Unconstrained Arabic & Urdu Text Recognition using Deep CNN-RNN Hybrid Networks

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

Mohit Jain
201202164
mohit.jain@research.iiit.ac.in

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2018

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Unconstrained Arabic & Urdu Text Recognition using Deep CNN-RNN Hybrid Networks” by Mohit Jain, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. C.V. Jawahar

To

Maa, Papa & Bhaiya

Acknowledgements

The success and final outcome of this thesis has been through continual guidance and assistance of many people. I feel privileged to have found such a good community of people all along who helped me take this project to its completion.

First of all, I would like to express my sincere gratitude to Prof. C.V. Jawahar for introducing me to research. His vision and zest for thoroughness in work are an inspiration and have taught me valuable lessons that I'll cherish throughout my professional career.

I feel blessed to have had Minesh Mathew as a friend and my PhD advisor at all decisive intersections of my research life. I couldn't have asked for better companions to cope with the pressures of research than Vinitha and Minesh. Thank you for keeping me under your wing like a younger sibling.

I would also like to extend my deepest regards to my lab mates and friends at CVIT, especially Praveen Krishnan, Sourabh Daptardar, Parikshit Sakurikar, Suriya Singh, Aditya Arun, Harish Krishna and Sirnam Swetha for their timely help and support at various stages of my work. I am also thankful to Silar Shaik, Varun Bhargawan, Siva Kumar and Ram Sharma for helping with the annotation of datasets, server handling issues and scheduling various meetings to fast track my work.

Thank you Himani and Urvashi for always keeping my back through tough times and motivating me to stay on-track in moments of self-doubt. Kalpit and Nihit, my brothers, we've done some crazy things and made even crazier memories together. Thanks for being an indispensable part of this ride, #BikerChokras forever! Somya, Ankita and Virali you've brought the magic of mischief in my life like none other have and I feel eternally grateful for having met you. Princu, Simran, Jigar, Anubhav, Niamat, Prabh, Aarshvi, Ankit, Devansh, Darshan and Aditya, thanks for making my college life one that I'll cherish forever and ever after.

Last and definitely not the least, I would like to thank my family for showering me with their love and blessings, at times understanding my needs better than me and never losing faith in me. I love you Maa, Papa, Nishant bhaiya and Arwa.

– Mohit Jain

Abstract

We demonstrate the effectiveness of an end-to-end trainable hybrid CNN-RNN architecture in recognizing Urdu text from printed documents, typically known as Urdu OCR, and from Arabic text embedded in videos and natural scenes. When dealing with low-resource languages like Arabic and Urdu, a major adversary in developing a robust recognizer is the lack of large quantity of annotated data. We overcome this problem by synthesizing millions of images from a large vocabulary of words and phrases scraped from Wikipedia's Arabic and Urdu versions, using a wide variety of fonts downloaded from various online resources.

Building robust recognizers for Arabic and Urdu text has always been a challenging task. Though a lot of research has been done in the field of text recognition, the focus of the vision community has been primarily on English. While, Arabic script has started to receive some spotlight as far as text recognition is concerned, works on other languages which use the *Nabatean* family of scripts, like Urdu and Persian, are very limited. Moreover, the quality of the works presented in this field generally lack a standardized structure making it hard to reproduce and verify the claims or results. This is quite surprising considering the fact that Arabic is the fifth most spoken language in the world after Chinese, English, Spanish and Hindi catering to 4.7% of the world's population, while Urdu has over a 100 million speakers and is spoken widely in Pakistan, where it is the national language, and India where it is recognized as one of the 22 official languages.

In this thesis, we introduce the problems related with text recognition of low-resource languages, namely Arabic and Urdu, in various scenarios. We propose a language independent Hybrid CNN-RNN architecture which can be trained in an end-to-end fashion and prove it's dominance over simple RNN based methods. Moreover, we dive deeper into the working of its convolutional layers and verify the robustness of convolutional-features through layer visualizations. We also propose a method to synthesize artificial text images to do away with the need of annotating large amounts of training data. We outperform previous state-of-the-art methods on existing benchmarks by quite some margin and release two new benchmark datasets for Arabic Scene Text and Urdu Printed Text Recognition to instill interest among fellow researchers of the field.

Contents

Chapter Page

List of Figures

Figure Page

List of Tables

Table Page

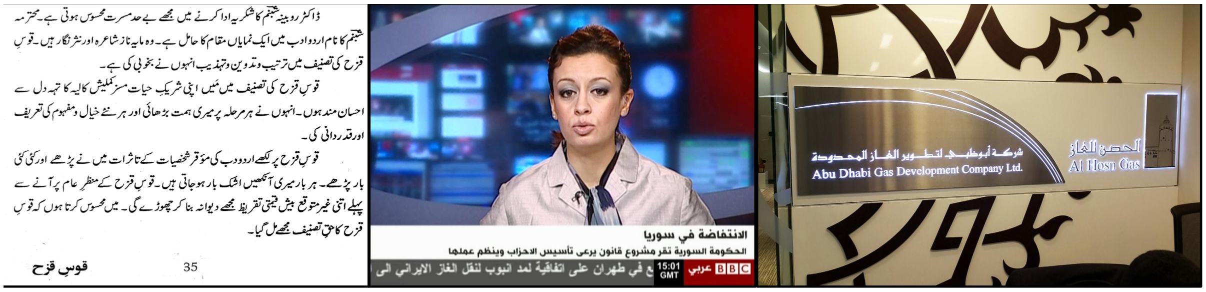
Chapter 1

Introduction

Research in text recognition for Arabic and Urdu has mostly been centered around the problem of printed text recognition; popularly known as Optical Character Recognition (OCR) [1, 2, 3, 4]. Lack of annotated data and inherent complexities of the script and language were the major challenges faced by the research community. Most of the early machine learning approaches which were effective for English OCR, were not easily adaptable to the Arabic or Urdu problem-setting for this reason. Even today, when modern machine learning methods could be used in a language/script agnostic manner, lack of annotated data remains a major challenge for Arabic and Urdu [5].

The problems of scene text and video text recognition deal with the recognition of text appearing in natural scene images and text embedded in video frames, respectively. Traditional OCR systems expect the images to be black-and-white with the text appearing in a proper structured fashion as seen in documents. Text images in scene and video text recognition tasks are in contrast to such assumptions and can have a huge variance in terms of font style, size, lighting conditions, perspective distortion, background noise and occlusions. Hence, the insignificant amount of research focus that Arabic and Urdu OCR received couldn't be scaled directly to more complex problems of scene and video text recognition.

The computer vision community experienced a strong revival of neural networks based solutions with the birth of Deep Learning in recent years. This process was stimulated by the success of models like Deep Convolutional Neural Networks (DCNNs) in feature extraction, object detection and classification tasks [6, 7]. However, these tasks only cater to problems where the subjects appear in isolation, rather than appearing as a sequence. Such problems often require solution models to predict a series of description labels instead of a single label. DCNNs are generally well suited for tasks where the inputs and outputs are bounded by fixed dimensions and hence are not well suited for such sequential recognition tasks. Moreover, any variation in the length of these sequence-like inputs and outputs further escalates the difficulty level. Recurrent Neural Networks (RNNs) tackle the problems faced by DCNNs for sequence-based learning by performing a forward pass on the network for each segment of the sequence-like input. Such models often involve a pre-processing or feature extraction step, where



ڈاکٹر روبینہ شبنم کا شکریہ ادا کرنے میں مجھے بے حد مسرت محسوس ہوتی ہے۔محترمہ
شبنم کا نام اردو ادب میں ایک نمایاں مقام کا حامل ہے۔ وہ مایہ ناز شاعرہ اور نثر نگار ہیں۔قوسِ
قزح کی تصنیف میں ترتیب وتدوین وتہذیب انہوں نے بخوبی کی ہے۔

قوسِ قزح کی تصنیف میں مَیں اپنی شریکِ حیات مسز کملیش کالیہ کا تہہ دل سے
احسان مند ہوں۔انہوں نے ہر مرحلہ پر میری ہمت بڑھائی اور ہر نئے خیال ومفہوم کی تعریف
اور قدر دانی کی۔

قوسِ قزح پر لکھے اردو ادب کی مؤقر شخصیات کے تاثرات میں نے پڑھے اور کئی کئی
بار پڑھے۔ ہر بار میری آنکھیں اشک بار ہوجاتی ہیں۔قوسِ قزح کے منظرِ عام پر آنے سے
پہلے اتنی غیر متوقع بیش قیمتی تقریظ مجھے دیوانہ بنا کر چھوڑے گی۔ میں محسوس کرتا ہوں کہ قوسِ
قزح کا حقِ تصنیف مجھے مل گیا۔

قوسِ قزح 35

الانتفاضة في سوريا
الحكومة السورية تقر مشروع قانون يرعى تأسيس الاحزاب وينظم عملها
ع في طهران على اتفاقية لمد انبوب لنقل الغاز الايراني الى
15:01 GMT
BBC عربي
شركة أبوظبي لتطوير الغاز المحدودة
Abu Dhabi Gas Development Company Ltd.
الحصن للغاز
Al Hosn Gas

Figure 1.1: Sample images showcasing the problem statements of Urdu OCR (*left*), Arabic video text recognition (*center*) and Arabic scene text recognition (*right*). Notice how as we move from left-to-right, the inherent problem complexity of recognizing text increases.

the input is first converted into a sequence of feature vectors [8, 9]. Such feature extraction stages are independent of the RNN-pipeline and hence they are not end-to-end trainable.

