a series of morphological operations. The proposed method first extracts vertical edges in the frame followed by computation of mean gradient magnitude around each pixel. The resulting image is binarized and to merge the candidate text pixels into regions, Run Length Smoothing Algorithm (RLSA) is applied. An edge-density filter is then applied to remove all regions where the edge density is below a pre-defined threshold. Finally, a set of geometrical constraints is applied on the candidate bounding boxes to eliminate the false detections. The key processing steps of the technique are summarized in Figure 2.4; the technique is evaluated on a small dataset of 150 video frames and an F-measure of 0.79 is reported.

A novel edge-based method, known as edge-ray filter was proposed by Huang et.al [82] to detect characters from camera-based images. The presented method works differently by filtering out the complex background in the image instead of directly detecting text lines. Edge Preserving Smoothing Filter (EPSF) followed by Canny edge detector is then applied. To speed up the filtering process, Edge Quasi-Connectivity Analysis (EQCA) is used to combine complex edges and

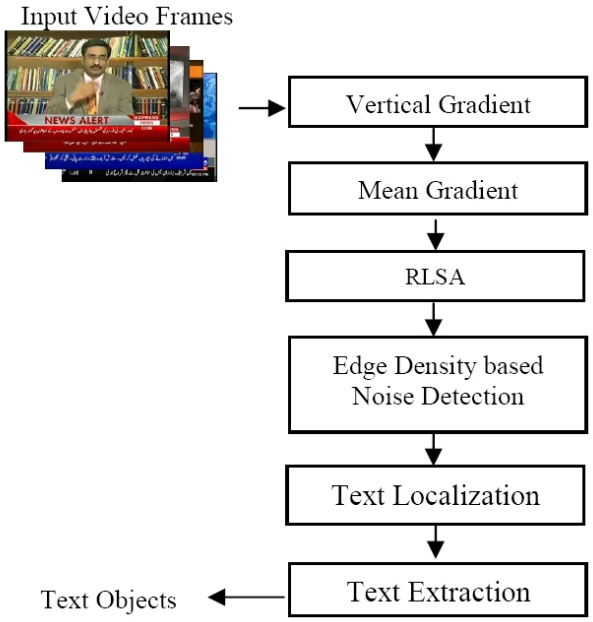


Figure 2.4: An overview of key processing steps for detection of Urdu caption text by Jamil et al. [44]

contours of separated characters. Noisy regions are eliminated using Label Histogram Analysis (Figure 2.5). The proposed technique works on dark text on bright background as well as the inverse scenario simultaneously. The proposed edge-ray filter reports an F-Measure of 63% on ICDAR2011 dataset.

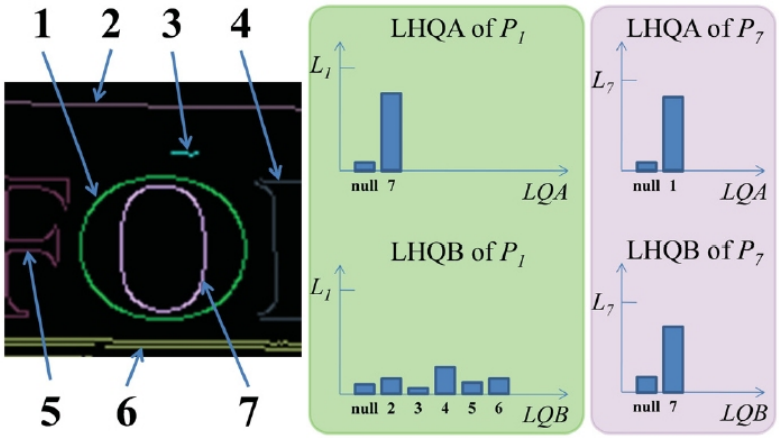


Figure 2.5: Label histogram Analysis for extraction of characters (Image Source [82])

Another notable text detection technique from camera images is presented by Banerjee et al. [83]. The authors identify the highly specular pixels in the image and apply connected component labeling on the identified set of pixels. Canny edge detector is then applied on the complete

image and bounded boxes corresponding to each component are generated followed by merging of overlapping boxes. For each bounding box, Otsu's thresholding is separately applied and finally, the specular components, which are extracted in the beginning, are replaced back in the corresponding regions of the image. A post-processing step is also carried out to enhance the detector performance. The technique is evaluated on the ICDAR 2013 dataset with an F-measure of 64%.

A method involving two edge-based techniques along with stroke width transform is proposed by Yu et al. [84]. The authors named these methods as edge classification and candidate edge recombination and exploit the concept of over-segmentation and region merging. As a first step, edges of text are extracted from the background by dividing the image into sub images. Based on color and stroke width, neighboring edges are combined resulting in bounded boxes over characters. Character boundaries are then merged into a sequence using chain features. Application of the proposed method on a sample image is show in Figure 2.6. Evaluations is done on ICDAR 2003 and ICDAR 2011 datasets and reported F-measures of 0.69 and 0.70 respectively. The work was later extended with similar edge filters and introduction of multi-channel processing [85]. The method is improved by storing all extracted edge features in a pool and then selected features are used to train the classifier. Multi-channel processing is used to verify the textual regions and duplication are removed using the non-maximal suppression method. The improved technique realized an F-measure of 0.73 on the ICDAR2011 dataset. Furthermore, experiments were also carried out on the SVT dataset [86] reading an F-measure of 0.31.

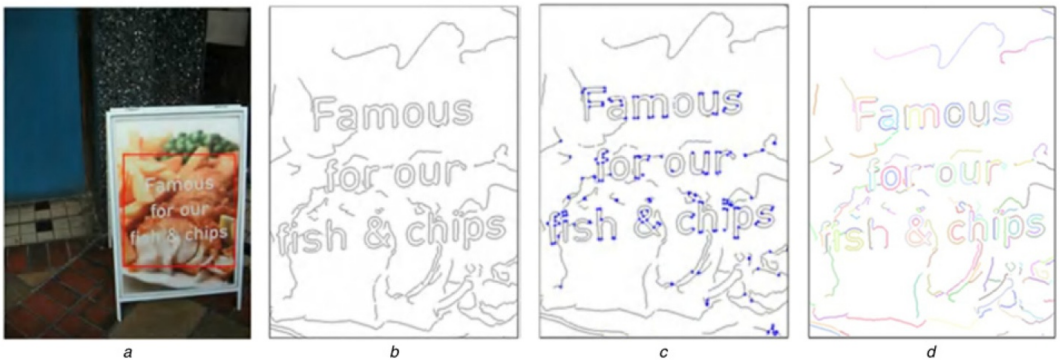


Figure 2.6: Edge segmentation example (Image Source [84]) (a): Original image (b): Edge image, (c): Segmentation points (d): Edge segmentation with different colors

A combination of saliency map and edge features for detection of video scene text is presented in [87]. The technique is claimed to be robust on cluttered backgrounds as well as low resolution text. The method first calculates saliency and edge maps from the image. The saliency map is used to retain saliency regions in the image and, is later employed to remove complex backgrounds. Ignoring the low resolution and light regions, the edge map calculates the edge features. Saliency map and edge map are then merged and the resulting image is termed as Saliency Edge Map (SEM). As a last step, connected component analysis are performed to extract text regions. Experimental

study of the system was carried out on multiple datasets including ICDAR 2011 and 2013, MSRA and SVT datasets and F-measures of more than 0.80 were reported in different evaluations.

2.2.1.2 Connected component based methods

Connected component based methods [88] exploit the color/intensity of text pixels generally accompanied with geometrical heuristics to distinguish text from the background. Pixels in the image are clustered into small regions based on homogeneity of color or intensity values. While CC-based methods heavily depend on the color/intensity information and geometrical properties to group pixels into clusters, these methods, like gradient based methods, do not perform well in case of low contrast between text and the background.

Connected components are exploited to localize characters in a technique presented by Wand and Kangasin [89]. Color clustering is employed to divide the image as a function of similar color layers. Connected-components in each color layer are processed using a graph and bounding boxes are calculated. An aligning-and-merging-analysis (AMA) method is then applied exploiting the color layers and the bounded boxes of connected components. The technique is evaluated on 325 camera-based images at a resolution of 640×480 captured from different perspectives and different lighting conditions. With 3,597 characters in 325 images, a detection rate of 92% is reported. Key steps in the proposed technique are summarized in Figure 2.7.

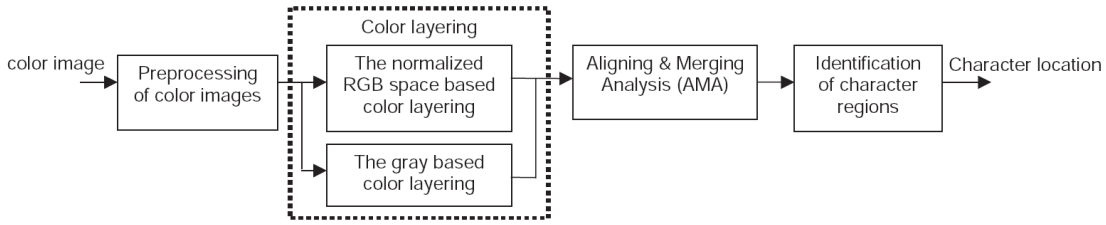


Figure 2.7: Character localization technique proposed in [89]

In another study, Liu et al. [90] employ Gaussian mixture model with learning of neighboring characters to localize multilingual text in images. Each connected component in the image is identified as text or non-text as a function of its neighbors. Parts of characters are connected together using a morphological process while Voronoi partition is employed to determine the component neighbors. Training is carried out using the maximum–minimum similarity (MMS) criterion and evaluations are conducted on Chinese and English text. A recall of more than 98% is reported on a set of 300 images. Another connected component based technique is presented in [91] for scene text detection. Components are extracted using the maximally stable extremal region (MSER) technique while AdaBoost is employed to find the pairwise adjacency relations between the clusters which are generated using CCs as potential text regions. A nearest neighbor based classifier is

developed to identify whether a given image contains text or not. ICDAR 2005 and ICDAR 2011 datasets are used in experiments reporting accuracies of 74% and 70% respectively.

Liu and Sarkar [92] introduce novel intensity and shape filters for text detection in scene images. The images are binarized using Niblack thresholding and components are grouped using typical geometrical properties. The intensity filter exploits the overlap between the intensity histograms of components while the shape filter serves to eliminate the non-text regions. An example image with output of both filters is presented in Figure 2.8. Experiments on 249 images of the ICDAR 2003 dataset report an F-measure of 54%.

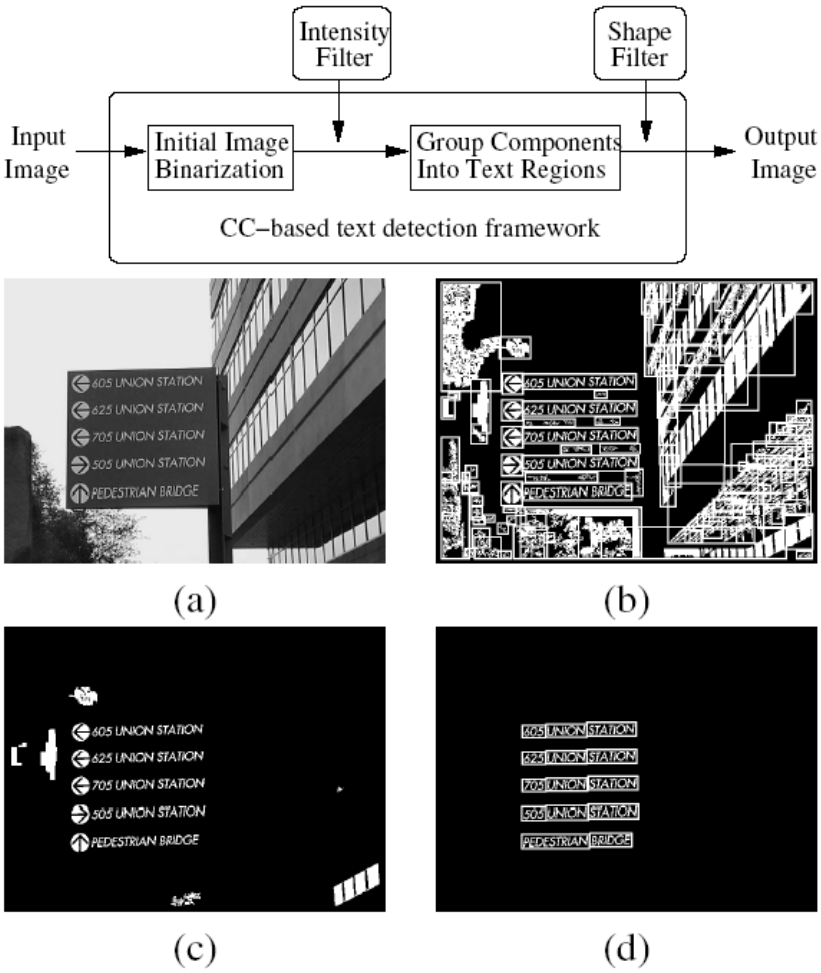


Figure 2.8: Application of Intensity and Shape filters for text detection in [92]

Laplacian operator is used in [93] for detection of text in video frames by computing maximum gradient difference value for each pixel. Text and non-text pixels are discriminated using k-means clustering. Projection profiles analysis is carried out on the input image to determine the boundaries of the textual regions while non-text regions are dropped by using the geometrical properties of text. Experiments on a small dataset of around 100 video frames report a detection rate of 93.3%.

A multi-channel connected component segmentation method (Figure 2.9) is investigated by Wang et al. [94]. The authors carry out connected component segmentation using the Markov Random Field exploiting color, contrast and gradients of image RGB channels. Non-text regions are removed from each channel of image separately while the remaining text components are merged and grouped into words. ICDAR 2003 and ICDAR 2011 datasets are used for experiments reporting an average F-measure of 70%.

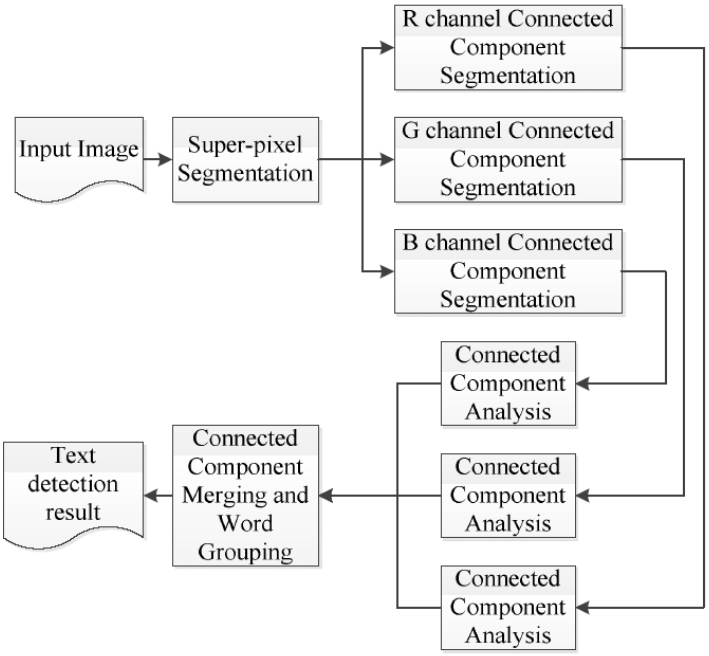


Figure 2.9: CC-based text segmentation method in [94]

2.2.1.3 Texture based methods

Texture-based methods consider the textual content in an image as a unique texture which distinguishes itself from the non-text regions. Texture features are generally computed from gray level images or by first transforming the image using filtering or applying frequency domain transformations. With complex backgrounds, texture-based methods perform better as compared to connected component or gradient based methods. These methods, however, generate more false positives when the background contains similar texture properties as text. Gabor filters, wavelets and spatial variance etc. have been investigated to capture texture properties of text. Along with this, Curvelets [95], LBP [96], HoG [97] and DCT [98] have also been used for detection of textual content.

Among one of the preliminary works on texture-based text detection from videos, Zohng et al. [98] proposed extraction of text from I-frames in JPEG compressed image of MPEG videos. Unlike the common practice of first decompressing the video and then performing text detection,

the technique relies on directly locating the candidate text regions in the DCT domain. In another texture based method [99], scale-space feature extractor is used to detect text in digital videos. The proposed method comprises of two steps, a sum of squared difference (SSD) which finds the initial position of text and a contour-based module, which refines the locations identified in first step. The technique reported an F-measure of 73% on a private dataset of videos. Another texture based method is presented by Kim et al. [100] where an SVM is used to analyze the textural pattern made by raw pixel values in an image. Output of texture analysis is fed to a continuously adaptive mean shift algorithm (CAMSHIFT) which refines and validates the text regions. Experiments on a private dataset of videos, web and document images reported an overall detection rate of 96%. Key steps in the proposed technique are illustrated in Figure 2.10.

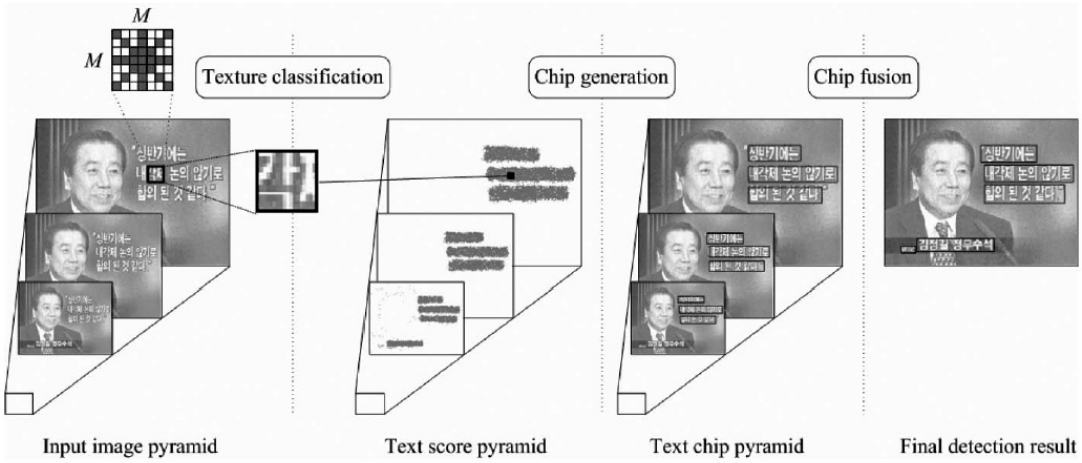


Figure 2.10: Text detection system presented by Kim et al. [100]

Wavelet transformation has been widely explored for detection of text in different types of images. Gallavata et al. [101], for instance, employed wavelet transformation on images and extracted the high-frequency wavelet coefficients to characterize text and non-text regions in the image. A k-means algorithm is then employed to group textual regions together and finally a projection analysis allows localizing the text boundaries. The technique is validated on a small dataset of 45 video frames [102] containing 145 readable text lines and an accuracy of 89% is reported. In a similar work [103], a multi-scale wavelet approach is investigated in a two-step detection process. As a first step, candidate text pixels are identified using wavelet energy features and are grouped together using a density-based region growing method. In the second step, textural measures computed from candidate text regions are used to refine the detection with an SVM classifier. The technique is evaluated on a private dataset containing 177 video frames as well as 44 images from Hua's dataset [80] and, a detection rate of 96.8% is reported.

Wonjun & Changick [104] hypothesize the existence of transient colors between text and the background and exploit this to detect textual content. A transition map is first produced and poten-

tial text regions are extracted using a reshaping method. The localization is refined by projecting the text pixels in the transition map. The proposed method is claimed to be invariant to changes in size, color, position and contrast of characters. Another texture-based technique for text detection is presented in [105] where the authors propose novel Fourier-statistical features (FSF) for this problem. The method first identifies text frames from a large collection of images and, in the second step, text regions in these frames are detected. Classification of text frames is carried out using visual cues while for detection FSF features are computed and are fed to k-means clustering to group pixels into text and not-text classes. Like many other methods, projection profiles are employed for fine localization while false alarms are reduced using a set of heuristics. Experimental study on a custom developed dataset reports an F-measure of 93%.

Among other texture-based methods, Das et al. [106] employed textural features to localize text regions in natural scene images. Authors first employ DCT for background suppression and subsequently extract textural features from the image. An F-measure of 64% is reported on a private dataset in this study. Likewise, texture features based on Gabor filters are employed in [107]. Experimental study on the ICDAR 2003 dataset along with 100 video frames collected by the authors reported a detection rate of 97.90%. In another work by Grzegorzek et al. [108], heuristics-based filtering is first carried out to discard the non-textual (background) regions in the image. Subsequently, textural features are used with SVM classifier to identify the text regions.

2.2.1.4 Color based methods

Color based methods [105, 109] are similar in many aspects to the component based methods and employ color information to distinguish text and non-text areas. These methods rely on the assumption that text pixels and the background contain separate color clusters and perform a color based segmentation to extract textual regions.

Among earlier works in color-based text detection, Garcia and Apostolidis [110] employ color quantization with clustering to identify candidate text regions. Character periodicity is then exploited to classify the regions as text or non-text. Experiments on DiVAN dataset of 200 JPEG images report an accuracy of 93%. A similar work is presented in [111], where in addition to color analysis, affine-rectification is applied to improve the detector performance. The technique is evaluated on road sign boards with text in Arabic, English and Chinese.

Among other color-based methods, Mancas and Gosselin [112] carry out text detection in natural scene images using color segmentation that is based on spatial information. Text pixels are combined together using clustering and the technique is claimed to be robust to uneven lighting, blur and complex backgrounds. The technique is validated on ICDAR 2003 dataset with an accuracy of 93%. The work was later extended [113] to include a selective metric-based clustering and the color information was complemented with intensity and spatial information computed using Log-Gabor

filters (Figure 2.11).

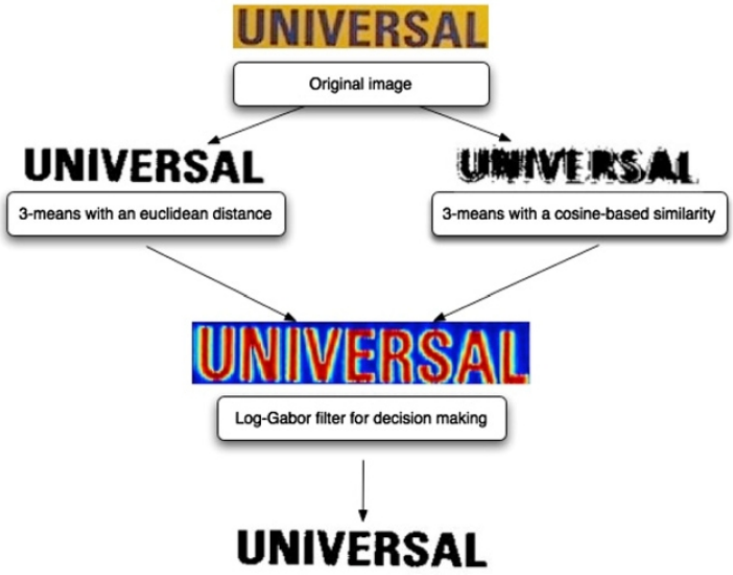


Figure 2.11: Combination of color and spatial information for text detection proposed in [113]

A framework for detection of text with arbitrary orientation is proposed in [114]. A set of components is extracted using local gradient features along with color uniformity in the image. Characters are grouped together using structural features of text such as, distance, size and alignment etc. Text lines are grouped by applying Hough transform that attempts to fit a line on the centroids of potential text regions. An overview of the proposed framework in illustrated in Figure 2.12. The Robust Reading and Oriented Scene Text dataset is used for experiments and F-measure of 62% is realized. The work was later extended [115] to present an enhanced method that relies on three steps. As a first step, a bigram-color-uniformity based technique is applied to group edge pixels based on color pairs. Character candidate regions are then extracted by applying stroke segmentation. Finally, texture features based on Gabor filters are employed for string fragment classification.

Karatzas and Antonacopoulos [116] exploit human color perception to extract text from complex backgrounds.Text is segmented by applying a split-and-merge approach on the hue-lightness-saturation (HLS) representation of color. The technique is evaluated on a dataset of 115 web images collected by the authors with a detection rate of almost 70%. Nikolaou and Papmarkos [117] argue that images with a large number of colors results into poor text detection. The authors proposed a method for color reduction in complex images with many colors and infer that it helps in better text detection. Using the 3D color histogram, generated using edge map and mean-shift method, significant reduction in number of colors is achieved. Furthermore, an edge-preserving-smoothing filter is applied as a pre-processing step to enhance the detections. The method is evaluated on different book covers with more than 200K colors. Impact of color reduction and pre-processing on

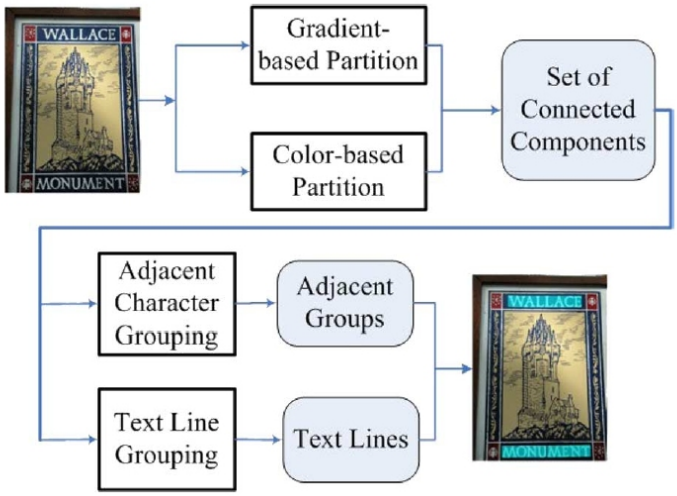
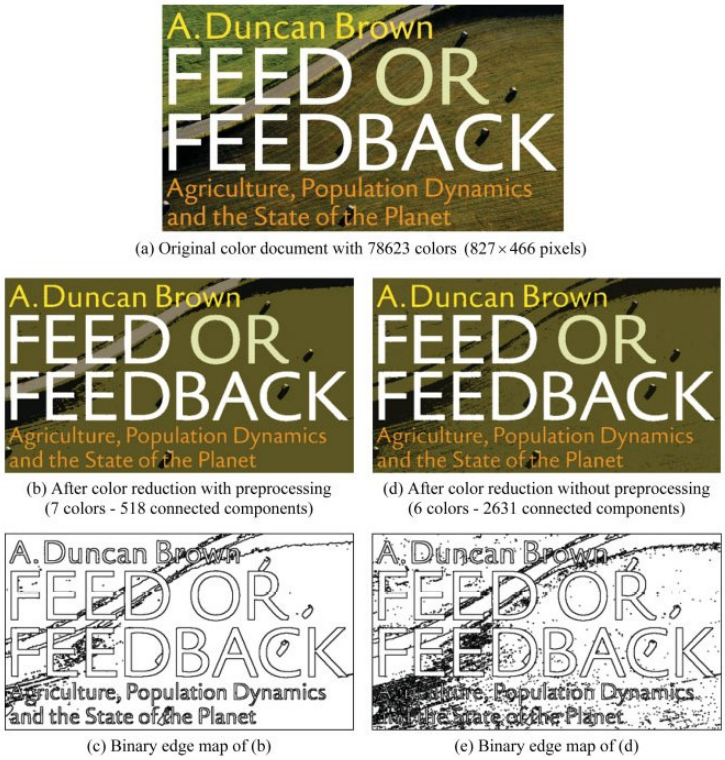


Figure 2.12: An overview of key steps for text detection in [114]

text segmentation for an example image is shown in Figure 2.13.



(a) Original color document with 78623 colors (827 × 466 pixels)

(b) After color reduction with preprocessing (7 colors - 518 connected components)

(d) After color reduction without preprocessing (6 colors - 2631 connected components)

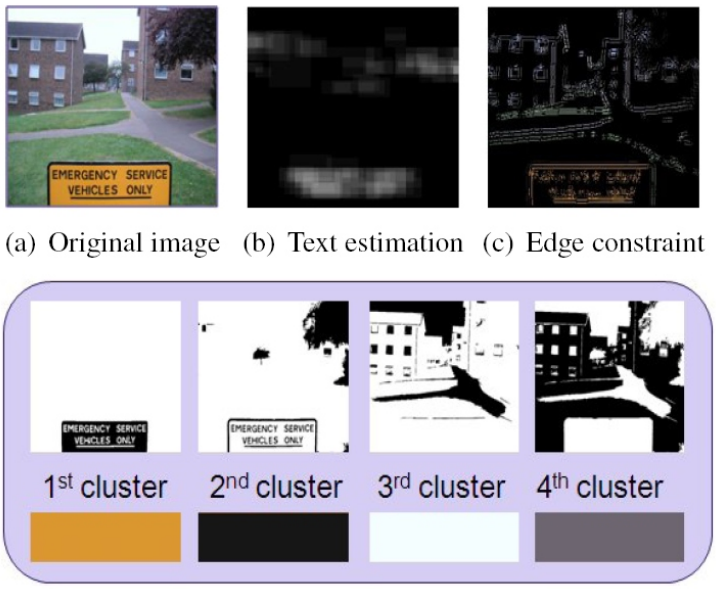
(c) Binary edge map of (b)

(e) Binary edge map of (d)

Figure 2.13: Color reduction and edge-preserving filtering for text detection in [117]

In another notable work by Song et al. [118], color-based k-means clustering is carried out for segmenting text in video frames. To detect text in different sizes, a multi-scale approach is adopted while projection profiles are employed to refine localization. Experimental study of the technique

is carried out using videos from the TRECVID dataset with occurrences of bilingual (English & Chinese) text, and a detection rate of more than 90% is reported. A similar work is reported by Lee et al. [119] where potential textual regions are highlighted by exploiting the color, edge and texture features using k-means clustering (Figure 2.14). Validation of possible text regions is carried out using Markov Random Field model and an F-measure of 64% is reported on the ICDAR 2003 dataset.



(a) Original image (b) Text estimation (c) Edge constraint

(d) Clustering result

Figure 2.14: Clustering based text detection reported in [119]

2.2.1.5 Discussion

An overview of well-known unsupervised techniques for text detection is presented in Table 2.1. As discussed earlier, these techniques do not involve any learning algorithm and text is segmented from rest of the image using a series of image analysis techniques and heuristics. The discussion was grouped into edge-based, CC-based, texture-based and color-based techniques. An analysis of the studies listed in Table 2.1 reveals that edge-based techniques have remained a popular choice of many researchers primarily due to the high edge density in text regions in all scripts. Color-based methods, though simplify the detection, are criticized by many due to reliance on color information which may not always be available (for instance if image acquisition is gray-scale). Component-based methods share many characteristics with color-based methods and represent a more attractive choice as they exploit the intensity rather than the chrominance information and can work on gray-scale images as well. Likewise, textural features have also been widely employed for text detection problem but are known to report high false detections in complex backgrounds.

From the view point of performance evaluation, the ICDAR datasets have been most widely employed for scene text detection. A series of competitions held in conjunction with different editions of ICDAR allowed researchers to objectively compare different techniques under the same experimental settings. From the perspective of caption text, in most cases, private datasets with limited number of images have been employed. For Urdu text Jamil et al. [44] employed an edge-based method with a series of morphological operations and reported a detection rate of 79% on a small set of 150 video frames.

Unsupervised text detection techniques, though report high detection rates in few cases, remain sensitive to the content of image and are mostly accompanied by heuristics and a (relatively) large number of parameters (which are empirically chosen). Supervised approaches, in general, tend to be more robust and effective and make the subject of our discussion in the next section.

Table 2.1: An overview of unsupervised text detection methods

Method	Study	Technique	Script	Dataset	Results
Edge Based	Cai et al. [75](2002)	Edge Strength & Density	Chinese	Private	93.6%
	Ye et al. [76](2003)	Edge Density	Chinese	Private	93.9%
	Shivakumara et al. [77](2008)	Edge Straightness	English	Private	82%
	Shivakumara et al. [78](2010)	Sobel & Canny Edge Detecor	English	Private	85.6%
	Shivakumara et al. [79](2010)	Color Differences	English	Hua's Dataset [80]	89.67%
	Guru et al. [81](2010)	Sobel Edge Detector	English	Private	84.5%
	Jamil et al. [44](2011)	Edge Features	Urdu	Private	79%
	Huang et al. [82](2013)	Edge Smoothing Filter	English	ICDAR2011	63%
	Banerjee et al. [83](2013)	Canny Edge Detector	English	ICDAR2013	64%
	Yu et al. [85](2016)	CERB	English	ICDAR2003&2011	69%, 70%
	Yu et al. [84](2015)	Multi-channel Processing	English	ICDAR2011,SVT [86]	73%,31%
	Huang et al. [87](2019)	Saliency Edge Map	English	CDAR2011&2013,SVT	83% to 88%
CC Based	Wand and Kangas [89](2003)	BAG& AMA	Chinese	Private	92%
	Koo and Duck [91](2013)	MSER	English	ICDAR 2005 & 2011	74%,70%
	Pan et al. [120](2011)	Conditional Random Field	English	ICDAR 2005	65.2%
	Liu et al. [90](2008)	Gaussian Mixture Model	English	Private+ICDAR2003	96%
	Phan et al. [93](2009)	Laplacian operator	English	Private	93.3%
	Liu and Sarkar [92](2008)	Intensity & Shape Filter	English	ICDAR2003	54%
	Shahzad & Khurshid [47](2017)	Image Analysis Techniques	Urdu	Private	88%
	Wang et al. [94](2013)	Multi-channel CCs	English	ICDAR 2003& 2011	70%
Texture Based	Zohng et al. [98](2002)	Discrete Cosine Transformation	English	Private	99%
	Li et al. [99](2000)	Scale-space Features	English	Private	73%
	Kim et al. [100](2003)	CAMSHIFT	English	Private	96%
	Gallavata [101](2004)	Wavelet transformation	English	Hua et al. [102]	89%
	Ye et al. [103](2005)	Multisacle Wavelets	English	Hua et al. [80]	96.8%
	Wonjun and Kim. [104](2008)	Transition Map	Chinese	Private	-
	Goto and Tanaka [121](2009)	Particle filter	Chinese	Private (1,730 video images)	-
	Shivakumara et al. [105](2010)	Fourier-statistical features	English	Private video dataset	93%
	Das et al. [106](2012)	DCT	English	Private	64%
	Aradhya et al. [107](2012)	Gabor filter	English	Private+ICDAR2003	97.9%
	Grzegorzek et al. [108](2013)	SVM filtering	English	ICDAR2003	36.45%
Color Based	Garcia & Apostolidis [110](2000)	Color Quantization	French	DiVAN dataset	93%
	Karatzas and Antona [116](2004)	Color Expression	English	Private	70%
	Mancas and Gosselin [112](2006)	Complementary Clustering	English	ICDAR2003	93%
	Mancas and Gosselin [113](2007)	Color metric Clustering	English	ICDAR2003	92%
	Yi and Tian [114](2011)	Color-uniformity	English	ICDAR2003	62%
	Yi and Tian [115](2012)	Bigram-color-uniformity	English	ICDAR2011	71%
	Song et al. [118](2008)	Color-based K-means	English	TRECVID 2005 & 2006	90%
	Lee et al. [119](2010)	K-means	English	ICDAR2003	64%
	Nikolaou and Nikos [117](2009)	Color reduction	English	ICDAR2003	96%

2.2.2 Text Detection using Supervised Techniques

Supervised techniques for text detection [122, 70, 123, 72, 71] rely on a learning algorithm to discriminate between text and non-text regions in an image. State-of-the-art classifiers like, Naïve Bayes (NB) [70], Support Vector Machine (SVM) [71], Artificial Neural Network (ANN) [72] and Deep Neural Networks (DNN) [73] have been typically employed for identification of text regions. Like any other pattern classification task, supervised methods for text detection consist of two phases, learning (training) and classification (inference). During training, features extracted from text and non-text regions are fed to a classifier to make it learn to discriminate between the two classes. In the inference phase, features extracted from the region in question are fed to the trained classifier which outputs the likelihood of the region as being text or non-text. In general, supervised approaches tend to be more sophisticated than the unsupervised methods. These methods, however, require significant training data (text and non-text regions) to achieve acceptable classification rates.

With conventional machine learning based classifiers (like SVM and ANN etc.), features are typically extracted by applying image analysis based techniques. Common examples of such hand-crafted features include Gabor filters [124], wavelets [103], curvelets [125], strokelets [126], local binary patterns (LBP) [96], discrete cosine transformation (DCT) [98], histograms of oriented gradients (HoG) [97] and Fourier transformation [79]. Deep learning based techniques, on the contrary, combine feature extraction and classification in a single model and features are learned as a part of training hence the term machine-learned features is commonly employed. A number of recent studies [127, 128, 129], validate the superiority of machine-learned over hand-engineered features (and raw pixel values).

In the following, we first discuss text detection using hand-crafted features with conventional classifiers followed by deep learning based techniques.

2.2.2.1 Machine Learning based Methods – Hand-crafted Features

As discussed earlier, in machine learning-based methods, features that serve to distinguish text and non-text regions are identified and extracted. These features are then employed to train a classifier and once trained, the model can be used for classification. We discuss these techniques grouping them as a function of the employed classifier.

Among one of the pioneer works on video text detection using neural networks, Jeong et al. [130] presented a method that classifies text and non-text pixels in video images using textural filters. Histogram analysis is subsequently carried out to remove errors in the first step. Experiments are carried out on 2000 frames collected from different Korean News channel videos. A similar technique is presented in [131] for detection of license plates from images. Neural network is employed as filter on small windows of image to locate the license plate. Reported accuracy is upto 90% on different images of cars. A post-processing step then combines the detected windows to

generate the final localization of license plate.

Li & Doermann [132] presented a neural network based dynamic system that adapts its parameters as a function of changing environments in the videos. The system is able to detect caption and scene text at arbitrary orientations in multiple languages. A similar technique called bootstrap artificial neural network (BANN) is presented by Hao et al. [133]. The method relies on a colored image edge operator which segments the image to extract candidate text blocks. A neural network is then employed to discriminate between text and non-text regions. In another work [123], a bootstrap polynomial neural network (PNN) (Figure 2.15) is employed for video text detection. The network is trained with textural features extracted using a modified version of the LBP operator. The experimental study of the system was carried out on a custom developed dataset comprising of 1,027 video frames and reported an F-measure of 87%.

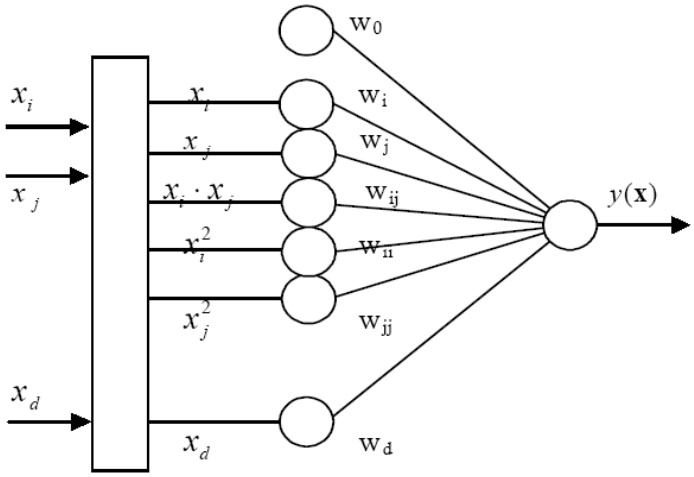


Figure 2.15: Structure of polynomial neural network (PNN) employed for text detection in [123]

Jamil et al. in [45] investigated the combination of unsupervised and supervised techniques for Urdu caption text detection. Edge based filtering with a series of morphological operations is applied to the image to extract potential text regions. The detected regions are then validated through an ANN. Evaluations on a set of 500 video frames collected from various News channels realize 85% F-measure. A similar approach is proposed by Thilagavathy et al. [134] where the authors first employ a combination of region-based and connected component-based methods to extract the potential text regions. Subsequently, an ANN validates the potential text regions to filter out the false positives.

In addition to ANN, SVM has also remained a popular choice as classifier for text detection problem. Among one of the earlier attempts, Shin et al. [135], instead of explicitly extracting any features, directly feed the raw-pixel values of the gray-scale images to an SVM. The idea is based on the ability of SVMs to learn from high-dimensional feature space and embed the implicit feature

extraction within its own architecture. A comparative study with ANNs revealed the superiority of SVM in detecting video text. In another study, Chen et al. [136] extract text lines from video frames with complex backgrounds using baseline location, edge analysis and a set of heuristic constraints. Verification of text lines is carried out using an SVM with edge-based distance map features. The technique is evaluated on a set of video frames containing English text and reported 98.7% correct detection rate. In a subsequent study by the same authors [137], the verification performance of SVM and ANN is compared. Similar to the findings of [135], the authors concluded that SVM outperforms on ANN in terms of detection performance.

Among other notable contributions, Anthimopoulos et al. [138] exploit an edge-map to identify the candidate text blocks. False alarms in the detection are then removed by validating each detected region through an SVM. The SVM is trained with a variant of LBP using a set of 3500 text and 6500 non-text lines from a dataset of 150 video frames. A detection rate of more than 96% is reported in the study. The work was later extended [96] to incorporate multi-resolution analysis and a more comprehensive series of experiments. Experiments on a dataset of 217 video frames collected from 10 different videos, containing 2,963 text regions reported an F-measure of 97%.

Wavelet transformation along with SVM is proposed in [139] for detection of text on complex backgrounds. The image is decomposed using wavelet transform and the high-frequency energy and the low-frequency approximate subspace are employed to train an SVM. Experiments are carried out on 300 images and 97% detection rate is achieved. The authors concluded that the combination of wavelet with SVM not only required fewer training examples but also allowed faster learning. A similar study by Zhen and Wei [71] also advocates that the combination of wavelets and SVM effectively detects text regions in video frames. Three different SVMs are trained by the authors with gray-level values extracted from 9 windows, 2D wavelet decomposition (Figure 2.16) and strokelets. Experiments on 550 video frames reported an accuracy of 92.78%.

Darab and Rahmati [140] also applied wavelet features combining them with HoG to detect Farsi text from scene images. Text and non-text distinction is carried out using an SVM and experiments on 800 images reported an F-measure of 86.5%.

Detection of Farsi text from video frames is also investigated by Moradi et al. in a series of studies [141, 141, 32]. The authors employed various features including text detectors, corners maps, projection profiles, Gaussian pyramid, corner histogram analysis and LBP. Like many other studies, SVM is employed as classifier. All experiments are carried out on a dataset of 50 videos collected from News channels containing Farsi text. Detection rates of close to 90% are reported in various evaluations. The authors claim generalization of the proposed techniques for Arabic text as well but quantified results are not presented.