In this thesis, we show how *state-of-the-art* deep learning techniques which are successful for English text recognition can be easily adapted to low-resource languages, like Arabic and Urdu, in a language-agnostic fashion. The solution proposed is not bounded by any language specific lexicon with the model following a segmentation-free, sequence-to-sequence transcription approach. For the difficulties related to lack of annotated data, we propose an approach of synthesizing large amounts of data for Urdu OCR and Arabic scene and video text recognition. The synthesized data resembles real-world scenarios closely and helps outperform previous solutions on transcription accuracy.

1.1 Motivation

Digitizing historical documents is crucial in preserving our literary heritage. With the availability of low cost mobile capturing devices, institutions all over the world are preserving their literature in the form of scanned documents. Huge amounts of valuable Urdu literature from philosophy to sciences is vanishing and being rendered useless because it has not been digitized till now. A major barrier to this process of digitization is the huge overhead introduced in indexing and retrieval of such documents. All these problems indicate towards the strong need for developing a robust OCR system for Urdu. While Arabic text recognition has started to grab some attention of the research community [1, 10], works for Urdu text recognition are very limited and severely lack quality. This is quite surprising considering the fact that Urdu is the national language of Pakistan and is considered as one of the 22 official languages of India. There are over a 100 million speakers of Urdu in the world [11]. Moreover, many of the native speakers of this language can only read and write in Urdu and hence there's a scarcity of information and data on the internet and in digitized form for them.

The upsurge in video sharing on social networking websites and the increasing number of television channels in today's world reveal videos to be a fundamental source of information. Effectively managing, storing and retrieving such video data is not a trivial task. Due to the huge memory overheads associated with storing videos, using descriptors which depict the contents of a video help perform analytic and storage tasks efficiently. Video text recognition can greatly aid video content analysis and understanding, with the recognized text giving a direct and concise description of the stories being depicted in the videos. In news videos, superimposed tickers running on the edges of the video frame are generally highly correlated to the people involved or the story being portrayed and hence provide a brief summary of the news event. Recognition of text embedded in video frames is generally not as trivial as compared to OCR due to variability in terms of font styles, colors, sizes and complex background structures.

Reading text in natural scenes is a relatively harder task compared to printed text recognition. The problem has been drawing increasing research interest in recent years. This can be partially attributed to the rapid development of wearable and mobile devices such as smart phones, Microsoft hololens, Oculus rift and self-driving cars, where scene text is a key module to a wide range of practical and useful applications. Typical printed text OCR methods do not generalize well to natural scene settings where factors like inconsistent lighting conditions, variable fonts, orientations, background noise and image distortions add to the complexity of the problem. In Fig. 1.1, we see a typical structure of a scanned page from a book on the left-section of the figure. It contains a page number, paragraphs with horizontal text lines evenly spaced and a single font type. Such characteristics can be exploited by text recognition systems to perform accurate transcriptions. However, we see that for the video text sample in the same figure (*middle region*), there is variability in terms of font style, size and color. Morover, the text is now randomly placed across the video frame. Moving further right, we see the scene text sample image which now has all sorts of noise factors in the form of lighting, perspective projections, occlusion, etc. Owing to these variations, a direct application of solutions proposed for text OCR is not feasible.

The lack of large quantities of structured and publicly available data has been a major cause for Arabic and Urdu text recognition community lagging behind it's Latin and Chinese counterparts [12]. Most works reported results on small datasets created privately and never made available publicly for other researchers to compare their solutions [13]. Hence, the accuracy numbers reported by most papers on text recognition is very high, even though the solutions may not scale to other similar problems. Most of the testing was done on very small datasets which were again curated for a specific task like recognizing text from bank cheques, postal address, etc [14]. For a fair comparison of Arabic and Urdu text recognition systems, a standard bench-marking dataset is of utmost importance. Such a dataset would automatically help the community to correctly rank all proposed solutions and establish a *state-of-the-art* solution.

1.2 Contributions

In this work, we start by providing solutions to the problems of Urdu printed text recognition, commonly referred to as Urdu OCR. Then we move to relatively harder problems of Arabic video text and scene text recognition and show how state-of-the-art deep learning research for English can be successfully adapted to Arabic and Urdu tasks, in a language-agnostic fashion. The major contributions of this thesis are as listed below,

- **Literature Survey :** We provide an extensive survey on the development of text recognition systems for Arabic and Urdu in various problem settings over the years. We try to classify the various *school-of-thought*'s that originated as the field developed and provide supporting or counter arguments for the same. We hope that a consolidated excerpt of the developments in Arabic and Urdu text recognition would help the community better compare their solutions and get an understanding of the areas that have a substantial scope for improvement.

- **Synthetic Data Generation Pipeline :** We propose a synthetic data generation pipeline to train our models for Arabic scene and video text recognition. For low resource languages, like Arabic and Urdu, getting large quantities of annotated data is often difficult and hence we feel such a large-scale synthetic data generation pipeline can tremendously help improve the research in this field. The synthetically generated images closely resemble real-world examples and have been created from a large vocabulary of text spanning the entire character/ligature sets of the language.

- **Establish new *state-of-the-art* accuracy :** We beat the current state-of-the-art (SOTA) solutions for Urdu OCR and Arabic video text recognition using a Hybrid CNN-RNN network, which is used quite often in English text recognition tasks. Thereby, we show how SOTA deep learning works for Latin and Chinese scripts can be successfully adapted to low-resource languages like Arabic and Urdu. Our model achieves transcription accuracy considerably higher than the previous benchmarks for both Urdu and Arabic text recognition tasks.

- **Insights into Convolutional layers :** We provide insights, by creating layer visualizations, into the workings of the convolutional layers in our Hybrid CNN-RNN network and thus verify it's dominance in terms of robustness over the traditional methods of using raw-image features and hand-crafted features.

- **Public Benchmark Datasets :** To further facilitate this field of research, we make available two bench-marking datasets; IIIT-Urdu OCR dataset and IIIT-Arabic dataset for problems of Urdu OCR and Arabic scene text recognition, respectively. To the best of our knowledge IIIT-Urdu OCR is the first line-level real-world image dataset for Urdu OCR task. Similarly, IIIT-Arabic dataset is the first publicly available word-level real-image dataset for Arabic scene text recognition.

1.3 Thesis Layout

The flow of this thesis from here is as follows. First, in Chapter 2 we take a look at the techniques for Arabic and Urdu text transcription. We discuss the development and branching of this field into multiple sub-categories and provide a literature survey of the methods devised to solve these sub-branches. We also try to categorize the solutions into abstract categories and provide supporting and counter arguments for most approaches in Section 2.2. In subsection 2.4.1, we describe the workings of CTC layer which enables us to train our Hybrid CNN-RNN model in an end-to-end fashion. Our solution architectures are discussed in depth with all model and implementation details in Section 2.5.

The focus of Chapter 3 is on the Urdu printed text recognition task (Urdu OCR). After describing the problem statement and its intricate details in Section 3.1 and Section 3.2, we discuss the existing datasets for this task and introduce our IIIT-Urdu OCR dataset in Section 3.4. Finally, we conclude with the transcription accuracy of our models and the observations made in Section 3.5.

Moving over to the more complex problem of scene text and video text recognition for Arabic in Chapter 4, we introduce the problems and their respective difficulties in subsections 4.1.1 and 4.1.2. Next, we discuss the synthetic rendering pipeline used for training our scene text and video text models along with the current benchmark datasets for Arabic video text recognition in Section 4.4. We also throw light on the details of the IIIT-Arabic dataset we release for the Arabic scene text recognition task in this section. Finally, we showcase the transcription accuracy of our model on video text and scene text recognition tasks in Section 4.5 and provide deeper insight into the workings of convolutional layers of the Hybrid CNN-RNN network.

Chapter 5 concludes the discussions of this thesis by consolidating all the contributions made by our work. We also leave pointers on the possible extensions of this field in Section 5.2 for any interested researchers of the community to pursue.

Chapter 2

Arabic and Urdu Text Transcription

2.1 Introduction

Text recognition can generally be divided into two categories; Online and Offline [15, 16]. In online recognition, the characters/glyph are recognized while the user writes the text - usually on a digitized pen tracer with a special stylus pen [17, 18]. Whereas offline recognition deals with the recognition of text from scanned copies of printed or handwritten texts. A comprehensive survey with its focus as the differences between online and offline text recognition was done by [19].

Printed texts generally have the same font styles and sizes across prints, while handwritten texts can have varying font styles and sizes for the same writer as well as among various writers. For languages like Arabic and Urdu, where the script has a complex cursive nature and also shows ligature, character-level segmentation is often an arduous task. Hence, segmentation-free techniques gained quite the popular appeal. Like, [20] segment words from the input script and then compute the discrete-cosine transform (DCT) features on a normalized input image. These DCT features are then used to train a neural network which performs word-classification. Another segmentation-free approach was suggested by [2], who describe a 1D HMM offline handwriting recognition system employing an analytic approach of extracting baseline dependent features from binarized input images.

In this chapter, we throw light on the process of development of text recognition systems for Arabic and Urdu, with focus on the various stages of the recognition pipeline. We also provide a literature survey discussing the previous works in this domain.

2.2 Literature Survey

Text recognition generally involve several steps for performing accurate transcriptions. Fig. 2.1 illustrates the various steps involved in a typical text recognition system. Each of these steps and the related work for Arabic and Urdu in those domains have been discussed in the subsections that follow.

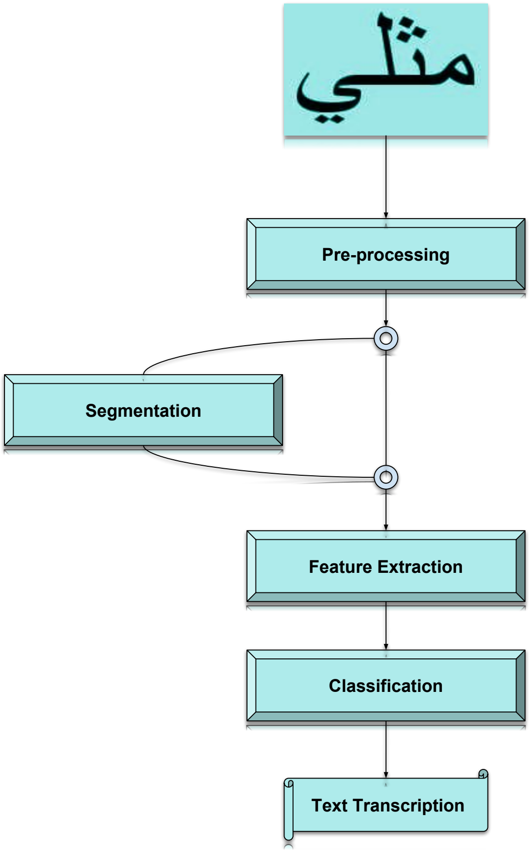


Figure 2.1: Flowchart representing the various stages of a text recognition system. Notice how the segmentation-block falls in an optional state depending on the type of solution approach utilised.

2.2.1 Pre-Processing Stage

Pre-processing and feature extraction are very important steps in automatic word recognition for cursive scripts [21]. This step is fundamental to improve discriminating nature of the pixels or raw features being computed from input images. There has been a lot of work done in the field of improving the pre-processing stage for Arabic and Urdu text recognition [22, 23, 24, 25, 26, 27] as it can turn out to be a bottleneck for the entire recognition process, specially since a large number of diacritics and dots are observed in these languages. However, we can generally categorize these efforts into four broad categories, namely; *binarization, noise removal, baseline detection* and *normalization*.

- **Binarization** is the process of converting colour or RGB images to a binary bit-map, generally with white pixels as background and black pixels as text. Binarization of text images has been an active field of research as it provides major speedups in computation [14]. Like, [28] suggest a Markov Random Field (MRF) model based method for scene text image binarization, inspired from the success of MRF models in object segmentation tasks.

- **Noise Removal** deals with the process of erasing pixels from the input image which hold negligible discriminative power. Such noise is generally added to the image during the scanning process. Some common methods to remove such unwanted noise utilize spatial and temporal filters using morphological operations like *opening, closing, erosion* and *dilation* to perform image operations like contour smoothing, contour or boundary extraction, text stroke reconstruction, etc. [29, 30, 31, 32, 33, 34]

- **Baseline Detection** helps gather lot of structural information from Arabic or Urdu text images such as dots and their positions, predecessor and successor. It also helps in correcting the slant and skew deformations. The most common way of predicting baseline is horizontal projection and it works exceptionally well for the OCR task [2, 22, 26, 35, 36, 37]. Other baseline detection methods utilize either contour information [23] or *Principal Component Analysis* (PCA) [38] to assign each image pixel into foreground or background.

- **Normalization** refers to the process of reducing the variation across text appearing in various images. The variation may be in terms of font sizes and styles or in terms of skew or rotation added during the scanning process of printed media. The most common normalization tactic is to resize character or word images to the same size [39]. Another derived approach is to divide the text image in multiple regions and then scale each of these regions separately [40]. For the slant/skew correction task, [41] suggest a method using Radon transform along with the image gradient for detecting slant angles.

2.2.2 Segmentation

For complex inflectional languages, like Arabic and Urdu, where the scripts are intricate, segmentation is often quite challenging. Segmentation of characters/ligature generally requires accurately finding its starting and ending point in the text stroke. Being prone to errors, the segmentation stage is generally a bottleneck for performing text transcription. Some of the common methods of performing text character segmentation are as follows,

- **Vertical Projection Techniques :** work on the assumption that most *connector-strokes* between characters of Arabic are usually thinner than their corresponding character-parts, when viewed from a vertical projection (1D view). Hence, by simply checking the pixel density along vertical lines, character end-points can be detected. Multiple algorithms incorporate this idea to segment words, ligatures and characters [26, 37, 42, 43]. However, these methods fail miserably when the writing style incorporated has a slant/skew.

- **Skeleton Extraction and Contour Tracing Techniques:** Accurate extraction of the text skeleton from an image can provide lot of insightful information. To further refine the extracted skeleton, many approaches [44, 45, 46] processed thinned versions of the texts to extract interesting key-points like edge-points, end-points or perform segmentation. Similarly, tracing the contours of main text body also helps in performing segmentation. Using contour tracing along with a set of topological rules, [47] propose an Arabic character segmentation rule by deciding whether the local minima obtained from contour tracing is actually a segmentation point or not.

- **Morphological Techniques :** use the operations of *opening, closing, erosion, dilation*, etc. on an image to perform character segmentation. By means of simple mathematical operations like addition and subtraction of morphologically operated image versions, segmentation points can be identified [48].
- **Neural Network Techniques :** are generally used to validate the correctness of segmentation candidates created by above techniques. ANNs have been trained by manually curating annotated data of valid and invalid segmentation candidates using above segmentation techniques [49, 50].

However, almost all of the above discussed techniques cannot solve the problem of overlapping characters, which occur quite often in Arabic and Urdu. Hence, there is still quite a lot of scope for improvement in this block of the text recognition pipeline.

2.2.3 Feature Extraction

Feature extraction is a key component for the classification stage. By extracting meaningful and robust features, we capture the intrinsic characteristics of the script which differentiate one character from the other. The classification stage can then make use of these features and perform accurate classification. However, the process of feature extraction is highly variable in terms of the problem being solved. Each problem has its own set of properties that need to be captured from the image and hence a feature that works for one problem might fail for another [51]. The categorization of most common feature extraction techniques is as follows,

- **Structural Features :** are generally formed using local and global properties of text image to capture the geometrical properties of an image. These are the most common types of features for performing text recognition [34, 52, 53, 54]. Some simple features for our Arabic and Urdu text recognition cases can be the position of diacritics and dots in the absolute pixel coordinates or relative to the baseline of text body, the weight of strokes, number of connected components or loops, etc.
- **Statistical Features :** try to fit a mathematical function over the spatial distribution of pixels in the text image. Generally, the function is built by deriving a set of statistical features at each image pixel [34]. The most common statistical feature for text recognition is zoning, where the characters are divided into overlapping and non-overlapping regions and analyzed for pixel density [55]. In a popular work, [56] divide the image into zones and measure the direction of text contours in each region. Thereafter, histograms of chain-codes define the direction of contour for each region which acts as the statistical feature for that region.

- **Global Features** are generally computed by using transformation techniques to move the image-features into a different vector space where the image signal can be described in a concise/compact format. This process generally involves representing the signal as a linear combination of simpler functions with the coefficients being given by the expansion of a linear combination [14, 29]. The most common global transformation techniques are *Fourier Transform* [57], *Discrete Cosine Transform* (DCT) [20], *Wavelets* [12], *Hough Transform* [58] and *Moments* [26].

2.2.4 Classification

The classification stage takes in features from the feature extraction stage and tries to assign a label to it from the given set of classes. The classification stage generally requires a training step where annotated input-output mappings are provided. This trained model is then used to predict the correct label class for a new input sample. The most popular methods for the classification stage are *K-nearest neighbour* (KNN) [59], *Hidden Markov Model* (HMM) [60, 61, 62] and *Artificial Neural Network* (ANN) [63, 64, 65].

2.3 Arabic and Urdu Text Databases

To develop and compare text recognition systems, existence of standard databases is essential. However, due to the focus of vision community being on English and Chinese scripts, insignificant amount of work has been done for Arabic and Urdu. Owing to a lack of standard benchmarks, most of the research in this field is done on private datasets without fair comparison. Hence, most works showcase high accuracy results while they may not scale to a large set of problems. We compile an extensive list of publicly available datasets in this subsection.

- **IFN/ENIT Database** [36] is the most popular Arabic handwritten words dataset. It contains 26,459 handwritten names written by 411 different writers representing 937 Tunisian town and village names. This dataset is available publicly for research purposes.

- **IFHCDB Database** [13] short for *Isolated Farsi Handwritten Character Database*, comprises of 52,380 character and 17,740 numerals scanned at a 300 dpi resolution and stored as 77x95 BMP images. This dataset is available publicly for research purposes.

- **AHDB database** [66] short for *Arabic Handwritten Database* contains numerals and entities used in cheques written in Arabic by 100 different writers.

- **Arabic Cheque Database** [67] is a handwritten cheques database containing 29,498 entities, 15,175 Indo-Arabic numerals and 2,499 samples each of legal and courtesy amounts curated from 3,000 real cheques.

- **ADBase and MADBase Dataset** [21] ADBase is a binary image dataset of 70,000 handwritten Arabic digits written by 700 different writers. MADBase is a modified version of ADBase following the conventions of MNIST dataset [68] to allow better comparison among Latin and Arabic scripts. It has grey-scaled images resized to 28x28 resolution. Both these datasets are available publicly for research purposes.

- **Handwritten Arabic Digit Database** [69] contains 21,120 scanned samples of digits written by 44 different writers. The images are saved in binary format along with horizontal and vertical histogram information to make available the digits locations.

- **Handwritten Arabic Character Database** [65] contains 15,800 Arabic character images written by about 500 writers. The handwritten pages were scanned at 300 dpi and saved as 7x7 resolution grey-scaled character images. However, this dataset isn't available publicly.

- **HACDB** [70] short for *Handwritten Arabic Characters Database*, contains 6,600 character shapes written by 50 writers of ages ranging 14-50 years. The focus of this database is to capture all possible shapes and ligatures that might occur in any Arabic text. This database is available publicly for research purposes.

- **UPTI Database** [71] short for *Urdu Printed Text Images* consists of 10,063 synthetically generated images of Urdu text lines. The dataset consists of both ligature and line versions. This dataset is available publicly for research purposes.