A relatively recent work on detection of Urdu text through SVM is proposed by Unar et al. [142]. Canny and Sobel operators are applied to the input video frame and combined with

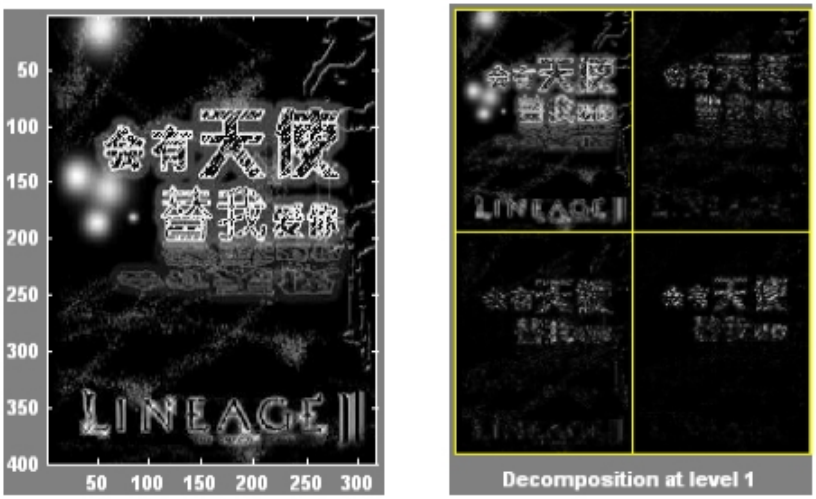


Figure 2.16: Illustration of 2D wavelet-decomposition on a video frame (Image Source [71])

MSER (Maximally Stable Extremal Region) detected candidates. Non-text regions are removed using geometric constraints and stroke width transformation. Finally, for verification of text and non-text blocks, SVM is used as a classifier. The technique is evaluated on 1000 video frames of the IPC dataset [143] and an F-measure of 85% is achieved. In another recent work, Francis and Sreenath [144] employ the least-Square SVM which is trained on 74,000 characters for text and CIFAR-10 dataset [145] for not-text examples. Possible textual regions are extracted from the pool of objects extracted from the input image and are validated by an SVM. The technique is evaluated on multiple datasets (ICDAR, MSRA500 and SVT) for detection of scene text and detection rates varying from 75% to 98% are reported in different experiments.

In general, supervised techniques are known to be more robust and effective in detecting the textual content from images as opposed to unsupervised methods. A major challenge, however, has been the choice of the right feature set that is fed to the classifiers. In the recent years, this challenge has been addressed through automatic feature learning using deep learning techniques. Such methods represent a major paradigm shift and are discussed in the following.

2.2.2.2 Machine Learning based Methods – Data Driven Features

Deep learning-based methods have emerged as one of the most influential solutions to almost all pattern classification problems. The landscape of video processing has also entirely changed with deep learning being the most dominant paradigm for solving a variety of problems.

Among deep learning-based techniques adapted for text detection, Huang et al. [146] employed sliding windows and MSER with CNNs to detect textual regions in low resolution scene images. The proposed technique improved the detection performance in low-quality images having text on complex backgrounds and diverse variations. The method was evaluated on ICDAR2011 dataset and

reported an F-measure of 78%. A fully convolutional network (FCN) is used to predict the salient-map of text blocks in an image by Zhang et al. [147]. The salient map and MSER components are then combined to estimate the text lines. As a last step, another FCN is employed that predicts the center of each character and removes the false positives. The proposed system is claimed to detect text at multiple orientations, fonts and languages. Evaluations on the MSRA-TD500, ICDAR2013 and ICDAR2015 datasets report F-measures of 0.74, 0.83 and 0.54, respectively. A similar work is presented by Gupta et al. [148] where a fully-convolutional regression network (FCRN) is trained using synthetic data for detection of text in natural images. FCRN is able to detect text and apply bounding box regression at multiple scales; and achieved an F-measure of 84.2% on the ICDAR2013 dataset.

Another method called 'SegLink', is proposed in [149] that relies on decomposing the text into segments (oriented boxes of words or lines) and links (connecting two adjacent segments). The segments and links are detected using fully convolutional networks at multiple scales and combined together to detect the complete text line. Figure 2.17 illustrates the architecture of the proposed network. Experiments are carried out on the ICDAR2015 dataset and an F-measure of 75% is reported. The authors also applied the same method to detection of non-Latin text and the system was able to perform equally well on long lines of Chinese text as well.

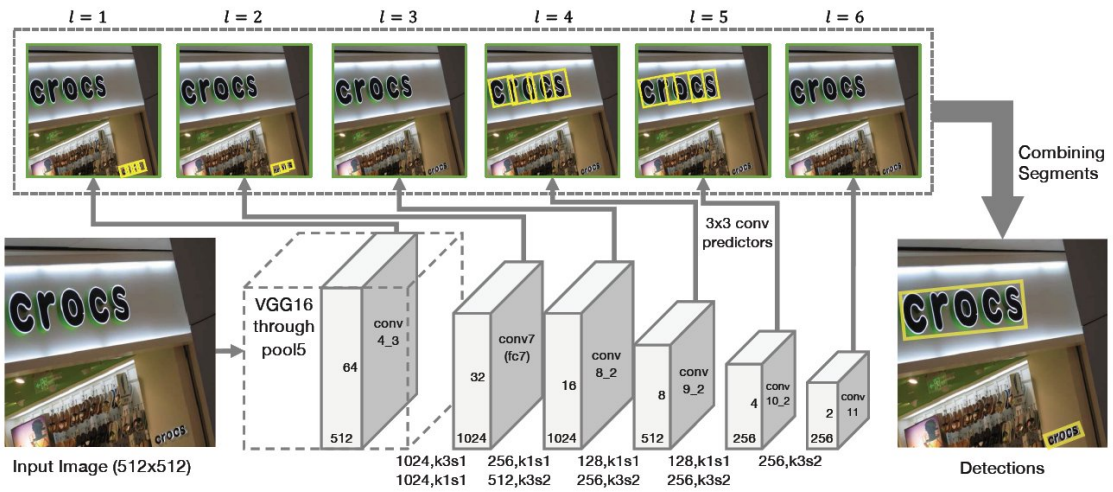


Figure 2.17: The 'SegLink' network architecture proposed by Shi et al. [149]

In [150], a vertical anchor-based method is reported that predicts text and non-text scores of fixed size regions. The proposed network is termed as Connectionist Text Proposal Network (CTPN) (Figure 2.18) and is able to effectively predict the candidate text and non-text scores in a fixed-width proposal. CTPN is able to detect multi-lingual text of any scale in natural images. ICDAR2013 and ICDAR2015 datasets are employed in the experimental study with F-measures reading 88% and 61% respectively. In another recent work, Wang et al. [58] present a framework based on conditional random field (CRF) to detect text in scene images. The authors define a cost function by considering

the color, stroke, shape and spatial features with CNN for effective detection of textual regions. Evaluations are carried out on various ICDAR datasets and a detection rate of up to 77% is achieved.

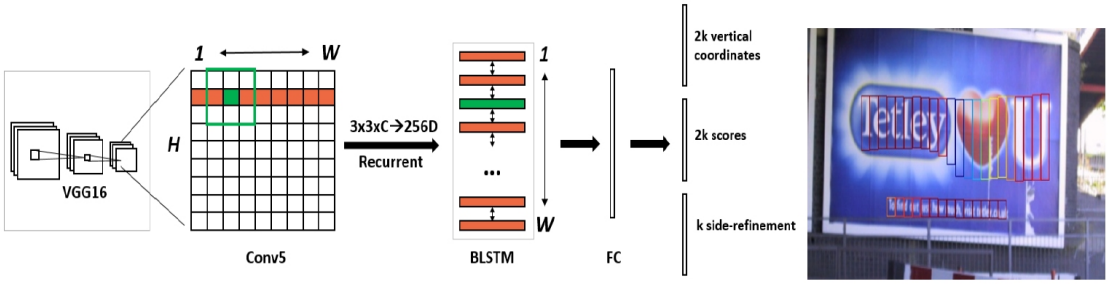


Figure 2.18: Connectionist Text Proposal Network (CTPN) (Image Source [150])

In another interesting work, Zhang et al. [151] presented a character proposal network (CPN) (Figure 2.19) which is inspired by the accelerated speed and high capacity of an FCN. The network is designed to predict the character-ness scores and localize the candidate characters. The score is subsequently employed to eliminate the non-text regions and improve the localization accuracy. The network is evaluated on SVT, ICDAR2013 and Chines-2K datasets with recall rates of 93.6%, 93.88% and 96.46% respectively. Likewise, Yao et al. [152] propose to produce a pixel-level prediction map which is subsequently employed for detection. An FCN is employed to estimate the information on text regions, characters and their relationship. The proposed network is able to handle horizontal as well as curved text in scene images. Experimental study reports F-measures of 84%, 65% and 76% on ICDAR2013, ICDAR2015 and MSRA-TD500 datasets respectively.

Among other end-to-end trainable deep neural networks based systems, Liao et al. [153] present a system called 'TextBoxes' which detects text in natural images in a single forward pass network (Figure 2.20). This approach is considered a fast text detector, as it takes 0.09 seconds per image. 'TextBoxes' achieved 85% accuracy on ICDAR2013 and ICDAR2015 datasets. The technique was later extended to 'TextBoxes++' [57] and evaluated on four public databases outperforming the state-of-the-art methods. He et al. [56] improved the convolutional layer of CNNs to detect text with arbitrary orientation by introducing a text-alignment layer that calculates features from text at arbitrary orientations. The proposed method significantly improved the results on ICDAR2013 and ICDAR2015 datasets with F-measure of 90% and 87% respectively. In another end-to-end trainable system [154], an ensemble of CNNs is trained on synthetic data to detect video text in East Asian languages. The proposed study employs consecutive sequences of video frames to identify textual regions. A set of 80 videos containing Chinese text is used in experiments and 98% accuracy is reported.

EAST [155] (Efficient and Accurate Scene Text Detector), is another well-known scene text detector that provides promising results in challenging scenarios. The detector can detect text of any orientation and draws quadrilateral shapes around the text. The CNN architecture of EAST

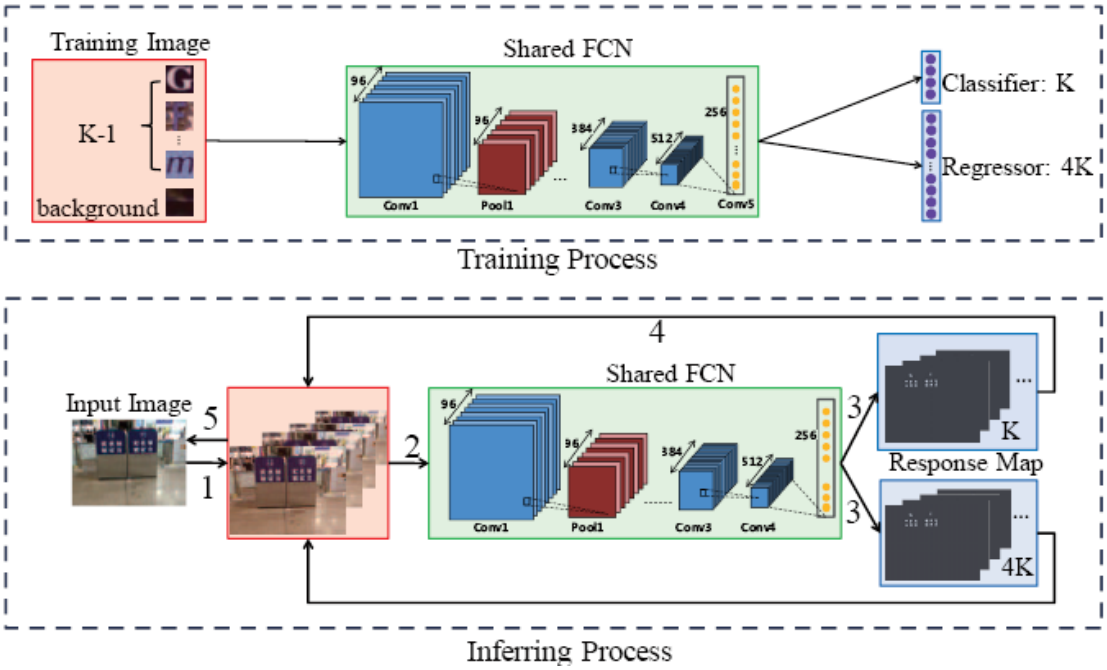


Figure 2.19: Character proposal network presented in [151]

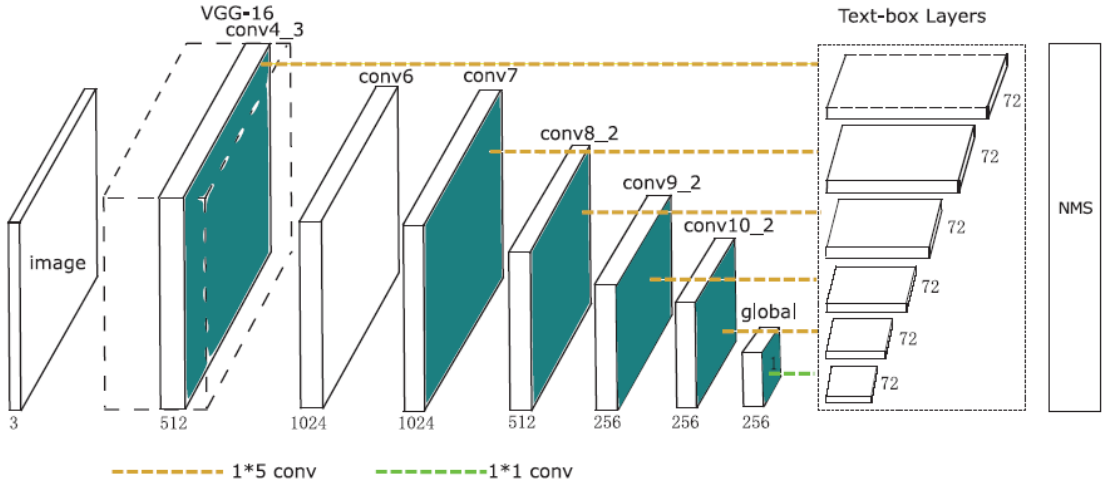


Figure 2.20: CNN architecture of TextBoxes [153]

contains two main phases for detection i.e. Multi-channel-FCN and Non Maximum Suppression (NMS) which finally produces the multi-orient text-lines and word boxes in the images. An F-measure of 78% on the ICDAR2015 and that of 76% on COCO-Text dataset is reported using EAST.

In a relatively recent work on detection of Arabic caption text, Zayene et al. [33] employ a combination of stroke width transform (SWT) with a convolutional auto-encoder (CAE). The method is evaluated on a publicly available dataset AcTiV-DB [156] which contains 1,843 frames with 5,133 text lines from Arabic News channels. The system realizes an F-measure of 84%.

Likewise, Yousfi et al. [157] employ CNNs and multi-exit asymmetric boosting cascade method to detect Arabic text in News videos and reports a detection rate of 97%. In another recent work [3], the authors target detection of moving caption text in videos. A sequence of frames is processed using Hough transform with color based filtering, and the candidate text regions are identified using a ConvNet. Likewise, the caption motion is analyzed using an LSTM and a correlation-based model. Experimental study on a self-collected dataset with multi-language captions News reported promising performance.

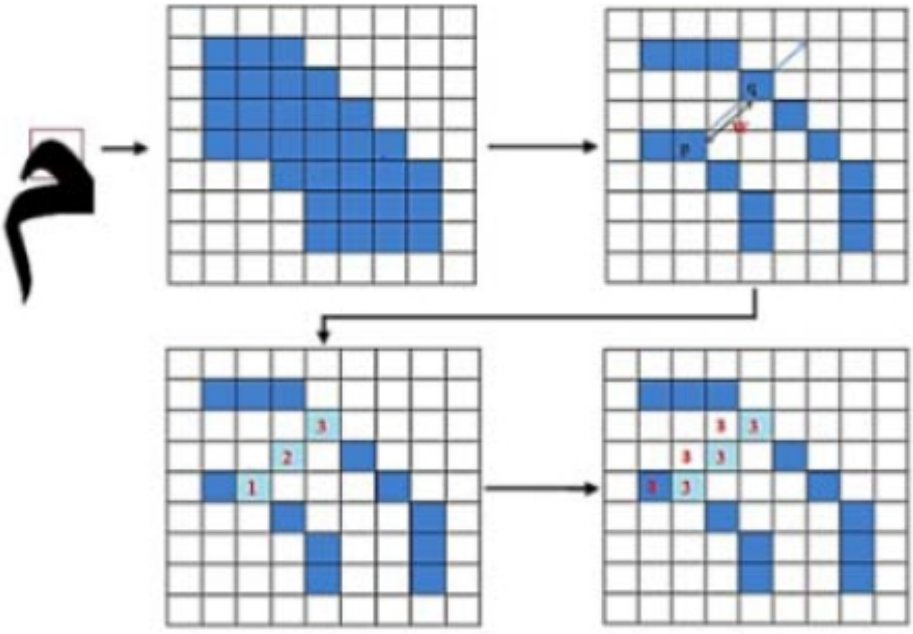


Figure 2.21: Application of stroke width transform (SWT) (Image Source [33])

2.2.2.3 Discussion

A summarized review of the well-known text detection methods is presented in Table 2.2. In pre-deep learning era, ANN and SVM have remained popular choices of researchers in classifying text and non-text regions. Though high detection rates are reported by many such studies, most of these have been evaluated on fairly limited sized datasets. In the recent years, it can be observed that the problem of text detection has been dominated by the application of different deep learning based techniques. The availability of benchmark datasets has also contributed to the rapid developments in this area. Different variants of convolutional neural networks have been thoroughly investigated on various ICDAR datasets. While detection of text in languages based on the Latin alphabet has received significant research attention and is very much mature, detection of cursive text still remains a relatively less addressed and challenging issue. Zayene et al. [33] and Yousfi et al. [157] investigated detection of Arabic text from News videos and the preliminary findings are indeed

promising. From the perspective of Urdu text which is the problem being addressed in our study, Jamil et al. [45] employed an ANN to validate the candidate text regions produced using image analysis techniques. Although an F-measure of 85% is reported, the technique is evaluated on a small set of 500 frames only.

Table 2.2: Summary of supervised text detection methods

Classifier	Study	Technique	Script	Dataset	Results
ANN	Park et al. [131](1999)	Window Filter	English	License plate dataset	90%
	Ye et al. [123](2009)	Bootstrap ANN	Chinese	Private (1,207 video frames)	87%
	Jamil et al. [45](2012)	Textural Features	Urdu	Private (500 video frames)	85%
SVM	Chen et al. [136](2001)	Edge-distance Map	English	Private	98.7%
	Anthimo. et al. [138](2008)	LBP,HAAR,DCT	English	Private (150 video frames)	96.7%
	Anthimo. et al. [96](2010)	LBP	English	Private (217 video frames)	97%
	Sun et al. [139](2006)	Wavelet Transform	Chinese	Private (300 images)	97%
	Zhen and Wei [71](2009)	Wavelet Transform	Chinese	Private (500 video frames)	92.78%
	Unar et al. [142](2018)	Canny & Sobel	Urdu	IPC [143]	85%
	Francis and Sreenath [144](2020)	Least-Square SVM	English	MSRA500,SVT	76%,98.5%
	Moradi et al. [141](2011)	Gaussian Pyramid	Farsi/Arabic	Private (2,871 video frames)	84.57%
	Darab and Rahmati [140](2012)	Wavelet+HoG	Farsi/Arabic	Private (800 images)	86.5%
	Moradi et al. [158, 32](2010,13)	Textural Features	Farsi/Arabic	Private (50 videos)	89.25%
CNN	Huang et al. [146](2014)	MSER	English	ICDAR2011	78%
	Zhang et al. [147](2016)	FCN, Salient-map	English	ICDAR2013,2015,MSRA500	83%,54%,74%
	Gupta et al. [148](2016)	Fully-CRN	English	ICDAR2013	84.2%
	Shi et al. [149](2017)	FCN,SegLink	English	ICDAR2015	75%
	Tian et al. [150](2016)	CTPN	English	ICDAR2013,2015	88%,61%
	Wang et al. [58](2018)	Conditional Random Field	English	ICDAR2015	77%
	Zhang et al. [151](2016)	Character Proposal Network	Chinese	SVT,ICDAR2013,Chines-2K	93.6%,94%,96.5%
	Yao et al. [152](2016)	Fully Convolutional Network	English	ICDAR2013,2015,MSRA500	84%,65%,76%
	Liao et al. [153](2017)	Single FPN	English	ICDAR2013,2015	85%,85%
	Liao et al. [57](2018)	Forward Pass Network	English	ICDAR2013,2015,COCO	80%,82%,56%
	He et al. [56](2018)	Alignment Layer	English	ICDAR2013,2015	90%,87%
	Xu et al. [154](2018)	Ensemble CNNs	Chinese	Private (80 video)	98%
	Zhou et al. [155](2017)	Multi- FCN,NMS	English	ICDAR2015,COCO	78%,76%
	Zayene et al. [33](2016)	SWT,CAE	Arabic	AcTiV [156]	84%
	Yousfi et al. [157](2014)	Multi-exit Boosting	Arabic	Private (Video images)	97%

2.3 Text Recognition Methods

Recognition of text commonly termed as Optical Character Recognition (OCR), is one the most classical pattern recognition problems that has been investigated in images, documents, natural scenes and videos for more than five decades [55]. From recognition of isolated characters and digits to complex end-to-end systems, the domain has matured significantly over the years. Formally, the task of an OCR system is to take a set of pixel data which contains textual information as input and convert it into corresponding string as output. Thanks to the extensive research endeavors, mature recognition systems like Google Tesseract [159] and Abbyy FineReader [160] etc. have been developed reporting near to 100% recognition rates on text in multiple scripts. However, as discussed earlier, recognition of text in cursive scripts still remains challenging especially when it comes to caption text [62].

We will discuss the recognition methods from the perspective of document text, scene text and video text (Figure 2.22). Since text recognition has been investigated for around half a century

now, our discussion will be more focused on cursive text in general and Urdu text in particular. For a comprehensive review on the history and development of OCR systems, readers may refer to resources like [161, 162].

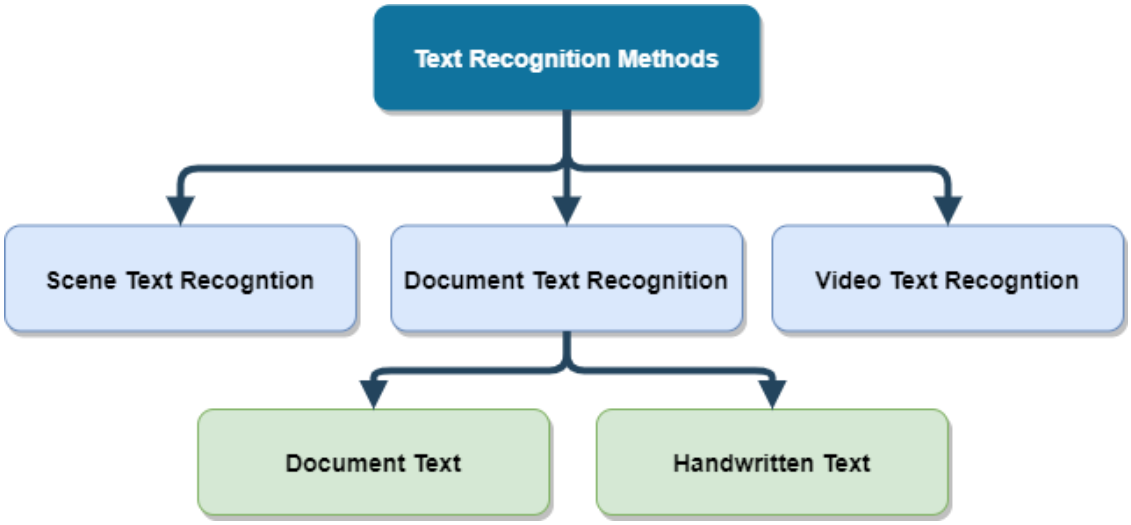


Figure 2.22: Taxonomy of text detection methods

2.3.1 Document Text Recognition

The earliest attempts towards development of recognition systems targeted text (printed as well as handwritten) in document images. As a function of recognition unit employed, recognition techniques are typically categorized into analytical (segmentation-based) [163, 164] and holistic (segmentation-free) [165, 166] methods. In segmentation-based approaches, the image of text is divided into individual characters which are then recognized. Segmentation-free approaches on the other hand recognize text at word or ligature level without segmenting them into characters. The main advantage of segmentation-based approaches is that the number of classes to be recognized is the same as number of characters (and their different shapes) in the alphabet. This number is much smaller when compared to the number of ligatures or words which are units of recognition in segmentation-free approaches (Urdu, for example, has more than 26,000 unique ligatures [167]). Segmentation of text into characters, however, is a complex and challenging problem. Segmentation-free approaches tend to be less complex than segmentation-based approaches in the sense that they do not require segmentation of text into individual characters. These methods are relatively easier to implement but are more prone to noise and minor variations in the patterns. The main challenge with segmentation-free approaches, however, is the large number of classes to be recognized. A recent trend in recognition is use of implicit-segmentation where the learning algorithm is fed with text line images as well as ground truth transcriptions to not only learn character shapes but also the segmentation points [168, 169].

For text recognition in document images, a large number of techniques have been presented both at character (analytical) and word (holistic) levels. Among methods employing characters as units of recognition graph-based models [170, 171, 172, 173], Bayesian classifier [174, 175, 176] and Hidden Markov Models (HHM) [177, 178, 179, 180] etc. have been typically investigated. Among holistic or word level recognition techniques, a wide variety of features as well as classifiers have been studied [181, 182, 183, 184] reporting high recognition rates. Deep learning has also been employed for feature extraction and classification at character and word levels [185, 186, 187, 188, 189].

From the view point of recognition of cursive scripts and more specifically Urdu text in document images, significant research efforts have been made in the last few years. These systems primarily target scanned document images of printed text in Nastaliq script. Due to challenges already discussed, implicit segmentation based techniques have remained a popular choice of researchers [168, 169, 190, 191]. Likewise, in case of holistic approaches, ligatures have been typically employed as recognition units [165, 38, 166].

The initial research endeavors on recognition of Urdu text mainly targeted isolated characters [192, 193] or already segmented ligatures [194]. Among significant holistic approaches, HMMs have been widely employed for recognition of ligatures [195, 196, 197, 198]. These techniques use the sliding windows to extract features from ligature images which are projected in the quantized feature space hence representing each ligature image as a sequence. In some cases, the main body and dots are separately recognized [165] to reduce the total number of unique classes which can be very high in case of Urdu text. A number of holistic techniques are based on word spotting [40, 199] rather than recognition, to retrieve documents containing words similar to those provided as query. Recently, recognition of Urdu handwriting in documents has also been explored in number of studies [200, 201, 41, 42, 43].

Among one of the earliest works on Urdu text, Pal et al. [163] presented structural, shape and water reservoir features for recognition of isolated characters and numerals. The document image is first binarized and skew is corrected using Hough transform. Traditional projection profile methods are then employed to segment the text lines. The authors report a character recognition rate of more than 97%. The system, however, works only on isolated characters and cannot be employed for words or lines directly. The authors in [192] propose a feed forward neural network-based method for recognition of individual printed Urdu characters. The system evaluated on isolated Urdu characters in Arial font reports 98% classification rate. In another study on isolated characters [71], the authors employ structural features to distinguish character shapes. Classification is followed by dictionary matching and an overall recognition rate of 97% is reported.

Sardar & Wahab [194] present a recognition system that performs skew correction, line extraction and ligature segmentation (into primary and secondary ligatures). Recognition is carried out

using Hu's moments and text to non-text pixel density features. These features are computed from sliding windows of varying sizes. Classification using nearest neighbor classifier reports an accuracy of 97%. Likewise, Nawaz et al. [202] remove dots and diacritics from isolated characters creating separate classes of secondary and primary ligatures. Chain code based features are then extracted to separately recognize main body and secondary ligatures which are subsequently re-associated to recognize the true characters. In another study, Ahmed et al. [203] classify each character as simple, semi-complex or complex using shape information and perform recognition using a neural network. The system reports 93.4% recognition rate in the absence of diacritics.

In another study [204], a holistic approach is employed for recognition of Arabic (Naskh) and Urdu (Nastaliq) text. Ligatures are characterized using shape descriptors and are classified using nearest neighbor classification. Recognition rates of 91% and 86% on Urdu and Arabic text respectively are reported. The paper also introduce the Urdu printed Text Images (UPTI) dataset containing around 10,000 text line images with ground truth transcription. UPTI is considered as one of the benchmark datasets for the evaluation of printed Urdu OCR systems. The text lines, however, are synthetically generated using an Urdu text tool hence the dataset does not offer the same kind of challenges as those encountered in scanned document images.

In a series of related studies [205, 164, 206], the Center of Language Engineering (CLE) team targeted recognition of Urdu ligatures in the CLE dataset [207]. Javed et al. in [205] present a holistic technique where features based on DCT are extracted from ligatures using sliding windows. Primary and secondary ligatures are separately recognized using HMMs and subsequently a set of rules is employed to associate the dots with the parent primary ligature. Experiments on a dataset of 3655 ligatures report a recognition rate of 92%. In a later study, Javed and Hussain [164] employ a segmentation-based technique with HMMs followed by a rule-based post processing to recognize main body ligatures (without diacritics). An accuracy of 92.7% is reported on printed and then scanned images. Subsequently, the Google Tesseract recognition engine was enhanced to recognize Nastaliq text in two font sizes (14 and 16) [206]. Experiments on a set of around 1500 unique ligature classes revealed that while the original Tesseract engine reported 66% recognition rate, the modification performed by the authors enhanced the recognition rates to 97%. The technique, however, suffers from the drawback that the recognition system needs to be trained for every font size separately.

Another holistic recognition method, where primary and secondary ligatures are separately recognized, is presented by Khattak et al. [208]. The authors employ sliding windows to extract features from ligature images which are then employed to train HMMs – a separate model for each ligature class. Primary and secondary ligatures, once recognized, are associated together using a comprehensive post processing step. A set of more than 2000 high frequency Urdu ligatures is employed in evaluations and a ligature recognition rate of 97.93% is reported. In a subsequent work, the authors replaced the HMM-based recognition with CNN-based recognition [38].

Similar to [208], the main body and secondary ligatures are separately recognized, by a CNN trained from scratch and then re-associated to recognize the complete ligature. The system is evaluated on UPTI [63] and CLE [207] datasets with recognition rates of 97.81% and 89.20%, respectively. A similar technique is presented by Rehman & Hussain [39] where the authors employ a CNN for font-independent ligature recognition with an accuracy of 84.2%. Likewise, Arafat & Iqbal [209] extracted features using Alexnet and VGG16 from 46,000 synthetically generated ligatures. Extracted features are then fed to BLSTM for recognition and a recognition rate of 70% is reported on a test set of 7,000 ligatures. Another recent study is presented in [210] where ligatures are segmented from text lines and a set of 15 features is extracted from each ligature image. Hierarchical clustering is carried out in the feature space and a genetic algorithm optimizes the classification rules. The technique reports a promising ligature recognition rate of 96.72%.

As discussed earlier, the large number of ligature classes and the challenges in segmentation of ligatures into characters resulted in shifting the research attention towards implicit segmentation based techniques. These methods exploit the deep learning architectures which are fed with images of text lines and the respective transcription to learn the character shapes and boundaries. Among these methods, Ahmed et al. [211] applied bidirectional Long Short-Term Memory (LSTM) network for character level evaluation of cursive as well as non-cursive scripts. Experiments were carried out on UPTI dataset for Urdu and UNLV-ISRI dataset for the Roman script and character-level recognition rates of 89% and 99.17% are reported for Urdu and Roman text respectively. In a similar work, Adnan et al. [212] fed raw pixels to a bidirectional LSTM with a connectionist temporal classification (CTC) layer. Experiments with and without considering the shape variations of characters reported recognition rates of 87.4% and 94.85% respectively.

In another series of significant contributions towards Urdu recognition systems, Naz et al. [168, 213] employ a set of statistical features that are fed to MD-RNNs for training and evaluation. Evaluations on the UPTI dataset with 6800 lines in the training and 1600 in the test set realize character recognition rates of 94.97% and 96.40% in [168] and [213] respectively. Later, the statistical features were replaced by features learned through a CNN which improved the recognition rate to 98.12% [190].

In addition to printed Urdu text, few recent endeavors have been made to recognize Urdu handwritten text as well. Hassan et al. [41], for instance, employ a combination of CNN with RNNs to recognize handwritten text lines in Urdu. Experimental study on a collection of 6000 text lines reported a character recognition rate of 84%. In another recent work, Anjum & Khan [214] employ a deep learning based encoder /decoder framework with attention mechanism to recognize Urdu text lines. Authors demonstrate that incorporating attention mechanism significantly improves the recognition performance both at character and word levels. Khan et al. [215] investigate different CNN architectures recognize isolated handwritten Urdu words. Although the study reports word recognition rates of up to 96%, the dataset comprises only 5 unique words with a little more than

1000 samples per word. In addition to recognition, few word spotting based systems have also been investigated for Urdu text. Among these, Sabbour et al. [216] proposed a set of compound features with SVM classifier to spot Urdu words in a set of documents. In another study, Abidi et al. [217] extract a set of profile and projection features from Urdu ligatures and employ Dynamic Time Warping (DTW) to compare two ligatures. Evaluations are carried out in a retrieval framework on a set of 50 Urdu documents and realized a recall of 95.17% while a precision of 94.3%.

Among other cursive scripts, recognition of Arabic text from document images has been studied in a number of studies [218]. Challenges in recognition of Arabic text are more or less similar to those of Urdu text. Arabic, however, is mostly printed in the Naskh style as opposed to the common Nastaliq style of Urdu. Segmentation of text into characters in Naskh is relatively less complex as compared to the diagonal Nastaliq text. IFN/ENIT [219] is the most widely employed dataset for evaluation of Arabic text recognition systems. Both hand-crafted [196, 220] and machine-learned [221] features have been investigated. HMM has also been applied in various studies [222, 223, 178, 179, 180] for Arabic handwriting. More details on recognition of Arabic and similar scripts can be found in [218, 224, 225, 226].

It can be observed from the above discussion that the literature is quite rich when it comes to recognition of text from document images. Indeed thanks the decades of research endeavors of the pattern recognition community, text recognition from document images is highly mature today with many commercial recognition engines available. On the contrary, recognition of text in scene images and video frames still remain open research problems and are discussed in the subsequent sections.

2.3.2 Scene Text Recognition

Contrary to document images, recognition of text from scene images is much more challenging due to camera perspective, varying lighting conditions and unconstrained backgrounds. Scene text recognition is typically employed for applications like robot navigation, self-driving cars and assistance to the visually impaired. Among well-known studies on this problem, Histogram of Oriented Gradients (HOG) [227, 228, 229], Strokelets [230, 126], and SIFT descriptors [231, 232] have been employed as popular features in a number of studies. For classification, similar to text detection, ANNs [233, 234, 235] and SVMs [236, 237] have been widely investigated. A special case of scene text recognition is the recognition of text on road signs and has also been investigated in a number of studies [238, 239, 240, 241, 242]. Word spotting based approaches have also been employed on scene text images [243, 244, 245] in the literature.

In a number of relatively recent studies, combination of CNNs with RNNs (and different variants) has been effectively applied to text recognition from scene images [246, 247, 248, 249, 250, 251]. Further advancements in deep learning techniques led to the development of more

sophisticated architectures. Notable of these include Binary Convolutional Encoder-Decoder Network (B-CEDNet) using Bidirectional-RNN [252] (Figure 2.23), Character-Aware Neural Network (Char-Net) using LSTM [253] (Figure 2.24), Character Attention Fully Convolutional Network (CA-FCN) [254] and Double Supervised Network (DSAN) [255]. These DNN based models claim to be robust and efficient for scene text recognition in challenging scenarios.

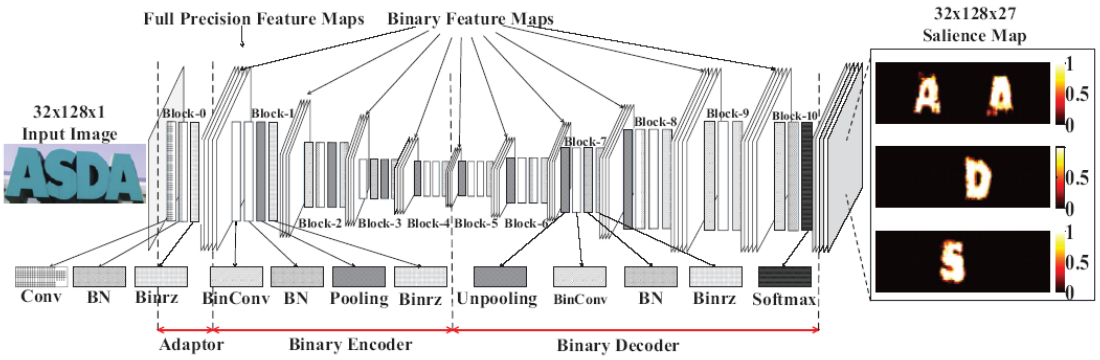


Figure 2.23: Binary Convolutional Encoder-Decoder network (B-CEDNet) [252]

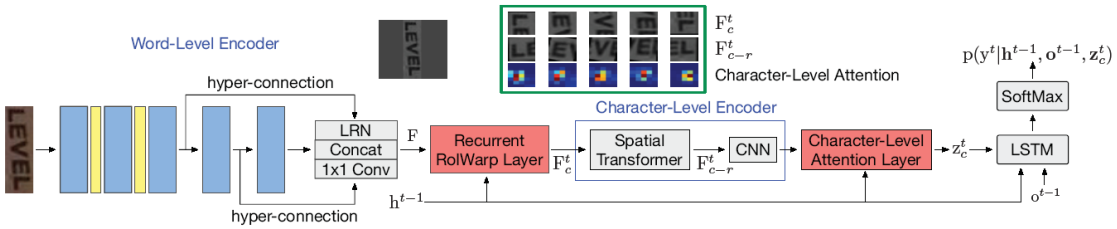


Figure 2.24: Architecture of Char-net (Image Source [253])

From the perspective of cursive text, a comprehensive survey on recognition of Arabic scene text is presented in [35]. the authors discuss the challenges in recognition of cursive text and emphasize on the need of a benchmark cursive scene text dataset. From the view point of Urdu text, Chandio et al. [256, 257] introduced a dataset of 2500 natural scene images with occurrences of Urdu text. In addition to Urdu, the images also contain instances of Sindhi and English text. The dataset is divided into three parts, images of isolated characters, cropped word images and the text spotting set of complete images. The dataset was evaluated by applying the latest deep learning based techniques for detection and recognition of text. In a subsequent study by the same authors [258], a hybrid deep neural network with skip connections is employed. The network combines a CNN with an LSTM and reports a recognition rate of 61.35% on 11,500 isolated word images.

Similarly, Panhwar et al. [259] target recognition of text in signboards, primarily focusing on English and Urdu text. An ANN is employed for recognition and a recognition rate of 85% is reported on a collection of 500 natural scene images. Likewise, Arafat & Iqbal [260] employed

a two-stream deep neural Nnetwork (TSDNN) to recognize Urdu ligatures from 4,200 natural scene images and 51,000 synthetically generated images and achieved recognition rates of 94.90%, 95.20% respectively. The work was later extended [261] to introduce a dataset of more than 30K natural scene images with 467 ligature categories. An 18-layer CNN was trained to recognize the ligatures and a classification rate of more than 97% was reported in this study.

After having discussed recognition of text in scene images, we now present a discussion on recognition of caption text from video frames in the following section.

2.3.3 Video Text Recognition

Caption (artificial) text, as elaborated previously, is superimposed on video and is typically employed for indexing and retrieval applications. While the challenges related to camera perspective and non-homogeneous backgrounds are not encountered in case of artificial video text, a major recognition challenge is the low resolution of text. In case of non-homogeneous backgrounds, segmentation of text from background prior to recognition can also be challenging.

Among one of the earlier endeavours towards the development of a video OCR, a combination of holistic and component-level approaches is presented in [262] for recognition of Korean characters appearing in video frames. The holistic approach exploits the global shape information of a character while the component-level analysis employs the local shape information in segments of characters. Recognition rate of more than 96% is reported on 50,000 character images. In another pilot study on recognition of Chinese caption text, Tang et al. [263] use a fuzzy-clustering neural network for recognition purposes and report an accuracy of 86% on News channel videos. Likewise, a step-wise language model is incorporated with an ANN to develop a character by character video OCR in [264]. A dataset of 12 videos from French News channels is used for evaluation purposes reporting 95% character and, 78% word recognition rate. Later on the work is improved by applying BLSTM with CTC. This time 32 videos containing french text are employed and 97.35% character recognition rate is achieved.

In another notable work [265], structural features of characters are employed for recognition of caption text. TRECVID dataset is used for validation with 1,462 different characters. The authors demonstrated that with only 10% samples of each character class in the training set, high recognition rate (94.5%) is achieved. Recognition of video text for indexing and retrieval applications has also been investigated in a number of studies. Khatri et al. [266], for instance, implemented a recognition system for lectures and News videos while Kulkarni et al. [267] targeted text recognition from cartoon, sports and military videos to develop a video-based search engine.

Among relatively recent studies, Bhunia et al. [61] employ an SVM to select and appropriate color channel for video text recognition. The recognizer is based on hidden Markov models with histogram of oriented gradients computed from the selected color channel. The technique is

evaluated on various public English datasets and a dataset of text in Devanagari script and, word recognition rates of 75.41% and 71.14% are reported on English and Devanagari text respectively.

In deep learning-based methods, Lu et al. [268] employ transfer learning with pre-trained CNNs for video text recognition. Models including InceptionV3, VGG16 and Resnet50 are considered for transfer learning and the performance is evaluated on multiple datasets. A similar study for recognition of video text in East Asian languages is presented in [154]. Characters in simplified as well as traditional Chinese are recognized using an ensemble of CNNs (Figure 2.25) and recognition rates of more than 98% are reported for both the scenarios. Dutta et al. [60] came up with an idea of recognizing the text from lecture videos. They developed a dataset called LectureVideoDB with 5,000 video frames of papers, slides and boards (black & white) with the word bounding boxes and their corresponding ground truth. The architecture of CNN and RNN is developed and fine-tuned on two well known datasets called IAM Handwriting [269] and MJSynthetic [233]. The proposed scheme is able to achieve 64.48% and 86.08% word and character recognition rates on the developed dataset, respectively.

From the view point of recognition of Arabic caption text, Halima et al. [270] applied KNN classifier on features extracted from segmented Arabic characters from News channel videos. The work was later improved with fuzzy-KNN [271] enhancing the recognition rate to 95%. Another notable study is presented by Yousfi et al. [272] where recognition is carried out using deep neural networks. A combination of CNN and deep auto-encoders is used for feature computation and the extracted features are subsequently fed to a BLSTM for recognition. A comprehensive dataset named 'ALIF' [273], with more than 6500 video frames containing Arabic caption text was also developed as a part of this study. Few sample images of the dataset are illustrated in Figure 2.26. A character recognition rate of 94.36% is reported on this dataset using the proposed technique. BLSTM with combination of CTC is also investigated by the authors in [34].

Another similar dataset called 'AcTiV' is presented by Zeyene et al [156] that comprises of video frames collected from four different Arabic News channel videos. The dataset contains bounding box information of textual regions (for evaluation of text detection) as well as the corresponding transcription (to evaluate recognition). The dataset was employed by the authors in a subsequent study [31] where Multidimensional LSTMs (MD-LSTM) were employed for recognition. In addition to their AcTiV dataset [156], the authors also evaluated the recognition engine on the ALIF dataset [273] with character recognition rates of 96.48% and 96.85% respectively.