- **ALIF Dataset** [72] is a dataset of text embedded in video frames. It consists of 6,532 cropped text line images from 8 popular Arabic News channels. A *fine* subsection of this dataset has been annotated at the character level. This dataset is available publicly for research purposes.

- **ACTIV Dataset** [73] is a dataset similar to ALIF but is larger in size. It consists of 21,520 images containing text embedded in video frames extracted from famous Arabic news channels. This dataset is also available publicly for research purposes.

From the above mentioned datasets, it is quite clear that more complex tasks like Arabic video text or scene text recognition at the word or line level have not received much attention. To the best of our knowledge, there are no publicly available datasets for word/line-level Arabic or Urdu text recognition.

2.4 Text Transcription

There have been a few works for complex cursive scripts similar to Arabic (like Indic and Persian scripts) which addressed the problem of data scarcity by using synthetic data. A synthetic dataset comprising of 28K word images was used in [74] for training a nearest neighbour based Telugu OCR. The images were rendered from a vocabulary of 1000 words by varying font, font size, kerning and other rendering parameters. Early attempts to recognize text in cursive scripts often required a segmentation module which could segment words into sub-word units like characters or glyphs.

In [3], a structural technique for recognition hand printed Arabic characters using thinning techniques [75] on the text skeleton was proposed. The feature extraction stage for this work involved identifying lines, curves and loops in the contour of text. In a more sophisticated work, [76] proposed an ANN based Arabic printed text recognition system. The pre-processing and feature extraction stage for this work involved binarizing the image and extracting global features from the image, respectively. A segmentation based approach [77] for recognition of printed Arabic characters split the process of segmentation and recognition into two steps. The first step incorporates segmenting each character based on the angle formed at the intersection of adjoining characters. Thereafter, the diacritics and dots are removed from the segmented characters to keep the number of distinct classes to a minimum. Finally, structural features are extracted from the segmented character images and a decision tree performs classification.

However, the works in OCR started following segmentation-free approaches. Like, [78] proposed a segmentation free technique for Arabic OCR by performing morphological operations on text images and comparing them with existing symbol models. Using Fourier Transform coefficients from normalized polar images, [12] proposed a method to recognize multi-font cursive Arabic words. Recognition was achieved by performing template-matching using Euclidean distance as the loss metric.

There have been many works using Hidden Markov Models (HMMs) [79] and Recurrent Neural Networks (RNNs) [80, 81]. A hybrid approach combining the powers of HMMs and RNNs used Hough transform features to perform multi-font Arabic character recognition [58]. Using HMMs, [62] propose a printed Arabic text recognition system utilizing 16 features extracted from non-overlapping hierarchical windows. Among these methods, RNNs became quite popular for transcribing text words or lines directly into a sequence of class labels. LSTM networks used along with CTC loss (*subsection 2.4.1*) [82] enabled end-to-end training of a network which can transcribe from a sequence of image features to a sequence of characters. This approach did not require segmenting words or lines into sub-units, and could handle variable length images and output label sequences naturally. Employing this strategy, [9] perform English scene text recognition by feeding Histogram-of-Gradients (HOG) features to a RNN trained using the CTC loss.

The most recent advances in the text recognition domain for Latin and Chinese scripts follow the segmentation-free ideology. Moreover, the heavy-lifting task of creating robust and descriptive features is done by a Convolutional Neural Network (CNN). For an image captioning task, [83] present a model combining CNN and Bidirectional-LSTM that generates natural language descriptions of natural scene images, using a structured objective that aligns the two modalities. Using a similar network pipeline, [84] propose the CRNN model for Image-based sequence recognition. They incorporate feature extraction (using CNN), sequence modelling (using Bi-directional LSTM) and transcription (using CTC) in a unified framework which can be trained in an end-to-end fashion. Another novel solution [85] tackled the text recognition problem from an encoder-decoder setting. The proposed RARE model (Robust text recognizer with Automatic REctification) consists of a Spatial Transformer Network with LSTM based encoder-decoder network to perform seqeunce-to-sequence transcription. Our solution architectures follow suit to the CRNN type of solutions.

2.4.1 Connectionist Temporal Classification

Most real-world sequence learning tasks require that prediction of sequences of labels be made from noisy, unsegmented input data. In our case of text prediction, for example, an image is transcribed into words or sub-word units. RNNs are the obvious choice for such tasks given their powerful sequence learning capabilities. However, a major challenge in using them for sequence classification tasks is that they require pre-segmented training data and post-processing to transform their output features into a label sequence. Since RNNs are trained to consider each classification as an independent event, each input feature needs to be mapped to its corresponding output feature before training the network. The independently recognized labels are joined post-classification appropriately to obtain the complete output sequence.

Segmentation of words into their corresponding classes, specially for inflectional languages like Arabic and Urdu, is extremely challenging and requires lot of effort and language knowledge. Hence, we decide to use a temporal classifier known as Connectionist Temporal Classification (CTC) [82]. The underlying idea of CTC is to interpret the network outputs as a probability distribution over all the possible label sequences, conditioned on a given input sequence. This probability distribution can then be directly used to derive an objective function that maximizes the probabilities of the correct label assignments. Moreover, since the objective function is differentiable, a CTC layer can be plugged into any network and trained with standard back-propagation through time (BPTT).

The output activation functions are normalized such that the resultant on their summing up is one. This activation can thus be treated as a probability vector of the characters present at that position. The output layer associates one node for each class label and another special ’null’ character node, to indicate a ’no character’ label on inputs where no decision can be made. Thus, for K classes, there are K+1 number of nodes in the output layer. The CTC objective function is defined as the negative log probability of the network correctly labelling the entire training set. Given a training set (S) consisting of paired input and target sequences (x,z), the objective function (O) can be expressed as follows,

$$O = -\sum_{(x,z)\in S} ln\, p(z|x)$$

The advantage of having such discriminative objective functions is that we can directly model output label sequence probabilities given the input sequences. Such functions perform better than generative

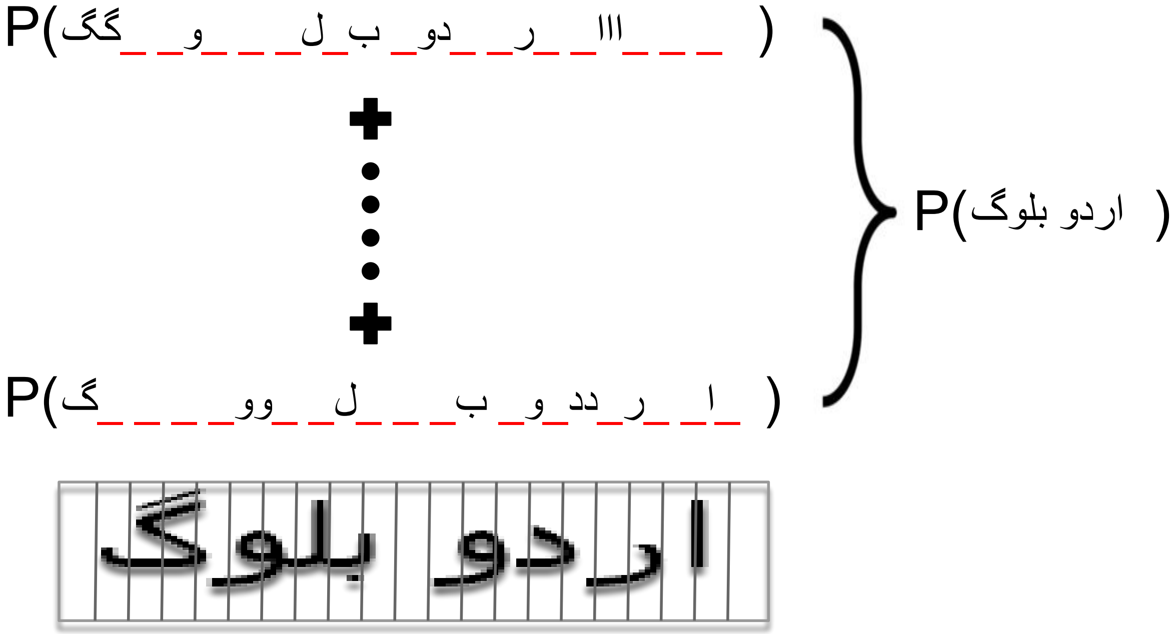


Figure 2.2: Visual representation of CTC loss in action. The illustration above shows CTC computing the probability of an output sequence ”Urdu Blog” (written in Urdu), as a sum over all possible alignments of input sequences that could map to it taking into account that labels may be duplicated because they may stretch over several time steps of the input data (represented by the split-image at the bottom of the figure).

function based systems like HMMs as shown in [86]. Moreover, HMM based systems assume that the probability of each observation is dependent only on its current state. Whereas, RNN based systems, specially with LSTM units, can easily model continuous trajectories and, in principle, extend the contextual information available over the entire input sequence.

2.5 Solution Architectures

The efficacy of a sequence-to-sequence transcription approach lies in the fact that the input sequence can be transcribed directly to the output sequence without the need for a target defined at each time-step. In addition, contextual modelling of the sequence in both directions, forward and backward, is of high utility for complex cursive scripts. Contextual information is critical in making accurate predictions for a language like Arabic or Urdu where there are many similar looking characters/glyphs, particularly vowel modifiers (dots above and below characters) and diacritics in Arabic and Urdu. Although we focus only on Arabic and Urdu in results and discussions, we provide an unconstrained solution architecture. By an *unconstrained* solution, we mean that our model is not bounded by any language-lexicon and any possible combination of the scripts character-set can be recognized. We discuss two solution

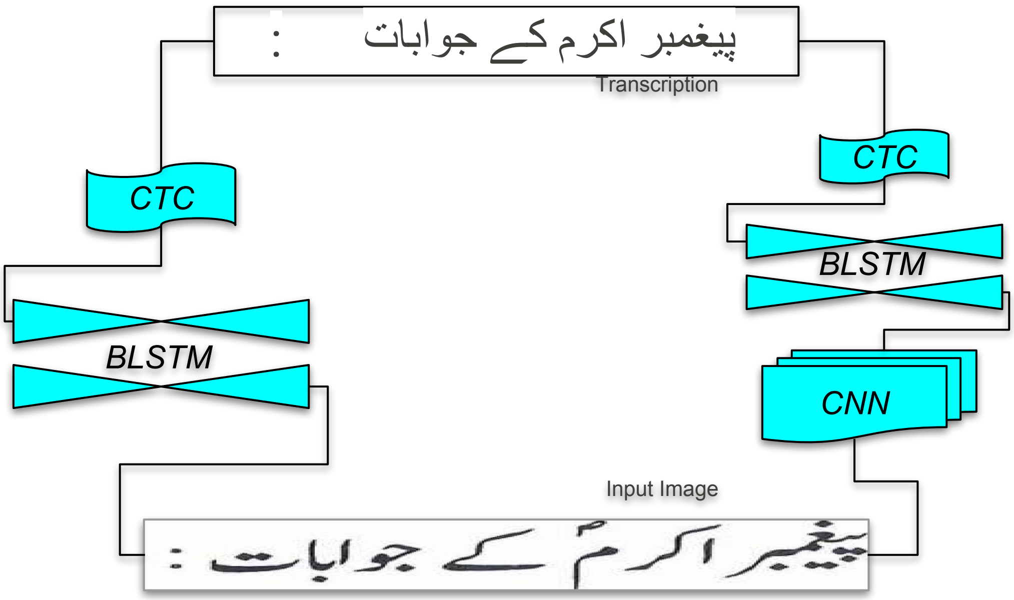


Figure 2.3: Flow diagram representing the two solution architectures; BLSTM (left) and HYBRID CNN-RNN (right) for an Urdu OCR task.

architectures for this text recognition setting in the following subsections. A visual comparison of the two methods can be seen in Fig. 2.3.

2.5.1 BLSTM Architecture

RNNs are an optimal choice for our unconstrained end-to-end system as they have a strong capability of capturing contextual information within a sequence. Additionally, RNNs are capable of handling variable length sequences. Since the number of parameters in a RNN is independent of the length of the sequence, we can simply unroll the network as many times as the number of time-steps in the input sequence. This helps us to perform unconstrained recognition, where the predicted output can be any sequence of labels from the entire label set (unique characters/glyphs and puntuation marks appearing in the Urdu and Arabic script).

Unfortunately, for standard RNN architectures (also called *vanilla* RNNs), the range of context that can be accessed at any given time step is limited. As the input activation cycles around the networks recurrent connections, it's influence on the hidden layer, and therefore on the network output, either decays or blows-up exponentially. This problem is often referred to as the vanishing gradient problem [87], making it difficult for an RNN to learn tasks containing delays of more than about 10 time-steps, in practice, between the relevant input and target events.

To tackle the problem of vanishing gradients, Long-Short Term Memory (LSTM) units are used, which were specifically designed to address the vanishing-gradients problem [88, 89]. Apart from the input and output, each LSTM unit has three multiplicative gates known as the *input*, *output* and the *forget gate*. The input gate decides when the current activation of the cell should be changed by taking input from network. Similarly, the output gate decides whether or not to propagate the node activation to the network. The forget gate helps in resetting activation value of the node. These gates help LSTM-based RNNs avoid the vanishing gradients problem. When the input gate has an activation value near to 0, no new inputs are made available to the node from the network. This ensures that the current activation value of the cell can be made available to the network at a much later stage without diminishing or exploding. In similar fashion, the output gate having a low activation value prevents the current activation of the node from getting released into the network. A visual comparison of *vanilla* and LSTM RNNs can be seen in Fig. 2.4.

In a text recognition setting, contexts from both directions (left-to-right and right-to-left) are useful and complimentary to each other in performing correct transcription. Therefore, a combined forward and backward oriented LSTM is used to create a *bi-directional* LSTM unit. Multiple such *bi-directional* LSTM layers can be stacked on top of each other to make the network deeper and gain higher levels of abstraction over the image-sequences as shown in [90].

The transcription layer at the top of the network is used to translate the predictions generated by the recurrent layers into label sequences for the target language. The CTC layer's conditional probability is used in the objective function as shown in [82]. This objective function calculates a cost value directly from an image and its ground truth label sequence, eliminating the need to manually label all the individual components in a sequence.

2.5.2 Hybrid CNN-RNN Architecture

A novel approach combining robust convolutional features and transcription abilities of RNNs was introduced for English scene text recognition by [84]. Our hybrid CNN-RNN solution is inspired from this work with changes made to cater for the intricacies of Arabic and Urdu scripts. The hybrid CNN-RNN networks have multiple convolutional layers stacked at the head of the BLSTM architecture described in the previous subsection. They consists of three major components; initial convolutional layers, middle recurrent layers and a final transcription layer but vary in the number of convolutional layers. The convolutional layers obtain robust feature representations from the input images. These features are then passed on to the recurrent layers which transcribe them into an output sequence of labels representing the Urdu characters/glyphs.

The convolutional layers follow a VGG [91] style architecture without the fully-connected layers. The input image first goes through the convolutional layers where feature maps are extracted from it. All the images are scaled to a fixed height before being fed to the convolutional layers. After the

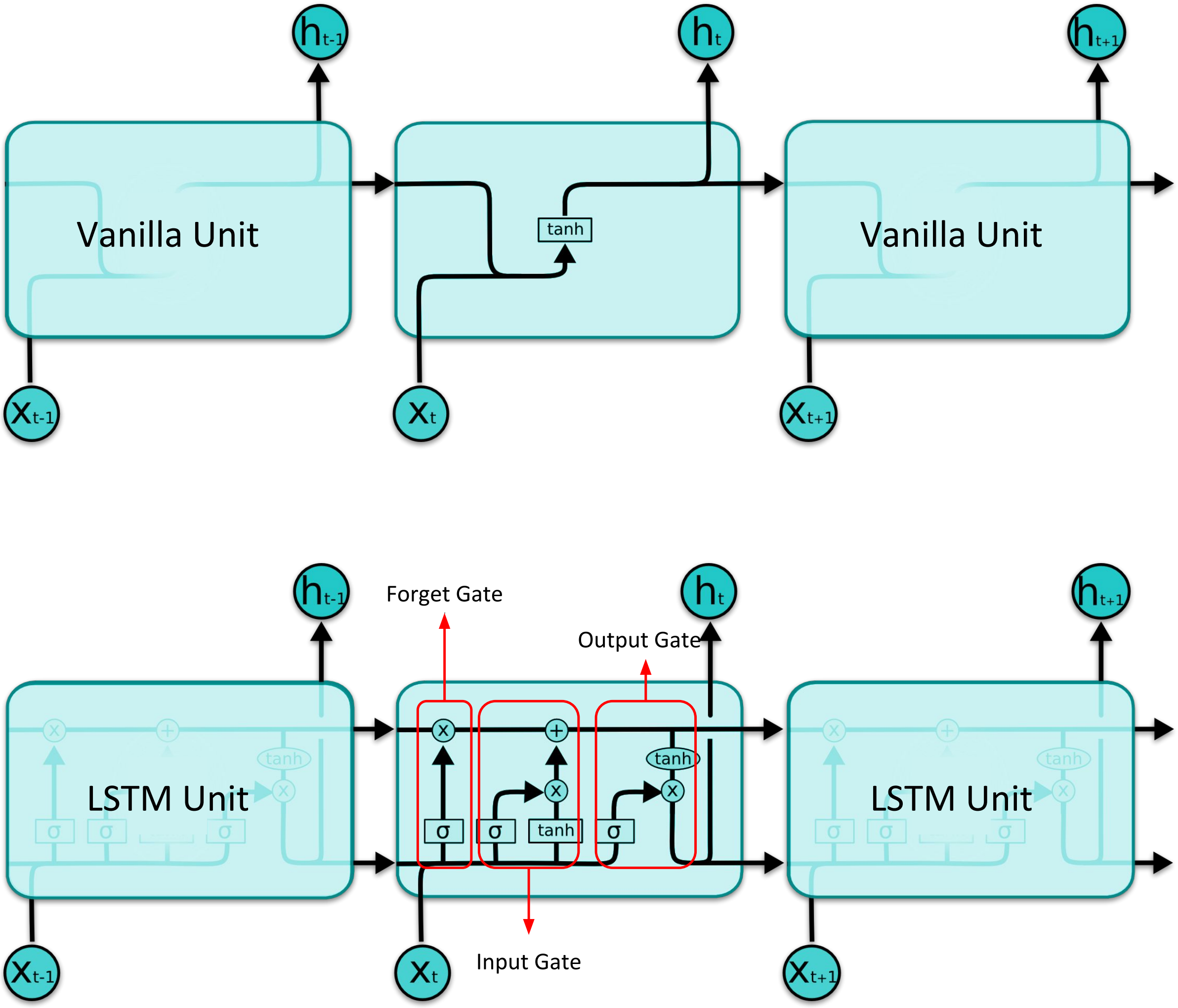


Figure 2.4: A standard (*vanilla* RNN network with a simple *tan h* non-linearity (*top*).
An LSTM units based RNN network (*bottom*). Notice the multiplicative input, output and forget gates.