In another work targeting recognition of Arabic caption text, Jain et al. [274] employed both the dataset (i.e. ALIF and AcTiV) in their study. An end-to-end, hybrid architecture of CNN and RNN is proposed that is able to recognize Arabic text from natural scenes as well as video frames. The reported character recognition rates read 98.17% on ALIF and 97.44% on AcTiV dataset.

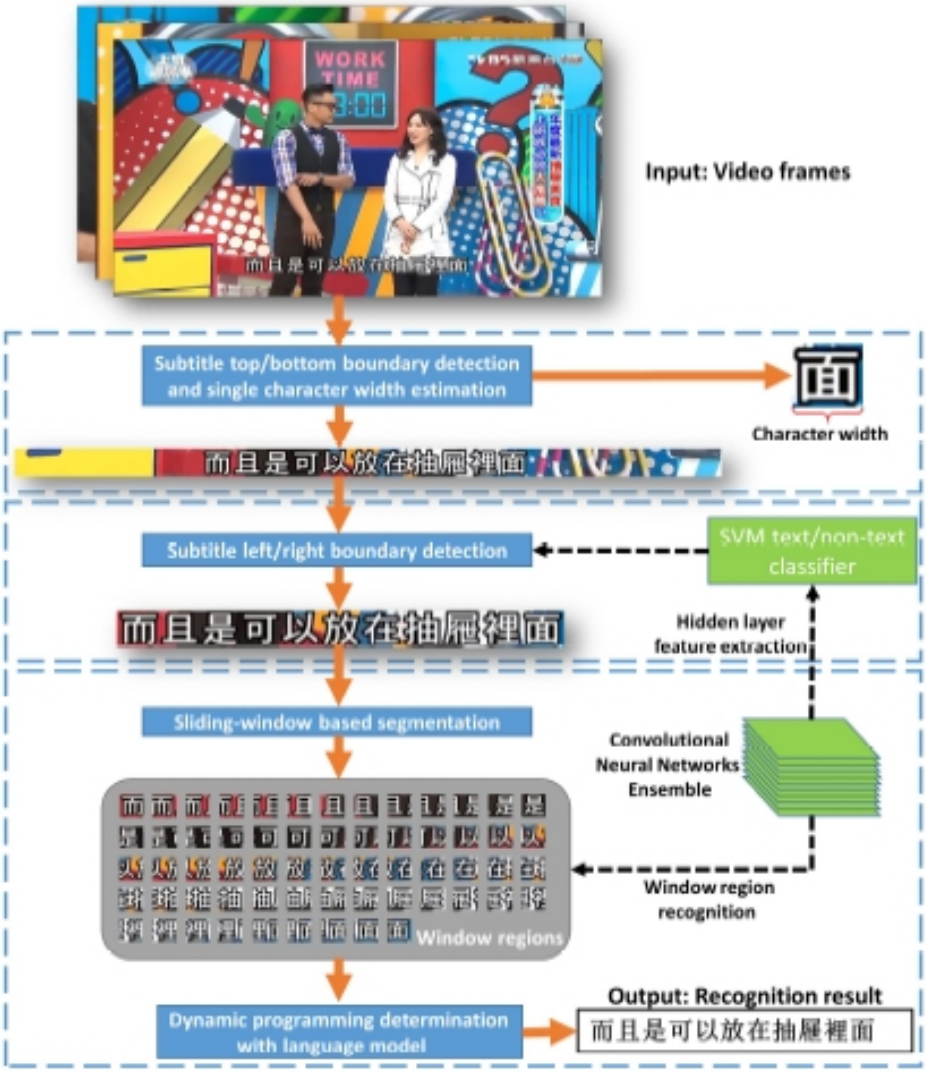


Figure 2.25: Overview of CNN Ensemble employed in [154]

Figure 2.26: Sample images in 'ALIF' dataset [273]

The literature is relatively limited when it comes to recognition of Urdu caption text. In a pilot study on this problem, holistic recognition technique is presented by Hayat et al. [48] where a number of pre-trained CNNs are employed to recognize a small set of 290 ligature classes. Though

very high ligature recognition rate (99.5%) is reported, the number of ligature classes is very small for employment in real world scenarios. In another recent study [49], Bi-Directional LSTMs are employed for recognition of Urdu News tickers. The technique is evaluated on a custom developed dataset and the performance is compared with a commercial recognition engine. Experiments on 19,824 text lines report a recognition rate of 93.02%.

2.3.4 Discussion

After having discussed text recognition methods, we now present a summary of prominent contributions to this problem in Table 2.3. Though the primary target of our current research is Urdu caption text, for completeness we have also discussed recognition of text in document images, natural scenes as well as handwriting images. Naturally, the objective is not to provide an exhaustive review of literature in all these related yet different problem areas, but to cover the breadth of knowledge in this highly investigated research problem. More focus, of course, lies on recognition of cursive text in general and Urdu text in particular.

From the perspective of text in languages based on the Roman script, recognition of caption text is relatively less challenging and major proportion of the recent research targets recognition of scene text [254, 254, 252, 259]. Similar to detection, different ICDAR datasets have been a popular choice of researchers for evaluation purposes [275, 276, 277, 146, 147, 148, 149, 150, 58]. Arabic text recognition has been investigated for printed document images as well as handwriting images for many years [224, 218, 223, 225, 226, 178, 179, 180, 221], the interest in caption text however, is relatively recent [271, 274, 34, 31]. The development of two benchmark datasets, ALIF [273] and AcTiv [156] also contributed to enhance the research attention in recognition of Arabic caption text.

From the view point of Urdu text, a number of techniques have been presented for printed document images [40, 62, 278, 37, 279]. While the initial research endeavors primarily relied on holistic methods, analytical techniques based on deep learning have mostly been proposed in the recent years [41, 42, 43, 213, 38, 39]. For video text, few pilot studies have been carried out for recognition of scene [209, 260, 256, 259, 258] as well as caption text [48, 49]. The work reported in [48] for recognition of Urdu caption text relies on a holistic approach and considers a very small subset of Urdu ligatures. Likewise, the work by Tayyab et al. [49] focuses on News tickers only which appear mostly on homogeneous backgrounds. The experimental study is also carried out on a private dataset.

Table 2.3: Summary of Text Recognition Methods

Image Type	Study	Technique	Script	Dataset	Results
Scene	Gao et al. [255](2018)	DSAN	English	IIIT5K,ICDAR2013	88.6%,92.3%
	Liao et al. [254](2018)	Char-Net LSTM	English	IIIT [280]	92%
	Liu et al. [252](2018)	B-CEDNet	English	ICDAR2003	98.4%
	Chandio et al. [256](2020)	BLSTM	Urdu,Sindhi	Private (2,500 images)	78%
	Ali et al. [258](2019)	CNN-LSTM	Urdu	Private (11,500 images)	61.35%
	Panhwar et al. [259](2019)	ANN	Urdu,English	Private (500 images)	85%
	Arafat and Iqbal [260](2020)	TSDNN	Urdu	Private (4,200 images)	94.90%
	Wang et al. [251](2017)	CNN-RNN	English	ICDAR2015	60.25% (WRR)
Handwritten	Ahmed et al. [200](2019)	BLSTM	Urdu	UNHD [200](2019)	90.72%
	Hassan et al. [41](2019)	CNN-LSTM	Urdu	Private (6,000 lines)	83.69%
	Husnain et al. [42](2019)	CNN	Urdu	Private (38,400 chars)	96.04%
	Ali et al. [43](2020)	DAE+CNN	Urdu	Private (45,000 chars)	82.7%
	Ahmed et al. [201](2019)	CNN-MDLSTM	Urdu	UNHD [200]	93%
	Khemiri et al. [196](2015)	HMM,DBN	Arabic	IFN/ENIT [219](2002)	94%
	Abandah et al. [221](2014)	BLSTM	Arabic	INF/ENIT [219]	98%
Document	Ahmed et al. [220](2016)	HMM	Arabic	Private	97.11%
	Pal et al. [163](2003)	Structural Features	Urdu	Private	97%
	Inam et al. [192](2007)	ANN	Urdu	Private	98%
	Sardar and Wahab [281](2010)	KNN	Urdu	Private	97%
	Ahmed et al. [203](2007)	ANN	Urdu	Private	93.4%
	Javed and Hussain [197](2013)	DCT,HMM	Urdu	Private (20 LC)	92.7%
	Hussain et al. [282](2015)	Shape Features	Urdu	Private	95%
	Javed et al. [283](2010)	DCT,HMM	Urdu	Private (4,937 Ligatures)	92%
	Khattak et al. [208](2015)	HMM	Urdu	Private (8,112 Ligatures)	97.93%
	Sabbour et al. [63](2013)	KNN	Urdu	UPTI [63]	89%
	Ahmed et al. [211](2016)	BLSTM	Urdu,English	UPTI [63],UNLV-ISRI	89%,99.2%
	Adnan et al. [212](2013)	BLSTM-CTC	Urdu	Private	94.85%
	Naz et al. [168](2016)	MDLSTM-CTC	Urdu	UPTI [63]	96.4%
	Uddin et al. [38](2019)	CNN	Urdu	UPTI [63],CLE [207]	97.8%,89.2%
	Rehman and Hussain [39](2020)	CNN	Urdu	Private	84.2%
	Arafat and Iqbal [209](2019)	CNN-BLSTM	Urdu	Private (46k Ligatures)	70%
Video	Lee et al. [262](2008)	Shape Features	Korean	Private (50 Videos)	96.5%
	Tang et al. [263](2002)	FCNN	Chinese	Private	86%
	Elagouni et al. [264](2011)	ANN	French	Private (12 Videos)	95%
	Elagouni et al. [59](2012)	BLSTM-CTC	French	Private (32 videos)	97.35%
	Halima et al. [270](2010)	KNN	Arabic	Private	91.85%
	Halima et al. [271](2013)	Fuzzy KNN	Arabic	Private	95%
	Shivakumara et al. [265](2011)	Structural Features	English	TRECVID [24]	94.5%
	Bhunia et al. [61](2018)	SVM,HMM	English,Indic	ICDAR2013,MSRA	75.4%,71.1%
	Xu et al. [154](2018)	CNN ensembles	Chinese	Private (80 Videos)	98.3%
	Yousfi et al. [272](2015)	CNN-DAE,BLSTM	Arabic	ALIF [273]	94.36%
	Yousfi et al. [34](2017)	BLSTM-CTC	Arabic	ALIF [273]	89.3%
	Dutta et al. [60](2018)	CNN-RNN	English	LectureVideoDB	86.08%
	Jain et al. [274](2017)	CNN-RNN	Arabic	ALIF [273],AcTiV [156]	98.2%,97.4%
	Zayene et al. [31](2018)	MDLSTM	Arabic	ALIF [273],AcTiV [156]	96.5%,96.9%
	Hayat et al. [48](2018)	CNN	Urdu	Private (290 LC)	99.5%
	Tayyab et al. [49](2018)	CNN-LSTM	Urdu	Private (19,824 text lines)	93.02%

2.4 Challenges in Video Text Detection and Recognition

A critical review of the literature presented in the previous sections reveals that for caption text in non-cursive scripts, detection and recognition have been thoroughly investigated and systems reporting high detection and recognition performance have been developed. The recent focus of the community from the perspective of text in non-cursive scripts is on the more challenging scene text detection and recognition which are marked by challenges like camera perspective, non-uniform illumination, complex backgrounds, text in different orientations and occlusion etc. The organization of different International competitions in conjunction with various editions of ICDAR as well as the public availability of the competition datasets has greatly contributed to

advance research efforts on this problem[284, 285, 286, 287]

As opposed to scanned documents, text in videos is of low resolution and may occur on complex backgrounds making its detection a challenging task, irrespective of the script. Text may occur in different font styles, sizes and at arbitrary positions in the video frame. While the latest deep learning based detectors are able to handle these challenges, many such techniques assume presence of text in a single script with a video frame. In most of the News channel videos however, it is common to have bi-lingual text (for instance English and Urdu text in most of our local News channels). While textual occurrences in different scripts are visually distinct, they also share some common properties. At the same time, script of detected text also needs to be identified so that subsequent processing (recognition) can be carried out accordingly. An open research question in such cases is whether to develop a generic script-independent text detector and subsequently identify the script or, to identify/learn script-dependent features allowing detection of text in the given script(s).

From the view point of recognition of Urdu text, a number of techniques exploiting the recent advancements in deep learning have been proposed in recent years [168, 213, 165]. It is however important to note that most of these techniques target document images and have been evaluated on either UPTI [204] or CLE dataset [207]. Though character recognition rates of as high as 98% have been reported on UPTI dataset, it is worth mentioning that the dataset comprises synthetically generated Urdu text line images. CLE, on the other hand, represents a more realistic scenario and consists of two parts, a collection of printed then scanned high frequency ligatures and digitized pages from Urdu books. Nevertheless, as mentioned earlier, text in document images does not offer the same kind of challenges as those of caption text. Videos, especially those uploaded on video sharing portals, are compressed and of relatively low resolution. While these aspects make recognition challenging in any script, the problem is much more complex in case of cursive scripts as illustrated in Figure 2.27.

Though detection and recognition of Arabic text has been investigated in the literature [31], the character set of Urdu (39 letters) is a super set of that of Arabic (28 letters) and the diagonal Nastaleeq style of Urdu is much more complex as opposed to the Naskh style of Arabic [62]. In addition to more complex character shapes, association of secondary ligatures (dots) with their parent primary ligature is much more challenging in the Nastaleeq style as compared to Naskh. An analysis of the existing work on Urdu text detection and recognition and, the associated challenges discussed in the preceding paragraphs, led us to identify the following research gaps which call for in-depth investigations.

- There is a need of a comprehensive dataset with ground truth information to support algorithmic development and evaluation of Urdu caption detection and recognition systems. Presently, a small dataset of about 1000 video frames is publicly available [288]; not only the

Figure 2.27: Low resolution caption text examples

dataset is too small to take advantage of the recent advancements in deep learning based solutions, the ground truth information is only provided for localization of textual regions and no transcription is available. The availability of a large labeled dataset is likely to trigger significant research on this problem similar to the impact of ICDAR scene text [289, 275, 276, 277], and Arabic caption text datasets ALIF [273] and AcTiV [156].

- Detection of Urdu caption text, especially in the context of our local News channels with bilingual textual occurrences, needs to be thoroughly investigated. Few preliminary studies based on unsupervised [47, 44, 46] as well as supervised [45, 142] techniques have been carried out but the methods are developed as well as evaluated on fairly limited sets of images. These conventional techniques rely on image analysis and traditional classifiers, supported by a set of heuristics, and are not robust to varying text sizes, complex backgrounds and low resolution of text.

- Few pilot studies [48, 49] targeted recognition of Urdu caption text. However, similar to the detection problem, a scalable and robust solution that can cope with the recognition challenges of video text as well as those of cursive scripts, needs to be investigated.

- The detection and recognition of Urdu caption text needs to be enhanced to an extent where rather than mere pilot studies real world applications can be developed on top of the detection and recognition engines. Textual content in videos is typically employed for applications like smart indexing and retrieval, generation of alerts on keywords, News summarization etc. Such systems are already in use by media houses and regulatory bodies in developed countries and a similar solution targeting local needs is very much desirable.

2.5 Summary

This chapter presented a comprehensive overview of detection as well as recognition of text. We first discussed the text detection methods as a function of detection technique i.e. unsupervised and supervised methods. It was concluded that deep learning based supervised techniques are much more robust and can better handle the variations in text size, font and contrast etc. Likewise, for recognition of text, we discussed the problem from the perspective of scene text, document images and caption text. Similar to detection, it was observed that deep learning-based analytical techniques based on implicit-segmentation outperform the holistic or explicit segmentation based techniques especially in case of cursive scripts. In the next chapter, we introduce the dataset collected and labeled as a part of this study.

Chapter 3

Data Collection and Labeling

3.1 Introduction

Availability of labeled dataset is of utmost importance for algorithmic development and evaluation of any computerized system. With reference to our problem of detection and recognition of Urdu text in videos, a dataset of 1000 video frames, the IPC dataset [143] is publicly available. However, the dataset is only labeled from the perspective of detection and does not support evaluation of recognition systems. Likewise, the size of the dataset is relatively small considering the requirements of deep learning methods. Consequently, we opted to collect and label a customized dataset supporting both detection and recognition. The first step towards development of a comprehensive labeled dataset is data collection. In our study, data refers to collection of videos. We have collected videos from multiple News channels (details are presented later in the chapter). All videos are recorded at a resolution of 900×600 and a frame rate of 25 fps. From the view point of textual content in the video, video frames need to be labeled from two perspectives, detection and recognition. Detection performance refers to how good the system is in locating the textual occurrences in a video frame while recognition performance refers to the effectiveness of the system to convert images into text. To evaluate the performance of detection system, bounding box of each textual region in the image must be identified and stored. Likewise, to evaluate the recognition performance, the transcription associated with each textual region needs to be stored as ground truth information.

In the following sections, we introduce the developed dataset which we have named as 'UTiV'. We also discuss in detail the evaluation metrics, the salient features for the collected data along with statistics and the developed labeling tool.

3.2 Evaluation Metrics

In the literature, several evaluation metrics have been proposed to evaluate the performance of text detection methods [44, 289, 290]. In our system, for evaluation of the text detection module, we

employ the most commonly used area based precision and recall measures reported in [44] and defined as follows.

Let A_E be the estimated text area given by the system and A_T be the ground truth text area, then the precision P and recall R are defined as:

$$P = \frac{A_E \cap A_T}{A_E} \tag{3.1}$$

$$R = \frac{A_E \cap A_T}{A_T} \tag{3.2}$$

The precision and recall measures can be combined in a single F-measure as follows.

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.3}$$

The same idea can be extended to multiple images by simply summing up area of intersection and dividing by the total ground truth area (in N images) for recall and the total detected area for precision. To compute these measures, for each frame, we need to store the actual location of the textual content. The text detected automatically by the system can then be compared with the ground truth text regions to compute precision, recall and F-measure. The idea is illustrated in Figure 3.1. Figure 3.1-a illustrates an example where the text regions detected by the system are shown while Figure 3.1-b illustrates the ground truth text locations for the given frame. The detected and ground truth text regions can be compared to compute the metrics defined earlier and quantify the detection performance.

Figure 3.1: Text regions in an image and the corresponding ground truth image

Unlike many other languages where text can be easily tokenized into words, segmenting Urdu text into words is highly challenging. Spaces appear between words as well as between the ligatures within a word. Consequently, text recognition performance is quantified using recognition rates computed either at ligature or character level (depending upon the recognition technique employed). In case of holistic techniques employing ligatures as recognition units, ligature recognition rate is computed as the ratio of correctly recognized ligatures to the total number of query ligatures. Analytical techniques employing individual characters as recognition units, however, require more sophisticated metrics for computation of character recognition rates. In most cases, the Levenshtein's edit distance [291] between the predicted and the ground truth

transcriptions is employed to compute the character recognition rates. Levenshtein's distance measures the difference between two sequences and corresponds to the smallest number of edits which change one word (sequence of characters) into the other. Formally, Levenshtein's distance between two sequences s_1 and s_2 of lengths $|s_1|$ and $|s_2|$ is given by $lev_{s_1,s_2}(|s_1|,|s_2|)$ where

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1 \end{cases} & otherwise. \end{cases} \tag{3.4}$$

An example of Levenshtein's edit distance with two Urdu strings have been solved in Appendix B.

In order to compute the recognition rate, the ground truth transcription of each textual region needs to be stored. Figure 3.2 illustrates this idea where a textual region in the image and the corresponding ground truth transcription are shown. The transcription produced by the recognition module can then be compared with the ground truth transcription to compute word or character recognition rates as summarized in Figure 3.3.

(a)

کراچی:ملوث ایف آئی اے انسپکٹر عدنان حبیبانی کو گرفتار کرنے کی ہدایت

(b)

Figure 3.2: (a) A text line in a video frame (b) Ground truth transcription of text

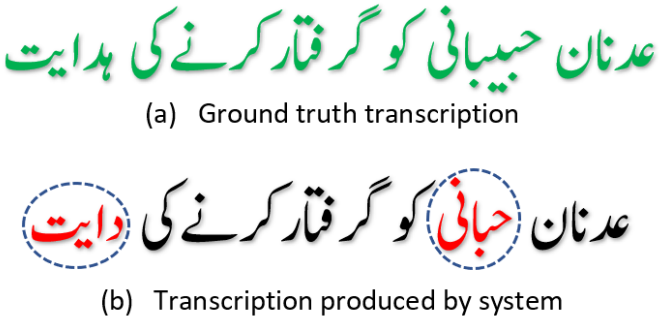


Figure 3.3: Comparison of ground truth and system produced transcriptions to quantify recognition performance

3.3 Ground Truth Labeling Tool

To facilitate the labeling process and standardize the ground truth data, a comprehensive labeling tool has been developed (in C#.NET) that allows storing the location of each textual region in a frame along with its ground truth transcription. A screen shot of the developed tool is presented in Figure 3.4. The tool allows loading video frames and labeling them one by one for text locations as well as ground truth transcription as discussed in the following. (Various features of the ground truth labeling tool are summarized in Appendix C).

Figure 3.4: Screen shot of ground truth labeling tool for text data

3.3.1 Labeling of Text Locations

As discussed earlier, evaluation of detection requires storing the actual location of textual content in each frame. For each textual region in the frame, the tool allows user to draw a rectangle encompassing the text. For each rectangle (bounding box), the localization information of text is stored in terms of the x and y coordinates, *width* and *height* of the rectangle. Each textual region is also identified as 'artificial' or 'scene' text. Artificial text refers to the caption text added to the video (for instance news tickers) while scene text refers to the text that occurs in the scene during the video capture (for example text on billboards). In addition to the type of text, the script information (English or Urdu text) is also stored by separating the English and Urdu text lines in the ground truth files. The ground truth information stored for each textual region is summarized in Table 3.1.

Table 3.1: Summary of attributes stored for each text line

Attribute	Description
ID	A unique identifier assigned to each text line a frame
Text Type	Type of text line, i.e. artificial or scene text
Location	Bounding box of text line in terms of x, y coordinates and *width* & *height* of the box

3.3.2 Transcription of Text

In order to evaluate the recognition performance, the transcription of each textual region needs to be stored as well. The labeling tool allows typing the transcription of each text line (in Urdu or English) and storing it in the ground turth file (Figure 3.5). Transcription is facilitated by providing a list of frequently used words and allowing users to add words to the list. The labeling process is carried out for each text region in a frame and the ground truth data is stored in an XML file.

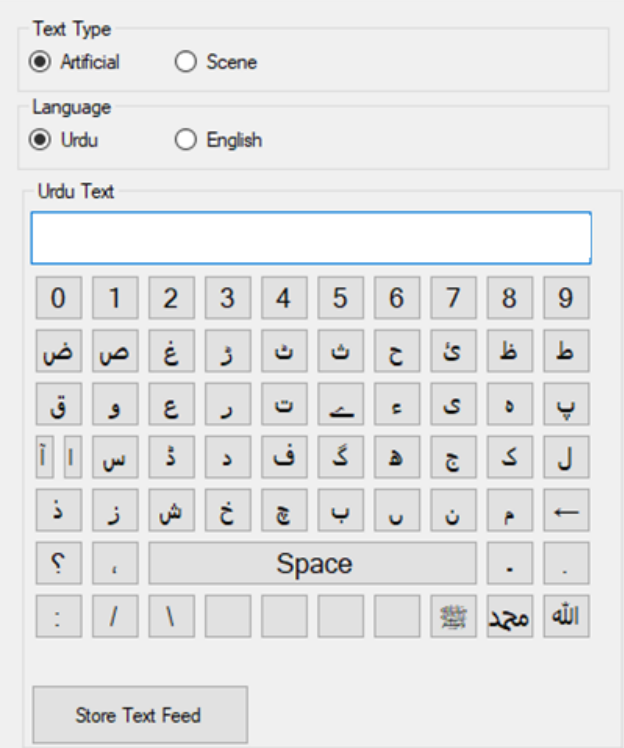


Figure 3.5: Interface to enter transcription of text

3.3.3 Ground Truth Data Organization

The ground truth information of each frame is stored as an XML file that comprises two parts, the frame meta data and the information on textual regions. The frame meta data contains information on video, channel and a unique code identifying the frame. The second part of XML file separately stores information of text in each script (Urdu and English in our case) in the frame. For each category, we store information on total number of text lines and for each line we store a unique ID, the type of text (scene text or artificial text), the location of text region within the frame and the transcription of text. The screen shot of ground truth information of an example frame is illustrated in Figure 3.6.

Figure 3.6: Screen shot of an XML file containing ground truth information of a frame

3.4 Statistics of Labeled Data

This section presents a summary of the labeled video frames. It is known that videos typically contain $25-30$ frames per second; consequently, successive frames in a video contain redundant information (both visual and textual content). From the view point of automatic analysis systems, frames with unique content are of interest. Hence, each single video frame does not need to be labeled as major proportions of such frames will have exactly the same textual information. In our study, we have extracted more than 11,000 frames from videos of different News channels with an attempt to have as much unique text as possible. Each frame is labeled for text location as well as transcription as discussed in the previous sections. The statistics of videos, frames and text lines of our dataset are presented in Table 3.2. Since the frames are primarily collected from Urdu News channels, major proportion of text lines in these images contain Urdu caption text. Nevertheless, some channels contain bilingual textual content with caption text appearing both in Urdu and English and all such occurrences are labeled.

Table 3.2: Statistics of labeled video frames

S#	Channel	Videos	Labeled Images	Urdu Lines	English Lines
1	Ary News	7	3,206	10,250	3,605
2	Samaa News	13	2,503	10,961	4,411
3	Dunya News	16	3,059	10,723	8,861
4	Express News	10	2,424	8,536	6,755
	Total	**46**	**11,192**	**40,470**	**23,632**

In an attempt to provide further insights in to the collected data, we provide additional statistics particularly from the perspective of Urdu text, the primary focus of our research. Figure 3.7 summarizes the distribution number of Urdu characters per line and the distribution of number of lines per frame. It is observed that on the average, each frame contains 3.62 ± 2.09 Urdu text

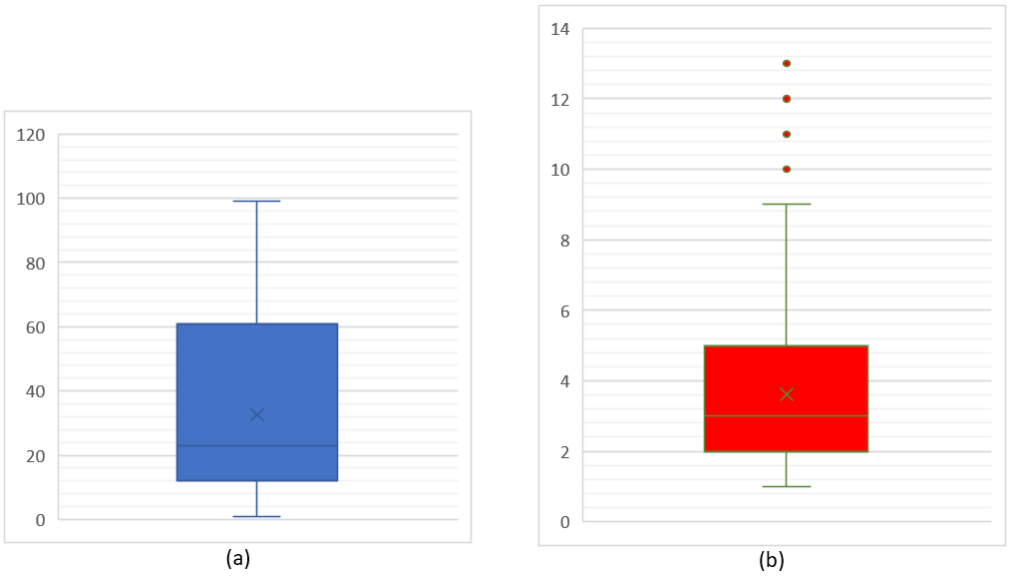


Figure 3.7: (a): Distribution of number of Urdu characters per line. (b): Distribution of Urdu lines per frame

lines while each text line contains 32.56 ± 26.66 characters. Likewise, the frequency of various characters is of interest and is outlined in Figure 3.8 for the Top-30 most frequent characters.

3.5 Synthetic Data Generation

As mentioned earlier, text lines extracted from video frames using the ground truth information are used to train the learning algorithm. Few examples of text lines are illustrated in Figure 3.9. In an attempt to enhance the size of training data (to ensure maximum representation of various character shapes and their combinations), we also generated a set of 50,000 synthetic text lines. In order to artificially generate the text line images, we first create a pool of text lines from different Urdu books and News portals. The textual content on these sources is parsed and each text line is stored in a pool. Although the content in books is semantically different from the content on News channels, it is important to mention that we strive to enhance the training data such that it contains various combinations of characters. Since we target an implicit segmentation based analytical approach, the semantic content itself is not important rather, the representation of various character shapes and their combinations is what contributes to the effectiveness of the learning algorithm. Likewise, to ensure close resemblance with the actual data, various backgrounds are extracted from actual News channel videos and a pool of background is created. Next, we randomly pick one of the text lines and one of the backgrounds from the respective pools and the text is superimposed on the background image. The process is repeated as many times as the required number of synthetic text lines. The overall process of generating the synthetic text lines is summarized in Figure 3.10 while samples of such synthetic text lines are presented in Figure 3.11 where it can be seen that the

generated text line images look very similar to the actual text lines extracted from video frames.

3.6 Summary

This chapter presented an overview of data collection and labeling and introduced the 'UTiV' dataset [1]. We also presented the ground truth labeling tool and its different features and presented a summary of the data that has been labeled along with different interesting statistics of the collected data. In the next chapter, we present the techniques investigated for detection of textual content in video frames.

[1] https://drive.google.com/drive/folders/1U3M6WTReCu4PYxk88aXlTQDqsSn4gHAq

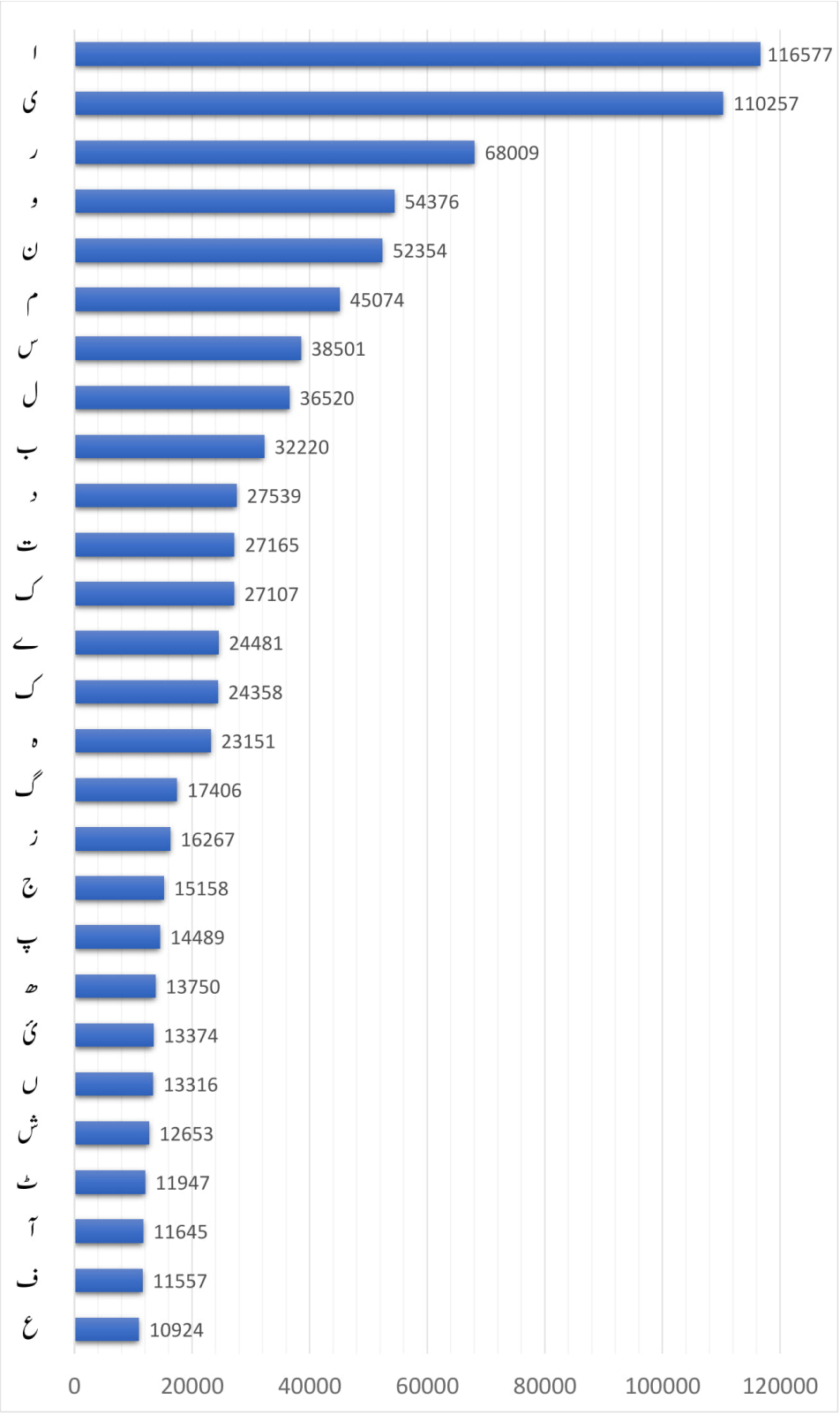


Figure 3.8: Frequency of Top-30 Urdu characters in the collected data

Figure 3.9: Sample text lines extracted from video frames

Figure 3.10: Generation of synthetic text lines

Figure 3.11: Synthetically generated text lines to resemble caption text

Chapter 4

Detection of Textual Content

4.1 Introduction

This chapter presents the details of techniques developed for detection of caption text appearing in video frames, the first step towards the development of textual content based applications. We first introduce our investigations based on hand-crafted features. A combination of unsupervised and supervised techniques was investigated for this purpose. The unsupervised detection relies on a series of image analysis operations exploiting the edge density of text characters to determine the candidate text regions. The identified regions are further enhanced by using a combination of morphological operations followed by application of geometrical constraints. The detected text regions are later validated through a supervised technique using textural measures as features. With the recent paradigm shift from hand-crafted to machine-learned features and, inspired by the superiority of these learned features over hand-engineered features, we adapted the detection methodology to include the latest and robust feature extraction using deep learning techniques. A number of object detectors were adapted for text detection problem and their performance was analyzed through a comprehensive series of experiments.

In the following sections, we first present text detection using image analysis techniques followed by the investigation of deep learning based techniques for this problem. We then present the findings of the experimental study along with a detailed analysis of the realized results. The chapter concludes with a recall of the key ideas and a summary of the findings.

4.2 Detection of Text using Image Analysis Techniques

Text in video or images has certain attributes which can be effectively employed to segment it from the background. Our initial research on detection of textual content targeted these properties to find potential text regions in the image. The text is supposed to be readable hence the contrast between the text and its background should be reasonably high. Similarly, text in any script can be characterized by a strong density of edges. With few exceptions, text is aligned horizontally and in

general has the same font style and size within the same line of text. We exploit these attributes to detect potential text regions as presented in the following.

4.2.1 Gray Scale Filtering

It is known that text appearing in video frames can be detected and read using intensity information only. Consequently, in order to make the processing of frames independent of the color information, the frames are first converted to gray scale keeping only the intensity information of each pixel [88].

4.2.2 Detection of Edges

It is known that edges are a common feature of text in all scripts [53, 44]. Although, many non-text objects may also have sharp edges, text is characterized by the presence of a large number of edges in the proximity of one another. Edges can be computed using first or second order derivatives termed as gradient and Laplacian methods respectively. Different scripts have different proportions of horizontal, vertical and diagonal edges corresponding to text strokes in each of these directions. These differences make it difficult to have a common method that could detect text occurrences in variety of languages (English and Urdu in our case). Analyzing samples of text in English and Urdu, it can be observed that vertical strokes represent the most dominant common attribute between the two types of text. We, therefore, extract vertical strokes through detection of vertical edges in the image.

Sobel operator [292] is applied on the grayscale image to detect the vertical edges. Effectively, performing convolution with the horizontal Sobel mask (Equation 4.1) computes the derivative of image in the horizontal direction which highlights the vertical edges in the image.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.1}$$

As illustrated in Figure 4.1, extracting vertical strokes retains most of the textual regions. Although a number of non-text regions are also retained and these non-text regions are likely to be high in case of complex backgrounds, the subsequent detection and validation steps reduce these false positives to as low as possible.

4.2.3 Mean Gradient

It is known that the textual content generally appears in groups rather than in isolation. As a result, a natural step after detection of (vertical) edges is to enhance the magnitude of image gradients in the text regions while suppressing it in the non-text areas. Generally, this is achieved by scanning the gradient image with a small window and performing some operations [44, 290, 77]. In our implementation, we have employed a sliding window based technique where each pixel value is

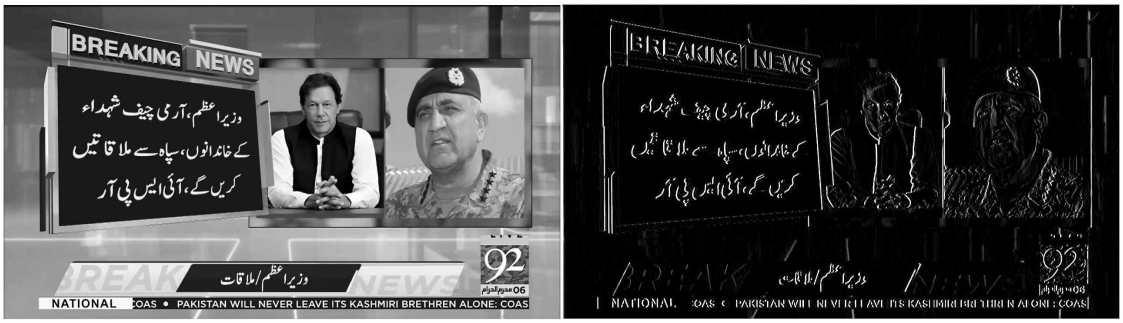


Figure 4.1: An image with occurrences of Urdu and English text and the corresponding vertical edges in the image

replaced with the mean gradient value within the window. The width of the window is determined as a function of the image resolution. The motivation behind this operation is that edges in text regions appear in clusters. Hence, computing the average gradient in windows over text regions is likely to maintain high values. On the other hand, isolated gradients in the non-text regions are likely to be suppressed.

4.2.4 Binarization

Once the gradients have been accumulated using the mean gradient filter, the resulting image is binarized to obtain the potential text regions as white pixels and the background as black pixels. This allows application of morphological operators in the subsequent steps. We have employed a global thresholding technique (Otsu's algorithm) [293] to binarize the frames. As a result of this step, the isolated or weak gradients become a part of the background.

4.2.5 Morphological Processing

Once the frames are binarized, the clusters of text regions are further discriminated from non-text regions by employing a series of morphological operations [44, 46, 294]. We first apply a horizontal run-length smoothing algorithm (RLSA) to the binarized image so that the white pixels within the proximity of one another are merged together. These components are likely to be characters or ligatures which are combined into a single component using the horizontal RLSA. To remove the noisy, text-like regions, we then apply morphological erosion. Erosion has the effect of shrinking the components while smaller components are completely removed from the frame. Finally, morphological closing is applied to the image to smooth component boundaries, fill the holes and merge the components joined through bridges.

4.2.6 Geometrical Constraints

Textual regions in images have some geometrical features that can be used to differentiate them from non-text areas [53, 44, 46]. Given the alignment of text (horizontal or vertical), the bounding box of textual content has a constrained aspect ratio. Similarly, thresholds can be applied to the

area, width and height of the text blocks. Since the size of text on the image is large enough to be read by the viewers, different thresholds can be determined as a function of image resolution. In our system, we mainly target the horizontal text. Consequently, we apply thresholds on the aspect ratio and size of the connected components to remove all components which are not likely to correspond to text regions. Finally, the connected components remaining in the frame are extracted which correspond to candidate text regions.

Different steps of unsupervised detection are illustrated on three diverse sample images in Figure 4.2 & Figure 4.3. It can be observed that detection of edges is one of the most critical steps which influences the subsequent stages. Relatively larger number of false regions are detected in images with high edge density. It should, however, be noted that the unsupervised, image analysis based processing steps discussed above are aimed at detecting the regions which are likely to contain textual occurrences. A precise localization of text is not required at this stage rather, the idea is to make an effort not to miss the textual regions. The next step of text validation will then be employed to eliminate the false positives and retain only textual regions.

4.2.7 Validation of Text Regions

Once the candidate text regions are identified, we validate them using a supervised approach (with hand-crafted features). Supervised techniques involve learning of a function that maps an input to an output based on the input-output pairs in the training set. From the perspective of text and non-text classification, supervised classification includes presenting a learning algorithm with examples of each category (i.e. text images and non-text images as illustrated in Figure 4.4) to make it learn the discrimination between the two classes.

Features extracted from video frames comprising text and non-text blocks are used to train classifiers to discriminate between the two classes as discussed in the following while an overview of the detection system is presented in Figure 4.5.

4.2.7.1 Feature Extraction

It is known that text has a unique texture that distinguishes itself from non-text regions. We, therefore, employ textural features to discriminate between the text and non-text regions. The textural measures considered in our study include Gabor filters and curvelets.

- **Gabor Filters:** One of the widely used and popular textural feature based filters are Gabor filters. Gabor filters share similarities with the visual cortex of mammalian cells. Mammals are able to use band pass and orientation selectivity as main characteristics of their visual cortex cells which make them respond to specific spatial frequency and direction. These cortex cells are found in pairs with odd and even symmetry respectively. Various image processing applications are developed based on these similarities of Gabor filters and visual cortex. In many applications, bank of Gabor filters is prepared using different scales and orientations.

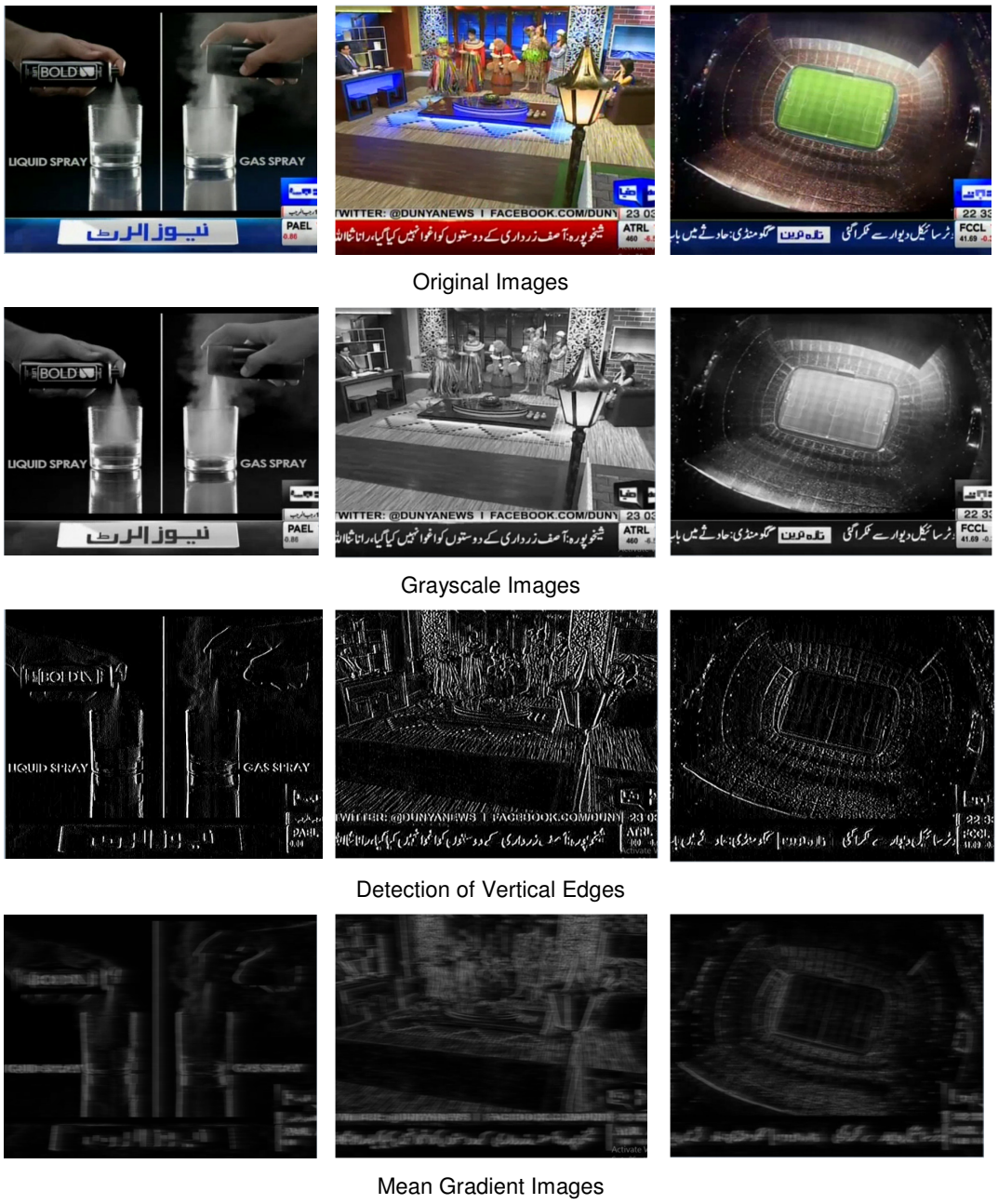


Figure 4.2: Steps I-III in unsupervised text detection

As an example, with four scales and six orientations, a bank of Gabor filter can be seen in Figure 4.6 and the same is employed in our study. Based on the combination of different orientations and scales, we get 24 filtered images. We calculate the mean and variance of each of the 24 images and place these values in two matrices. Fast Fourier Transform (FFT) is then applied on the mean and variance matrices to generate a 48 dimensional feature vector.

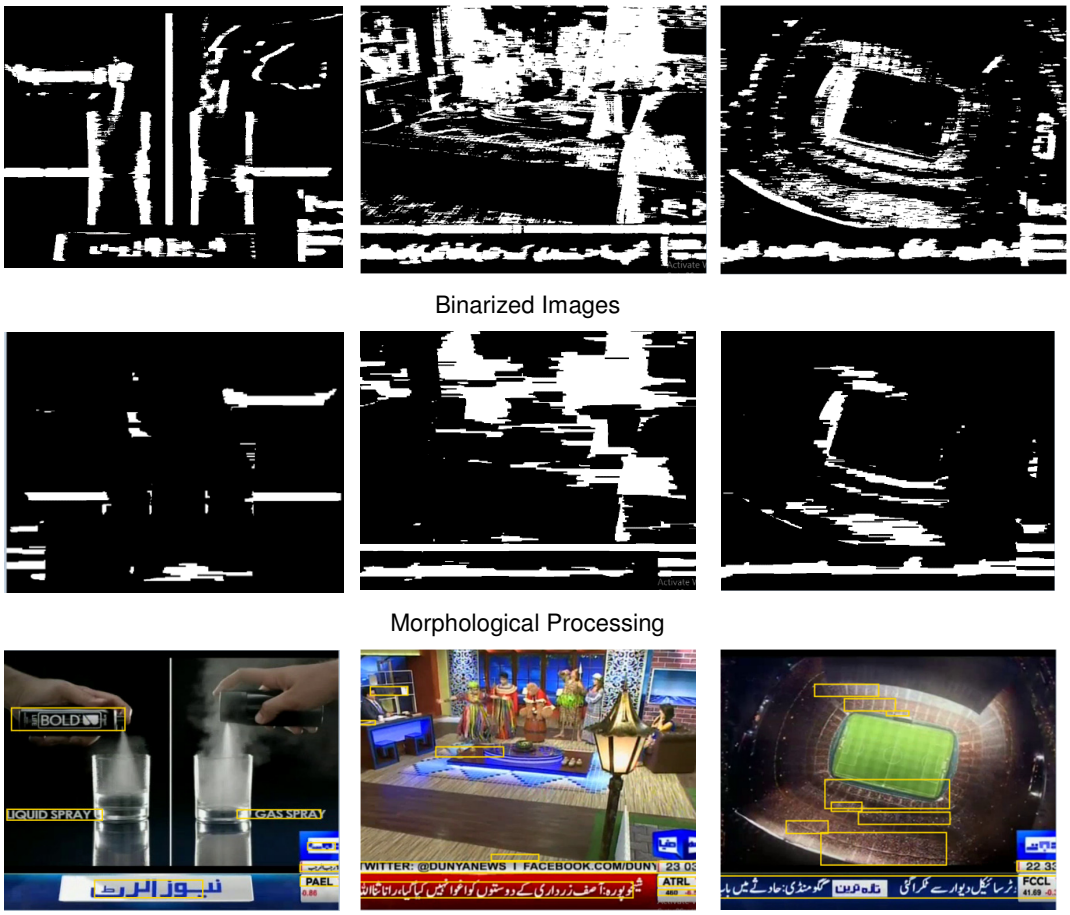


Figure 4.3: Steps IV-VI in unsupervised text detection

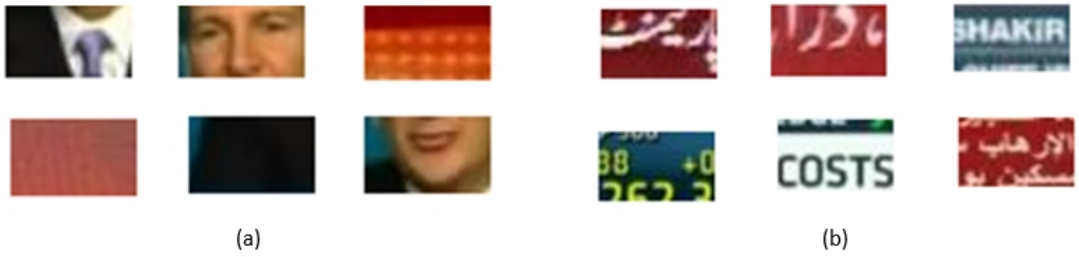


Figure 4.4: Sample blocks of (a) Non-text and (b) text regions which are employed to train a classifier

- **Curvelets:** Curvelets [95] are known to be effective as a feature descriptor for images containing textual occurrences. Pixels in the close proximity of one another give rise to edges, the strokes of text in our case. The 2D Fast Fourier Transform (FFT) of the curvelet transformed image is taken and is employed as feature.

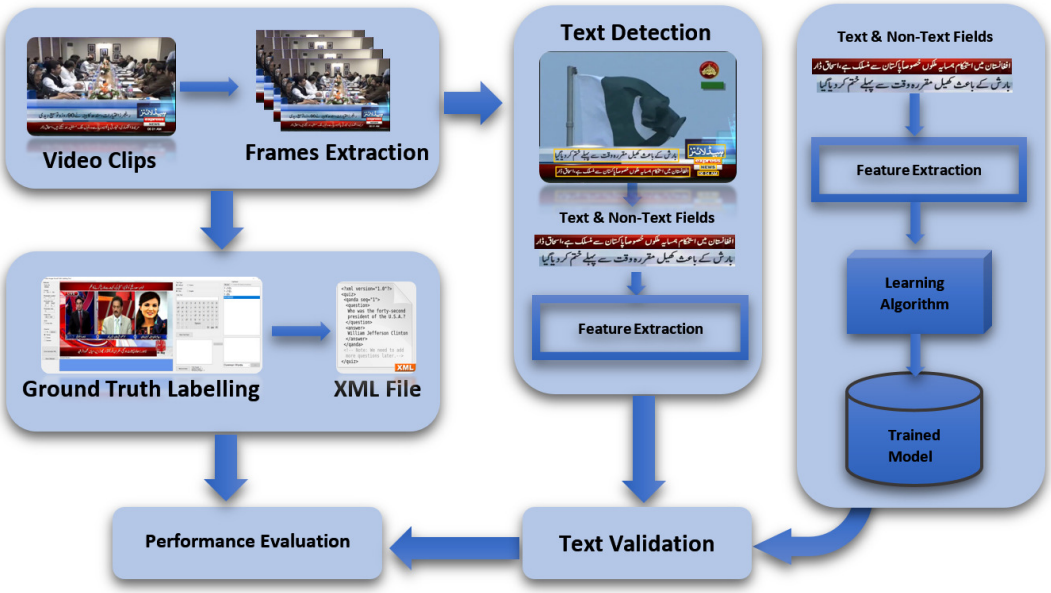


Figure 4.5: Overview of detection system using hand-crafted features

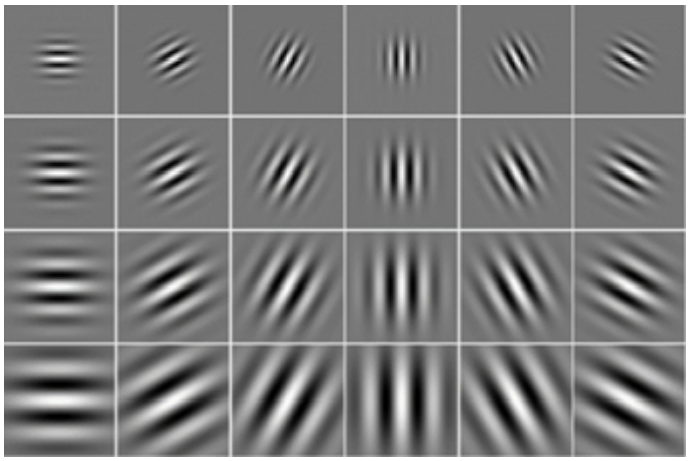


Figure 4.6: Bank of Gabor filters with 4 scales and 6 orientations

4.2.7.2 Classification

For classification, two state-of-the-art classifiers namely Support Vector Machine (SVM) and feed forward Artificial Neural Network (ANN) have been employed. Gabor and curvelet features extracted from text and non-text blocks are used to train these classifiers which are later evaluated on unseen blocks.

4.2.7.3 Evaluation

In the preliminary experiments, we studied the effectiveness of the proposed scheme using a set of 1000 video frames taken from 'UTiV' dataset. The system realized a precision of 0.72 and a recall

of 0.89. More details on supervised text validation using hand-crafted features can be found in [295].

As discussed earlier, the latest trend in machine learning is to replace hand-crafted features with machine-learned features which are known to be more robust and also outperform the traditional feature extraction techniques. Consequently, after initial investigations with hand-crafted features, we explored machine learned features using convolutional neural networks (CNNs). Details on CNNs and their adaptation to our problem are presented in the following sections.

4.3 Detection of Text using Deep Learning Techniques

Deep neural networks enjoy a renewed interest of the machine learning community thanks primarily to the availability of high performance computing hardware (GPUs) as well as large data sets to train these systems. A major development contributing to the current fame of deep learning was the application of Convolutional Neural Networks (CNNs) by Krizhevsky et al. [296] on the ImageNet Large Scale Visual Recognition competition [297], which greatly reduced the error rates. Since then, CNNs are considered to be state-of-the-art feature extractors and classifiers [298, 299] and have been applied to a variety of recognition tasks [300, 301, 302]. In addition to classification, CNNs have also been adapted for object detection and are known to outperform the conventional computer vision algorithms for detecting and localizing objects in images. Inspired by their robustness, we have chosen to adapt deep learning based object detectors for detection of textual content in the video frames. We first present an overview of the well-known object detectors followed by details on how they are adapted for detection of textual content in video frames.

4.3.1 Deep Learning based Object Detectors

While traditional CNNs are typically employed for object classification, Region-based Convolutional Networks (R-CNN) [303] and their further enhancements Fast R-CNN [304] and Faster R-CNN [305] adapt CNNs for object detection. In addition to different variants of R-CNN, a number of new architectures have also been proposed in the recent years for real time object detection. The most notable of these include YOLO (You Only Look Once) [306] and SSD (Single Shot Detector) [307]. Each of these object detectors can be trained to detect C object classes (plus one for the background). The output of the detector is the location of the bounding box (four coordinates) containing one of the C classes as well as the class confidence score.

In our study, for detection of textual content in a given frame, we investigated a number of CNN based object detectors. Although, many object detectors are trained with thousands of class examples and provide high accuracy in detection and recognition of different objects, these object detectors can not be directly applied to identify text regions in images. These models have to be tuned to the specific problem of discrimination of text from non-text regions. The convolutional base of these models can be trained from scratch or known pre-trained models can be fine-tuned by

training them on text and non-text regions. The following object detectors were adapted for text detection in our study.

- Faster R-CNN
- You Only Look Once (YOLO)
- Single Shot Detector (SSD)
- Region-Based Fully Convolutional Networks (R-FCN)

In the next sections, for completeness, we provide a brief overview of these object detectors.

4.3.1.1 Faster R-CNN

Faster R-CNN [305] is an enhanced version of its predecessors R-CNN [303] and Fast R-CNN [304]. Each of these detectors exploits the powerful features of ConvNets for object localization as well as classification. R-CNN was one of the first attempts to apply ConvNets for object detection. An R-CNN scans the input image for potential objects using Selective Search [301] that generates around 2,000 region proposals. Each of these region proposals is then fed to a CNN for feature extraction. The output of the CNN is finally employed by an SVM to classify the object and a linear regressor to tighten the bounding box. R-CNN was enhanced in terms of training efficiency by extending it to Fast R-CNN [304]. In Fast R-CNN, rather than separately feeding each region proposal to the ConvNet, convolution is performed only once on the complete image and the region proposals are projected on the feature maps. Furthermore, the SVM in R-CNN was replaced by a softmax layer extending the network to predict the class labels rather than using a separate model. While Fast R-CNN significantly reduced the time complexity of the basic R-CNN, a major bottleneck was the selective search algorithm to generate the region proposals. This was addressed through Region Proposal Network (RPN) in Faster R-CNN [305] which shares convolutional features with the detection network. RPN predicts region proposals which are then fed to the detection network to identify the object class and refine the bounding boxes produced by the RPN. A summary of various R-CNN models in presented in Figure 4.7.

4.3.1.2 You Only Look Once (YOLO)

YOLO [306] takes a different approach to object detection primarily focusing on improving the detection speed (rather than accuracy). As the name suggests, YOLO employs a single pass of the convolutional network for localization and classification of objects from the the input images. The input image is divided into a grid and an object is expected to be detected by the grid which holds the center of the object. Each cell in the grid predicts up to two bounding boxes (and class probabilities) (Figure 4.8). The network comprises 24 convolutional and fully connected layers. YOLO works in real time but in terms of accuracy, it is known to make significant localization

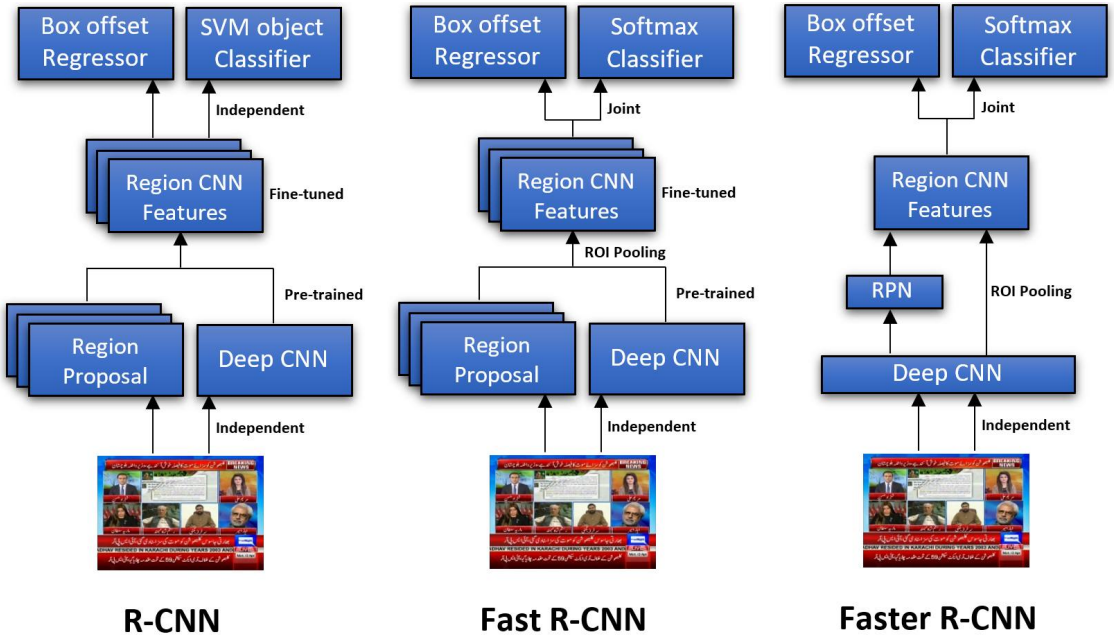


Figure 4.7: Summary of R-CNN Family Models

errors in comparison to region based object detectors (Faster R-CNN for instance). YOLO was later enhanced to YOLO9000 [308] by including batch normalization, increasing the resolution of the input image (by a factor of 2) and introducing the concept of anchor boxes. YOLO9000 employs Darknet 19 architecture with 19 convolutional layers and 5 max pooling layers and a softmax layer for classification objects. Incremental improvements in YOLO v2 resulted in YOLO v3 [309] that uses logistic regerssion to predict the score of objectness for each bounding box. Furthermore, it employs class-wise logistic classifiers (rather than softmax) allowing multi-label classification.

4.3.1.3 Single Shot Detector (SSD)

Unlike the R-CNN series object detectors which require two shots to detect objects in an image, Single Shot Multi-box Detector [307], as the name suggests, requires a single shot to detect objects (similar to YOLO). SSD relies on the idea of default boxes and multi-scale predictions and directly applies bounding box regression to the default boxes without generating the region proposals. Detection at multiple scales are handled by exploiting the feature maps of different convolutional layers corresponding to different receptive fields in the input image. The architecture (Figure 4.9) has an input size of $300 \times 300 \times 3$ and primarily builds on the VGG-16 architecture discarding the fully connected layers. VGG-16 is used as base network mainly due to its robust performance of image classification tasks. The bounding box regression technique of SSD is inspired by [299] while the MultiBox relies on priors, the pre-computed fixed size bounding boxes. The priors are selected in such a way that their Intersection over Union ratio (with ground truth objects) is greater than 0.5. The MultiBox starts with the priors as predictions and attempt to regress closer to the

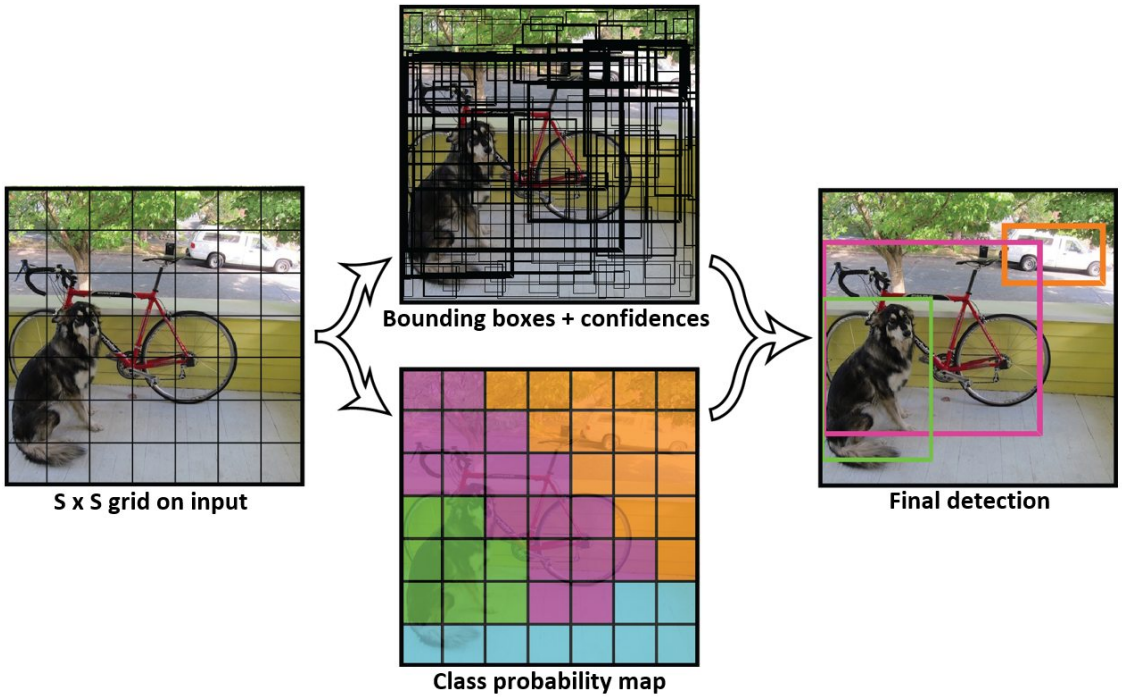


Figure 4.8: Detection method of YOLO, (Image Source [306])

ground truth bounding boxes. SSD works in real time but requires images of fixed square size and is known to miss small objects in the image.

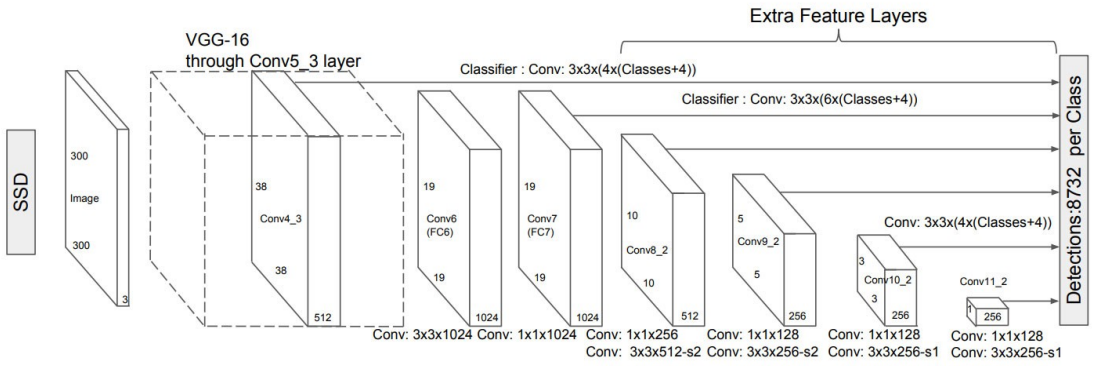


Figure 4.9: Architecture of Single Shot Detector, (Image Source [307])

4.3.1.4 Region-Based Fully Convolutional Networks (R-FCN)

R-FCN [310] builds on the idea of increasing the detection accuracy by maximizing the shared calculations. R-FCN generates position-sensitive score maps to represent different relative positions of an object. An object is represented by k^2 relative positions dividing it into a grid of size $k \times k$. A ConvNet (ResNet in the original R-FCN paper) sweeps the input image and an additional fully convoltional layer produces the position-sensitive scores in $k^2 \times (C+1)$ score maps where C is the number of classes plus 1 class for the background. A fully convolutional proposal network

generates regions of interest which are divided in k^2 bins and the corresponding class probabilities are obtained from the score maps. The scores are averaged to convert the $k^2 \times (C+1)$ values into a one dimensional $(C+1)$ sized vector which is finally fed to the softmax layer for classification. Localization is carried out using the bounding box regression similar to other object detectors. R-FCN speeds up the detection in comparison to Faster R-CNN but compared to other Single Shot methods, it requires more computational resources. An overview of R-FCN based object detection is presented in Figure 4.10.

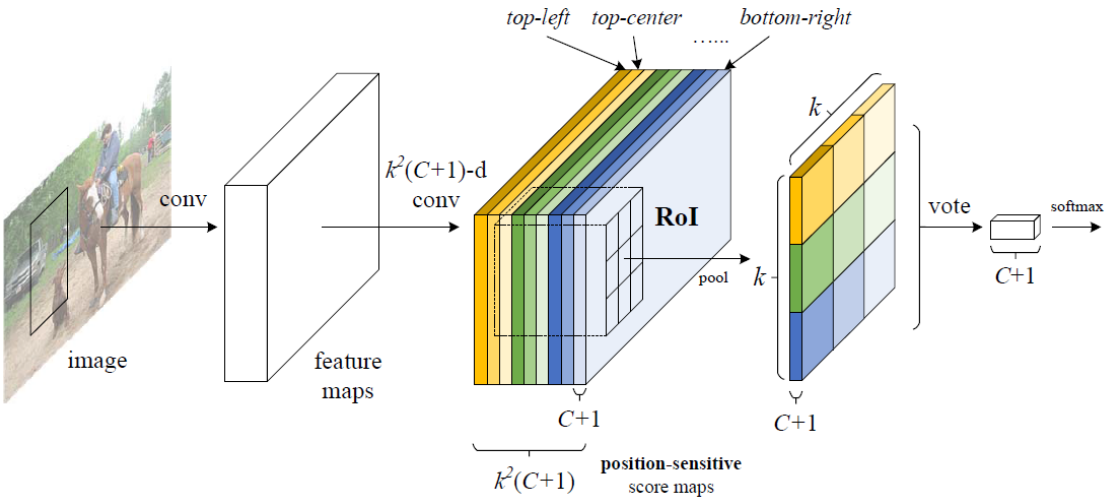


Figure 4.10: Region-based Fully Convolutional Networks (R-FCN) for Object Detection (Image Source [310])

4.3.2 Adapting Object Detectors for Text Detection

In the context of object detection, the problem of text detection can be formulated as a two class problem. The text regions represent the object of interest while the non-text regions need to be ignored. The object detectors discussed in the previous section are adapted for text detection using two pre-trained models, ResNet 101 [311] and Inception v3 [312]. The pre-trained models are trained on the very large scale Microsoft COCO (Common Objects in Context) database [313]. The database contains images of 91 different object types with a total of 2.5 million labeled instances in 328K images. The pre-trained network serves as starting point rather than random weight initialization and the network is made to learn the specific class labels (text or non-text) by continuing back propagation (Figure 4.11). The ground truth localization information of the textual regions in the video frames is employed for training the models. A critical aspect in employing object detectors for text detection is the choice of anchor boxes. The anchor boxes in all the detectors have been designed to detect general object categories. Text appearing in videos has specific geometric properties in terms of size and aspect ratio hence the default anchor boxes of the detectors need to be adapted to detect textual regions. We carried out a comprehensive analysis of the textual regions in terms of width, height and aspect ratios of the bounding boxes. As a result of this analysis we have chosen a base anchor of size 256×256. To each anchor box we apply three scales $(1.0, 2.0, 5.0)$

and five aspect ratios $(0.125, 0.1875, 0.25, 0.375, 0.50)$ as illustrated in Figure 4.12. Among the various investigated detectors, we finally adapted Faster R-CNN for our study; more details are presented in Section 4.4.

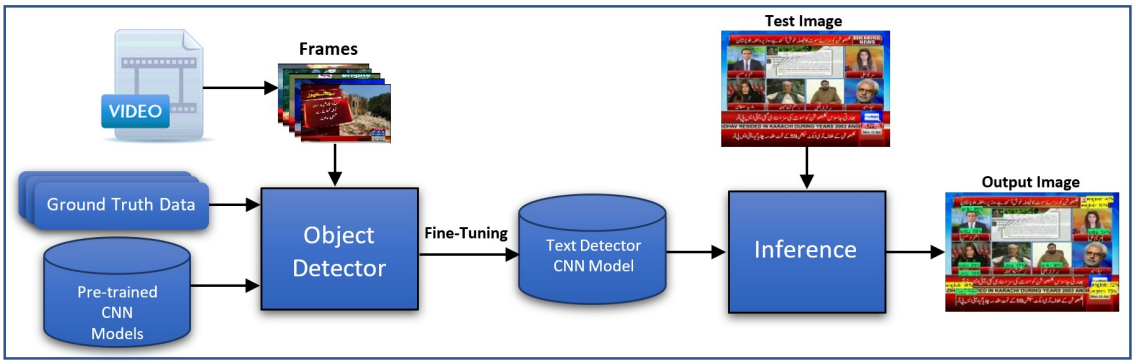


Figure 4.11: Overview of adapting object detectors for text detection

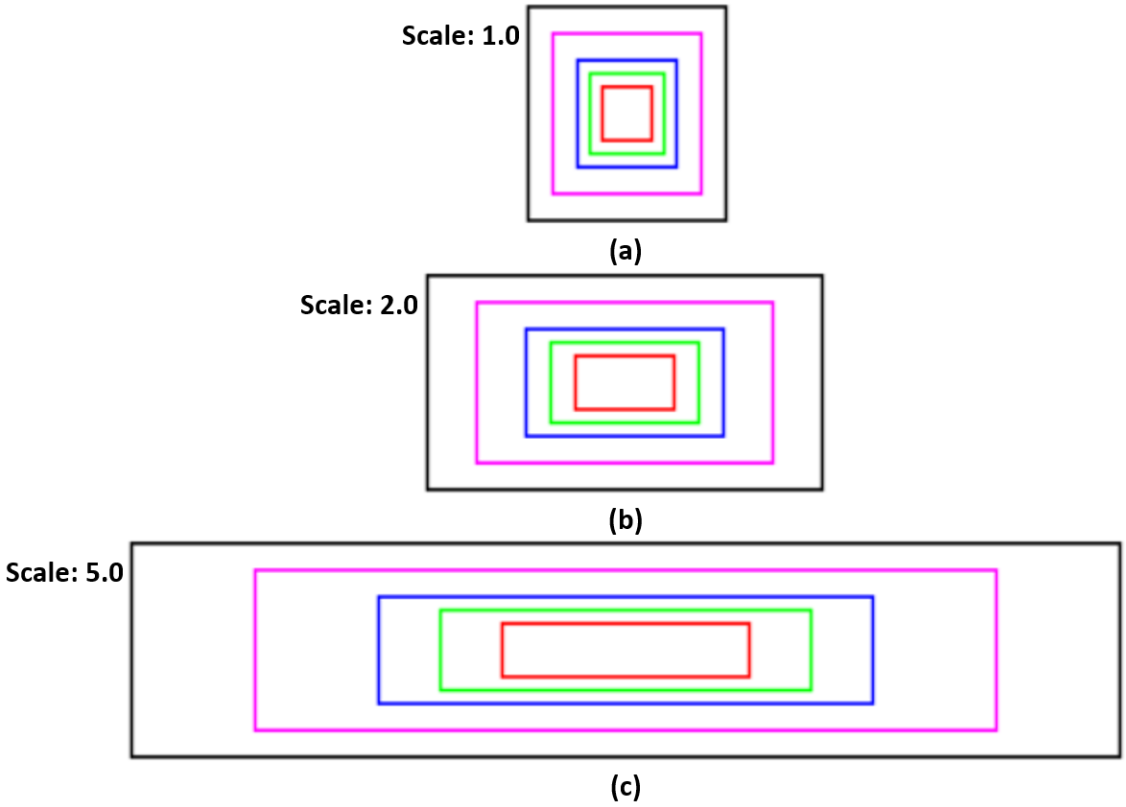


Figure 4.12: Anchor boxes (base size 256×256) at three scales $(1.0, 2.0, 5.0)$ and five aspect ratios $(0.125, 0.1875, 0.25, 0.375, 0.50)$

4.3.3 Script Identification

Since we primarily target videos from the local News channels, video frames are likely to contain bilingual text (Urdu & English) in most cases. Consequently, once the text is detected, we need to identify the script of each detected region (Figure 4.13) so that the subsequent processing (recognition) of each type of script can be carried out by the respective recognition engine. For script identification, we employ CNNs in a classification framework (rather than detection). Urdu and English text lines are employed to fine-tune CNNs to discriminate between the two classes. Once trained, the model is able to separate text lines as a function of the script. Similar to detection, rather than training the networks from scratch, we fine-tune known pre-trained models (Inception and ResNet) to solve the two-class classification problem.

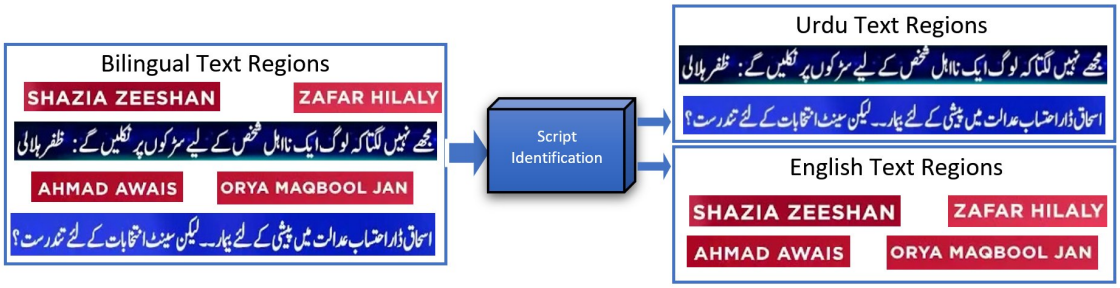


Figure 4.13: Example of script identification

4.3.4 Hybrid Text Detector & Script Identifier

Detection of text and identification script, as discussed previously, can be implemented in a cascaded framework where the output of text detector is fed to the script identifier. A deep learning framework can be tuned to discriminate between text and non-text regions and the extracted text regions can be fed to a separate script recognition model that identifies the script of the detected text. This, however, introduces a bottleneck of training two separate networks. Furthermore, the cascaded solution also implies that errors in detection are propagated to the next step as well. We, therefore, propose to combine the text detector and script identifier in a single hybrid model. Rather than treating detection as a two-class problem (text and non-text), we consider it as a three class problem, i.e. non-text regions, English text and Urdu text. This not only avoids training two separate models but also eliminates the accumulation of errors in a cascaded solution. The superiority of the combined text detector and script identifier is also supported through quantitative evaluations as discussed in the next section.

All detectors are trained in an end-to-end manner with a multi-task objective function that combines the classification and regression losses. Formally, the loss function (of the chosen Faster R-CNN based detector) is defined as a combination of the region proposal and the detection network loss functions as follows.

$$\mathscr{L} = \mathscr{L}^{\text{RPN}} + \mathscr{L}^{\text{DET}}$$

Where $\mathscr{L}^{\text{RPN}}$ is the loss of region proposal network while $\mathscr{L}^{\text{DET}}$ is the loss of detection network. Each of these are defined in the following. The multi-task loss function $\mathscr{L}^{\text{RPN}}$ combines the losses of the region proposal classification and bounding box regression.

$$\mathscr{L}^{\text{RPN}} = \mathscr{L}^{\text{RPN}}_{\text{cls}} + \mathscr{L}^{\text{RPN}}_{\text{box}}$$
$$\mathscr{L}^{\text{RPN}} = \frac{1}{N_{\text{cls}}} \sum_i \mathscr{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*)$$

Where $\mathscr{L}^{\text{RPN}}_{\text{cls}}$ is the log loss function over two classes, p_i is the predicted probability of anchor i being an object, p_i^* is the ground truth label (binary) of anchor i being an object, t_i are the predicted four coordinates of the bounding box and t_i^* are the ground truth coordinates. Likewise, N_{cls} is the normalization term set to the mini-batch size, N_{box} is the normalization term set to the number of anchors and λ is the balancing parameter set to the default value of 10. L_1^{smooth} is the smooth $L1$ loss.

$$\mathscr{L}^{\text{RPN}}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i)$$
$$L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

In a similar fashion, the multi-task loss function of the detection network ($\mathscr{L}^{\text{DET}}$) also combines the losses of classification and bounding box regression.

$$\mathscr{L}^{\text{DET}} = \mathscr{L}^{\text{DET}}_{\text{cls}} + \mathscr{L}^{\text{DET}}_{\text{box}}$$

For background, $\mathscr{L}^{\text{DET}}_{\text{box}}$ is ignored by the indicator function $1[u \geq 1]$, defined as:

$$1[u >= 1] = \begin{cases} 1 & \text{if } u \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

Where u is the true class label, $u \in 0, 1, 2$; in our study. For background we have u=0. The overall

detector loss function is defined in the following.

$$\mathscr{L}^{\text{DET}} = \mathscr{L}^{\text{DET}}_{\text{cls}}(p,u) + 1[u \geq 1]\mathscr{L}^{\text{DET}}_{\text{box}}(t^u, v)$$
$$\mathscr{L}^{\text{DET}}_{\text{cls}} = -\log p_u$$
$$\mathscr{L}^{\text{DET}}_{\text{box}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{\text{smooth}}(t_i^u - v_i)$$

p_u is the probability of region belonging to class u, $v = (v_x, v_y, v_w, v_h)$ are the ground truth bounding box coordinates while $(t^u$ are the predicted coordinates.

The evolution of training loss for the investigated detectors (with Inception and ResNet) is illustrated in Figure 4.14 where it can be seen that the loss begins to stabilize from 40 epochs on wards. A summary of different hyper-parameters employed for training is presented in Table 4.1 while the number of tuned parameters in our adapted and standard Faster R-CNN are presented in Table 4.2.

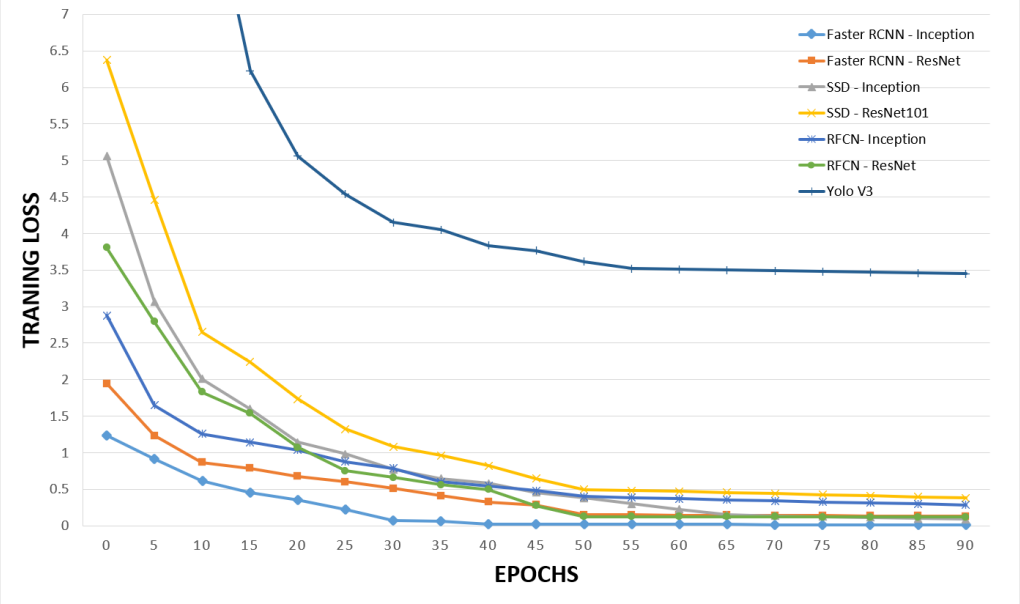


Figure 4.14: Training loss of various detectors

4.4 Experiments and Results

The detection performance is evaluated through a series of experiments carried out on the collected set of video frames. We first present the experimental protocol followed by the detection results of various object detectors. We then present the script identification results and the performance of the combined text detector and script identifier. Furthermore, performance sensitivity of the system as well as a comparison with state-of-the-art is also presented.

Table 4.1: Training parameters of hybrid text detector & script identifier network

Training Parameters	
Parameter	**Value**
Number of Classes	2
Learning Rate	0.0001
Batch Size	32
Anchor Box Size	256×256
Scales	1.0, 2.0, 5.0
Aspect Ratios	0.125, 0.1875, 0.25, 0.375, 0.50

Table 4.2: Tuned Parameters of adapted and standard Faster R-CNN models

			Parameters			
	Image Size	**Base Model**	**Base Layers**	**RPN**	**Detection**	**Total**
Orignal	$224 \times 224 \times 3$	VGG16	14.7 M	2.4 M	39.4 M	56.5 M
Adapted	$900 \times 600 \times 3$	InceptionV3	21.5 M	9.5 M	25 M	56 M

4.4.1 Experimental Settings

As introduced in Chapter 3, we collected a total of 11,192 video frames from four different News channel videos. The localization information of text regions in these frames is used to train and subsequently evaluate the text detection and script identification performance. The distribution of frames into training and test sets along with the number of text lines in each set is summarized in Table 4.3. The split of data into training and test sets for machine learning-based systems has remained a subject of thorough discussion in the literature. The generally recommended splits of data into training and test sets are 80%-20%, 75%-25% or 70%-30% [314, 315]. In some cases, a split of 60%-40% is also suggested [316, 317]. From the perspective of text detection and recognition, a split of 80%-20% is employed in [33, 318] while a distribution of 70%-30% into training and test sets is carried out in [190, 168, 213]. Taking into account the common split ratios suggested by machine learning researchers in general and employed by text detection and recognition community in particular, we split the data into train and test sets with a ratio of 75:25.

The details of detection performance are presented in the next section.

Table 4.3: Data distribution for text detection experiments

	Train		**Test**	
	Frames	**Lines**	**Frames**	**Lines**
Urdu	8,500	31,321	2,692	9,149
English		16,207		7,425
Total		47,528		16,574

4.4.2 Text Detection Results

Object detectors including Faster R-CNN, YOLO, SSD and R-FCN are adapted to detect textual content by fine-tuning the Inception and ResNet pre-trained models and changing the anchor boxes as discussed previously. Performance of each of these detectors in terms of precision, recall and F-measure is summarized in Table 4.4. It can be seen that in all cases, detectors pre-trained on Inception outperform those trained on ResNet. Among various detectors, Faster R-CNN reports the highest F-measure of 0.90. The lowest performance is reported by Yolo reading an F-measure of 0.66. A comprehensive study on the trade-off between speed and accuracy of various object detectors is presented in [319] and our findings on detection of text are consistent with those of [319]. It is also important to recap that precision and recall are computed using area based metrics. As a result, if the detected bounding box is larger (smaller) than the ground truth, it results in penalizing the precision (recall) of the detector as illustrated in Figure 4.15. The output of the Faster R-CNN based text detector for few sample frames in our dataset is illustrated in Figure 4.16.

Table 4.4: Text Detection Results

	RestNet			**Inception**		
Model	**Precision**	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
SSD	0.83	0.71	0.77	0.82	0.77	0.80
R-FCN	0.79	0.86	0.82	0.84	0.89	0.86
Faster R-CNN	0.82	0.90	0.85	**0.86**	**0.95**	**0.90**
Yolo	-	-	-	0.63	0.69	0.66

In an attempt to provide an insight into the detection errors, few of the errors are illustrated in Figure 4.17. It can be seen that in most cases, the detector is able to detect the textual region but the localization is not perfect i.e. in some cases the bounding box is larger (smaller) than the actual content leading to a reduced precision (recall).