convolutional operations, the sequence of feature maps obtained are split column-wise to create feature vectors which act as time-steps for the recurrent layers. These feature descriptors are highly robust and most importantly can be trained to be adopted to a wide variety of problems [6, 7, 92]. Since the convolution, max-pooling and element-wise activation function layers operate on local regions, they are invariant to translation. Hence, each column in the convolutional feature map obtained corresponds to a rectangular region in the input image. Moreover, these rectangular regions are in the same order as their corresponding columns on the feature maps from left to right. We can consider each rectangular patch on our input image to be the *receptive field* of its corresponding convolutional feature column vector as shown in Fig. 2.5

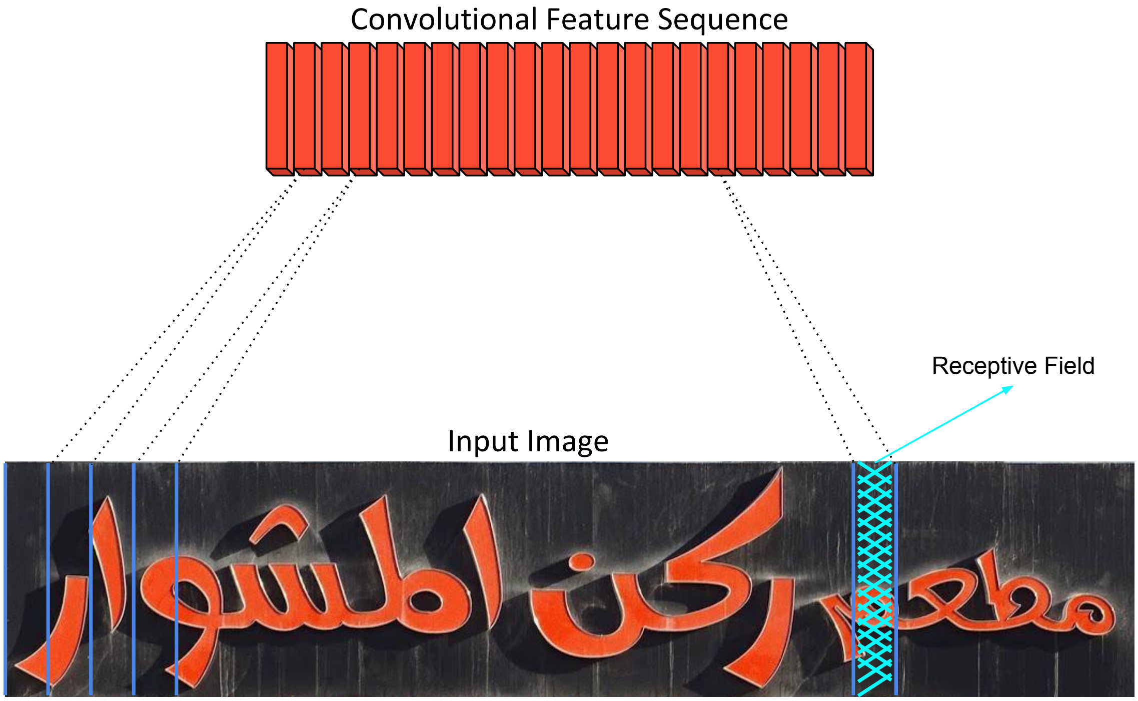


Figure 2.5: Receptive field of convolutional feature maps. Each column feature is associated with a corresponding column patch on the input image.

The deep bidirectional RNN is placed on top of the convolutional layers, as recurrent layers. RNNs have a strong capability of capturing contextual information within a sequence. Using such contextual cues for image-based text recognition is more stable and helpful than treating each character independently. Wider characters generally require more than one successive frame to be correctly described. Moreover, ambiguous characters are easier to distinguish when contextual information is available. This is particularly helpful for our case of Arabic and Urdu, or any other inflectional language, where intricacies of the script make separation of individual characters extremely difficult. Also, RNN layers can back-propagate the error differentials to convolutional layers and hence the model can be trained in an end-to-end fashion. They can also operate on input sequences of arbitrary lengths and hence are optimal for our case of performing unconstrained transcription. A visual representation of the BLSTM layers can be seen in Fig. 2.6

To summarize the overall network, the convolutional layers extract robust and descriptive features from raw input image. The recurrent layers take each feature vector from the feature sequence generated by convolutional layers and make predictions. The sequence-to-sequence transcription is achieved by using a CTC loss layer at the output. A visualization of this process with complete network configurations can be seen in Fig. 2.7.

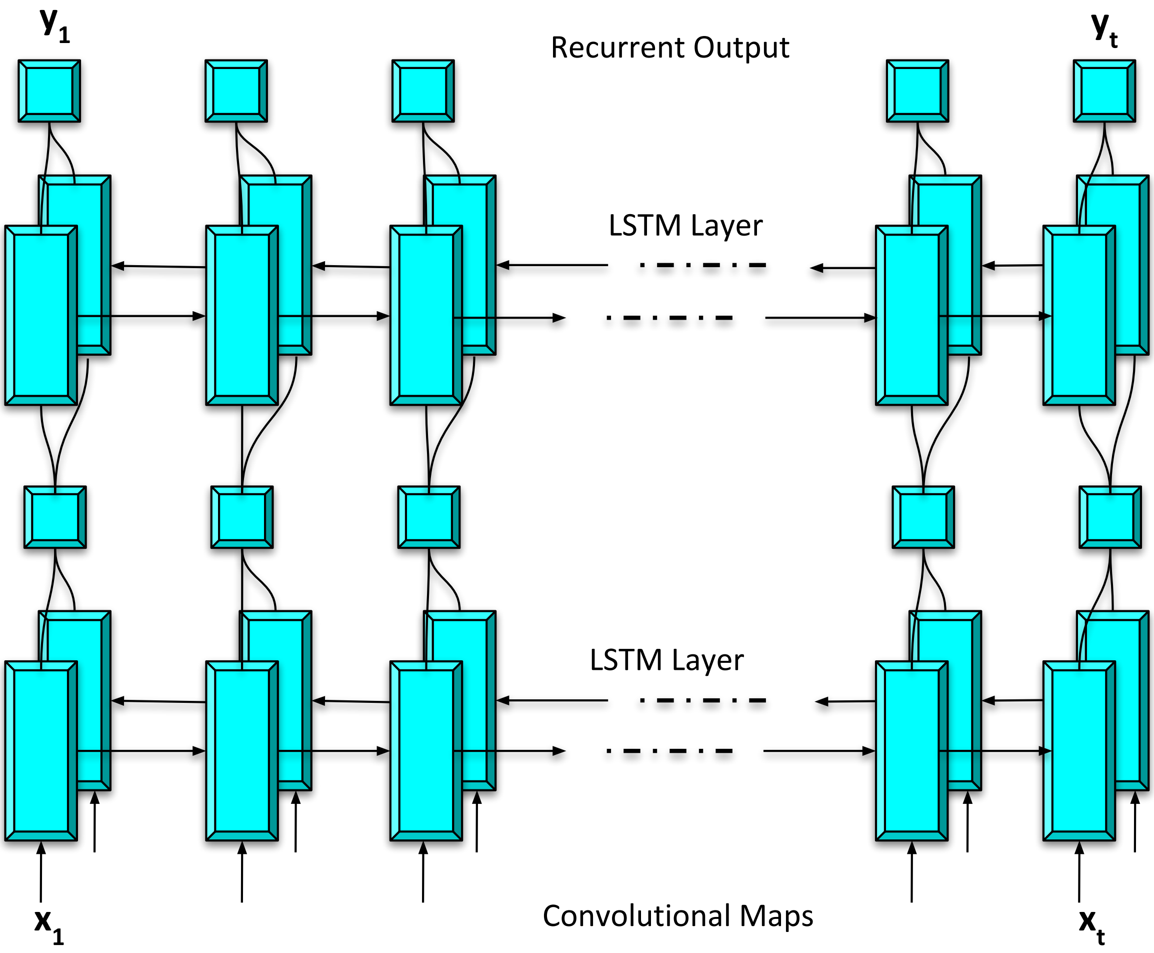


Figure 2.6: The deep BLSTM layers. Combining a forward (left-to-right) and a backward (right-to-left) LSTM yeilds improved transcription performance as contextual information from both directions is captured. Stacking multiple BLSTM layers results in a deep BLSTM layer.

2.6 Implementation Details

The convolutional blocks follow a VGG style architecture [91]. However, certain tweaks are made to the architecture to better cater to the language intricacies and type of problem being solved; OCR, video text or scene text. In the 3rd and 4th max-pooling layers, the pooling windows used are rectangular in shape, instead of the usual square windows as used in VGG. The added advantage of this tweak is that the feature maps obtained after the convolutional layers are wider and hence create longer feature sequences for the recurrent layers that follow. Moreover, rectangular windows yield wider receptive fields (illustrated in Fig. 2.5) which are beneficial for performing transcription among confusing characters/glyphs since contextual information is preserved better.

For our case of Arabic and Urdu transcription, since these languages are read from right-to-left, all input images are flipped horizontally before being fed to the convolutional layers. Moreover, the images are first converted into gray-scale. The convolutional stack is followed by a recurrent stack consisting of 2 BLSTM layers each having 512 hidden nodes. The second BLSTM layer is connected to a fully connected layer of size equivalent to the number of distinct labels in our language vocabulary and an additional label denoting the *blank* label for CTC transcription. Labels in our experiments are the Unicode representations of each character/glyph appearing in the language vocabulary used to create the training and testing datasets and basic punctuation symbols. Finally, a Softmax activation is applied to the outputs of the last layer and the CTC loss is computed between the output probabilities and the expected target label sequence.