4.4.3 Script Identification Results

For script identification, we employ the same distribution of frames into training and test sets as that of the detection protocol. Text lines from the video frames in the training set are employed to fine-tune the pre-trained ConvNets while the identification rates are computed on text lines from the frames in the test set. A total of $31,321$ Urdu and $16,207$ English text lines are used in the training set while the test set comprises $9,9149$ and $7,425$ text lines in Urdu and English respectively. The resulting confusion matrix is presented in Table 4.5 while the precision, recall and F-measure are summarized in Table 4.6. It can be seen that the model was able to correctly identify the scripts with an accuracy of more than 94%.

Figure 4.15: Computation of precision and recall (a):Ground Truth Bounding Box (b): Detected region is larger than ground truth (c):Detected region is smaller than ground truth (d):Detected region overlaps perfectly with the ground truth

Figure 4.16: Text detection results on sample images (Faster R-CNN with Inception)

Figure 4.17: Imperfect Localization of Text Regions

Table 4.5: Script identification confusion matrix

	Urdu	**English**
Urdu	8763	386
English	551	6874

Table 4.6: Performance of Script Identification

	Precision	**Recall**	**F–Measure**
Urdu	0.94	0.96	0.95
English	0.95	0.93	0.94

4.4.4 Hybrid Text Detection & Script Identification Results

As discussed previously, text detection and script identification can be combined in a single model treating detection as a three (rather than two) class problem. The results of these experiments are summarized in Table 4.7 keeping the same distribution of training and test frames as in the previous experiments. Many interesting observations can be drawn from the results in Table 4.7. Similar to the script independent detectors, models pre-trained on Inception outperform those trained on ResNet and the observation is consistent for all four detectors. Comprehensive analytical studies [320, 321] that aim to compare state-of-the-art CNN models also report the superior performance of Inception over ResNet for the image recognition task.

Comparing the performance of different detectors, Faster R-CNN reports the highest F-measure both for detection of Urdu and English text reading 0.91 and 0.87 respectively. SSD and YOLO are single shot detectors where the prime objective is real time object detection and in pursuit of speed, the accuracy is compromised. Faster R-CNN, on the other hand is a two-stage detector which is not as fast as YOLO or SSD but reports higher F-measure. In all cases, the performances on detection of Urdu text are better than those on English text. This can be attributed to the fact that the data is collected primarily from Urdu News channels which have limited amount of English text.

It is also interesting to note that by combining text detection and script identification in a single model, not only the cascaded solution is avoided, the detection F-measures have also improved (in most cases). This can be attributed to the fact that errors are encountered in both text detection and script identification, and once a cascaded solution is employed, errors of the two stages are accumulated. The total error in such a solution is a combination of detector error and the script identifier error. On the other hand, the hybrid solution detects the textual regions with script information in a single step reducing the error rates. As an example, assuming there are G ground truth text lines out of which D are detected, the detected text lines are fed to the script identification module. The script identifier introduces its own errors correctly recognizing S out of D lines. On the other hand, the hybrid solution detects the textual regions with script information in a single step reducing the error rates i.e. D out of G lines are detected along with script information and additional errors are not introduced. Though the improvement is marginal, eliminating the separate processing of detected text regions to identify the script offers a much simplified (yet effective) solution. Detection outputs on sample frames for the four detectors are illustrated in Figure 4.18. While some sample output of Faster R-CNN on different news channels frames is presented in Appendix D.

To study the effectiveness of the proposed set of anchor boxes, we also (statistically) compared the performance of the detector with default anchor boxes with that of the adapted anchor boxes. F-measure was computed by running the detector on multiple splits of training and test sets (keeping their ratio same) and the t-test was performed. Average F-measures of 0.86 and 0.91 were reported with default and adapted anchor boxes respectively and the performance of the proposed anchor boxes was confirmed to be statistically significant with respect to default anchors.

Table 4.7: Performance of hybrid text detector and script identifier

		RestNet			Inception		
Method	**Script**	**Precision**	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
SSD	**Urdu**	0.83	0.72	0.77	0.82	0.78	0.80
	English	0.80	0.63	0.70	0.82	0.70	0.75
R-FCN	**Urdu**	0.80	0.87	0.83	0.85	0.90	0.87
	English	0.73	0.81	0.77	0.77	0.84	0.81
Faster R-CNN	**Urdu**	0.82	0.92	0.86	**0.87**	**0.95**	**0.91**
	English	0.80	0.81	0.80	**0.81**	**0.94**	**0.87**
Yolo	**Urdu**	-	-	-	0.64	0.70	0.67
	English	-	-	-	0.62	0.67	0.64

In an attempt to carry out an in-depth analysis of the detection performance and its evolution with respect to important system parameters, we carried out another series of experiments using Faster R-CNN (with Inception). In the first such experiment, we study the performance sensitivity to the amount of training data. We train the model by varying the number of text line images (from 10K to 47K) and compute the detector F-measure. Naturally, the detector performance enhances with the increase in the amount of training data (Figure 4.19) and begins to stabilize from around

Figure 4.18: Detection output of hybrid text detection and script identification for different detectors (a): SSD (b): R-FCN (c): Faster RCNN (d): Yolo

30K-35K training lines.

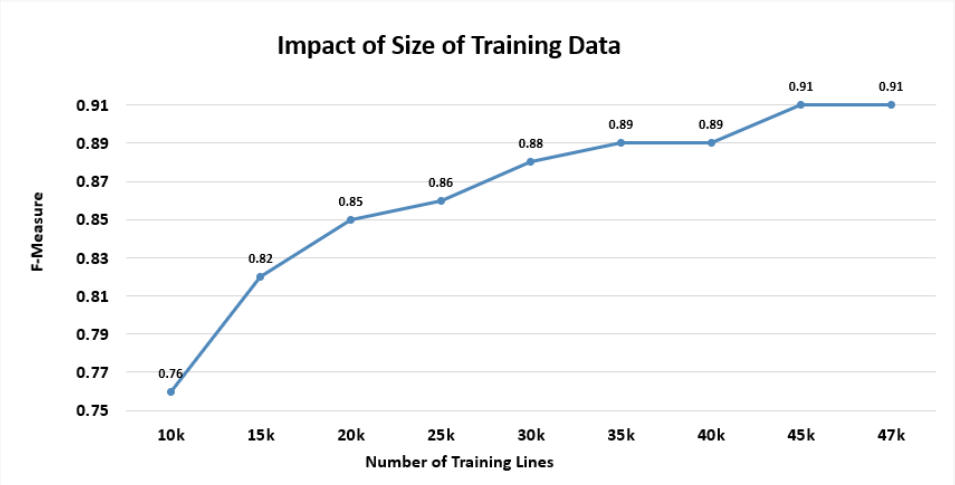


Figure 4.19: Impact of size of training data on text detection performance (Faster R-CNN with Inception)

Resolution of input video frames is an important parameter that might affect the detector performance. To study the detector sensitivity to image resolution, we varied the image resolution from 256×144 to 1920×1080. The resolution was varied only in the test set and all sets of images were evaluated on the detector trained on a single resolution (900×600). The F-measures in Figure 4.20 are more less consistent for varied image resolutions reflecting the robustness of the detector. The proposed anchor boxes adapted for textual content play a key role in achieving this scale invariance. An overall average F-measure of 0.90 ± 0.015 is reported taking into account multiple frame resolutions. Likewise, average precision and recall read 0.90 ± 0.027 and 0.90 ± 0.017 respectively.

From the perspective of computation time, on the average, detection takes 0.36 seconds per frame (at resolution of 900×600) once the model is executed on Tesla K40 GPU Computing Processor with 12GB RAM. Unlike real time object detection, localization of textual content does not require every frame to be processed due to redundancy of content in subsequent frames. The processing time of 0.36 seconds per frame in our system maps to a little less than 3 frames per second making it an appropriate model for retrieval and other related applications.

4.4.5 Performance Comparison

In an attempt to compare the performance of our detector with those reported in the literature, we present a comparative overview of various text detectors targeting cursive caption text in Table 4.8. It is important to note that since different studies are evaluated on different datasets, a direct comparison of these techniques is difficult. Most of the listed studies employ a small set of images (≤ 1000). Moradi et al. [32] and Zayene et al. [33] report results on relatively larger datasets with F-measures of 0.89 and 0.84 respectively. In comparison to other studies, we employ a significantly larger set

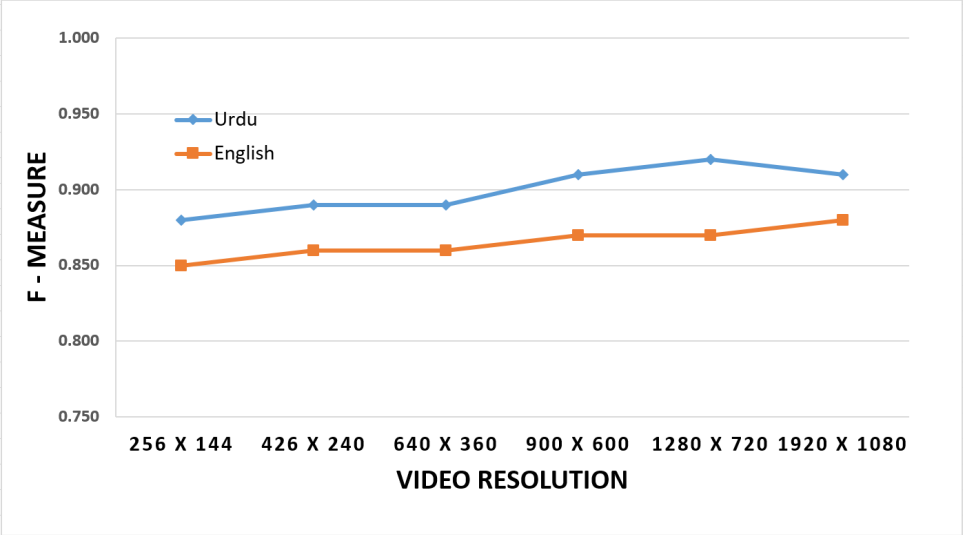


Figure 4.20: Impact of video resolution on text detection performance (Faster R-CNN with Inception)

of images with an F-measure of 0.91. Furthermore, for a fair comparison, we also evaluated our system on the set 1000 images in the publicly available IPC dataset [143] for Urdu and English text. The corresponding F-measures read 0.92 and 0.89 for Urdu and English respectively outperforming other methods evaluated on this dataset hence validating the effectiveness of our detection technique.

Table 4.8: Performance comparison with other techniques

Study	Method	Dataset	Script	Video Frames	Precision	Recall	F-Measure
Jamil et al.(2011) [44]	Edge-based Features	IPC	Urdu	150	0.77	0.81	0.79
Siddiqi and Raza(2012) [143]	Image Analysis	IPC	Urdu	1,000	0.71	0.80	0.75
Moradi et al.(2013) [32]	LBP with SVM	-	Farsi/Arabic	4971	0.91	0.87	0.89
Raza et al.(2013) [46]	Cascade of Transforms	IPC	Urdu	1,000	0.80	0.89	0.84
Raza et al.(2013) [46]	Cascade of Transforms	IPC	Arabic	300	0.81	0.93	0.86
Yousfi et al.(2014) [157]	ConvNet	-	Arabic	201	0.75	0.80	0.77
Zayene et al.(2015) [156]	SWT	AcTiV	Arabic	425	0.67	0.73	0.70
Zayene et al.(2016) [33]	SWT & CAE	AcTiV	Arabic	1843	0.83	0.85	0.84
Shahzad et al.(2017) [47]	Image Analysis	-	Urdu/Arabic	240	0.83	0.93	0.88
Mirza et al.(2018) [295]	Textural Features	UTiV	Urdu	1,000	0.72	0.89	0.80
Unar et al.(2018) [142]	Image Analysis+SVM	IPC	Urdu	1,000	0.83	0.88	0.85
Proposed Method	**Deep ConvNets**	UTiV	**Urdu**	**11,192**	**0.87**	**0.95**	**0.91**
		IPC	**Urdu**	**1,000**	**0.91**	**0.93**	**0.92**
		IPC	**English**	**1,000**	**0.84**	**0.94**	**0.89**

4.5 Summary

This chapter presented the details of techniques developed for detection of caption text appearing in video frames. We fist introduced the image analysis based detection technique and highlighted our motivation of migrating to deep learning based object detectors. A number of object detectors were adapted for detection of textual regions and among these, based on the findings of a comprehensive series of experiments, Faster R-CNN with Inception was eventually chosen. We also demonstrated

the effectiveness of combining text detection and script identification in a single model. The findings of this study are published in [322]. In the next chapter, we introduce the technique developed for recognition of textual regions.

Chapter 5

Recognition of Textual Content

5.1 Introduction

This chapter presents the details of the techniques developed for recognition of cursive (Urdu) caption text. We first discuss the pros and cons of holistic and analytical recognition techniques and the motivation of choosing an implicit segmentation-based analytical technique in our study. We next introduce the details of the recognition technique including the pre-processing, feature extraction and classification. Recognition is modeled as a sequence-to-sequence mapping problem where a Convolutional Neural Network is employed for feature extraction while different variants of Recurrent Neural Networks are investigated for (sequence) classification. We then present the experimental protocol, the realized results and a detailed analysis of the recognition performance with some insights into causes of recognition errors. The chapter concludes with a summary of the key findings.

5.2 Choice of Recognition Unit

As a function of recognition unit, recognition techniques for cursive scripts are categorized into holistic and analytical methods as discussed in Chapter 2. Holistic or segmentation-free techniques employ sub-words (also known as ligatures) as units of recognition while analytical approaches recognize individual characters. Holistic methods avoid the complex segmentation part but a major challenge in these techniques is the large number of unique ligature classes. On the other hand, analytical methods need to discriminate among a small number of character classes which is equal to the number of unique characters in the alphabet and their various context-dependent shapes. Segmentation of cursive text into characters, however, is itself a highly complex problem.

In Urdu, characters may appear in isolated form or are joined with other characters using the joiner rules. The shape of a character, therefore, varies depending upon whether it appears in isolation or, at the start, end or middle of a sequence of joined characters. Since word boundaries

are hard to identify in Urdu and other cursive scripts, holistic recognition techniques typically rely on extracting partial words or ligatures from text line images. Ligatures are further categorized into primary and secondary ligatures. Primary ligatures are the main body component while secondary ligatures correspond to dots and diacritics (Figure 5.1). As a first step, ligatures are extracted from input text images typically using connected component labeling. Secondary ligatures (dots and diacritics) are then associated with their parent primary ligatures by applying morphological operations. Subsequently, ligatures are grouped in to clusters to produce the training data. While the total number of unique ligatures in Urdu is more than 26,000 [167], it has been shown [167] that more than 90% of Urdu corpus can be covered with around 2,000 frequent ligatures only. In some cases, primary and secondary ligatures are separately recognized (to reduce the number of unique classes) and are re-associated in a post processing step [165].

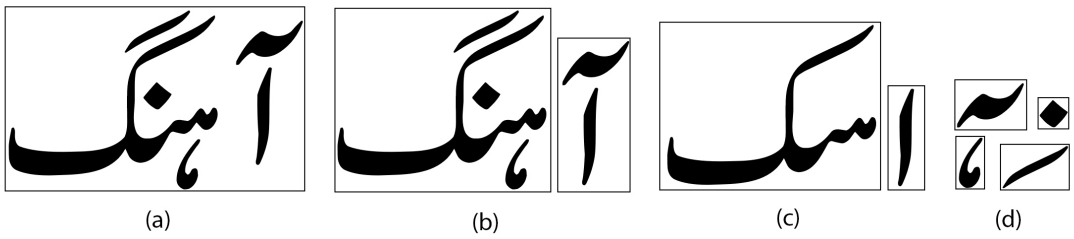


Figure 5.1: (a):A complete Urdu word (b):Ligatures (c):Main body (primary ligature) (d):Dots and diacritics (secondary ligatures)

There are, however, also some challenges associated with recognition using holistic techniques. Generation of ligature clusters along with labels for training the model is a tedious and prohibitively time consuming task. Furthermore, in videos, segmentation of caption text in to ligatures is difficult due to low resolution. In cases, where primary and secondary ligatures are recognized separately, the re-association of secondary ligatures with the parent body is also known to be a major cause of recognition errors [165].

In analytical recognition techniques, characters are used as units of recognition. Both explicit and implicit segmentation of characters can be employed in analytical recognition techniques. However, explicit segmentation of text into characters is a highly challenging problem, therefore, most of the analytical techniques rely on implicit segmentation [323, 37, 211, 213]. With the recent advancements in deep learning, learning algorithms can be provided with text line images along with the corresponding textual transcriptions. The algorithm not only learns various character shapes but also character boundaries. The only challenging part in such implicit segmentation based techniques is the need of 'sufficient' labeled training data.

In our study, we employed an analytical technique for recognition of Urdu caption text. The decision is also supported by our preliminary investigations on holistic recognition (details are provided in Appendix E) which do not scale up well as the vocabulary size increases. Unlike

holistic techniques, analytical recognition also allows recognizing the words which the system might not have seen in the actual training set. The details of the proposed recognition technique are presented in the next section.

5.3 Recognition using Analytical Technique

This section presents the details of the proposed recognition technique. Text line images are first pre-processed to segment text from the background. The binarized images of text lines are then fed to a convolutional neural network for feature extraction. The generated feature map is subsequently provided as input to a recurrent neural network using sliding windows. Finally, being a sequence-to-sequence mapping problem, we introduce a connectionist temporal classification layer for sequence alignment. An overview of these steps is presented in Figure 5.2 while each of these is detailed in the following.

5.3.1 Pre-Processing

While the recognition engine can be fed with colored or grayscale images, removing the background information and binarizing the image allows the learning algorithm better learn the character shapes and boundaries. For images with simple and homogeneous backgrounds, a global thresholding suffices. Video frames, however, often contain text on multiple, non-homogeneous backgrounds. Furthermore, there are two scenarios in which text may appear; dark text on bright background or bright text on dark background. Once the image is binarized, we need all text lines to follow one of the two conventions. In our study, we assume dark text on bright background and if this is not the case, we invert the polarity of the grayscale image prior to binarization.

As a first step, we need to detect the polarity of the text. Canny edge detector is applied to the grayscale text line image and blobs are identified. These blobs correspond to (approximate) text regions in the image. Region filling is applied to these blobs and the generated binary image is used as mask on the grayscale image to extract potential text regions (characters or ligatures). We then compute the median gray value (Med_{text}) of the extracted blobs as well as the median gray value of the background (all pixels which do not belong to any blob), Med_{back}. If $Med_{text} < Med_{back}$ we have dark text on bright background and the polarity agrees with our assumed convention. On the other hand, if $Med_{text} > Med_{back}$, this corresponds to bright text on dark background. In such cases, the polarity of the image is reversed prior to any further processing [324]. The process is summarized in Figure 5.3.

Once all text lines contain text in the same polarity, we binarize the images to contain only textual information. For binarization, we investigated a number of thresholding techniques. These

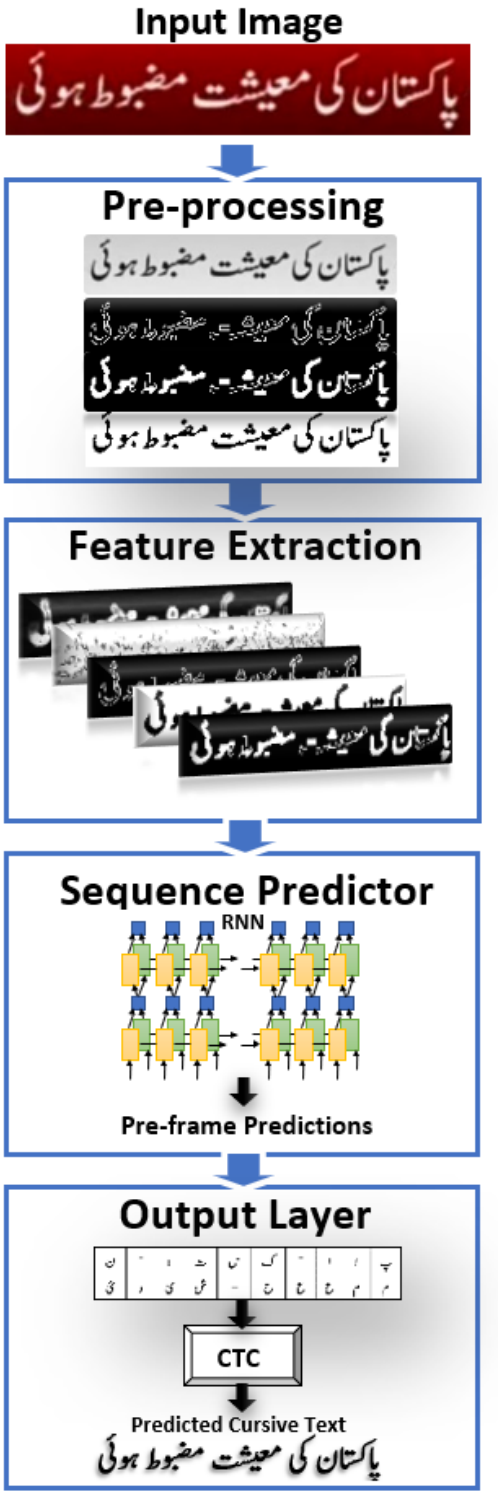


Figure 5.2: An overview of the key processing steps

include Otsu's global thresholding method [293] as well as a number of local thresholding algorithms. The local thresholding algorithms are adaptive techniques where the threshold value of each pixel is computed as a function of the neighboring pixels. Most of these algorithms are inspired

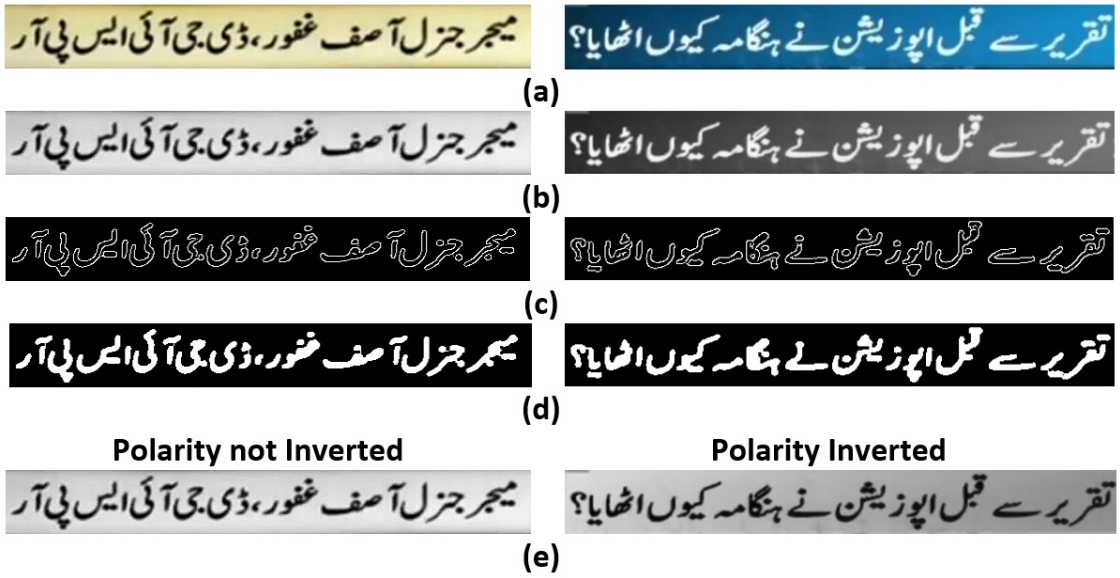


Figure 5.3: Identification of polarity of text (a):Original image (b):Gray scale image (c):Text blobs (d):Filled text blobs serving as a mask to extract corresponding blobs from the gray image (e): Final image (Image on the right is inverted while the one on left remains unchanged)

from the classical Niblack thresholding [88] where the threshold is computed as a function of the mean and standard deviation of the gray values in the neighborhood of a reference pixel. Other algorithms investigated in our study include Sauvola [325], Feng's [326] and Wolf's thresholding algorithm [53]. A brief description of these methods is presented in the following.

Otsu's Global Thresholding [293], named after Nobuyuki Otsu, is a classical binarization method that converts an image into two classes based on a single threshold value. As a function of the distribution of gray values in the input image I, Otsu's method computes a threshold value (T_{Otsu}) which is then employed to binarize the image.

$$O(i,j) = \begin{cases} 1 & if\ I(i,j) \geq T_{Otsu} \\ 0 & otherwise. \end{cases} \tag{5.1}$$

Niblack thresholding [88] is one of the earliest local binarization algorithms that computes a threshold for every pixel in the input image as a function of the the neighboring pixels. The threshold is calculated by considering a small rectangular window around each pixel using the mean m and standard deviation s of the pixel values within a window as outlined in the following.

$$T_{Niblack} = m + k \times s \tag{5.2}$$

where m is the mean gray value of pixels within a window, s is the standard deviation and k represents the Niblack constant that is fixed to -0.2. While Niblack's thresholding is known to

correctly binarize the text regions in an input image, it tends to introduce noise in non-text regions of the image.

Sauvola's Algorithm [325] is an improvement of Niblack's thresholding and incorporates the dynamic range of image gray level values in calculating the threshold.

$$T_{Sauvola} = m \times (1 - k \times (1 - \frac{s}{R})) \tag{5.3}$$

Similar to Niblack's algorithm, m and s represent respectively the mean and standard deviation of the gray values within a window while k and R are fixed to 0.5 and 128 respectively. This method provides good binarization results in case the pixel values in the input grayscale image are near to black or white. If the values of text and non-text pixels are close to each other, the results degrade significantly.

Wolf's Algorithm [53], an enhancement of Sauvola's method, was specifically developed for binarization of multimedia document images. The algorithm normalizes the average gray and contrast value of the image and computes the binarization threshold as follows.

$$T_{Wolf} = (1-k) \times m + k \times M + k \times \frac{s}{R}(m - M) \tag{5.4}$$

Where k is fixed to 0.5, M is the minimum gray value in the image while R represents the maximum value of standard deviations in all windows.

Feng's Binarization [326] presented the idea of using two local windows to calculate the dynamic standard deviation of the grayscale image (in contrast to a single window in Wolf's algorithm). The local mean m, minimum gray-level M, and standard deviation s are calculated in the first (smaller) window while the dynamic range standard deviation R_s is computed in the second (larger) window. Threshold for binarization is then calculated as follows.

$$T_{Feng} = (1 - \alpha_1) \times m + \alpha_2 \times \left(\frac{s}{R_s}\right) \times (m - M) + \alpha_3 \times M \tag{5.5}$$

Where $\alpha_2 = k_1(s/R_s)^{\gamma}$ and $\alpha_3 = k_2(s/R_s)^{\gamma}$; the value of γ is set to 2 by the authors while for other parameters, α_1 varies from 0.1 to 0.2, k_1 from 0.15 to 0.25 and k_2 form 0.01 to 0.05. These ranges have been empirically determined by the authors based on a comprehensive series of experiments.

Prior to binarizing the images, we also apply a smoothing (median) filter on each text line to remove/suppress any noisy patterns in the image. Binarization results of applying various thresholding techniques to a sample text line image are illustrated in Figure 5.4. From the subjective analysis of these results, Wolf's algorithm that was specifically proposed for low resolution video text, seems to outperform other techniques. Nevertheless, it is hard to generalize from visual inspection of few

sample images and the recognition rates on images generated by each of these techniques could be a better indicative of the effectiveness of the method. Following binarization, we normalize the height of each text line to a fixed size (32 pixels in our case) while the width of the line is a function of the textual content it contains.

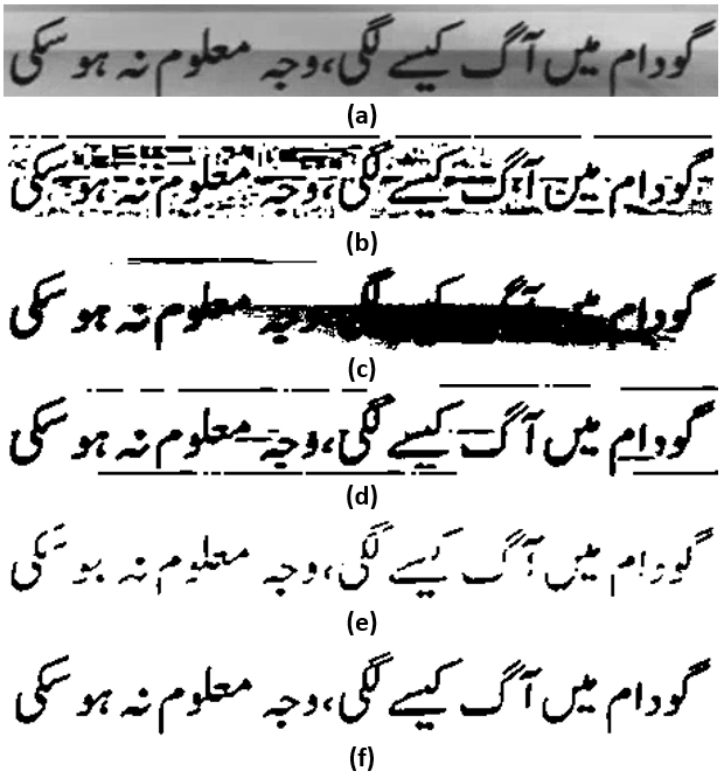


Figure 5.4: Binarization results on a sample text line (a): Grayscale Image (b):Niblack (c):Otsu's Global Thresholding (d):Feng's Algorithm (e):Sauvola's Algorithm (f):Wolf's Algorithm

5.3.2 Feature Extraction

Once the text lines are pre-processed, we proceed to the next step of feature extraction. As mentioned earlier, a given text line image contains a sequence of characters which needs to be mapped to the corresponding sequence of characters in the ground truth transcription. The problem is hence formulated in a sequence-to-sequence mapping framework. The input sequences to the model can be raw pixel values or features (hand-crafted or machine learned) extracted using a sliding window protocol. A number of recent studies [127, 128, 129], validate the superiority of machine-learned features over hand-engineered features (and raw pixel values). We, therefore, employ a convolutional neural network as feature extractor. The architecture of a CNN is a function of many hyper-parameters. In our study, we employed a deep neural network architecture which contains seven convolutional layers (with max pooling). The configuration of the designed network including filter sizes, strides, padding and output volume shapes is summarized in Table 5.1 while

the architecture is visually presented in Figure 5.5. The cascaded convolutional layers serve as (hierarchical) feature extractors mapping the input text line image to feature maps. The output of the CNN is a 128×512 dimensional feature vector which is fed to an RNN for sequence classification.

Table 5.1: Architectural details of the proposed CNN

Layer Type	Filter Size	Stride	Padding	Output Volume	Trainable Parameters
Input	-	-	-	$32 \times 512 \times 1$	-
Convolution	3 x 3	1	1	$32 \times 512 \times 64$	640
MaxPooling	2 x 2	2	-	$32 \times 256 \times 64$	-
Convolution	3 x 3	1	1	$16 \times 256 \times 128$	73,856
MaxPooling	2 x 2	2	-	$8 \times 128 \times 128$	-
Convolution	3 x 3	1	1	$8 \times 128 \times 256$	295,168
Convolution	3 x 3	1	1	$8 \times 128 \times 256$	590,080
MaxPooling	2 x 1	2,1	-	$4 \times 128 \times 256$	-
Convolution	3 x 3	1	1	$4 \times 128 \times 512$	1,180,160
Convolution	3 x 3	1	1	$4 \times 128 \times 512$	2,359,808
MaxPooling	2 x 1	2,1	-	$2 \times 128 \times 512$	-
Convolution	2 x 2	1	2,1	$1 \times 128 \times 512$	1,049,088
				Total Parameters	**5,548,800**

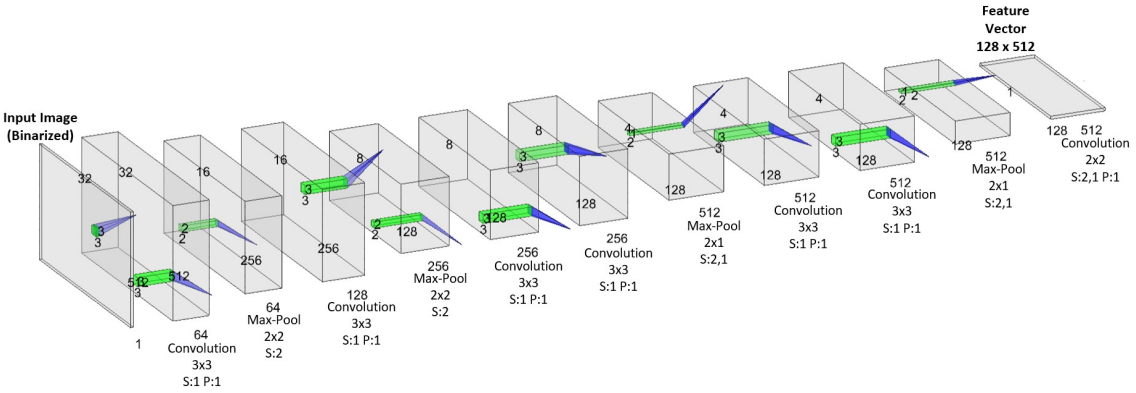


Figure 5.5: Architecture of the convolutional neural network employed for feature extraction

5.3.3 Sequence Prediction with Recurrent Nets

Unlike traditional classification tasks (one-to-one mapping), CNNs cannot be directly employed to recognize the complete text lines. The input in text recognition is a sequence of characters and the predictor is required to produce the corresponding string as output. The shape of a character depends on the preceding as well as the subsequent characters within a word (ligature). These dependencies in the input can be modeled using the recurrent neural networks. RNNs take the input, process it through multiple time steps and hidden layers, and produce output. The functional dependencies of hidden layers on previous hidden layers (in depth as well as time) allow RNNs to exploit the contextual dependencies in the input. RNNs can be employed to solve a variety of

sequence mapping problems including one-to-many (image captioning), many-to-one (sentiment analysis) many-to-many (language translation) mappings.

The hidden state of a simple (single layer) RNN (Figure 5.6-a) at time-step t is defined as a function of the input x_t and the previous hidden state h_{t-1}.

$$h_t = tanh(W_{hh}\, h_{t-1} + W_{xh}\, x_t + b_h) \tag{5.6}$$

The typical activation function in RNNs is $tanh$ while W_{hh} and W_{xh} represent the weight matrices corresponding to the previous hidden state and the current input respectively. Once the hidden state is updated, the output is calculated as follows.

$$y_t = g(W_{hy}\, h_t + b_y) \tag{5.7}$$

The output y_t can be calculated at multiple time-steps (many-to-many or one-to-many mappings) or the final time-step only (many-to-one mapping). An inherent problem of RNNs is the inability to model long-term dependencies in the input. This is due to vanishing (exploding) gradients during back propagation through time. The vanishing gradient problem refers to the scenario when the gradient shrinks as it back propagates through time and if it becomes extremely small, it does not contribute to learning. To address these issues, Long Short-Term Memory Networks (LSTMs) were introduced [327] to model long-term dependencies in the input. LSTMs (Figure 5.6-b) are a special kind of RNN, capable of learning long-term dependencies. The key to LSTMs is the cell state. LSTMs have the ability to add or remove information to the cell state using *gates*. Gates are composed of a sigmoid layer followed by a point-wise multiplication operation which can optionally let information to flow from one state to another.The sigmoid layer outputs numbers between 0 and 1, describing how much of each component should be let through.

A typical LSTM cell has a forget gate, an input gate and an output gate. The forget gate as a function of current input x_t and the previous hidden state h_{t-1} controls what information should be removed from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

To update the cell state, the input gate first decides which values in the cell state are updated.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

A $tanh$ layer next produces a vector of candidate values that could be added to the cell state.

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Finally, the cell state is updated from C_{t-1} to C_t regulated by the forget and input gates.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

The cell output h_t is a filtered version of the cell state C_t regulated by the output gate o_t.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot tanh(C_t) \tag{5.8}$$

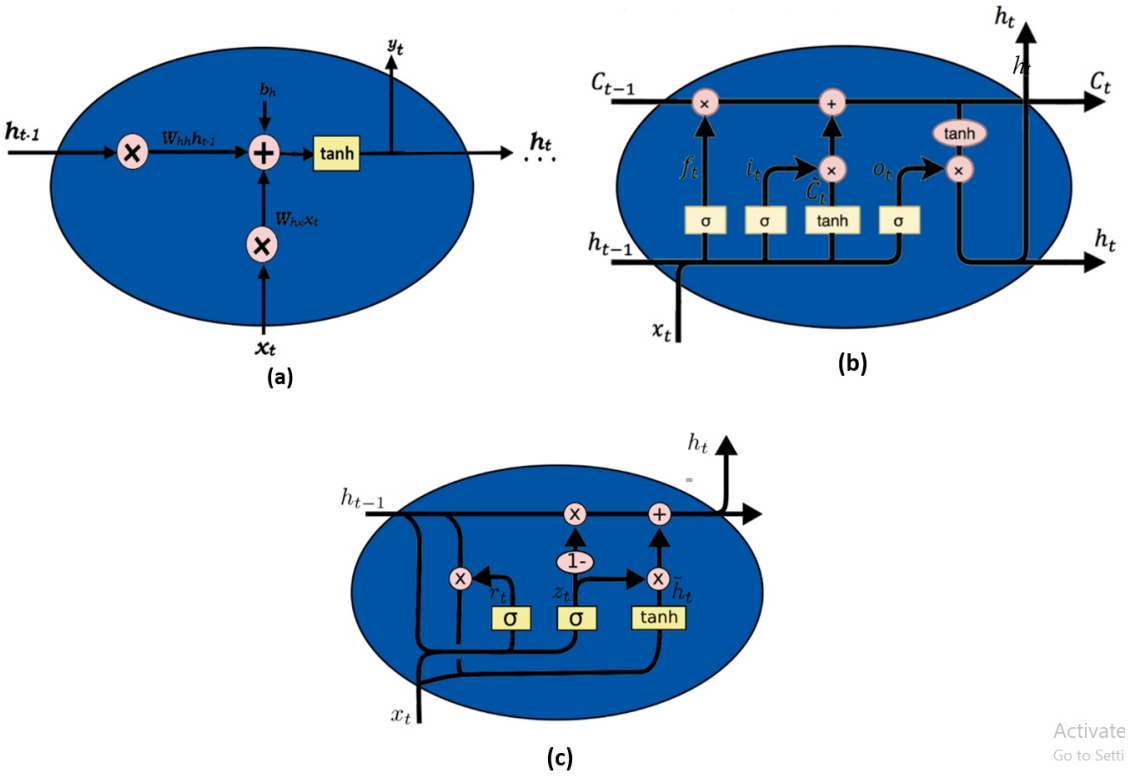


Figure 5.6: Architectures of (a): Simple RNN (b): GRU (c): LSTM

LSTMs were followed by Gated Recurrent Units (GRUs) [328] which significantly simplified the cell architecture (Figure 5.6-c). Unlike three gates in an LSTM cell, GRU employs only two gates, a reset gate and an update gate. Likewise, GRUs also eliminate separate cell and hidden sates and only the hidden state is employed to transfer information to the next time-step. The update gate in a GRU regulates how much of the past information needs to be passed to the next time-step.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

Likewise, the reset gate helps the model determine the past information to forget.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

The candidate memory state is computed using the reset gate as follows.

$$\tilde{h_t} = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + bt)$$

As a final step, the current memory state is updated using the update gate which regulates what to collect from the candidate state $\tilde{h_t}$ and what from the previous time-step h_{t-1}.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \tag{5.9}$$

In our study, we investigated the performance of simple RNN as well as its advanced variants (LSTMs and GRUs) for sequence prediction. Since character shapes are dependent both on preceding and subsequent characters in the sequence, bi-directional RNNs are employed. The output of the CNN (128×512) is fed as a sequence (with 512 time-steps and input at each time-step being a vector of size 128) to the RNN. The proposed RNN architecture contains two stacks of hidden layers where each stack contains a forward and a backward layer with 256 hidden units each. The recurrent layers are connected to a fully connected layer which outputs the predicted sequence of characters. The overall architecture is summarized in Figure 5.7 while the configuration details are presented in Table 5.2. The impact of changing the model design on recognition rates can be found in Appendix G. The predictions of the recurrent network are passed to a Connectionist Temporal Classification (CTC) layer [329] for text alignment as discussed in the next section.

Table 5.2: Architectural details of the recurrent network

Cell Type	Layer Type	No. of Neuron	Input Vector	Trainable Parameters	Total Trainable Parameters
RNN	Bidirectional Stack-1	256	128×512	393,728	1,050,112
	Bidirectional Stack-2	256	512×512	393,728	
	Dense	80	512	262,656	
GRU	Bidirectional Stack-1	256	128×512	1,181,184	2,625,024
	Bidirectional Stack-2	256	512×512	1,181,184	
	Dense	80	512	262,656	
LSTM	Bidirectional Stack-1	256	128×512	1,574,912	3,412,480
	Bidirectional Stack-2	256	512×512	1,574,912	
	Dense	80	512	262,656	

5.3.4 Connectionist Temporal Classification (CTC) Layer

A CTC layer serves to convert the raw predictions of RNN into the actual transcription of a given text line image aligning the output sequence of RNN with the target labels. The alignment of labels is learned during the training process. The CTC layer keeps record of all labels in the transcription along with a special extra character which separates the consecutive occurrence of characters in transcription. CTC layer predicts the most probable sequence of labels against the sequence predicted by the the RNN.

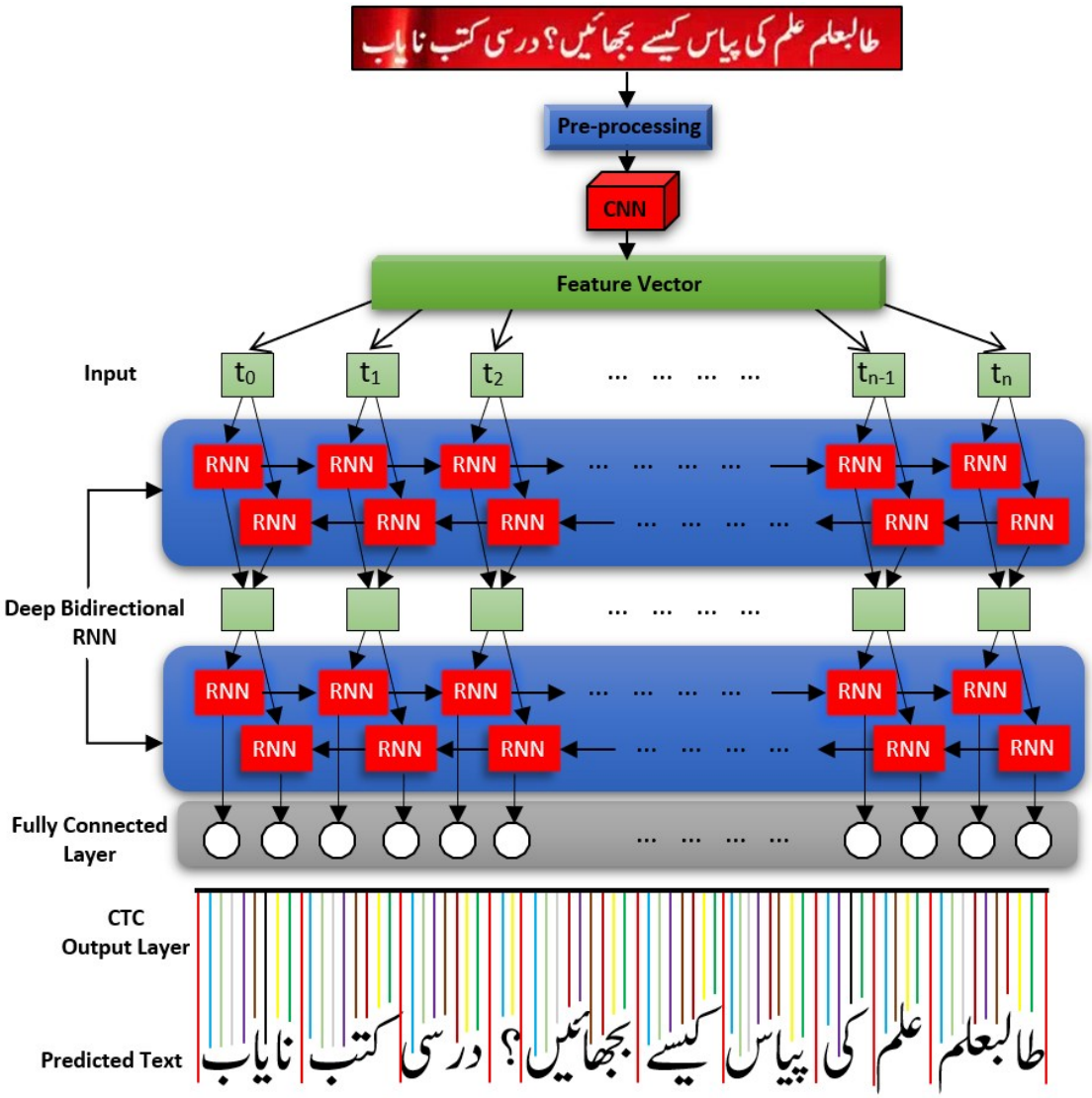


Figure 5.7: Architecture of the bidirectional (left-to-right & right-to-left) RNN model with CTC output layer

Figure 5.8 shows an example of a sequence (of characters) that is produced by the sequence predictor and contains repetition of characters, the special character ('-') and a character for white space. The extra pseudo character ('-') is to be distinguished from a white space character and is used to solve the duplicate character problem. While encoding the text, a character may be repeated any number of times and, any number of blanks can be inserted between any of the characters. The blank character must be inserted between duplicate characters. During the decoding, the first step is to remove all repeated characters. The special character (blank) is then removed to produce the final transcription.

As discussed previously, the CTC layer allows the model to be trained in an end-to-end manner by providing the text line images and the respective transcriptions. Training is guided by the CTC

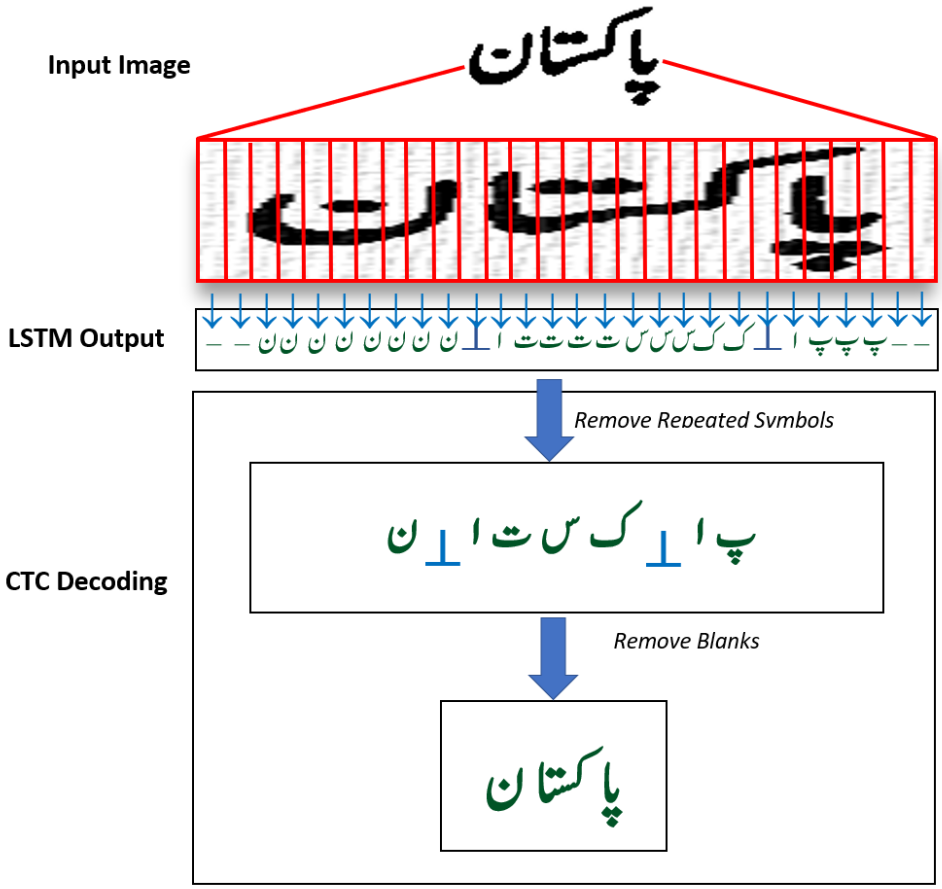


Figure 5.8: CTC Decoding Example

loss function which calculates probabilities of all possible alignments of the ground truth text in the image and takes the sum of all scores. Given a training set with pairs of text line images and the corresponding transcriptions, the model parameters are updated to minimize the negative log-likelihood of the probability of producing the output transcription:

$$\sum_{(X,Y)\in\mathscr{D}} -\log\ p(Y \mid X) \tag{5.10}$$

Where X is the sequence of input features and Y is the output transcription.

The process is elaborated through Figures 5.9 and 5.10. The vocabulary is assumed to be two characters ('pay' and 'alif'), Figure 5.9-a shows the ground truth transcription of an input image while Figures 5.9-b summarizes the output probabilities of the characters in the vocabulary at three time-steps. Figure 5.10-a shows all possible alignments which produce the ground truth text. The probability of a sequence is computed by multiplying the probabilities of the corresponding characters at the respective time-steps. These probabilities are then added and the cumulative probability is converted into loss by taking its negative logarithm (Figure 5.10-b). The loss value is

then back-propagated through all layers and the parameters are updated.

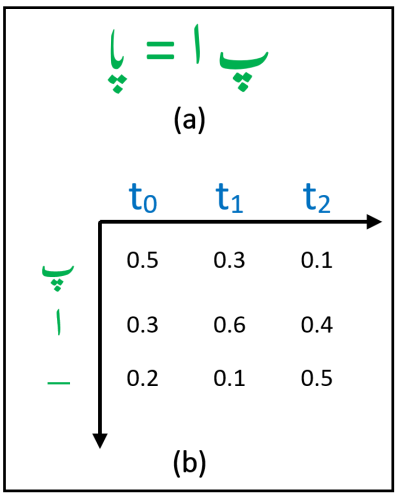


Figure 5.9: RNN output (probabilities) with three time-steps and two characters (given in (a)) along with CTC blank ('-')

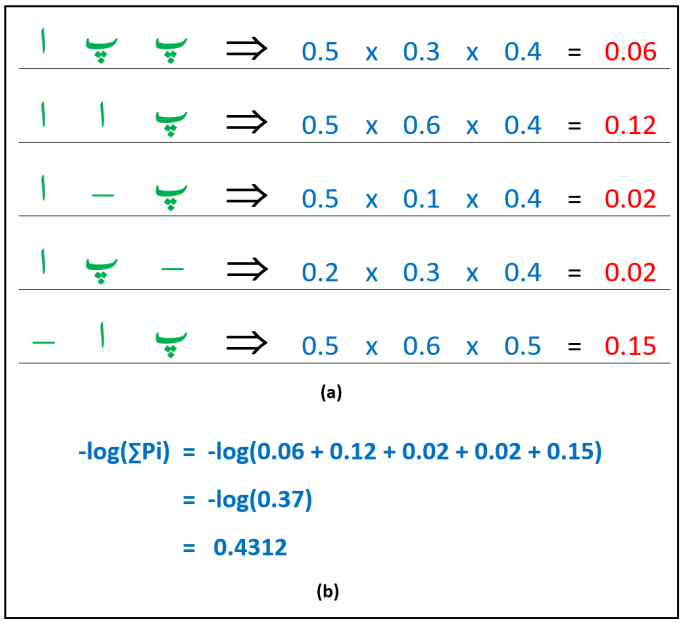


Figure 5.10: (a): All possible alignments of character sequences producing the ground truth text in Figure 5.9 (b): Summary of CTC loss calculation

5.4 Model Training and Recognition

The CNN, bidirectional LSTM and CTC layer are combined in a single end-to-end trainable network. The model is provided with the text line images along with the corresponding transcriptions

to learn different character shapes and boundaries. CNN layers extract features from text lines and pass them to the bidirectional recurrent layers which predict the sequence and feed it to the CTC layer for alignment.The model is trained using the CTC loss as discussed in the previous section while other parameters involved in the training are summarized in Table 5.3. Once the model is trained, it can be fed with the query text line images to predict the transcription.

Table 5.3: Training parameters of recognition network

Training Parameters	
Parameter	**Value**
Optimizer	Adam
Learning Rate	0.0001
Learning Rate Decay	0.96
Batch Size	128

After having discussed the details of the recognition engine, we present the experimental protocol and the realized results in the next section.

5.5 Experiments and Results

To evaluate the effectiveness of the proposed recognition technique, we carried out a comprehensive series of experiments. We first introduce the experimental protocol followed by the recognition results as a function of pre-processing, type of RNN cell and various combinations of training data. Finally, we present a comparative analysis of the reported results with respect to other similar studies.

5.5.1 Experimental Protocol

The experimental study of the system is carried out on text lines extracted from the video frames using ground truth information. The total number of text lines extracted from video frames of four different News channels sum up to a total of 40,470. Among these, 27,321 text lines are used in the training set, 4,000 text lines (1,000 from each channel) in the validation set while 9,149 text lines are used in the test set. It is ensured that text lines from a given video are only in one of the training or test sets. The distribution of text lines into training, validation and test sets along with the statistics on the number of words and characters in each, are summarized in Table 5.4. In some of the experiments, we also employed the 50,000 synthetic text lines. These synthetic text lines, however, are only employed in the training set (Table 5.4) while the validation and test sets for all experiments are kept the same.

Table 5.4: Distribution of dataset including synthetically generated text lines

	Train		Validation		Test	
Type of Data	**Lines**	**Chars**	**Lines**	**Chars**	**Lines**	**Chars**
Video Text	27,321	556,773	4,000	81,516	9,149	277,819
Synthetic Text	50,000	1,259,339	-	-	-	-
Total	77,321	1,816,112	4,000	81,516	9,149	277,819

5.5.2 Recognition Results

In the first series of experiments, we studied the recognition performance as a function of pre-processing (binarization) technique and the type of RNN cell (Simple RNN, GRU, and LSTM). These experiments are carried out on actual text lines only and synthetic data is not included at this stage, i.e. 27,321 text lines in the training set, 4,000 in validation and 9,149 in the test set (Table 5.4). Furthermore, we also compared the performance of feeding the RNN with raw pixel values and CNN based features. For all experiments we quantify the system performance by computing the character recognition rates. The recognition engine outputs the predicted transcription of the query text line. Recognition rates are calculated by computing the Levenshtein distance between the predicted and the ground truth transcription. The recognition rates corresponding to the first series of experiments are summarized in Table 5.5.

A number of interesting observations can be made from the reported recognition rates. First of all, it can be seen that features computed using CNN report higher recognition rates in all experiments as compared to raw pixels. Secondly, RNNs implemented with Gated Recurrent Units perform better than simple RNN cells while LSTM based model outperforms the other two in all cases for a given pre-processing (binarization) technique. Comparing the various binarization techniques, the grayscale text lines report higher recognition rates when compared to those obtained on text lines binarized using Niblack and Otsu's thresholding algorithms. This observation is consistent with our initial assessment of binarization algorithms where, in general, Niblack's binarization introduces a lot of noise in the binarized images while global thresholding fails once the text images have non-homogeneous backgrounds. The performance of Feng's and Sauvola's binarization methods is more or less similar. Text lines binarized using Wolf's algorithm report the highest recognition rates. This observation is also consistent with the subjective analysis of binarization techniques where Wolf's algorithm produced relatively cleaner versions of binarized images. Overall, the highest reported recognition rate reads 95.98% when using the CNN-LSTM combination and binarizing the text lines using Wolf's algorithm. We also carried out the analysis of variance (ANOVA) test to confirm if the results obtained by CNN-LSTM combination are statistically significant as opposed to those reported with other combinations and raw pixels. The binarization technique was fixed to Wolf's algorithm and multiple splits of training and test data were employed keeping their ratio same. The test confirmed that the reported superiority of the CNN-LSTM combination is statistically significant. Based on these observations, the subsequent experiments are carried out with Wolf's binarization technique as the pre-processing step and the

combination of CNN and LSTM as the recognition model.

Table 5.5: Summary of character recognition rates in different experimental settings

	Character Recognition Rate, %					
	Raw Pixels			**CNN Features**		
	RNN	**GRU**	**LSTM**	**RNN**	**GRU**	**LSTM**
Grayscale	72.57	75.19	77.68	80.09	82.35	84.1
Niblack [88]	70.66	73.39	74.81	76.87	79.91	82.13
Otsu [293]	69.36	71.45	75.23	76.54	79.11	80.39
Feng [326]	76.67	79.85	81.31	82.94	84.56	87.91
Sauvola [325]	74.27	78.36	80.19	80.65	83.41	86.25
Wolf [53]	81.78	83.88	86.06	90.18	92.76	95.98

In the second series of experiments, we study the impact of training data on the recognition performance (using CNN-LSTM with Wolf's binarization). Furthermore, to provide deeper insights, in addition to character recognition rates, we also computed text line recognition rate. A text line is considered to be correctly recognized if all characters constituting the line are classified correctly. The models are trained using three different scenarios, using text lines from video frames, using synthetic text lines only and by combining the video text lines with synthetically generated text lines.

The results of these experiments are presented in Table 5.6. It is interesting to note that when the system is trained using only synthetic data, it still reports acceptable recognition rates reading 69.68% and 89.32%, at line and character levels respectively. Combining the video text lines with synthetic text lines improves the character recognition rate from 95.98% to 97.63% demonstrating the effectiveness of the generated text lines. The enhanced recognition rates when using synthetic data can be attributed to the fact that some of the character combinations which could not be captured in the original text lines are represented in the synthetic text lines leading to improved recognition rates. Training with synthetic data, naturally, took slightly longer to converge (Figure 5.11) as the learning algorithm has more number of character combinations to learn.

Table 5.6: Recognition rates as a function of training data

Item	**Recognition Rate (%)**	
Training Data	**Line**	**Chars**
Videos	77.53	95.98
Synthetic	69.68	89.32
Videos + Synthetic	81.34	97.63

In the last series of experiments, we studied the impact of size of training data on the recognition performance. Keeping the test size (and all other system parameters) fixed, we varied the number of training text lines from 3,000 to 27,321. The corresponding recognition rates are illustrated in Figure 5.12 where it can be seen that the recognition rates begin to stabilize from 15,000 lines of

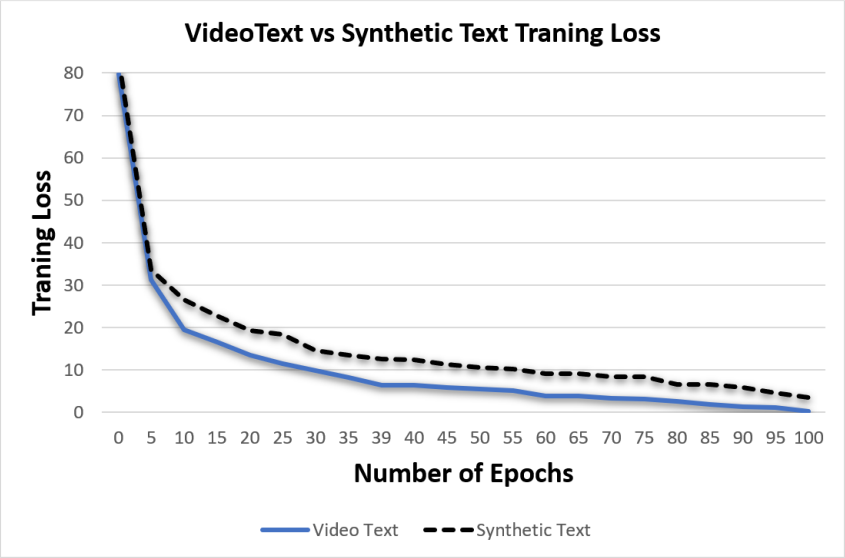


Figure 5.11: Training loss for video and synthetically generated text lines

text on wards which is a manageable size for such applications.
From the view point of recognition time, the time to recognize a text line is naturally a function of the length of text. We report the average recognition time per line, the average being computed on all text lines in our test set. Recognition takes on average 0.18 seconds per text line on Tesla K40 GPU Computing Processor with 12GB RAM. Video frames, on the average, contain 4 to 5 text lines hence the recognition engine can process one frame per second allowing it to be employed for indexing and retrieval applications.

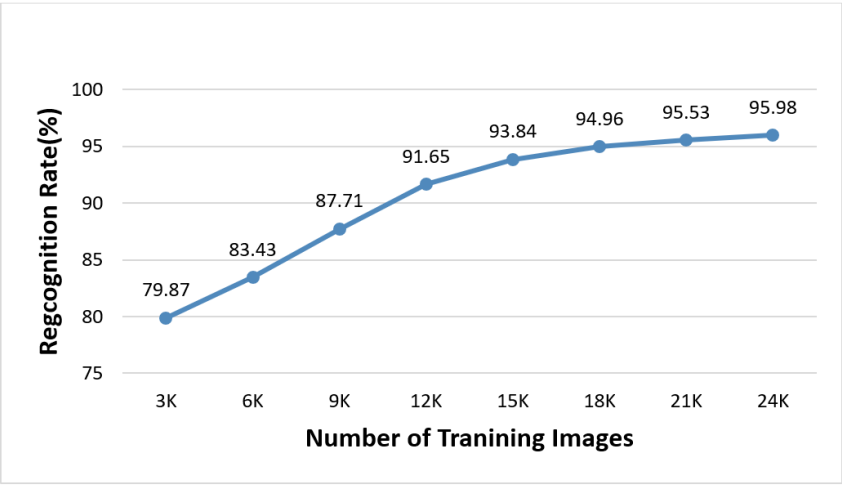


Figure 5.12: Recognition rates as a function of size of training data

5.5.3 Performance Comparison

To provide an idea of the effectiveness of the proposed recognition technique, we present a comparative analysis of various recent studies. Naturally, a meaningful quantitative comparison is only possible if all techniques are evaluated on the same dataset using the same experimental protocol. However, unfortunately, due to lack of benchmark datasets for this problem, the reported techniques are mostly evaluated on custom developed datasets. Nevertheless, for completeness we present these results to give readers an idea of the current state-of-the-art on this problem. Furthermore, in addition to caption text, we also list the recognition rates reported on printed document text in well-known recent studies. These results are summarized in Table 5.7 with a summary of techniques and the size of dataset employed. In case of printed text, the highest reported recognition rate is 98.12% on the UPTI dataset [63]. It is however important to mention that UPTI is a synthetically generated dataset that does not offer the same kind of challenges as those encountered in scanned images of documents or caption text. In case of caption text, a recognition rate of 96.85% is reported on a relatively smaller set of Arabic text lines. For Urdu caption text, Tayyab et al. [49] achieve 93% recognition rate on approximately 20,000 text lines while Hayyat et al. [48] report a ligature recognition rate of 99.5%. The dataset considered in [48] however, is fairly limited with only 290 unique ligature classes. In our experiments, we report a recognition rate of 95.98% (97.63% with synthetic data in training) which, though not directly comparable with reported studies, is indeed very promising considering the complexity of the problem. Furthermore, since the UPTI dataset [63] is publicly available, we also trained our model using the text line images in the UPTI dataset and realized a character recognition rate of 99.14% outperforming other studies evaluated on the same dataset [168, 213, 190] and validating the effectiveness of our proposed model.

Table 5.7: Results comparison with other recognition techniques

Image Type	Study	Language	Technique	Database	Data Size	Results
Document	Ahmed et al.(2007) [203]	Urdu	ANN	Private	56 LC	93.40%
	Hassan et al.(2013) [330]	Urdu	BLSTM	UPTI	10,000 Lines	94.85%
	Akram et al.(2014) [206]	Urdu	DCT & HMMs	CLE	224 Images	86.15%
	Hussain et al.(2015) [331]	Urdu	DCT & HMMs	CLE	5,249 Ligatures	87.44%
	Ahmed et al.(2016) [332]	Urdu	BLSTM	UPTI	15,251 Lines	96.00%
	Naz et al.(2016) [168]	Urdu	MDLSTM	UPTI	10,000 Lines	96.40%
	Naz et al.(2017) [213]	Urdu	MDLSTM	UPTI	10,000 Lines	94.97%
	Naz et al.(2017) [190]	Urdu	CNN & MDLSTM	UPTI	10,000 Lines	98.12%
Videos	Zayene et al.(2018) [31]	Arabic	MDLSTM	AcTiV-R	7,843 Lines	96.85%
	Tayyab et al.(2018) [49]	Urdu	RNN	Private	19,824 Lines	93.02%
	Hayat et al.(2018) [48]	Urdu	CNN	Private	290 LC	99.50%
	Proposed	**Urdu**	**CNN+LSTM**	**UTiV**	**40,470 Lines**	**95.98% (97.63%)**
		Urdu	**CNN+LSTM**	**UPTI**	**10,000 Lines**	**99.14%**

Figure 5.13 presents a screen shot of the visual application that was developed for recognition of text lines. Furthermore, to provide insights into recognition errors, some common errors produced by the system are illustrated in Figure 5.14 where it can be seen that a major proportion of errors results due to false recognition of secondary ligatures (dots and diacritics) while the main body ligatures is correctly recognized in most cases.

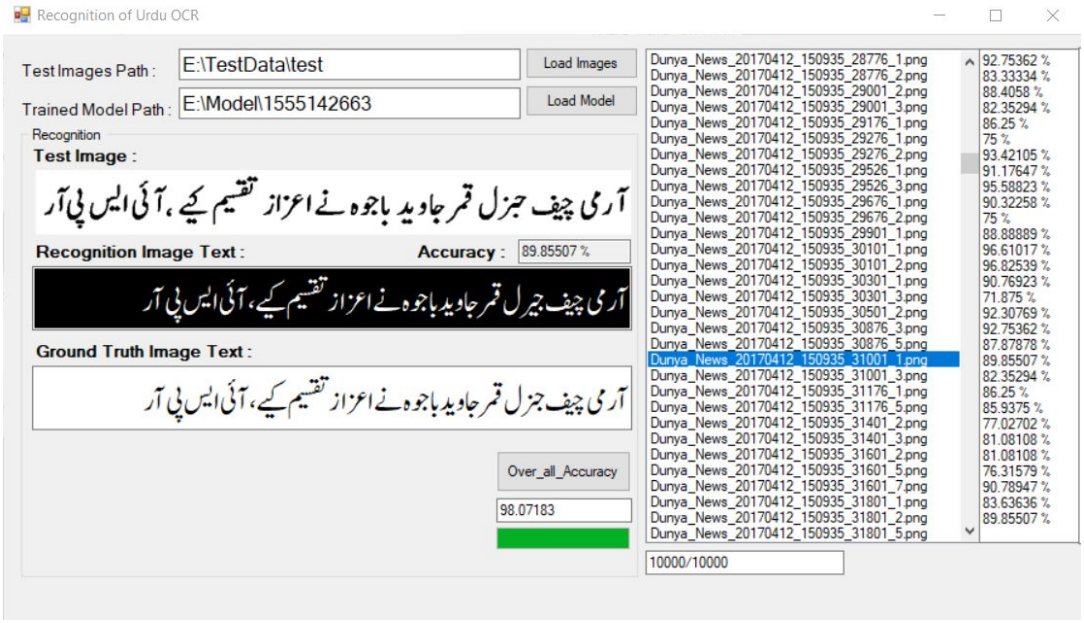


Figure 5.13: Screen shot of the recognition application developed in C#.NET and Python

5.6 Summary

This chapter presented the technical details of the text recognition module for Urdu caption text. Although the methodology is developed using Urdu text, the technique can be adapted to other cursive scripts as well. The proposed recognition technique relies on pre-processing the text line images and feeding the binarized images along with the ground truth transcriptions to an end-to-end trainable CNN+RNN model. The convolutional layers convert the raw images into feature maps while the recurrent layers carry out the sequence prediction. Finally, a CTC layer is employed to convert the raw predictions into actual transcriptions. The technique was evaluated through a comprehensive series of experiments and the reported performance was compared with the similar recent studies. High character recognition rates on a large set of test line images validated the robustness of the proposed recognition engine. The findings of this study are published in [333, 334]. In the next chapter, we present potential applications by combining the detection and recognition modules into a single system.

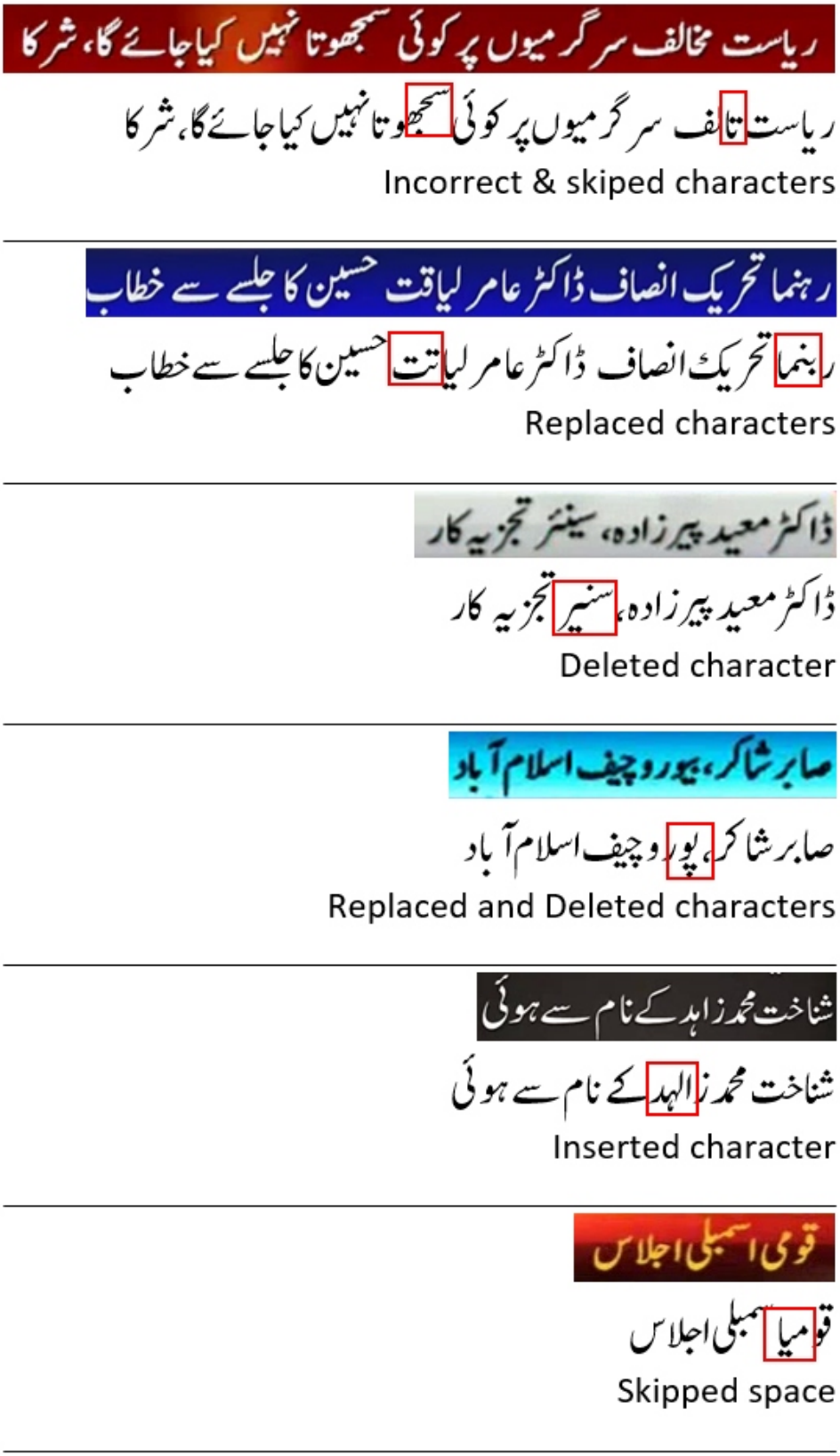


Figure 5.14: Examples of recognition errors

Chapter 6

Text Detection & Recognition: Application

6.1 Introduction

The previous chapters introduced the techniques developed for detection and recognition of textual content appearing in video frames with focus on cursive (Urdu) caption text. Combining the detection and recognition in a single system, a number of interesting and useful applications can be developed. Example applications include keyword based indexing and retrieval of videos, generation of summaries of News tickers for a given time period, analysis of the frequency of News related to a particular theme and a comparison of News related to a given theme across various News channels etc. Furthermore, performing detection and recognition of text on live video streams rather than archived videos can be exploited to generate user alerts on specific keywords, 'Breaking News' for instance. In this chapter, we present one such application that was developed by combining the text detector and the V-OCR into a single system to perform indexing of videos on specific keywords. The videos can then be retrieved on provided query keyword. Details of the developed application are presented in the next section.

6.2 Video Indexing & Retrieval

Text appearing in videos can be exploited as a semantic index for content based retrieval. Such retrieval systems allow users input (query) keywords and retrieve all videos where the keyword has appeared. Keywords refer to the words that are provided as query by the user (Figure 6.1). Once the keyword is provided the system queries the database that contains videos indexed on keywords. The videos containing the relevant keyword are then provided to the user in the retrieval phase (Figure 6.1).

We combine the text detector with the recognition engine to index videos on specific keywords. To demonstrate the idea, list of each 100 English and Urdu keywords (listed in Appendix F) is

Figure 6.1: Keyword supplied as query by user

created and maintained in the database (MS SQL). Frames extracted from a video to be indexed are provided to the text detector and the output of the detector is fed to the recognizer. From the view point of application, in addition to Urdu text, we also recognize English text using off-the-shelf Google's Tesseract recognition engine. The recognizer outputs the text in a given frame which is parsed for appearance of any of the keywords. If one or more keywords appear in a frame, the database is updated to record the identity of the video and the time-stamp where the keyword has appeared. The process is repeated for all frames in the video and for all videos to be indexed.

During the retrieval phase, user provides a query keyword (Figure 6.2) along with other meta-data (dates, channels etc.). The system queries the database and returns all videos and all instances within each video where the keyword appears. Retrieval results for a query keyword returned by the system are illustrated in Figure 6.3.

From the view point of quantitative evaluation of the retrieval application, it is important to

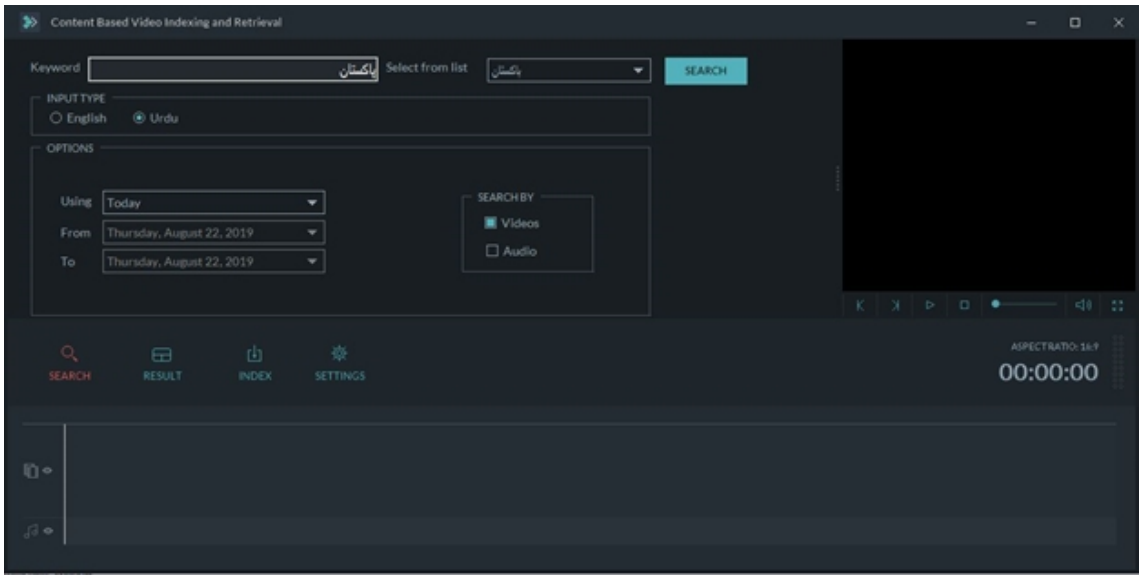


Figure 6.2: Search Screen of the Retrieval Application

Figure 6.3: Retrieval Results for a Query Keyword

mention that in the previous chapters, we reported the results of text detection and text recognition independently. When detection and recognition are combined, the output of the detector is fed to the recognizer hence any missed or incorrectly detected text regions cannot be recognized correctly. To provide an idea of the overall system performance, we report the character recognition rates on text regions extracted by the detector in Table 6.1. The Detection is carried out using Faster R-CNN while recognition is implemented using CNN-LSTM combination and Wolf's binarization. It can be seen from Table 6.1 that an end-to-end character recognition rate of 92.56% is reported as opposed to 97.63% when evaluated on text lines segmented using the ground truth information. Considering the fact that output of detector (which may not always be perfect) is fed to the recognition engine,

the reported character recognition rate is indeed very promising.

Table 6.1: Character Recognition Rate

Recognizer CRR	**End-to-end CRR**
97.63%	92.56%

In addition to the character recognition rate, we also evaluate the system from the perspective of a retrieval engine. The system is provided with query keywords and the precision, recall and F-measure are computed based on the retrieval results. The results are reported in Table 6.2 where it can be seen that an F-measure of 0.89 is reported on more than 4000 instances of Urdu queried keywords. Likewise, an F-measure of 0.84 is realized for 624 English query words.

From the view point of time complexity of retrieval, given a query keyword, the retrieval involves a join over two tables, the table containing indices against the keywords and the table containing video information. The overall retrieval complexity is $O(M+N)$ where M is total number of rows in frame (index) table and N is the total number rows in video table. We have employed MS SQL Server as DBMS which sorts tables according to the columns that are used for joining the two tables. Due to sorting of the tables. A merge operation is carried out resulting in time complexity of $O(M+N)$

Table 6.2: Results

		Query Instances	**Precision**	**Recall**	**F–Measure**
Text	**Urdu**	4061	0.92	0.85	0.89
	English	624	0.90	0.80	0.84

The indexing and retrieval application developed on top of the detection and recognition modules was supported by IGNITE, National Technology Fund and was successfully deployed at the Associated Press of Pakistan. Efforts are being made to commercialize the application targeting local media houses as well as regulatory bodies.

In addition to smart retrieval, the extracted and recognized text can also be employed to develop a number of useful applications. The system, for instance, can be extended to work on live video streams rather than archived content. This in turn would allow development of keyword based user-alert systems where an alert is generated whenever one of the keywords of interest for a given user appears in the video stream. Likewise, the extracted textual content can be employed to develop a summary of News flashed on a given News channel in a given duration of time. Natural language processing techniques can also be incorporated to compare and analyze the reporting of same events by multiple News channels.

6.3 Summary

This chapter introduced potential applications that can be developed exploiting the text detected and recognized from video frames. We presented the details of one such application, textual content based video indexing and retrieval, that was developed in our study. In addition, system level performance of the retrieval application was also discussed. In the next chapter, we provide our concluding remarks and discuss possible future directions both from the research and application perspectives.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Text appearing in videos contains rich semantic information that can be exploited to develop a number of useful applications. The core modules of such applications include the extraction and recognition of textual information from video frames and made the subject of our study. More specifically, our research was aimed at development of techniques for detection and recognition of Urdu caption text appearing in video frames. We investigated the latest deep learning based techniques for detection as well as recognition of caption text and developed techniques reporting high performance using standard evaluation metrics.

For detection of textual content, we adapted deep learning based object detectors by tuning the models to learn to discriminate between text and non-text regions. The investigated techniques included Faster R-CNN, YOLO, SSD and R-FCN while models like Inception and ResNet trained on Miscrosoft COCO dataset were employed as the base networks. Text detection and script identification were then combined into a single hybrid model. For recognition, an implicit segmentation based technique was employed that relies on a combination of CNN and RNN followed by the CTC layer. Text line images extracted from video frames along with the ground truth transcription, are fed to a CNN for feature extraction and the extracted feature sequences are provided as input to a recurrent net for predicting the most likely character sequence. Finally, CTC decoding is applied to convert the raw network predictions into meaningful text. The experimental study of the system was carried out on more than 11,000 video frames and an over all F-measure of 0.91 was reported by the detector using a Faster R-CNN with Inception. Likewise, a character recognition rate of 97.63% is realized by the CNN–LSTM combination. Furthermore, we also integrated the detector and the recognizer to develop an indexing and retrieval application. System level retrieval experiments also reported a high F-measure reading 0.89 demonstrating the effectiveness of the proposed techniques and their potential employment in real world applications. The UTiV dataset of video frames collected and labeled as a part of this study has been made publicly available and the ground truth information allows evaluation of text localization, text recognition and script

identification tasks.

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

- UTiV (Urdu Text in Videos) dataset has been collected, labeled and made publicly available.
- A hybrid text detector and script identifier has been developed primarily targeting the content on our local News channels.
- A number of pre-processing techniques were investigated to segment text from background for effective recognition.
- A joint convolutional-recurrent network based recognition engine has been developed to recognize Urdu caption text.
- Detection and recognition modules are combined in a single system to support development of high level applications.

7.2 Future Work

The presented study proposed text detection and recognition techniques using Urdu caption text as a case study, however, the findings can be generalized to other cursive scripts as well. In our further work on this subject, we intend to extend the detector and recognizer to process wild scene text as well. In addition, the present study primarily focused on horizontally aligned text. Techniques can be developed to detect and recognize text at other orientations as well. From the view point of recognition, the performance can be further enhanced by incorporating a post-processing stage that may include dictionary validation as well as the semantic contextual information to predict the most likely word given a sequence of words and hence improve the recognition performance.

From the view point of application development, we intend to optimize the system to work in real time allowing users to set keyword based alerts in live video streams. Likewise, the transcription of textual data in videos can be processed further to develop News category classification, automatic summarization and content mining systems. Furthermore, in addition to textual content, the visual and audio content can also be exploited to complement the text-based applications. Spoken keywords contain useful information that can complement the textual content. Likewise, the visual information containing key individuals, objects and locations etc. can also serve as a useful index. This can lead to a comprehensive video analytics system that can serve regulatory bodies, media houses and general public. It is expected that the findings of this study would be useful for the pattern classification community in general and researchers targeting detection and recognition of text in particular.

Appendix A

Research Publications

Journal Publications

- Mirza, Ali, and Imran Siddiqi. "Recognition of cursive video text using a deep learning framework." IET Image Processing (2020). DOI: https://doi.org/10.1049/iet-ipr.2019.1070 (Impact Factor: 1.995)
- Mirza, Ali, Ossama Zeshan, Muhammad Atif, and Imran Siddiqi. "Detection and recognition of cursive text from video frames." EURASIP Journal on Image and Video Processing 2020, no. 1 (2020): 1-19. DOI: https://doi.org/10.1186/s13640-020-00523-5 (Impact Factor: 1.474)
- Mirza, Ali, and Imran Siddiqi, Cursive Caption Text Detection in Videos, Submitted to Journal of Electronic Imaging (Under Review) – (Impact Factor: 1.505)

Book Chapters

- Mirza A., Siddiqi I., Hayat U., Atif M., Mustufa S.G. (2020) “Recognition of Cursive Caption Text Using Deep Learning - A Comparative Study on Recognition Units.” In: Pattern Recognition and Artificial Intelligence. ICPRAI 2020. Lecture Notes in Computer Science, vol 12068. Springer, Cham.
- Mirza A., Siddiqi I., Mustufa S.G., Hussain M. (2019) “Impact of Pre-Processing on Recognition of Cursive Video Text.” In: Pattern Recognition and Image Analysis. IbPRIA 2019. Lecture Notes in Computer Science, vol 11867. Springer, Cham.

Conference Publications

- Mirza, Ali, Marium Fayyaz, Zunera Seher, and Imran Siddiqi. "Urdu caption text detection using textural features." In Proceedings of the 2nd Mediterranean Conference on Pattern Recognition and Artificial Intelligence, pp. 70-75. 2018. https://doi.org/10.1145/3177148.3180098 (Best Paper Award).

Appendix B

Edit Distance Example

Algorithm Edit distance

Input: $\alpha = \alpha_1 \dots \alpha_n$ and $\beta = \beta_1 \dots \beta_m$

1: `for` $i \leftarrow$ `0 to` n `do`
2: $\quad D_{i,0} \leftarrow i$;
3: `end for`
4: `for` $j \leftarrow$ `0 to` m `do`
5: $\quad D_{0,j} \leftarrow j$;
6: `end for`
7: `for` $i \leftarrow$ `1 to` n `do`
8: $\quad$ `for` $j \leftarrow$ `1 to` m `do`
9: $\quad\quad t \leftarrow (\alpha_i = \beta_j)? \; 0 : 1$;
10: $\quad\quad D_{i,j} \leftarrow \min\{D_{i-1,j-1} + t, D_{i,j-1} + 1, D_{i-1,j} + 1\}$;
11: $\quad$ `end for`
12: `end for`
13. `return` $D_{n,m}$

Example of Levenshtein's distance between two strings پاکستان and باکسان

	0	1	2	3	4	5	6	7
ب	1	**1**	2	3	4	5	6	7
ا	2	2	**1**	2	3	4	5	6
ک	3	3	2	**1**	2	3	4	5
س	4	4	3	2	**1**	**2**	3	4
ا	5	5	4	3	2	2	**2**	3
ن	6	6	5	4	3	3	3	**2**

Figure B.1

Appendix C

Ground Truth Labeling Tool

This appendix presents screen shots illustrating different features of the ground truth labeling tool.