To enable faster batch learning, all the inputs are re-scaled to a fixed height (32 pixels), while keeping the aspect ratio same, before being fed to the Hybrid network. Experimentally, we observe that re-scaling of images only marginally affects the transcription performance but a major speedup is observed in the time taken for training convergence. With a batch size of 64, training the Hybrid CNN-RNN architecture reached convergence in about 14 hours on a single *Nvidia-TitanX* GPU occupying less than 3GBs of GPU memory when the images fed in are all of fixed aspect ratio. However, incorporating zero-padding and allowing variable sized images shoots up the training time to 24 hours and occupies close to 24 hours to reach convergence and takes upto 8GBs of space on the same *TitanX* GPU.

To tackle the problems of training such deep convolutional and recurrent layers, we used the *batch normalization* [93] technique. Adding two batch-norm layers after the 5th and 6th convolutional layers respectively, accelerated the training process greatly. The network is trained with *stochastic gradient descent* (SGD) algorithms. Gradients are calculated by the back-propagation algorithm. Precisely, the transcription layers' error differentials are back-propagated with the forward-backward algorithm, as shown in [82]. While in the recurrent layers, the *back-propagation through time* (BPTT) [94] is applied to calculate the error differentials. The hassle of manually setting the *learning-rate* parameter is taken care of by using ADADELTA optimization [95].

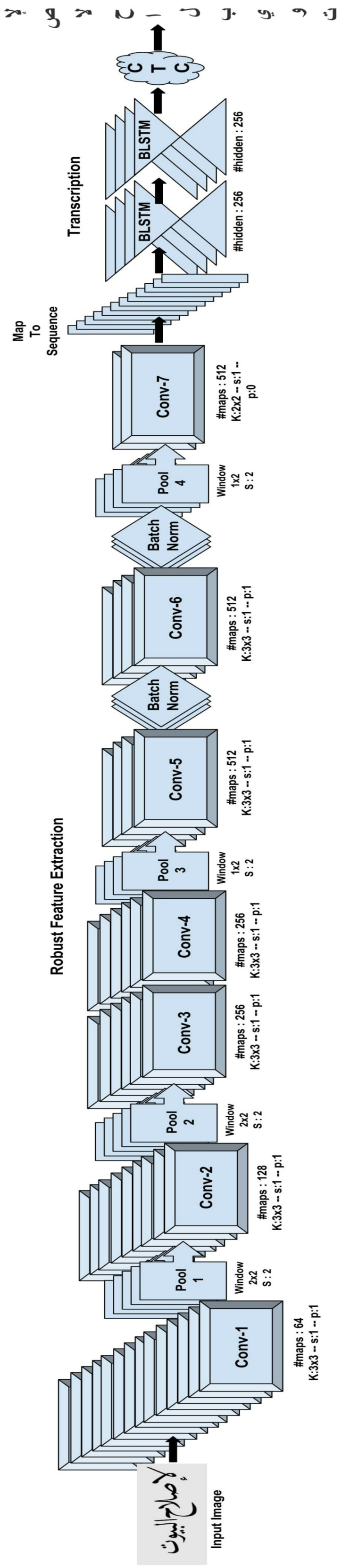


Figure 2.7: Visualization of the hybrid CNN-RNN architecture with a 7-layer-deep convolutional block. The symbols ‘k’, ‘s’ and ‘p’ stand for kernel size, stride and padding size respectively.

Chapter 3

Urdu Printed Text Recognition

3.1 Introduction

Building robust text recognition systems for languages with cursive scripts like Urdu has always been challenging. Intricacies of the script and the absence of ample annotated data further act as adversaries to this task. In this chapter, we demonstrate the effectiveness of an end-to-end trainable hybrid CNN-RNN architecture in recognizing Urdu text from printed documents, also commonly known as Urdu OCR. An example of Urdu OCR can be seen in Fig. 3.1. The solution proposed is not bounded by any language specific lexicon with the model following a segmentation-free, sequence-to-sequence transcription approach, as discussed in the previous chapter. We outperform previous state-of-the-art results on existing benchmark datasets and publish a new dataset curated by scanning printed Urdu publications in various writing styles and fonts, annotated at the line level. We also provide benchmark results of our model on this dataset.

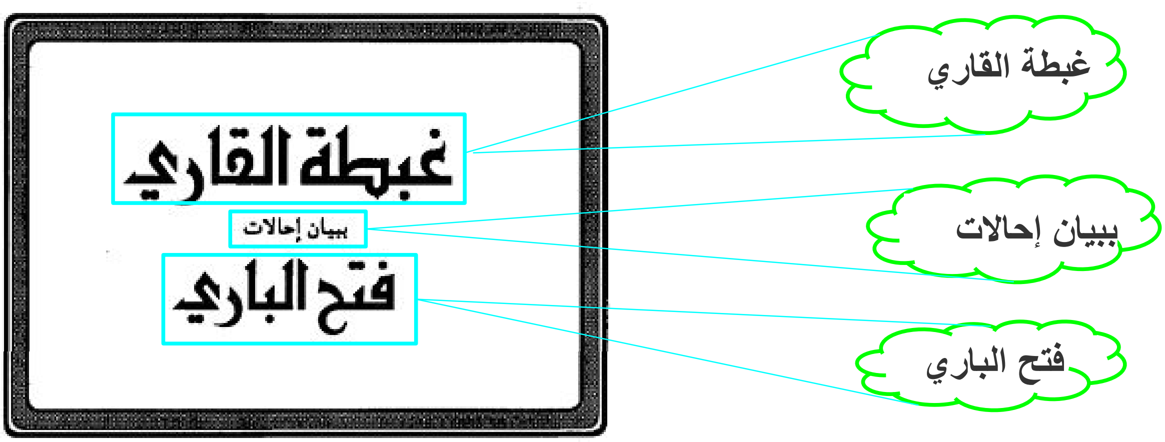


Figure 3.1: Example of Urdu Optical Character Recognition (OCR). Our work deals only with the recognition of cropped words/lines. Bounding boxes were provided manually.

3.1.1 Optical Character Recognition

Optical Character Recognition (OCR) refers to the process of converting a document image to its corresponding text. The document image can be acquired from various sources like newspapers, magazines, text books, novels, etc. Some major challenges that generally occur with this problem are the variation seen across various font styles and writing styles, font sizes, low resolution of characters and lack of large quantities of annotated data. There are several areas where OCR technology is of high utility. Vehicle number plate recognition for video surveillance, recognition of road signs for self-navigating driving, helping visually challenged people, automatic filling of forms, compression of digital image to Unicode text, etc.

Dividing the development cycle of OCR into generations, the first generation of Urdu OCRs used rule based solutions and intuitive features for the recognition of characters. The second generation of OCRs extended on the definition of characters, but now incorporated better principled features based on signal processing techniques and statistical modelling. Urdu OCRs using traditional machine learning relied on multi-class classification schemes implemented with neural networks or Support Vector Machines (SVMs). However, these solutions required the combination of two separate modules. The first module would comprise of the image segmentation stage that would create isolated character samples. The second module would convert the class labels into a Unicode sequence depending on the ordering of these characters.

Most Urdu OCR systems developed until recently would incorporate a segmentation step prior to recognition. However, due to the intricacies of Urdu script, discussed in the next subsection, creating a robust segmentation module has been the most difficult block in this pipeline. Posing the OCR task from a machine learning perspective, the problem can be thought of as a sequence-to-sequence translation task. Here, the input sequence is a series of feature vectors of varying length while the output sequence is series of characters/glyphs of arbitrary length. In the recent years, there has been a shift in ideology where segmentation-free methods for OCR, generally based on HMMs or RNNs, are better appreciated. Such methods generally follow a direct transcription of input features to output label sequence scheme.

3.2 Intricacies of Urdu Script

While Urdu can be written in various styles like *Naskh, Nastaliq, Kofi, Thuluth, Devani* and *Riqa*, the most commonly used writing styles are *Naskh* and *Nastaliq*. Printed media like magazines, newspapers and books generally follow the *Nastaliq* style of writing, whereas online material is mostly available in the *Naskh* style of writing. Both these styles are written in a semi-cursive fashion from right-to-left, similar to Arabic. However, a prominent distinction between the two styles is the flow in which these scripts are written. *Naskh* has a horizontal writing flow from right to left while *Nastaliq*'s flow is diagonal from right-top to left-bottom as seen in Fig. 3.2. Another peculiar characteristics of Urdu

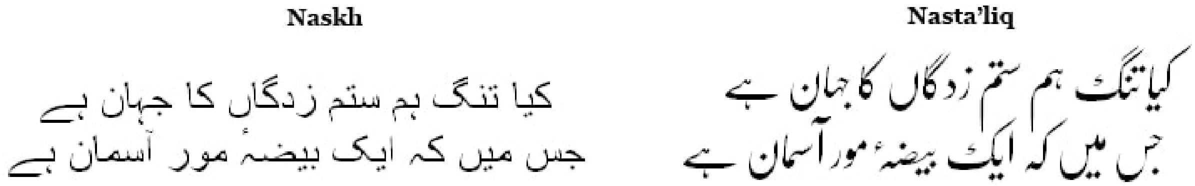

Naskh
Nasta'liq
کیا تنگ ہم ستم زدگاں کا جہان ہے
جس میں کہ ایک بیضۂ مور آسمان ہے

Figure 3.2: Two commonly used styles for Urdu scripts; *Nastaliq* (notice the diagonal flow from right-top to left-bottom) and *Naskh* (horizontal flow from right to left).