Figure C.1: A single frame loaded in the labeling Software and key components of the tool

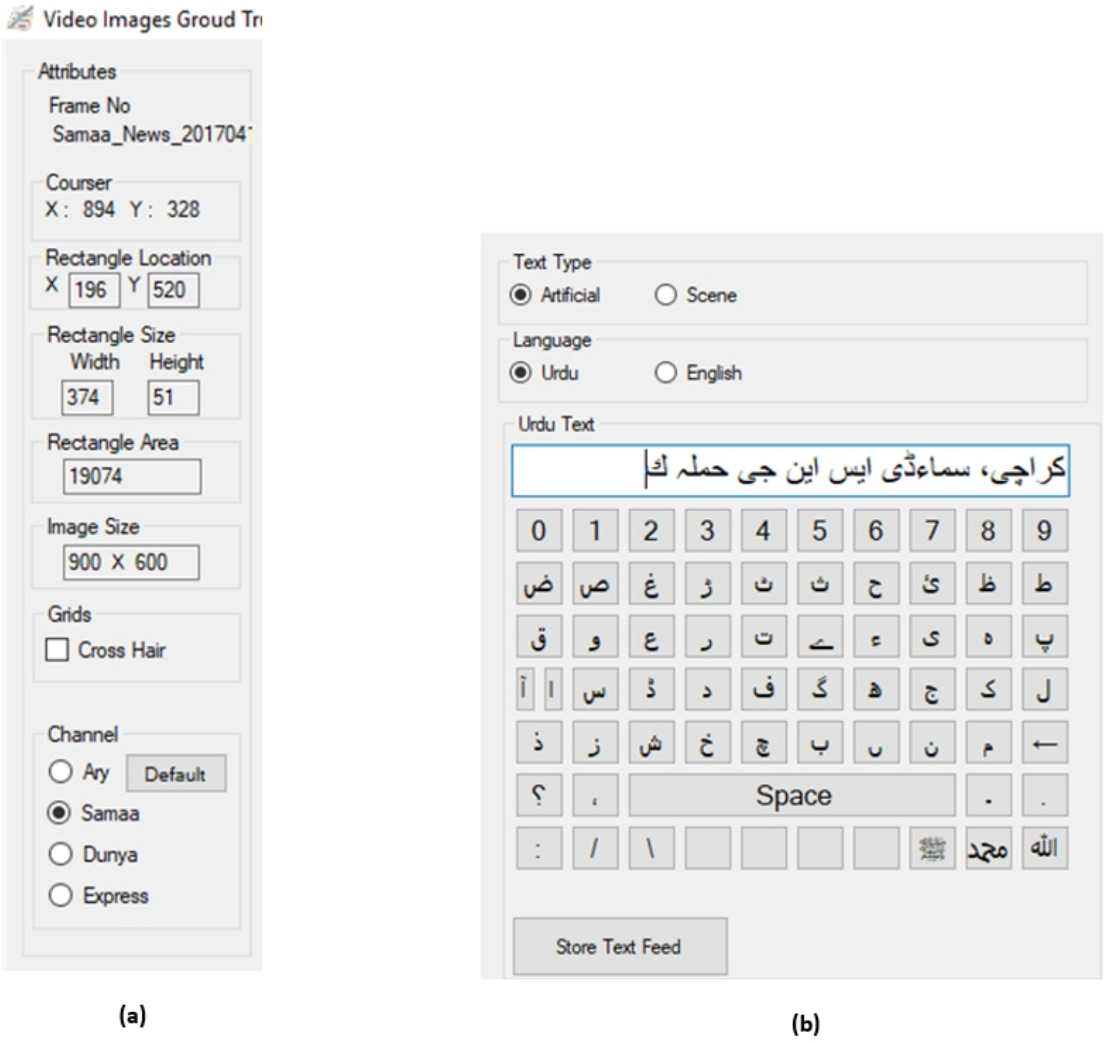


Figure C.2: (a): Ground truth information on bounding box of a text region and the frame (b): Transcription of text with information on text type and text script

Appendix D

Sample Images of Hybrid Text Detector and Script Identifier

Figure D.1: Hybrid text detector and script identifier output: Express News

Figure D.2: Hybrid text detector and script identifier output: Samaa News

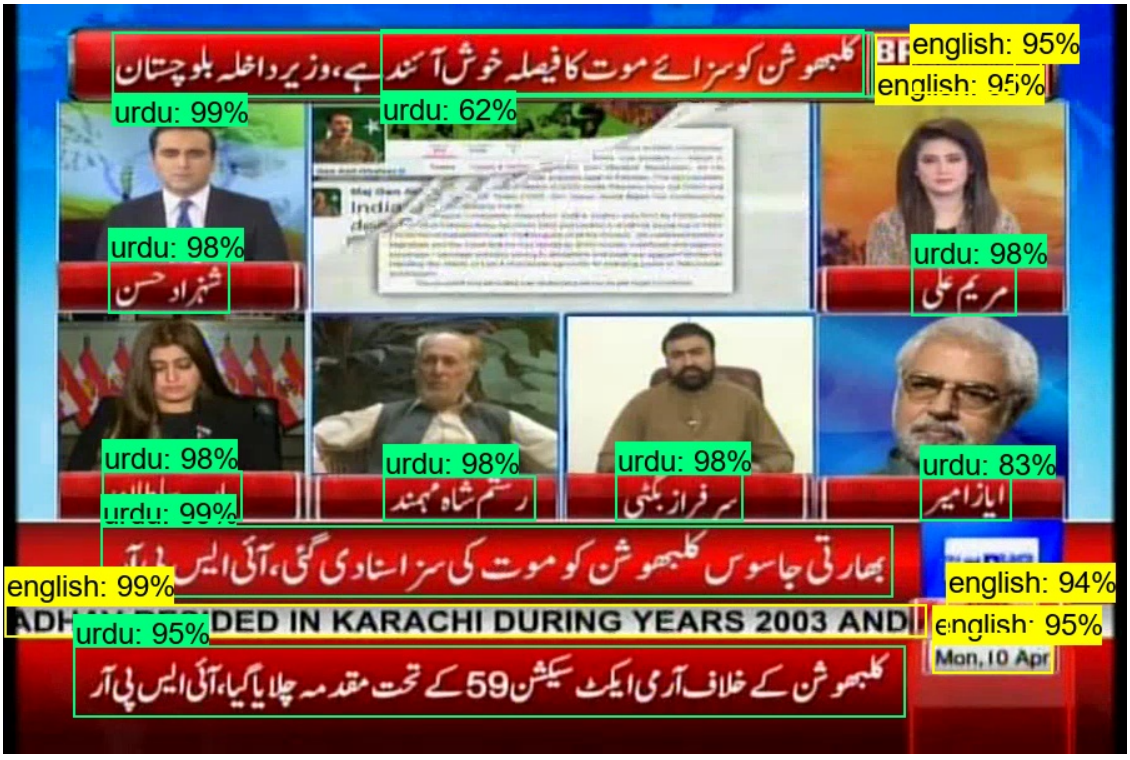


Figure D.3: Hybrid text detector and script identifier output: Dunya News

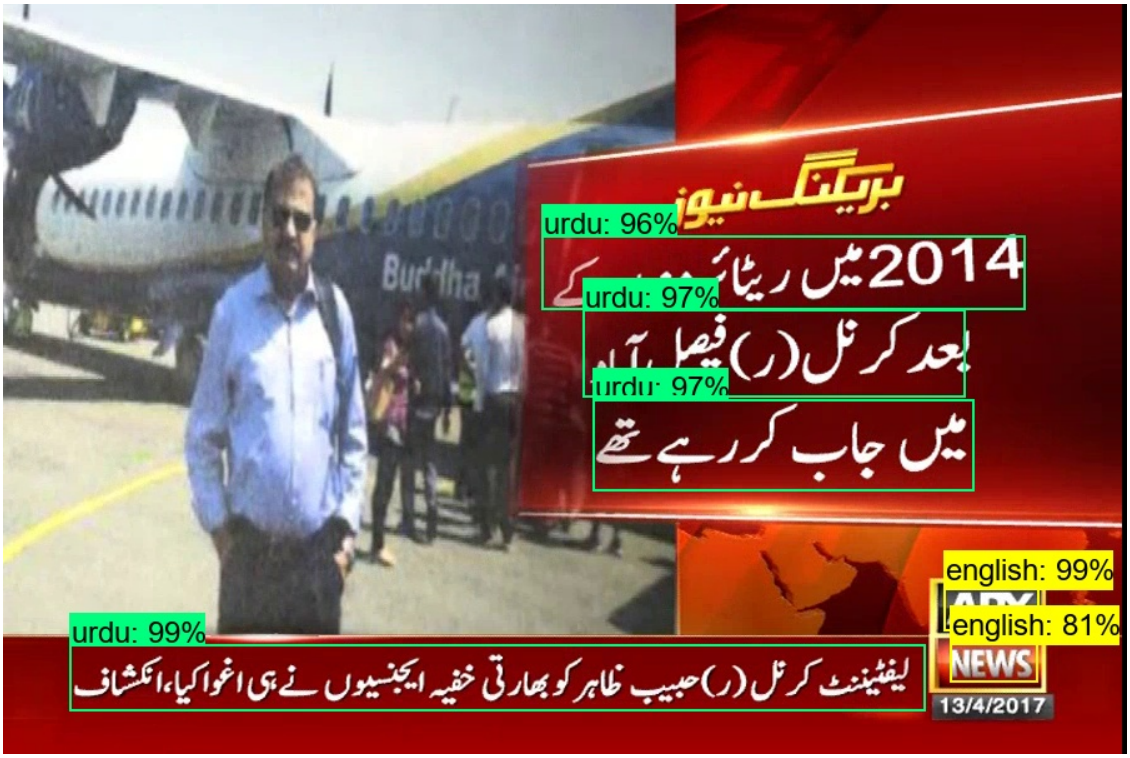


Figure D.4: Hybrid text detector and script identifier output: Ary News

Appendix E

Preliminary Experiments–Recognition using Holistic Technique

Holistic recognition technique employs ligatures as units of recognition. As a first step, ligatures need to be extracted from the binarized text line images. Ligatures are extracted using connected component labeling and the secondary ligatures (dots and diacritics) are associated with their parent primary ligatures by performing morphological dilation (with a vertical structuring element). For the preliminary experiments, a total of 130,000 ligatures are extracted from 8000 text lines. In order to prepare the training and test data, these ligatures are organized into classes (clusters) where each class is a collection of images that correspond to a single ligature. The total number of unique ligature classes in our study sums up to 900 with an average of 70 images per class.

For recognition of ligatures, we investigated a number of deep convolutional neural network architectures. More specifically, we employed a number of pre-trained CNN models using the transfer learning framework. These include the classical AlexNet [296], VGG Nets [298], GoogLeNet [299], InceptionV3 [335] and ResNet101 [311]. In transfer learning, a pre-trained model can be used as a feature extractor (using convolutional layers only) and these features can be fed to a separate classifier similar to the traditional machine learning framework. Another common technique is to replace the last fully connected layer of a pre-trained network with class labels of the dataset under study and continue back propagation (either on all or few last layers of the network) to adjust the network weights. In our study, we investigate both the possibilities employ pre-trained models as feature extractors as well as fine-tuned them to our set of ligatures. A summary of the networks considered in our work is presented in Table E.1.

For experimental study, we employ 6,500 text lines in the training set and 1,500 in the test set. The ligature classes corresponding to the same set of 6,500 and 1,500 lines are employed in the training and test sets respectively. Performance is quantified using ligature recognition rate computed as the fraction of ligatures correctly recognized by the system.

Table E.1: Summary of pre-trained models employed in our study

Model	**Input Size**	**Depth**	**FC Layers**
Alex-net [296]	$227 \times 227 \times 3$	8	3
VGG-16 [298]	$224 \times 224 \times 3$	16	3
VGG-19 [298]	$224 \times 224 \times 3$	19	3
Google-net [299]	$224 \times 224 \times 3$	22	1
Inceptionv3 [335]	$299 \times 299 \times 3$	48	1
Resnet-101 [311]	$224 \times 224 \times 3$	101	1

The recognition rates realized in our experiments are summarized in Table E.2. The results are presented for both fine-tuning and feature extraction using multiple pre-trained models. Comparing the performance of various pre-trained models, it can be seen that fine-tuning outperforms feature extraction for all models. The observation is natural as fine-tuning allows adjusting the weights of the network according to images under study hence the extracted features are likely to be more effective. The highest recognition rate is reported by AlexNet reading 83.50%. It is interesting to note that AlexNet has the least depth among the investigated pre-trained models. This observation is consistent with previous findings on the recognition of ligatures in video [48] as well as printed (scanned) documents [278]. This observation can be attributed to the fact that all these models are trained on the ImageNet [336] dataset which contains colored images of objects. We, on the other hand, deal with binary images of ligatures representing a different scenario. Consequently, networks with relatively fewer convolutional layers are able to learn the discriminative features reporting acceptable recognition rates.

The initial study using ligatures as recognition units led us to the following findings.

- Segmentation of text into ligatures in caption text is highly error prone due to low resolution of text as opposed to scanned document images.
- Preparing training data for such a technique is a highly tedious task as ligature clusters are required to be created.
- The total number of unique classes in such a technique would be very high even if dots and diacritics are removed.
- Re-association of secondary ligatures with primary ligatures can introduce post-recognition errors.
- The recognition rates in preliminary experiments are relatively very low (when compared to printed text) even with a small set of ligature classes.

These findings suggested us to investigated analytical recognition techniques which do not require an explicit segmentation, the training data needs to be labeled only with transcription of text and the number of unique classes remains manageable.

Table E.2: Recognition Rates of Analytical and Holistic Techniques

Model	**Recognition Rate**	
	Feature Extraction	**Fine-Tuning**
AlexNet	78.27	83.50
VGG16	76.60	82.95
VGG19	76.79	82.96
GoogleNet	79.28	82.60
InceptionV3	76.91	81.47
ResNet	67.94	75.15

Appendix F

List of Keywords used in Indexing Application

پاکستان	لاہور	چیف آف جوائنٹ اسٹاف	ہاکی ٹیم	چوہدری شجاعت
پنجاب	کراچی	عدلیہ	موسم کی خبریں	چوہدری پرویز الٰہی
سندھ	پشاور	سپریم کورٹ	سپورٹس نیوز	ڈونلڈ ٹرمپ
بلوچستان	کوئٹہ	جج	پاکستان تحریک انصاف	نریندر مودی
خیبر پختونخواہ	گوادر	عدالت	جماعت اسلامی	اسپیکر قومی اسمبلی
گلگت	گورنمنٹ	قومی اسمبلی	پاکستان مسلم لیگ نون	بین اقوامی
بلتستان	چیف جسٹس	وزیر اعظم ہاؤس	پاکستان مسلم لیگ ق	قومی
آزاد کشمیر	صدر	اعوانِ صدر	پاکستان پیپلز پارٹی	رد الفساد
جموں کشمیر	وزیر اعظم	الیکشن کمیشن	متحدہ قومی موومنٹ	بنی گالا
صبح	گورنر	دفتر خارجہ	قمر جاوید باجوہ	رائے ونڈ
چائنا	وزیر اعلی	آئی ایس پی آر	ممنون حسین	میٹرو بس
انڈیا	وزیر داخلہ	نیب	عمران خان	اورنج ٹرین
بھارت	وزیر خارجہ	ایف آئی اے	شیخ رشید	پاکستان انٹرنیشنل ایئر لائن
امریکہ	وزیر دفاع	ایس سی ای پی	سراج الحق	پاکستان ریلوے
سعودی عرب	وزیر قانون	جے آئی ٹی	نواز شریف	نظر ثانی
افغانستان	وزیر خزانہ	بریکنگ نیوز	شاہد خاقان عباسی	دہشت گردی
ایران	آرمی چیف	بری خبر	آصف علی زرداری	خود کش حملہ آور
روس	نیول چیف	نیوز الرٹ	شہباز شریف	دہشت گرد
اسلام آباد	ایئر چیف	ہیڈ لائنز	مریم نواز	وار آن ٹیرر
راولپنڈی	آئی ایس آئی چیف	کرکٹ ٹیم	طاہر القادری	ضرب عزب

Figure F.1: List of 100 Urdu keywords for indexing and retrieval application

Pakistan	Queta	Judge	Jamat e Islami	Narinder Modi
Punjab	Gawadar	Maryam Nawaz	Pakistan Muslim League	Siraj ul Haq
Sindh	Government	Donald Trump	Minister of Foreign Affairs	International
Blochistan	Chief Justice	President House	Pakistan Muslim League Qaaf	National
Imran Khan	President	Shabaz Sharif	Pakistan People's Party	CPEC
Gilgit Bultistan	Prime Minister	Noon	Mutahada Qomi Moment	Bani Gala
Azad Kashmir	Governor	ISPR	Qamar Javeed Bajwa	Raiwind
Jamu Kashmir	Chief Minister	NAB	Mamnoon Hussain	Metro Bus
China	Interior Minister	FIA	Khayber Pakhtoon Khwa	Orange Train
India	Foreign Minister	SCEP	Pakistan Tehreek e Insaf	Nawaz Sharif
America	Defense Minister	JIT	Speaker Parliament House	Pakistan Railway
Saudi Arabia	Law Minister	Breaking News	Pakistan International Airline	Reference
Afghanistan	Finance Minister	Big News	Shahid Khaqan Abbasi	Petition
Iran	Army Chief	News Alert	Asif Ali Zardari	Terror Attach
Russia	Naval Chief	Headlines	Election Commission	Suicide Bombing
Islamabad	Air Chief	Cricket Team	Parliament House	Terrorist
Rawalpindi	ISI Chief	Hockey Team	Chief of Joint Staff	War on Terror
Lahore	Tahir ul Qadri	Weather News	Choudary Shujaat	Zarb e Azab
Karachi	Judiciary	Sports News	Choudary Peviz Elahi	Rad al Fasad
Peshwar	Supreme Court	Sheikh Rasheed	Prime Minister House	People Killed

Figure F.2: List of 100 English keywords for indexing and retrieval application

Appendix G

Recognizer Performance with Different CNN-LSTM Designs

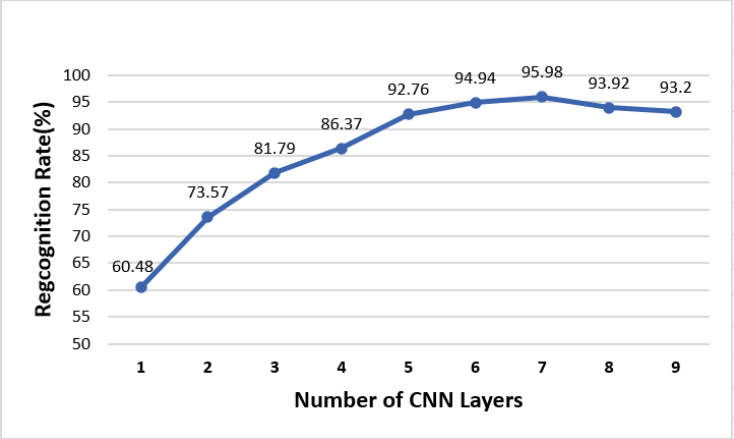


Figure G.1: Recognition rates as a function of number of convolutional layers

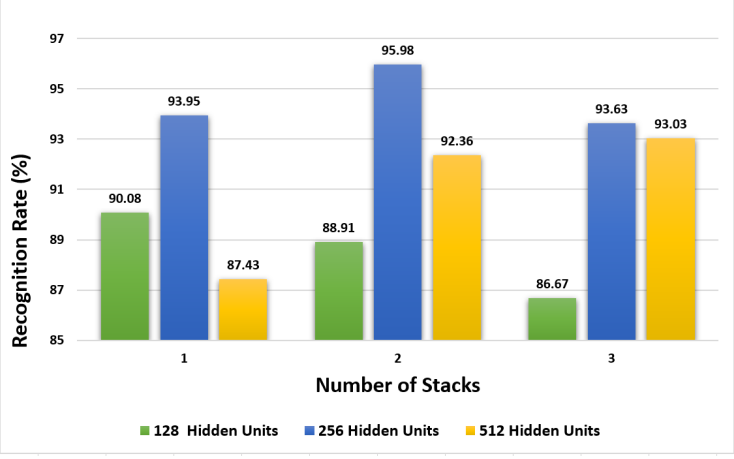


Figure G.2: Recognition rates as a function of number of LSTM stacks and hidden units

Appendix H

Awards & Achievements

- Winner of Best (PhD) Poster Award–3rd IAPR International Summer School on Document Analysis (SSDA)–Islamabad, Pakistan, 2019.
- Won the IAPR Full Funding to attend the 2nd IAPR International Summer School on Document Analysis (SSDA)–La Rochelle, France, 2018.
- Received the Best Paper Award for the paper titled "Urdu Caption Detection using Textural Features" at the 2nd IAPR International Mediterranean Conference on Pattern Recognition and Artificial Intelligence, MedPRAI, Morocco, 2018.
- Received Full Funding (from the Higher Education Commission, Pakistan) to attend the 2nd IAPR International Mediterranean Conference on Pattern Recognition and Artificial Intelligence, MedPRAI, Morocco, 2018.

Bibliography

[1] Jean Burgess and Joshua Green. *YouTube: Online video and participatory culture*. John Wiley & Sons, 2018. Cited on p. 1.

[2] Unknown. Cisco visual networking index: Forecast and trends, 2017–2022. In *White Paper*, 2019. Cited on p. 1.

[3] Hossam Elshahaby and Mohsen Rashwan. A system for detection of moving caption text in videos: a news use case. *Multimedia Tools and Applications*, pages 1–25, 2021. Cited on pp. 1 and 35.

[4] Xiaoqing Liu and Jagath Samarabandu. An edge-based text region extraction algorithm for indoor mobile robot navigation. In *IEEE International Conference Mechatronics and Automation, 2005*, volume 2, pages 701–706. IEEE, 2005. Cited on p. 1.

[5] Kazem Qazanfari and Saeed Shiri. Real time text localization for indoor mobile robot navigation. *arXiv preprint arXiv:1709.09634*, 2017. Cited on p. 1.

[6] Nobuo Ezaki, Marius Bulacu, and Lambert Schomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 683–686. IEEE, 2004. Cited on p. 1.

[7] Shehzad Muhammad Hanif and Lionel Prevost. Texture based text detection in natural scene images-a help to blind and visually impaired persons. In *CVHI*, 2007. Cited on p. 1.

[8] Dhruv Dahiya, Ashish Issac, Malay Kishore Dutta, Kamil Říha, and Petr Kříž. Computer vision technique for scene captioning to provide assistance to visually impaired. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–4. IEEE, 2018. Cited on p. 1.

[9] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819, 2011. Cited on p. 2.

[10] Maria Tzelepi and Anastasios Tefas. Deep convolutional learning for content based image retrieval. *Neurocomputing*, 275:2467–2478, 2018. Cited on p. 2.

[11] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Deep video hashing. *IEEE Transactions on Multimedia*, 19(6):1209–1219, 2016. Cited on p. 2.

[12] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2): 237–254, 2017. Cited on p. 2.

[13] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu. Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Transactions on Image Processing*, 25 (6):2752–2773, 2016. Cited on pp. 2 and 13.

[14] Zhen Dong, Su Jia, Tianfu Wu, and Mingtao Pei. Face video retrieval via deep learning of binary hash representations. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. Cited on p. 2.

[15] Themos Stafylakis and Georgios Tzimiropoulos. Zero-shot keyword spotting for visual speech recognition in-the-wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–529, 2018. Cited on p. 2.

[16] Da Zhang, Hamid Maei, Xin Wang, and Yuan-Fang Wang. Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*, 2017. Cited on p. 2.

[17] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016. Cited on p. 2.

[18] Aasif Ansari and Muzammil H Mohammed. Content based video retrieval systems-methods, techniques, trends and challenges. *International Journal of Computer Applications*, 112(7), 2015. Cited on pp. 3 and 4.

[19] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, et al. The qbic system. *IEEE computer*, 28(9):23–32, 1995. Cited on p. 4.

[20] Alex Pentland, Rosalind W Picard, and Stan Sclaroff. Photobook: Content-based manipulation of image databases. *International journal of computer vision*, 18(3):233–254, 1996. Cited on p. 4.

[21] Elena Stringa, Paul Meylemans, and João GM Gonçalves. Image retrieval by example: Techniques and demonstrations. In *23rd ESARDA Symposium on Safeguards and Nuclear Material Management*, volume 2, 2001. Cited on p. 4.

[22] John R Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *ACM multimedia*, volume 96, pages 87–98. Citeseer, 1996. Cited on p. 4.

[23] Michael Christel, Scott Stevens, and Howard Wactlar. Informedia digital video library. In *Proceedings of the second ACM international conference on Multimedia*, pages 480–481. ACM, 1994. Cited on p. 4.

[24] Alan F Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330. ACM, 2006. Cited on pp. 4 and 47.

[25] R Anuranji and H Srimathi. A supervised deep convolutional based bidirectional long short term memory video hashing for large scale video retrieval applications. *Digital Signal Processing*, 102:102729, 2020. Cited on p. 4.

[26] T Prathiba and R Shantha Selva Kumari. Eagle eye cbvr based on unique key frame extraction and deep belief neural network. *Wireless Personal Communications*, 116(1):411–441, 2021. Cited on p. 4.

[27] Haojin Yang and Christoph Meinel. Content based lecture video retrieval using speech and video text information. *IEEE Transactions on Learning Technologies*, 7(2):142–154, 2014. Cited on p. 4.

[28] Tatiana Tommasi, Robin Aly, K McGuinness, K Chatfield, R Arandjelovic, O Parkhi, R Ordelman, A Zisserman, and Tinne Tuytelaars. Beyond metadata: searching your archive based on its audio-visual content. *Unknown*, 2014. Cited on p. 4.

[29] Rachid Benmokhtar and Benoit Huet. An ontology-based evidential framework for video indexing using high-level multimodal fusion. *Multimedia Tools and Applications*, 73(2): 663–689, 2014. Cited on p. 4.

[30] Anil K Jain and Bin Yu. Automatic text location in images and video frames. *Pattern recognition*, 31(12):2055–2076, 1998. Cited on p. 4.

[31] Oussama Zayene, Sameh Masmoudi Touj, Jean Hennebert, Rolf Ingold, and Najoua Essoukri Ben Amara. Multi-dimensional long short-term memory networks for artificial arabic text recognition in news video. *IET Computer Vision*, 2018. Cited on pp. 4, 44, 46, 47, 48, and 104.

[32] Mohieddin Moradi and Saeed Mozaffari. Hybrid approach for farsi/arabic text detection and localisation in video frames. *IET Image Processing*, 7(2):154–164, 2013. Cited on pp. 4, 30, 36, 83, and 84.

[33] Oussama Zayene, Jean Hennebert, Mathias Seuret, Sameh M. Touj, Rolf Ingold, and Najoua Essoukri Ben Amara. Text detection in arabic news video based on swt operator and convolutional auto-encoders. *2016 12th IAPR Workshop on Document Analysis Systems*, 2016. Cited on pp. viii, 4, 34, 35, 36, 77, 83, and 84.

[34] Sonia Yousfi, Sid-Ahmed Berrani, and Christophe Garcia. Contribution of recurrent connectionist language models in improving lstm-based arabic text recognition in videos. *Pattern Recognition*, 64:245–254, 2017. Cited on pp. 4, 44, 46, and 47.

[35] Saad Bin Ahmed, Saeeda Naz, Muhammad Imran Razzak, and Rubiyah Yusof. Arabic cursive text recognition from natural scene images. *Applied Sciences*, 9(2):236, 2019. Cited on pp. 4 and 42.

[36] Saad Bin Ahmed, Saeeda Naz, Muhammad Imran Razzak, and Rubiyah Yousaf. Deep learning based isolated arabic scene character recognition. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 46–51. IEEE, 2017. Cited on p. 4.

[37] Saeeda Naz, Arif Iqbal Umar, Riaz Ahmed, Muhammad Imran Razzak, Sheikh Faisal Rashid, and Faisal Shafait. Urdu nasta'liq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks. *SpringerPlus*, 5(1):2010, 2016. Cited on pp. 6, 46, and 87.

[38] Israr Uddin, Nizwa Javed, Imran A Siddiqi, Shehzad Khalid, and Khurram Khurshid. Recognition of printed urdu ligatures using convolutional neural networks. *Journal of Electronic Imaging*, 28(3):033004, 2019. Cited on pp. 6, 38, 39, 46, and 47.

[39] Atique Ur Rehman and Sibt Ul Hussain. Large scale font independent urdu text recognition system. *arXiv preprint arXiv:2005.06752*, 2020. Cited on pp. 6, 40, 46, and 47.

[40] Malik Waqas Sagheer, Nicola Nobile, Chun Lei He, and Ching Y Suen. A novel handwritten urdu word spotting based on connected components analysis. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2013–2016. IEEE, 2010. Cited on pp. 6, 38, and 46.

[41] Shahbaz Hassan, Ayesha Irfan, Ali Mirza, and Imran Siddiqi. Cursive handwritten text recognition using bi-directional lstms: A case study on urdu handwriting. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pages 67–72. IEEE, 2019. Cited on pp. 6, 38, 40, 46, and 47.

[42] Mujtaba Husnain, Malik Muhammad Saad Missen, Shahzad Mumtaz, Muhammad Zeeshan Jhanidr, Mickaël Coustaty, Muhammad Muzzamil Luqman, Jean-Marc Ogier, and Gyu Sang Choi. Recognition of urdu handwritten characters using convolutional neural network. *Applied Sciences*, 9(13):2758, 2019. Cited on pp. 6, 38, 46, and 47.

[43] Hazrat Ali, Ahsan Ullah, Talha Iqbal, and Shahid Khattak. Pioneer dataset and automatic recognition of urdu handwritten characters using a deep autoencoder and convolutional neural network. *SN Applied Sciences*, 2(2):152, 2020. Cited on pp. 6, 38, 46, and 47.

[44] Akhtar Jamil, Imran Siddiqi, Fahim Arif, and Ahsen Raza. Edge-based features for localization of artificial urdu text in video images. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1120–1124. IEEE, 2011. Cited on pp. viii, 6, 16, 17, 27, 49, 51, 52, 62, 63, and 84.

[45] Akhtar Jamil, Ali Abidi, Imran Siddiqi, and Fahim Arif. A hybrid approach for artificial urdu text detection in video images. In *Proc. ICPR*, pages 1944–1947, 2012. Cited on pp. 6, 29, 36, and 49.

[46] Ahsen Raza, Imran Siddiqi, Chawki Djeddi, and Abdellatif Ennaji. Multilingual artificial text detection using a cascade of transforms. In *2013 12th International Conference on Document Analysis and Recognition*, pages 309–313. IEEE, 2013. Cited on pp. 6, 49, 63, and 84.

[47] Usman Shahzad and Khurram Khurshid. Oriental-script text detection and extraction in videos. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 15–20. IEEE, 2017. Cited on pp. 6, 27, 49, and 84.

[48] Umar Hayat, Muhammad Aatif, Osama Zeeshan, and Imran Siddiqi. Ligature recognition in urdu caption text using deep convolutional neural networks. In *2018 14th International Conference on Emerging Technologies (ICET)*, pages 1–6. IEEE, 2018. Cited on pp. 6, 45, 46, 47, 49, 104, and 123.

[49] Burhan Ul Tayyab, Muhammad Ferjad Naeem, Adnan Ul-Hasan, Faisal Shafait, et al. A multi-faceted ocr framework for artificial urdu news ticker text recognition. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 211–216. IEEE, 2018. Cited on pp. 6, 46, 47, 49, and 104.

[50] Gurpreet Singh Lehal. Choice of recognizable units for urdu ocr. In *Proceeding of the workshop on Document Analysis and Recognition*, pages 79–85. ACM, 2012. Cited on p. 6.

[51] G Ciardiello, G Scafuro, MT Degrandi, MR Spada, and MP Roccotelli. An experimental system for office document handling and text recognition. In *Proc 9th Int. Conf. on Pattern Recognition*, pages 739–743, 1988. Cited on pp. 7 and 12.

[52] David G Elliman and Ian T Lancaster. A review of segmentation and contextual analysis techniques for text recognition. *Pattern Recognition*, 23(3-4):337–346, 1990. Cited on pp. 7 and 12.

[53] Christian Wolf and J-M Jolion. Extraction and recognition of artificial text in multimedia documents. *Formal Pattern Analysis & Applications*, 6(4):309–326, 2004. Cited on pp. 7, 62, 63, 90, 91, and 102.

[54] Srinandan Komanduri, Y Mohana Roopa, and M Madhu Bala. Novel approach for image text recognition and translation. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 596–599. IEEE, 2019. Cited on pp. 7 and 12.

[55] Ye Qiaoyang and David Doermann. Text detection and recognition in imagery: A survey. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 37, July 2015. Cited on pp. 7, 12, 13, and 36.

[56] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. Single shot textspotter with explicit alignment and attention. *arXiv preprint arXiv:1803.03474*, 2018. Cited on pp. 7, 33, and 36.

[57] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *arXiv preprint arXiv:1801.02765*, 2018. Cited on pp. 7, 33, and 36.

[58] Yanna Wang, Cunzhao Shi, Baihua Xiao, Chunheng Wang, and Chengzuo Qi. Crf based text detection for natural scene images using convolutional neural network and context information. *Neurocomputing*, 295:46–58, 2018. Cited on pp. 7, 32, 36, and 46.

[59] Khaoula Elagouni, Christophe Garcia, Franck Mamalet, and Pascale Sébillot. Text recognition in videos using a recurrent connectionist approach. In *International Conference on Artificial Neural Networks*, pages 172–179. Springer, 2012. Cited on pp. 7 and 47.

[60] Kartik Dutta, Minesh Mathew, Praveen Krishnan, and CV Jawahar. Localizing and recognizing text in lecture videos. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 235–240. IEEE, 2018. Cited on pp. 7, 44, and 47.

[61] Ayan Kumar Bhunia, Gautam Kumar, Partha Pratim Roy, R Balasubramanian, and Umapada Pal. Text recognition in scene image and video frame using color channel selection. *Multimedia Tools and Applications*, 77(7):8551–8578, 2018. Cited on pp. 7, 43, and 47.

[62] Saeeda Naz, Khizar Hayat, Muhammad Imran Razzak, Muhammad Waqas Anwar, Sajjad A Madani, and Samee U Khan. The optical character recognition of urdu-like cursive scripts. *Pattern Recognition*, 47(3):1229–1248, 2014. Cited on pp. 12, 36, 46, and 48.

[63] Nazly Sabbour and Faisal Shafait. A segmentation-free approach to arabic and urdu ocr. In *IS&T/SPIE Electronic Imaging*, pages 86580N–86580N. International Society for Optics and Photonics, 2013. Cited on pp. 12, 40, 47, and 104.

[64] Shupeng Wang, Chenglin Fu, and Qi Li. Text detection in natural scene image: A survey. In *International Conference on Machine Learning and Intelligent Communications*, pages 257–264. Springer, 2016. Cited on p. 13.

[65] Honggang Zhang, Kaili Zhao, Yi-Zhe Song, and Jun Guo. Text extraction from natural scene image: A survey. *Neurocomputing*, 122:310–323, 2013. Cited on p. 13.

[66] Nabin Sharma, Umapada Pal, and Michael Blumenstein. Recent advances in video based document processing: a review. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 63–68. IEEE, 2012. Cited on p. 13.

[67] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *arXiv preprint arXiv:1811.04256*, 2018. Cited on p. 13.

[68] Han Lin, Peng Yang, and Fanlong Zhang. Review of scene text detection and recognition. *Archives of Computational Methods in Engineering*, pages 1–22, 2019. Cited on p. 13.

[69] VN Manjunath Aradhya, HT Basavaraju, and Devanur S Guru. Decade research on text detection in images/videos: a review. *Evolutionary Intelligence*, pages 1–27, 2019. Cited on p. 13.

[70] Palaiahnakote Shivakumara, Rushi Padhuman Sreedhar, Trung Quy Phan, Shijian Lu, and Chew Lim Tan. Multioriented video scene text detection through bayesian classification and boundary growing. *IEEE Transactions on Circuits and systems for Video Technology*, 22(8): 1227–1235, 2012. Cited on pp. 13 and 28.

[71] Wang Zhen and Wei Zhiqiang. A comparative study of feature selection for svm in video text detection. In *2009 Second International Symposium on Computational Intelligence and Design*, volume 2, pages 552–556. IEEE, 2009. Cited on pp. viii, 13, 28, 30, 31, 36, and 38.

[72] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, 36 (5):970–983, 2013. Cited on pp. 13 and 28.

[73] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. Cited on pp. 13 and 28.

[74] Victor Wu, Raghavan Manmatha, and Edward M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Transactions on pattern analysis and machine intelligence*, 21(11):1224–1229, 1999. Cited on p. 15.

[75] Min Cai, Jiqiang Song, and Michael R Lyu. A new approach for video text detection. In *Proceedings. International conference on image processing*, volume 1, pages I–I. IEEE, 2002. Cited on pp. 15 and 27.

[76] Qixiang Ye, Wen Gao, Weiqiang Wang, and Wei Zeng. A robust text detection algorithm in images and video frames. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 2, pages 802–806. IEEE, 2003. Cited on pp. 16 and 27.

[77] Palaiahnakote Shivakumara, Weihua Huang, and Chew Lim Tan. An efficient edge based technique for text detection in video frames. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 307–314. IEEE, 2008. Cited on pp. 16, 27, and 62.

[78] Palaiahnakote Shivakumara, Weihua Huang, Trung Quy Phan, and Chew Lim Tan. Accurate video text detection through classification of low and high contrast images. *Pattern Recognition*, 43(6):2165–2185, 2010. Cited on pp. 16 and 27.

[79] Palaiahnakote Shivakumara, Anjan Dutta, Umapada Pal, and Chew Lim Tan. A new method for handwritten scene text detection in video. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 387–392. IEEE, 2010. Cited on pp. 16, 27, and 28.

[80] Xian-Sheng Hua, Liu Wenyin, and Hong-Jiang Zhang. An automatic performance evaluation protocol for video text detection algorithms. *IEEE transactions on circuits and systems for video technology*, 14(4):498–507, 2004. Cited on pp. 16, 22, and 27.

[81] DS Guru, S Manjunath, P Shivakumara, and Chew Lim Tan. An eigen value based approach for text detection in video. In *Proceedings of the 9th IAPR international workshop on document analysis systems*, pages 501–506, 2010. Cited on pp. 16 and 27.

[82] Rong Huang, Palaiahnakote Shivakumara, and Seiichi Uchida. Scene character detection by an edge-ray filter. In *2013 12th International Conference on Document Analysis and Recognition*, pages 462–466. IEEE, 2013. Cited on pp. viii, 16, 17, and 27.

[83] Sudipto Banerjee, Koustav Mullick, and Ujjwal Bhattacharya. A robust approach to extraction of texts from camera captured images. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 30–46. Springer, 2013. Cited on pp. 17 and 27.

[84] Chong Yu, Yonghong Song, Quan Meng, Yuanlin Zhang, and Yang Liu. Text detection and recognition in natural scene with edge analysis. *IET Computer Vision*, 9(4):603–613, 2015. Cited on pp. viii, 18, and 27.

[85] Chong Yu, Yonghong Song, and Yuanlin Zhang. Scene text localization using edge analysis and feature pool. *Neurocomputing*, 175:652–661, 2016. Cited on pp. 18 and 27.

[86] Kai Wang and Serge Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer, 2010. Cited on pp. 18 and 27.

[87] Xiaodong Huang. Automatic video scene text detection based on saliency edge map. *Multimedia Tools and Applications*, 78(24):34819–34838, 2019. Cited on pp. 18 and 27.

[88] Wayne Niblack et al. *An introduction to digital image processing*, volume 34. Prentice-Hall Englewood Cliffs, 1986. Cited on pp. 19, 62, 90, and 102.

[89] Kongqiao Wang and Jari A Kangas. Character location in scene images from digital camera. *Pattern recognition*, 36(10):2287–2299, 2003. Cited on pp. viii, 19, and 27.

[90] Xiabi Liu, Hui Fu, and Yunde Jia. Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images. *Pattern Recognition*, 41(2):484–493, 2008. Cited on pp. 19 and 27.

[91] Hyung Il Koo and Duck Hoon Kim. Scene text detection via connected component clustering and nontext filtering. *IEEE transactions on image processing*, 22(6):2296–2305, 2013. Cited on pp. 19 and 27.

[92] Zongyi Liu and Sudeep Sarkar. Robust outdoor text detection using text intensity and shape features. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. Cited on pp. viii, 20, and 27.

[93] Trung Quy Phan, Palaiahnakote Shivakumara, and Chew Lim Tan. A laplacian method for video text detection. In *2009 10th International Conference on Document Analysis and Recognition*, pages 66–70. IEEE, 2009. Cited on pp. 20 and 27.

[94] Xiaobing Wang, Yonghong Song, and Yuanlin Zhang. Natural scene text detection with multi-channel connected component segmentation. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1375–1379. IEEE, 2013. Cited on pp. viii, 21, and 27.

[95] Emmanuel J Candes and David L Donoho. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report, DTIC Document, 2000. Cited on pp. 21 and 66.

[96] Marios Anthimopoulos, Basilis Gatos, and Ioannis Pratikakis. A two-stage scheme for text detection in video images. *Image and Vision Computing*, 28(9):1413–1426, 2010. Cited on pp. 21, 28, 30, and 36.

[97] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. Cited on pp. 21 and 28.

[98] Yu Zhong, Hongjiang Zhang, and Anil K Jain. Automatic caption localization in compressed video. *IEEE transactions on pattern analysis and machine intelligence*, 22(4):385–392, 2000. Cited on pp. 21, 27, and 28.

[99] Huiping Li, David Doermann, and Omid Kia. Automatic text detection and tracking in digital video. *IEEE transactions on image processing*, 9(1):147–156, 2000. Cited on pp. 22 and 27.

[100] Kwang In Kim, Keechul Jung, and Jin Hyung Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1631–1639, 2003. Cited on pp. viii, 22, and 27.

[101] Julinda Gllavata, Ralph Ewerth, and Bernd Freisleben. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 425–428. IEEE, 2004. Cited on pp. 22 and 27.

[102] Xian-Sheng Hua, Liu Wenyin, and Hong-Jiang Zhang. Automatic performance evaluation for video text detection. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 545–550. IEEE, 2001. Cited on pp. 22 and 27.

[103] Qixiang Ye, Qingming Huang, Wen Gao, and Debin Zhao. Fast and robust text detection in images and video frames. *Image and Vision Computing*, 23(6):565–576, 2005. Cited on pp. 22, 27, and 28.

[104] Wonjun Kim and Changick Kim. A new approach for overlay text detection and extraction from complex video scene. *IEEE transactions on image processing*, 18(2):401–411, 2008. Cited on pp. 22 and 27.

[105] Palaiahnakote Shivakumara, Trung Quy Phan, and Chew Lim Tan. New fourier-statistical features in rgb space for video text detection. *IEEE transactions on circuits and systems for video technology*, 20(11):1520–1532, 2010. Cited on pp. 23 and 27.

[106] M Swamy Das, B Hima Bindhu, and A Govardhan. Evaluation of text detection and localization methods in natural images. *International Journal of Emerging Technology and Advanced Engineering*, 2(6):277–282, 2012. Cited on pp. 23 and 27.

[107] VN Manjunath Aradhya, MS Pavithra, and C Naveena. A robust multilingual text detection approach based on transforms and wavelet entropy. *Procedia Technology*, 4:232–237, 2012. Cited on pp. 23 and 27.

[108] Marcin Grzegorzek, Chen Li, Johann Raskatow, Dietrich Paulus, and Natalia Vassilieva. Texture-based text detection in digital images with wavelet features and support vector machines. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 857–866. Springer, 2013. Cited on pp. 23 and 27.

[109] Jian Yi, Yuxin Peng, and Jianguo Xiao. Color-based clustering for text detection and extraction in image. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 847–850, 2007. Cited on p. 23.

[110] C Garcia and X Apostolidis. Text detection and segmentation in complex color images. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 4, pages 2326–2329. IEEE, 2000. Cited on pp. 23 and 27.

[111] Xilin Chen, Jie Yang, Jing Zhang, and Alex Waibel. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on image processing*, 13(1):87–99, 2004. Cited on p. 23.

[112] Céline Mancas-Thillou and Bernard Gosselin. Spatial and color spaces combination for natural scene text extraction. In *2006 International Conference on Image Processing*, pages 985–988. IEEE, 2006. Cited on pp. 23 and 27.

[113] Céline Mancas-Thillou and Bernard Gosselin. Color text extraction with selective metric-based clustering. *Computer Vision and Image Understanding*, 107(1-2):97–107, 2007. Cited on pp. viii, 23, 24, and 27.

[114] Chucai Yi and YingLi Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 20(9):2594–2605, 2011. Cited on pp. viii, 24, 25, and 27.

[115] Chucai Yi and Yingli Tian. Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification. *IEEE Transactions on Image Processing*, 21(9):4256–4268, 2012. Cited on pp. 24 and 27.

[116] Dimosthenis Karatzas and Apostolos Antonacopoulos. Text extraction from web images based on a split-and-merge segmentation method using colour perception. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 634–637. IEEE, 2004. Cited on pp. 24 and 27.

[117] Nikos Nikolaou and Nikos Papamarkos. Color reduction for complex document images. *International Journal of Imaging Systems and Technology*, 19(1):14–26, 2009. Cited on pp. viii, 24, 25, and 27.

[118] Yan Song, Anan Liu, Lin Pang, Shouxun Lin, Yongdong Zhang, and Sheng Tang. A novel image text extraction method based on k-means clustering. In *Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008)*, pages 185–190. IEEE, 2008. Cited on pp. 25 and 27.

[119] SeongHun Lee, Min Su Cho, Kyomin Jung, and Jin Hyung Kim. Scene text extraction with edge constraint and text collinearity. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3983–3986. IEEE, 2010. Cited on pp. viii, 26, and 27.

[120] Yi-Feng Pan, Xinwen Hou, and Cheng-Lin Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing*, 20(3):800–813, 2011. Cited on p. 27.

[121] Hideaki Goto and Makoto Tanaka. Text-tracking wearable camera system for the blind. In *2009 10th International Conference on Document Analysis and Recognition*, pages 141–145. IEEE, 2009. Cited on p. 27.

[122] Tong He, Weilin Huang, Yu Qiao, and Jian Yao. Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing*, 25(6):2529–2541, 2016. Cited on p. 28.

[123] Jun Ye, Lin-Lin Huang, and XiaoLi Hao. Neural network based text detection in videos using local binary patterns. In *2009 Chinese Conference on Pattern Recognition*, pages 1–5. IEEE, 2009. Cited on pp. viii, 28, 29, and 36.

[124] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93 (26):429–441, 1946. Cited on p. 28.

[125] Joutel Guillaume, Eglin Véronique, Bres Stéphane, and Emptoz Hubert. Curvelets based feature extraction of handwritten shapes for ancient manuscripts classification. In *Electronic Imaging 2007*, pages 65000D–65000D. International Society for Optics and Photonics, 2007. Cited on p. 28.

[126] Xiang Bai, Cong Yao, and Wenyu Liu. Strokelets: A learned multi-scale mid-level representation for scene text recognition. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 25, 2016. Cited on pp. 28 and 41.

[127] Jyostna Devi Bodapati, B Suvarna, and Veeranjaneyulu N. Role of deep neural features vs hand crafted features for hand written digit recognition. *International Journal of Recent Technology and Engineering (IJRTE)*, 7, 2010. Cited on pp. 28 and 92.

[128] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017. Cited on pp. 28 and 92.

[129] Hammam Alshazly, Christoph Linse, Erhardt Barth, and Thomas Martinetz. Handcrafted versus cnn features for ear recognition. *Symmetry*, 11(12):1493, 2019. Cited on pp. 28 and 92.

[130] Ki-Young Jeong, Keechul Jung, Eun Yi Kim, and Hang Joon Kim. Neural network-based text location for news video indexing. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 3, pages 319–323. IEEE, 1999. Cited on p. 28.

[131] Sung Han Park, Kwang In Kim, Keechul Jung, and Hyung Jin Kim. Locating car license plates using neural networks. *Electronics Letters*, 35(17):1475–1477, 1999. Cited on pp. 28 and 36.

[132] Huiping Li and David Doermann. A video text detection system based on automated training. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 2, pages 223–226. IEEE, 2000. Cited on p. 29.

[133] Yan Hao, Zhang Yi, Hou Zeng-guang, and Tan Min. Automatic text detection in video frames based on bootstrap artificial neural network and ced. *Journal of WSCG*, 11(1-3): 2298–2304, 2003. Cited on p. 29.

[134] A Thilagavathy, K Aarthi, and A Chilambuchelvan. Text detection and extraction from videos using ann based network. *International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)*, 1(1), 2012. Cited on p. 29.

[135] CS Shin, KI Kim, MH Park, and Hang Joon Kim. Support vector machine-based text detection in digital video. In *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No. 00TH8501)*, volume 2, pages 634–641. IEEE, 2000. Cited on pp. 29 and 30.

[136] Datong Chen, Hervé Bourlard, and J-P Thiran. Text identification in complex background using svm. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001. Cited on pp. 30 and 36.

[137] Datong Chen and Jean-Marc Odobez. Comparison of support vector machine and neural network for text texture verification. Technical report, IDIAP, 2002. Cited on p. 30.

[138] Marios Anthimopoulos, Basilis Gatos, and Ioannis Pratikakis. A hybrid system for text detection in video frames. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pages 286–292. IEEE, 2008. Cited on pp. 30 and 36.

[139] Hongxing Sun, Nannan Zhao, and Xinhe Xu. Extraction of text under complex background using wavelet transform and support vector machine. In *2006 International Conference on Mechatronics and Automation*, pages 1493–1497. IEEE, 2006. Cited on pp. 30 and 36.

[140] Maryam Darab and Mohammad Rahmati. A hybrid approach to localize farsi text in natural scene images. *Procedia Computer Science*, 13:171–184, 2012. Cited on pp. 30 and 36.

[141] Mohieddin Moradi, Saeed Mozaffari, and Ali Asghar Orouji. Farsi/arabic text extraction from video images. In *2011 19th Iranian Conference on Electrical Engineering*, pages 1–6. IEEE, 2011. Cited on pp. 30 and 36.

[142] Salahuddin Unar, Akhtar Hussain Jalbani, Muhammad Moazzam Jawaid, Mohsin Shaikh, and Asghar Ali Chandio. Artificial urdu text detection and localization from individual video frames. *Mehran University Research Journal of Engineering and Technology*, 37(2): 429–438, 2018. Cited on pp. 30, 36, 49, and 84.

[143] Imran Siddiqi and Ahsen Raza. A database of artificial urdu text in video images with semi-automatic text line labeling scheme. In *MMEDIA 2012, The Fourth International Conferences on Advances in Multimedia*, pages 75–81, 2012. Cited on pp. 31, 36, 51, and 84.

[144] Leena Mary Francis and N Sreenath. Tedless–text detection using least-square svm from natural scene. *Journal of King Saud University-Computer and Information Sciences*, 32(3): 287–299, 2020. Cited on pp. 31 and 36.

[145] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55, 2014. Cited on p. 31.

[146] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *European Conference on Computer Vision*, pages 497–511. Springer, 2014. Cited on pp. 31, 36, and 46.

[147] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4159–4167, 2016. Cited on pp. 32, 36, and 46.

[148] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016. Cited on pp. 32, 36, and 46.

[149] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. *arXiv preprint arXiv:1703.06520*, 2017. Cited on pp. viii, 32, 36, and 46.

[150] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016. Cited on pp. viii, 32, 33, 36, and 46.

[151] Shuye Zhang, Mude Lin, Tianshui Chen, Lianwen Jin, and Liang Lin. Character proposal network for robust text extraction. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2633–2637. IEEE, 2016. Cited on pp. viii, 33, 34, and 36.

[152] Cong Yao, Xiang Bai, Nong Sang, Xinyu Zhou, Shuchang Zhou, and Zhimin Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016. Cited on pp. 33 and 36.

[153] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167, 2017. Cited on pp. viii, 33, 34, and 36.

[154] Xu Yan, Shan Siyuan, Qiu Ziming, Jia Zhipeng, Shen Zhengyang, Wang Yipei, Shi Mengfei, Eric I, and Chang Chao. End-to-end subtitle detection and recognition for videos in east asian languages via cnn ensemble. *Signal Processing: Image Communication*, 60:131–143, 2018. Cited on pp. viii, 33, 36, 44, 45, and 47.

[155] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proc. CVPR*, pages 2642–2651, 2017. Cited on pp. 33 and 36.

[156] Oussama Zayene, Jean Hennebert, Sameh Masmoudi Touj, Rolf Ingold, and Najoua Essoukri Ben Amara. A dataset for arabic text detection, tracking and recognition in news videos - activ. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015. Cited on pp. 34, 36, 44, 46, 47, 49, and 84.

[157] Sonia Yousfi, Sid-Ahmed Berrani, and Christophe Garcia. Arabic text detection in videos using neural and boosting-based approaches: Application to video indexing. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3028–3032. IEEE, 2014. Cited on pp. 35, 36, and 84.

[158] Mohieddin Moradi, Saeed Mozaffari, and Ali Asghar Orouji. Farsi/arabic text extraction from video images by corner detection. In *2010 6th Iranian Conference on Machine Vision and Image Processing*, pages 1–6. IEEE, 2010. Cited on p. 36.

[159] Ray Smith. An overview of the tesseract ocr engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007. Cited on p. 36.

[160] Edward Mendelson. Abbyy finereader professional 9.0. *PC Magazine*, 2008. Cited on p. 36.

[161] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, and Ching Suen. *Character recognition systems: a guide for students and practitioners*. John Wiley & Sons, 2007. Cited on p. 37.

[162] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2014. Cited on p. 37.

[163] U Pal and Anirban Sarkar. Recognition of printed urdu script. In *Proc. 7th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1183–1187, 2003. Cited on pp. 37, 38, and 47.

[164] Sobia Tariq Javed and Sarmad Hussain. Segmentation based urdu nastalique ocr. In *Proc. of Iberoamerican Congress on Pattern Recognition CIARP*, pages 41–49, 2013. Cited on pp. 37 and 39.

[165] Israr Ud Din, Imran Siddiqi, Shehzad Khalid, and Tahir Azam. Segmentation-free optical character recognition for printed urdu text. *EURASIP Journal on Image and Video Processing*, 2017(1):62, 2017. Cited on pp. 37, 38, 48, and 87.

[166] Muhammad Ferjad Naeem, Aqsa Ahmed Awan, Faisal Shafait, Adnan ul Hasan, et al. Impact of ligature coverage on training practical urdu ocr systems. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 131–136. IEEE, 2017. Cited on pp. 37 and 38.

[167] Gurpreet Singh Lehal. Choice of recognizable units for urdu ocr. In *Proceeding of the workshop on document analysis and recognition*, pages 79–85. ACM, 2012. Cited on pp. 37 and 87.

[168] Saeeda Naz, Arif I Umar, Riaz Ahmad, Saad B Ahmed, Syed H Shirazi, Imran Siddiqi, and Muhammad I Razzak. Offline cursive urdu-nastaliq script recognition using multidimensional recurrent neural networks. *Neurocomputing*, 177:228–241, 2016. Cited on pp. 37, 38, 40, 47, 48, 77, and 104.

[169] Nizwa Javed, Safia Shabbir, Imran Siddiqi, and Khurram Khurshid. Classification of urdu ligatures using convolutional neural networks-a novel approach. In *Frontiers of Information Technology (FIT), 2017 International Conference on*, pages 93–97. IEEE, 2017. Cited on pp. 37 and 38.

[170] Shivakumara Palaiahnakote, Phan Trung Quy, and Tan Chew Lim. A laplacian approach to multi-oriented text detection in video. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):412–419, 2011. Cited on p. 38.

[171] Angelika Garz, Mathias Seuret, Fotini Simistira, Andreas Fischer, and Rolf Ingold. Creating ground truth for historical manuscripts with document graphs and scribbling interaction. In *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*, pages 126–131. IEEE, 2016. Cited on p. 38.

[172] Alejandro Héctor Toselli, Enrique Vidal, Verónica Romero, and Volkmar Frinken. Hmm word graph based keyword spotting in handwritten document images. *Information Sciences*, 370:497–518, 2016. Cited on p. 38.

[173] Jean-Baptiste Fasquel and Nicolas Delanoue. A graph based image interpretation method using a priori qualitative inclusion and photometric relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. Cited on p. 38.

[174] Weinman Jerod J and Learned-Miller Erik. Improving recognition of novel input with similarity. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 308–315. IEEE, 2006. Cited on p. 38.

[175] Noman Islam, Zeeshan Islam, and Nazia Noor. A survey on optical character recognition system. *arXiv preprint arXiv:1710.05703*, 2017. Cited on p. 38.

[176] Bangjun Lei, Guangzhu Xu, Ming Feng, Ferdinand Van der Heijden, Yaobin Zou, Dick de Ridder, and David MJ Tax. *Classification, parameter estimation and state estimation: an engineering approach using MATLAB*. John Wiley & Sons, 2017. Cited on p. 38.

[177] Weinman Jerod J, Butler Zachary, Knoll Dugan, and Feild Jacqueline. Toward integrated scene text reading. *IEEE transactions on pattern analysis and machine intelligence*, 36(2): 375–387, 2014. Cited on p. 38.

[178] Ahmed H Metwally, Mahmoud I Khalil, and Hazem M Abbas. Offline arabic handwriting recognition using hidden markov models and post-recognition lexicon matching. In *Computer Engineering and Systems (ICCES), 2017 12th International Conference on*, pages 238–243. IEEE, 2017. Cited on pp. 38, 41, and 46.

[179] Mouhcine Rabi, Mustapha Amrouch, and Zouhir Mahani. Recognition of cursive arabic handwritten text using embedded training based on hidden markov models. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(01):1860007, 2018. Cited on pp. 38, 41, and 46.

[180] Mouhcine Rabi, Mustapha Amrouch, and Zouhair Mahani. Cursive arabic handwriting recognition system without explicit segmentation based on hidden markov models. *Journal of Data Mining and Digital Humanities*, 2018. Cited on pp. 38, 41, and 46.

[181] Peng Zhou, Linlin Li, and Chew Lim Tan. Character recognition under severe perspective distortion. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 676–680. IEEE, 2009. Cited on p. 38.

[182] Gulcin Caner and Ismail Haritaoglu. Shape-dna: effective character restoration and enhancement for arabic text documents. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2053–2056. IEEE, 2010. Cited on p. 38.

[183] Pramod Sankar Kompalli. Image document processing in a client-server system including privacy-preserving text recognition, December 19 2017. US Patent 9,847,974. Cited on p. 38.

[184] V Märgner, U Pal, A Antonacopoulos, et al. Document analysis and text recognition, 2018. Cited on p. 38.

[185] Wang Tao, Wu David J, Coates Adam, and Ng Andrew Y. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3304–3308. IEEE, 2012. Cited on p. 38.

[186] Sebastian Sudholt and Gernot A Fink. Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 277–282. IEEE, 2016. Cited on p. 38.

[187] Cheng-Lin Liu, Gernot A Fink, Venu Govindaraju, and Lianwen Jin. Special issue on deep learning for document analysis and recognition, 2018. Cited on p. 38.

[188] Mohamed Yousef, Khaled F Hussain, and Usama S Mohammed. Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. *Pattern Recognition*, page 107482, 2020. Cited on p. 38.

[189] Hassan El Bahi and Abdelkarim Zatni. Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network. *Multimedia tools and applications*, 78(18):26453–26481, 2019. Cited on p. 38.

[190] Saeeda Naz, Arif I Umar, Riaz Ahmad, Imran Siddiqi, Saad B Ahmed, Muhammad I Razzak, and Faisal Shafait. Urdu nastaliq recognition using convolutional–recursive deep learning. *Neurocomputing*, 243:80–87, 2017. Cited on pp. 38, 40, 77, and 104.

[191] Asma Naseer and Kashif Zafar. Comparative analysis of raw images and meta feature based urdu ocr using cnn and lstm. *Int J Adv Comput Sci Appl*, 9(1):419–424, 2018. Cited on p. 38.

[192] Inam Shamsher, Zaheer Ahmad, Jehanzeb Khan Orakzai, and Awais Adnan. Ocr for printed urdu script using feed forward neural network. *the Proceedings of World Academy of Science, Engineering and Technology*, 23, 2007. Cited on pp. 38 and 47.

[193] Junaid Tariq, Umar Nauman, and Muhammad Umair Naru. Softconverter: A novel approach to construct ocr for printed urdu isolated characters. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 3, pages V3–495. IEEE, 2010. Cited on p. 38.

[194] Shuwair Sardar and Abdul Wahab. Optical character recognition system for urdu. In *2010 International Conference on Information and Emerging Technologies*, pages 1–5. IEEE, 2010. Cited on p. 38.

[195] Irfan Ahmad, Sabri A Mahmoud, and Gernot A Fink. Open-vocabulary recognition of machine-printed arabic text using hidden markov models. *Pattern Recognition*, 51:97–111, 2016. Cited on p. 38.

[196] Akram Khémiri, Afef Kacem Echi, Abdel Belaïd, and Mourad Elloumi. Arabic handwritten words off-line recognition based on hmms and dbns. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 51–55. IEEE, 2015. Cited on pp. 38, 41, and 47.

[197] Sobia Tariq Javed and Sarmad Hussain. Segmentation based urdu nastalique ocr. In *Iberoamerican Congress on Pattern Recognition*, pages 41–49. Springer, 2013. Cited on pp. 38 and 47.

[198] Sobia T aved, Sarmad Hussain, Ameera Maqbool, Samia Asloob, Sehrish Jamil, and Huma Moin. Segmentation free nastalique urdu ocr. *World Academy of Science, Engineering and Technology*, 46:456–461, 2010. Cited on p. 38.

[199] Ali Abidi, Akhtar Jamil, Imran Siddiqi, and Khurram Khurshid. Word spotting based retrieval of urdu handwritten documents. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 331–336. IEEE, 2012. Cited on p. 38.

[200] Saad Bin Ahmed, Saeeda Naz, Salahuddin Swati, and Muhammad Imran Razzak. Handwritten urdu character recognition using one-dimensional blstm classifier. *Neural Computing and Applications*, 31(4):1143–1151, 2019. Cited on pp. 38 and 47.

[201] Saad Bin Ahmed, Ibrahim A Hameed, Saeeda Naz, Muhammad Imran Razzak, and Rubiyah Yusof. Evaluation of handwritten urdu text by integration of mnist dataset learning experience. *IEEE access*, 7:153566–153578, 2019. Cited on pp. 38 and 47.

[202] Tabassam Nawaz, SAHS Naqvi, Habib ur Rehman, and Anoshia Faiz. Optical character recognition system for urdu (naskh font) using pattern matching technique. *International Journal of Image Processing (IJIP)*, 3(3):92, 2009. Cited on p. 39.

[203] Zaheer Ahmad, Jehanzeb Khan Orakzai, Inam Shamsher, and Awais Adnan. Urdu nastaleeq optical character recognition. In *Proc. of world academy of science, engineering and technology*, pages 249–252, 2007. Cited on pp. 39, 47, and 104.

[204] Nazly Sabbour and Faisal Shafait. A segmentation-free approach to arabic and urdu ocr. In *Document Recognition and Retrieval XX*, volume 8658, page 86580N. International Society for Optics and Photonics, 2013. Cited on pp. 39 and 48.

[205] Sobia T Javed, Sarmad Hussain, Ameera Maqbool, Samia Asloob, Sehrish Jamil, and Huma Moin. Segmentation free nastalique urdu ocr. *Proc. of World Academy of Science, Engineering and Technology*, pages 456–461, 2010. Cited on p. 39.

[206] Q Akram, S Hussain, F Adeeba, S Rehman, and M Saeed. Framework of urdu nastalique optical character recognition system. In *Proc. of Conference on Language and Technology (CLT)*, pages 1–7, 2014. Cited on pp. 39 and 104.

[207] Center for language engineering. http://http://www.cle.org.pk/, 2019. Accessed: 2019-04-15. Cited on pp. 39, 40, 47, and 48.

[208] Israr Uddin Khattak, Imran Siddiqi, Shehzad Khalid, and Chawki Djeddi. Recognition of urdu ligatures-a holistic approach. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 71–75. IEEE, 2015. Cited on pp. 39, 40, and 47.

[209] Syed Yasser Arafat and Muhammad Javed Iqbal. Two stream deep neural network for sequence-based urdu ligature recognition. *IEEE Access*, 7:159090–159099, 2019. Cited on pp. 40, 46, and 47.

[210] Naila Habib Khan, Awais Adnan, Abdul Waheed, Mahdi Zareei, Abdallah Aldosary, and Ehab Mahmoud Mohamed. Urdu ligature recognition system: An evolutionary approach. *CMC-COMPUTERS MATERIALS & CONTINUA*, 66(2):1347–1367, 2021. Cited on p. 40.

[211] Saad Bin Ahmed, Saeeda Naz, Muhammad Imran Razzak, Shiekh Faisal Rashid, Muhammad Zeeshan Afzal, and Thomas M Breuel. Evaluation of cursive and non-cursive scripts using recurrent neural networks. *Neural Computing and Applications*, 27(3):603–613, 2016. Cited on pp. 40, 47, and 87.

[212] Adnan Ul-Hasan, Saad Bin Ahmed, Faisal Rashid, Faisal Shafait, and Thomas M Breuel. Offline printed urdu nastaleeq script recognition with bidirectional lstm networks. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1061–1065. IEEE, 2013. Cited on pp. 40 and 47.

[213] Saeeda Naz, Arif I Umar, Riaz Ahmad, Saad B Ahmed, Syed H Shirazi, and Muhammad I Razzak. Urdu nasta'liq text recognition system based on multi-dimensional recurrent neural network and statistical features. *Neural Computing and Applications*, 28(2):219–231, 2017. Cited on pp. 40, 46, 48, 77, 87, and 104.

[214] Tayaba Anjum and Nazar Khan. An attention based method for offline handwritten urdu text recognition. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 169–174. IEEE, 2020. Cited on p. 40.

[215] Hashim Raza Khan, MA Hasan, M Kazmi, Nabeel Fayyaz, H Khalid, and SA Qazi. A holistic approach to urdu language word recognition using deep neural networks. *Engineering, Technology & Applied Science Research*, 11(3):7140–7145, 2021. Cited on p. 40.

[216] Malik Waqas Sagheer, Nicola Nobile, Chun Lei He, and Ching Y Suen. A novel handwritten urdu word spotting based on connected components analysis. In *2010 20th International Conference on Pattern Recognition*, pages 2013–2016. IEEE, 2010. Cited on p. 41.

[217] Ali Abidi, Imran Siddiqi, and Khurram Khurshid. Towards searchable digital urdu libraries-a word spotting based retrieval approach. In *2011 International Conference on Document Analysis and Recognition*, pages 1344–1348. IEEE, 2011. Cited on p. 41.

[218] Ahmed Lawgali. A survey on arabic character recognition. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(2):401–426, 2015. Cited on pp. 41 and 46.

[219] Mario Pechwitz, S Snoussi Maddouri, Volker Märgner, Noureddine Ellouze, Hamid Amiri, et al. Ifn/enit-database of handwritten arabic words. In *Proc. of CIFED*, volume 2, pages 127–136. Citeseer, 2002. Cited on pp. 41 and 47.

[220] Irfan Ahmad, Sabri A Mahmoud, and Gernot A Fink. Open-vocabulary recognition of machine-printed arabic text using hidden markov models. *Pattern recognition*, 51:97–111, 2016. Cited on pp. 41 and 47.

[221] Gheith A Abandah, Fuad T Jamour, and Esam A Qaralleh. Recognizing handwritten arabic words using grapheme segmentation and recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):275–291, 2014. Cited on pp. 41, 46, and 47.

[222] et al. Saudagar, Abdul Khader Jilani. Efficient arabic text extraction and recognition using thinning and dataset comparison technique. *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on. IEEE*, 2015. Cited on p. 41.

[223] Irfan Ahmad, Sabri A Mahmoud, and Gernot A Fink. Open-vocabulary recognition of machine-printed arabic text using hidden markov models. *Pattern Recognition*, 51:97–111, 2016. Cited on pp. 41 and 46.

[224] Saeeda Naz, Saad Ahmed, Riaz Ahmad, and Muhammad Razza. Arabic script based digit recognition systems. In *International Conference on Recent Advances in Computer Systems*. Atlantis Press, 2015. Cited on pp. 41 and 46.

[225] Saeeda Naz, Naila H Khan, Shizza Zahoor, and Muhammad I Razzak. Deep ocr for arabic script-based language like pastho. *Expert Systems*, page e12565, 2020. Cited on pp. 41 and 46.

[226] Riaz Ahmad, Saeeda Naz, Muhammad Zeshan Afzal, Sheikh Faisal Rashid, Marcus Liwicki, and Andreas Dengel. A deep learning based arabic script recognition system: benchmark on khat. *Int. Arab J. Inf. Technol.*, 17(3):299–305, 2020. Cited on pp. 41 and 46.

[227] Chen-Yu Lee, Anurag Bhardwaj, Wei Di, Vignesh Jagadeesh, and Robinson Piramuthu. Region-based discriminative feature pooling for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4050–4057, 2014. Cited on p. 41.

[228] Shangxuan Tian, Ujjwal Bhattacharya, Shijian Lu, Bolan Su, Qingqing Wang, Xiaohua Wei, Yue Lu, and Chew Lim Tan. Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. *Pattern Recognition*, 51:125–134, 2016. Cited on p. 41.

[229] Boran Yu and Hongjie Wan. Chinese text detection and recognition in natural scene using hog and svm. *DEStech Transactions on Computer Science and Engineering*, 2016. Cited on p. 41.

[230] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4042–4049, 2014. Cited on p. 41.

[231] Chucai Yi and Yingli Tian. Scene text recognition in mobile applications by character descriptor and structure configuration. *IEEE transactions on image processing*, 23(7): 2972–2982, 2014. Cited on p. 41.

[232] Bowornrat Sriman and Lambert Schomaker. Object attention patches for text detection and recognition in scene images using sift. In *ICPRAM (1)*, pages 304–311, 2015. Cited on p. 41.

[233] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. Cited on pp. 41 and 44.

[234] Sunil Kumar, Krishan Kumar, Rahul Kumar Mishra, et al. Scene text recognition using artificial neural network: A survey. *International Journal of Computer Applications*, 137(6), 2016. Cited on p. 41.

[235] Siyu Zhu and Richard Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–632, 2016. Cited on p. 41.

[236] Shijian Lu, Tao Chen, Shangxuan Tian, Joo-Hwee Lim, and Chew-Lim Tan. Scene text extraction based on edges and support vector regression. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):125–135, 2015. Cited on p. 41.

[237] Luk Neumann. *Scene text localization and recognition in images and videos*. PhD thesis, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University, 2017. Cited on p. 41.

[238] Ahmed Hechri, Rihab Hmida, and Abdellatif Mtibaa. Robust road lanes and traffic signs recognition for driver assistance system. *International Journal of Computational Science and Engineering*, 10(1-2):202–209, 2015. Cited on p. 41.

[239] Ayoub Ellahyani, Mohamed El Ansari, and Ilyas El Jaafari. Traffic sign detection and recognition based on random forests. *Applied Soft Computing*, 46:805–815, 2016. Cited on p. 41.

[240] A Salhi, B Minaoui, M Fakir, H Chakib, and H Grimech. Traffic signs recognition using hp and hog descriptors combined to mlp and svm classifiers. *Traffic*, 8(11), 2017. Cited on p. 41.

[241] Karattupalayam Chidambaram Saranya and Vutsal Singhal. Real-time prototype of driver assistance system for indian road signs. In *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, pages 147–155. Springer, 2018. Cited on p. 41.

[242] Yan Lai, Nanxin Wang, Yusi Yang, and Lan Lin. Traffic signs recognition and classification based on deep feature learning. In *7th International Conference on Pattern Recognition Applications and Methods (ICPRAM), Madeira, Portugal*, pages 622–629, 2018. Cited on p. 41.

[243] Kai Wang and Serge Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer, 2010. Cited on p. 41.

[244] Vibhor Goel, Anand Mishra, Karteek Alahari, and CV Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 398–402. IEEE, 2013. Cited on p. 41.

[245] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European conference on computer vision*, pages 512–528. Springer, 2014. Cited on p. 41.

[246] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016. Cited on p. 41.

[247] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2017. Cited on p. 41.

[248] Haodong Yang, Shuohao Li, Xiaoqing Yin, Anqi Han, and Jun Zhang. Recurrent highway networks with attention mechanism for scene text recognition. In *Digital Image Computing: Techniques and Applications (DICTA), 2017 International Conference on*, pages 1–8. IEEE, 2017. Cited on p. 41.

[249] Michal Bušta, Lukáš Neumann, and Jirı Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2223–2231. IEEE, 2017. Cited on p. 41.

[250] Zhengchao Lei, Sanyuan Zhao, Hongmei Song, and Jianbing Shen. Scene text recognition using residual convolutional recurrent neural network. *Machine Vision and Applications*, pages 1–11, 2018. Cited on p. 41.

[251] Xiaobing Wang, Yingying Jiang, Shuli Yang, Xiangyu Zhu, Wei Li, Pei Fu, Hua Wang, and Zhenbo Luo. End-to-end scene text recognition in videos based on multi frame tracking. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1255–1260. IEEE, 2017. Cited on pp. 41 and 47.

[252] Zichuan Liu, Yixing Li, Fengbo Ren, Wang Ling Goh, and Hao Yu. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 23–28, 2018. Cited on pp. viii, 42, 46, and 47.

[253] Wei Liu, Chaofeng Chen, and Kwan-Yee K Wong. Char-net: A character-aware neural network for distorted scene text recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 23–29, 2018. Cited on pp. viii and 42.

[254] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. *arXiv preprint arXiv:1809.06508*, 2018. Cited on pp. 42, 46, and 47.

[255] Yuting Gao, Zheng Huang, and Yuchen Dai. Double supervised network with attention mechanism for scene text recognition. *arXiv preprint arXiv:1808.00677*, 2018. Cited on pp. 42 and 47.

[256] Asghar Ali Chandio, Md Asikuzzaman, Mark Pickering, and Mehwish Leghari. Cursive-text: A comprehensive dataset for end-to-end urdu text recognition in natural scene images. *Data in Brief*, page 105749, 2020. Cited on pp. 42, 46, and 47.

[257] Asghar Ali and Mark Pickering. Urdu-text: A dataset and benchmark for urdu text detection and recognition in natural scenes. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 323–328. IEEE, 2019. Cited on p. 42.

[258] Asghar Ali and Mark Pickering. A hybrid deep neural network for urdu text recognition in natural images. In *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, pages 321–325. IEEE, 2019. Cited on pp. 42, 46, and 47.

[259] Muhammad A Panhwar, Kamran A Memon, Adeel Abro, Deng Zhongliang, Sijjad A Khuhro, and Saleemullah Memon. Signboard detection and text recognition using artificial neural networks. In *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 16–19. IEEE, 2019. Cited on pp. 42, 46, and 47.

[260] Syed Yasser Arafat and Muhmmad Javed Iqbal. Urdu-text detection and recognition in natural scene images using deep learning. *IEEE Access*, 2020. Cited on pp. 42, 46, and 47.

[261] Syed Yasser Arafat, Nabeel Ashraf, Muhammad Javed Iqbal, Iftikhar Ahmad, Suleman Khan, and Joel JPC Rodrigues. Urdu signboard detection and recognition using deep learning. *Multimedia Tools and Applications*, pages 1–23, 2021. Cited on p. 43.

[262] SeongHun Lee and JinHyung Kim. Complementary combination of holistic and component analysis for recognition of low-resolution video character images. *Pattern Recognition Letters*, 29(4):383–391, 2008. Cited on pp. 43 and 47.

[263] Xiaoou Tang, Xinbo Gao, Jianzhuang Liu, and Hongjiang Zhang. A spatial-temporal approach for video caption detection and recognition. *IEEE transactions on neural networks*, 13(4):961–971, 2002. Cited on pp. 43 and 47.

[264] Khaoula Elagouni, Christophe Garcia, and Pascale Sébillot. A comprehensive neural-based approach for text recognition in videos using natural language processing. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, 2011. Cited on pp. 43 and 47.

[265] Palaiahnakote Shivakumara, Trung Quy Phan, Shijian Lu, and Chew Lim Tan. Video character recognition through hierarchical classification. In *2011 International Conference on Document Analysis and Recognition*, pages 131–135. IEEE, 2011. Cited on pp. 43 and 47.

[266] Mohd Javed Khatri, Abhishek Shetty, Ajay Gupta, and Grishma Sharma. Video ocr for indexing and retrieval. *International Journal of Computer Applications*, 118(2), 2015. Cited on p. 43.

[267] Parag Kulkarni, Bhagyashri Patil, and Bela Joglekar. An effective content based video analysis and retrieval using pattern indexing techniques. In *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pages 87–92. IEEE, 2015. Cited on p. 43.

[268] Wei Lu, Hongbo Sun, Jinghui Chu, Xiangdong Huang, and Jiexiao Yu. A novel approach for video text detection and recognition based on a corner response feature map and transferred deep convolutional neural network. *IEEE Access*, 6:40198–40211, 2018. Cited on p. 44.

[269] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5 (1):39–46, 2002. Cited on p. 44.

[270] M Ben Halima, Hichem Karray, and Adel M Alimi. A comprehensive method for arabic video text detection, localization, extraction and recognition. In *Pacific-Rim Conference on Multimedia*, pages 648–659. Springer, 2010. Cited on pp. 44 and 47.

[271] M Ben Halima, Hichem Karray, and Adel M Alimi. Arabic text recognition in video sequences. *arXiv preprint arXiv:1308.3243*, 2013. Cited on pp. 44, 46, and 47.

[272] Sonia Yousfi, Sid-Ahmed Berrani, and Christophe Garcia. Deep learning and recurrent connectionist-based approaches for arabic text recognition in videos. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1026–1030. IEEE, 2015. Cited on pp. 44 and 47.

[273] Sonia Yousfi, Sid-Ahmed Berrani, and Christophe Garcia. Alif: A dataset for arabic embedded text recognition in tv broadcast. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1221–1225. IEEE, 2015. Cited on pp. viii, 44, 45, 46, 47, and 49.

[274] Mohit Jain, Minesh Mathew, and CV Jawahar. Unconstrained scene text and video text recognition for arabic script. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 26–30. IEEE, 2017. Cited on pp. 44, 46, and 47.

[275] Asif Shahab, Faisal Shafait, and Andreas Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *2011 international conference on document analysis and recognition*, pages 1491–1496. IEEE, 2011. Cited on pp. 46 and 49.

[276] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. Cited on pp. 46 and 49.

[277] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. Cited on pp. 46 and 49.

[278] Israr Uddin Khattak, Imran Siddiqi, Shehzad Khalid, and Chawki Djeddi. Recognition of urdu ligatures- a holistic approach. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 71–75. IEEE, 2015. Cited on pp. 46 and 123.

[279] Shabbir Safia and Siddiqi Imran. Optical character recognition system for urdu words in nastaliq font. *BU*, 2016. Cited on p. 46.

[280] Anand Mishra, Karteek Alahari, and CV Jawahar. Top-down and bottom-up cues for scene text recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2687–2694. IEEE, 2012. Cited on p. 47.

[281] Shuwair Sardar and Abdul Wahab. Optical character recognition system for urdu. In *Information and Emerging Technologies (ICIET), 2010 International Conference on*, pages 1–5. IEEE, 2010. Cited on p. 47.

[282] Raashid Hussain, Haris Ahmad Khan, Imran Siddiqi, Khurram Khurshid, and Asif Masood. Keyword based information retrieval system for urdu document images. In *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 27–33. IEEE, 2015. Cited on p. 47.

[283] Sobia T Javed, Sarmad Hussain, Ameera Maqbool, Samia Asloob, Sehrish Jamil, and Huma Moin. Segmentation free nastalique urdu ocr. *World Academy of Science, Engineering and Technology*, 46:456–461, 2010. Cited on p. 47.

[284] Qingqing Wang, Wenjing Jia, Xiangjian He, Yue Lu, Michael Blumenstein, Ye Huang, and Shujing Lyu. Deeptext: Detecting text from the wild with multi-aspp-assembled deeplab. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 208–213. IEEE, 2019. Cited on p. 48.

[285] Zhuoyao Zhong, Lei Sun, and Qiang Huo. A teacher-student learning based born-again training approach to improving scene text detection accuracy. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 281–286. IEEE, 2019. Cited on p. 48.

[286] Chen Du, Chunheng Wang, Yanna Wang, Zipeng Feng, and Jiyuan Zhang. Textedge: Multi-oriented scene text detection via region segmentation and edge classification. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 375–380. IEEE, 2019. Cited on p. 48.

[287] Xi Liu, Rui Zhang, Yongsheng Zhou, and Dong Wang. Scene text detection with feature pyramid network and linking segments. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 508–513. IEEE, 2019. Cited on p. 48.

[288] Ahsen Raza and Imran Siddiqi. A database of artificial urdu text in video images with semi-automatic text line labeling scheme. In *Proc. 4th Int. Conf. Adv. Multimedia (MMEDIA)*, pages 75–81, 2012. Cited on p. 48.

[289] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, Robert Young, Kazuki Ashida, Hiroki Nagai, Masayuki Okamoto, Hiroaki Yamamoto, et al. Icdar 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(2-3):105–122, 2005. Cited on pp. 49 and 51.

[290] Christian Wolf and Jean-Michel Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4):280–296, 2006. Cited on pp. 51 and 62.

[291] Vladimir Iosifovich Levenshtein. Binary codes with correction of drops, insertions and substitutions of characters. In *Reports of the Academy of Sciences*, volume 163-4, pages 845–848. Russian Academy of Sciences, 1965. Cited on p. 52.

[292] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988. Cited on p. 62.

[293] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. Cited on pp. 63, 89, 90, and 102.

[294] Raza Ahsen, Ali Abidi, and Imran Siddiqi. Multilingual artificial text detection and extraction from still images. *IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics*, 2013. Cited on p. 63.

[295] Ali Mirza, Marium Fayyaz, Zunera Seher, and Imran Siddiqi. Urdu caption text detection using textural features. In *Proceedings of the 2nd Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, pages 70–75. ACM, 2018. Cited on pp. 68 and 84.

[296] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. Cited on pp. 68, 122, and 123.

[297] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. Cited on p. 68.

[298] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Cited on pp. 68, 122, and 123.

[299] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. Cited on pp. 68, 70, 122, and 123.

[300] David Bouchain. Character recognition using convolutional neural networks. *Institute for Neural Information Processing*, 2007, 2006. Cited on p. 68.

[301] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171, 2013. Cited on pp. 68 and 69.

[302] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 643–650. ACM, 2015. Cited on p. 68.

[303] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. Cited on pp. 68 and 69.

[304] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. Cited on pp. 68 and 69.

[305] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. Cited on pp. 68 and 69.

[306] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. Cited on pp. ix, 68, 69, and 71.

[307] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. Cited on pp. ix, 68, 70, and 71.

[308] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. Cited on p. 70.

[309] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. Cited on p. 70.

[310] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. Cited on pp. ix, 71, and 72.

[311] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. Cited on pp. 72, 122, and 123.

[312] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. Cited on p. 72.

[313] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. Cited on p. 72.

[314] Jason Brownlee. What is the difference between test and validation datasets. *Machine Learning Mastery*, 14, 2017. Cited on p. 77.

[315] Tarang Shah. About train, validation and test sets in machine learning. *Towards Data Science*, 6, 2017. Cited on p. 77.

[316] Ayedh Alqahtani and Andrew Whyte. Estimation of life-cycle costs of buildings: regression vs artificial neural network. *Built Environment Project and Asset Management*, 2016. Cited on p. 77.

[317] R Draelos. Best use of train/val/test splits, with tips for medical data. *GLASS BOX*, 15, 2019. Cited on p. 77.

[318] Zayene Oussama, Hennebert Jean, Touj Sameh Masmoudi, Ingold Rolf, and Amara Najoua Essoukri Ben. A dataset for arabic text detection, tracking and recognition in news

videos-activ. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 996–1000. IEEE, 2015. Cited on p. 77.

[319] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017. Cited on p. 78.

[320] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018. Cited on p. 80.

[321] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. Cited on p. 80.

[322] Ali Mirza, Ossama Zeshan, Muhammad Atif, and Imran Siddiqi. Detection and recognition of cursive text from video frames. *EURASIP Journal on Image and Video Processing*, 2020 (1):1–19, 2020. Cited on p. 85.

[323] Saeeda Naz, Arif I Umar, Riaz Ahmad, Saad B Ahmed, Syed H Shirazi, and Muhammad I Razzak. Urdu nastaliq text recognition system based on multi-dimensional recurrent neural network and statistical features. *Neural Computing and Applications*, pages 1–13, 2015. Cited on p. 87.

[324] Ali Mirza, Imran Siddiqi, Syed G.Mustufa, and Mazahir Hussain. Impact of pre-processing on recognition of cursive video text. In *9th Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2019. Cited on p. 88.

[325] Jaakko Sauvola and Matti Pietikäinen. Adaptive document image binarization. *Pattern recognition*, 33(2):225–236, 2000. Cited on pp. 90, 91, and 102.

[326] Meng-Ling Feng and Yap-Peng Tan. Contrast adaptive binarization of low quality document images. *IEICE Electronics Express*, 1(16):501–506, 2004. Cited on pp. 90, 91, and 102.

[327] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997. Cited on p. 94.

[328] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. Cited on p. 95.

[329] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006. Cited on p. 96.

[330] Adnan Ul-Hasan, Saad Bin Ahmed, Faisal Rashid, Faisal Shafait, and Thomas M Breuel. Offline printed urdu nastaleeq script recognition with bidirectional lstm networks. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1061–1065. IEEE, 2013. Cited on p. 104.

[331] Sarmad Hussain, Salman Ali, and Qurat ul ain Akram. Nastalique segmentation-based approach for urdu ocr. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(4):357–374, 2015. Cited on p. 104.

[332] Saad Bin Ahmed, Saeeda Naz, Muhammad Imran Razzak, Shiekh Faisal Rashid, Muhammad Zeeshan Afzal, and Thomas M Breuel. Evaluation of cursive and non-cursive scripts using recurrent neural networks. *Neural Computing and Applications*, 27(3):603–613, 2016. Cited on p. 104.

[333] Ali Mirza and Imran Siddiqi. Recognition of cursive video text using a deep learning framework. *IET Image Processing*, 14(14):3444–3455, 2020. Cited on p. 105.

[334] Ali Mirza, Imran Siddiqi, Umar Hayat, Muhammad Atif, and Syed Ghulam Mustufa. Recognition of cursive caption text using deep learning-a comparative study on recognition units. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 156–167. Springer, 2020. Cited on p. 105.

[335] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. Cited on pp. 122 and 123.

[336] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. Cited on p. 123.