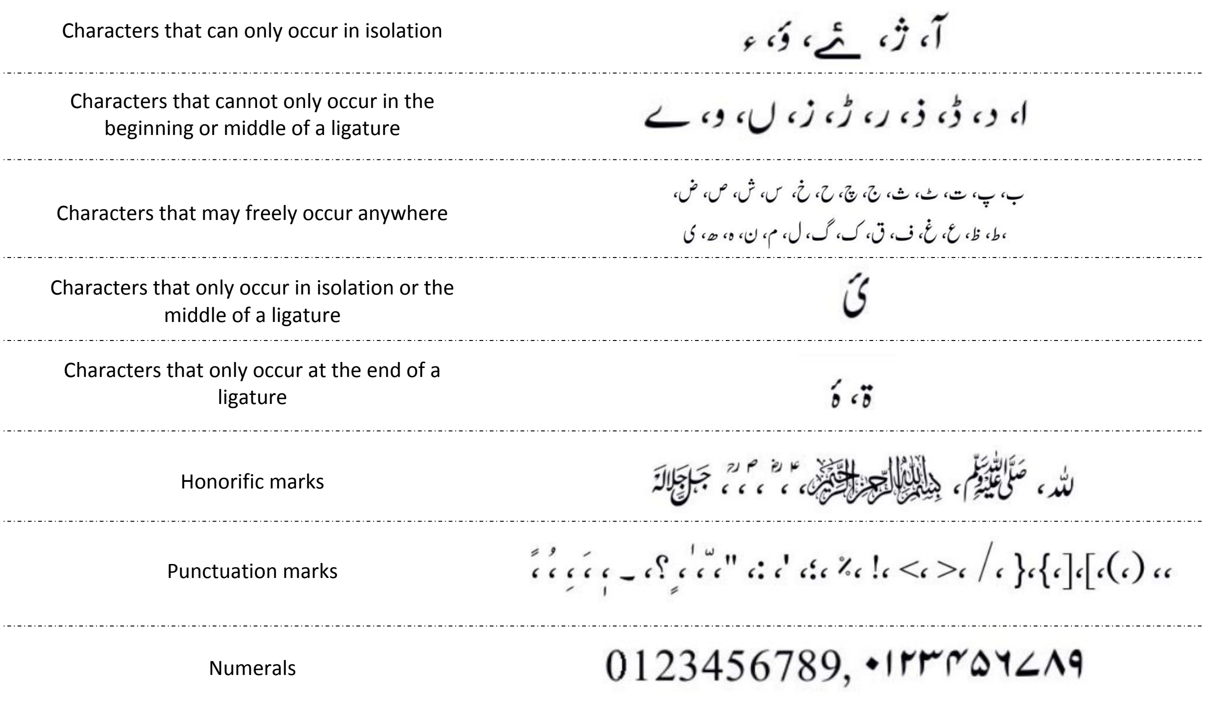
script is that unlike words, numerals are written from left-to-right. Since the problem of OCR caters to printed text in documents, we would primarily be dealing with the *Nastaliq* style of writing.

The script for Urdu uses 45 characters as their basic building blocks. Of these, 5 characters are bound to appear in isolation, 10 appear only at the beginning or at the end of a word and 1 character is limited solely to middle positions. The other 27 characters are free to occur in isolation or at the beginning, middle or end of a word. Additionally, there are 26 punctuation marks, 8 honorific marks and 10 digits that complete the character-set for Urdu. However, from on OCR systems perspective, English numerals and punctuation marks are also a common occurrence in the printed Urdu documents domain and hence need to be recognized by any practical solution. Also, the position of a character in a word (at the beginning, middle or end of the word) changes the shape of the glyph used to represent the character completely. Accounting for all the above mentioned variations, there are a total of 192 distinct glyphs that might occur in an Urdu publication. The segregation of characters based on the inherent script rules can be seen in Fig. 3.3.

Non-standardization of fonts and their rendering schemes, especially for the publications printed prior to the emergence of Unicode, has made the development of an Urdu OCR further challenging. Moreover, due to the mismatch between the basic units for representation and rendering (Unicode characters v/s various fonts and writing styles), creation of synthetically rendered data samples to employ fully supervised machine learning methods is all the more difficult.

3.3 Related Work

Recognizing cursive scripts has been an active field of research. Initial works in this domain like [96] presented an OCR solution for languages with large character sets like Japanese and Chinese. They used an approximate character shape similarity on top of a word segmentation algorithm using language models. Later on, [97] proposed a segmentation based approach for OCR using SVMs for classification of individual segmented units into character label classes. They computed local and global features on top of these segmented cursive characters.



Characters that can only occur in isolation	آ، ژ، ئے، ؤ، ء
Characters that cannot only occur in the beginning or middle of a ligature	ا، د، ڈ، ذ، ر، ڑ، ز، ں، و، ے
Characters that may freely occur anywhere	ب، پ، ت، ٹ، ث، ج، چ، ح، خ، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ک، گ، ل، م، ن، ہ، ھ، ی
Characters that only occur in isolation or the middle of a ligature	ئ
Characters that only occur at the end of a ligature	ۃ، ۂ
Honorific marks	ﷲ، ﷺ، ﷽، ؑ، ؓ، ؒ، ؔ، جل جلالہ
Punctuation marks	،، (،)، [،]، {، }، /، >، <، !، ٪، ؛، '، :، "، ٓ، ٰ، ٍ، ؟، ـ، ٖ، ِ، َ، ُ، ً
Numerals	0123456789, ۰۱۲۳۴۵۶۷۸۹

Figure 3.3: Character categorization based on Urdu script rules and ligature behaviours.

Urdu OCR still remains in a nascent stage as compared to other cursive scripts used in the Asian continent. Among initial works, [98] used morphological operations and character-specific filters to pre-process each segmented character/glyph from a line image. They used a heuristics based approach on the character-chain-code created to figure out which class label (Urdu glyph) the segmented image must be assigned to.

Most of the research in the domain of Urdu OCR utilised handcrafted features and used nearest-neighbour techniques to perform text classification. Using connected components information and extracting stroke information by detecting the baseline, [99] proposed an Urdu OCR system. Features were also obtained by passing sliding windows of various filters over the raw input image. Similarly, [71] used contour extraction extraction techniques and utilised contextual information derived from glyph-shape to create feature descriptors for each ligature/glyph. Finally, classification was done using k-Nearest Neighbour algorithm in both [99] and [71].

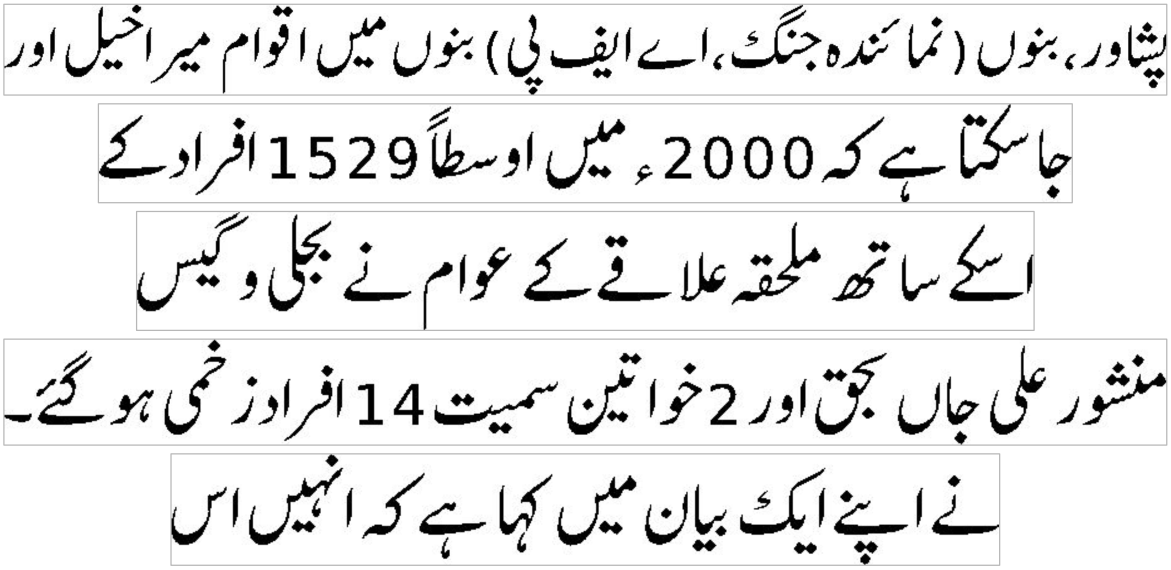
Segmentation-free methods have come into light only recently. These methods are generally based on Hidden Markov Models (HMMs) or Recurrent Neural Networks (RNNs). In one such work, [100] train multiple HMMs for each type of ligature and body-text of the characters. These models are used to create a feature matrix based on the number and positions of diacritics to perform classification on the text in a segmentation-free fashion. Some methods improved transcription accuracy with the help of language models [101]. They used uni-gram, bi-gram and tri-gram counts for words and ligatures to rank possible word predictions based on probabilities derived from a lexicon of the most frequently used Urdu words.

3.4 Datasets

When dealing with low-resource and inflectional languages like Urdu, curating large amounts annotated data is a challenging task. Lack of mainstream focus by the vision community on such languages makes the scenario worse. In this section, we discuss the various existing datasets for Urdu OCR and describe in detail the data used for training our solution architectures. Finally, we introduce the new IIIT-Urdu OCR dataset we release publicly for future researchers to test their solutions.

3.4.1 UPTI Dataset

The current benchmark for Urdu OCR task is the Urdu Printed Text Images (UPTI) dataset [71]. It consists of 10,063 synthetically generated Urdu text line images and their corresponding annotations. The dataset consists of both ligature-level and line-level versions, however, we only use the line-level version in our experiments for better comparison with other datasets. Sample images from the UPTI dataset can be seen in Fig. 3.4.



پشاور، بنوں (نمائندہ جنگ، اے ایف پی) بنوں میں اقوام میراخیل اور

جا سکتا ہے کہ 2000ء میں اوسطاً 1529 افراد کے

اسکے ساتھ ملحقہ علاقے کے عوام نے بجلی و گیس

منشور علی جاں بحق اور 2 خواتین سمیت 14 افراد زخمی ہو گئے۔

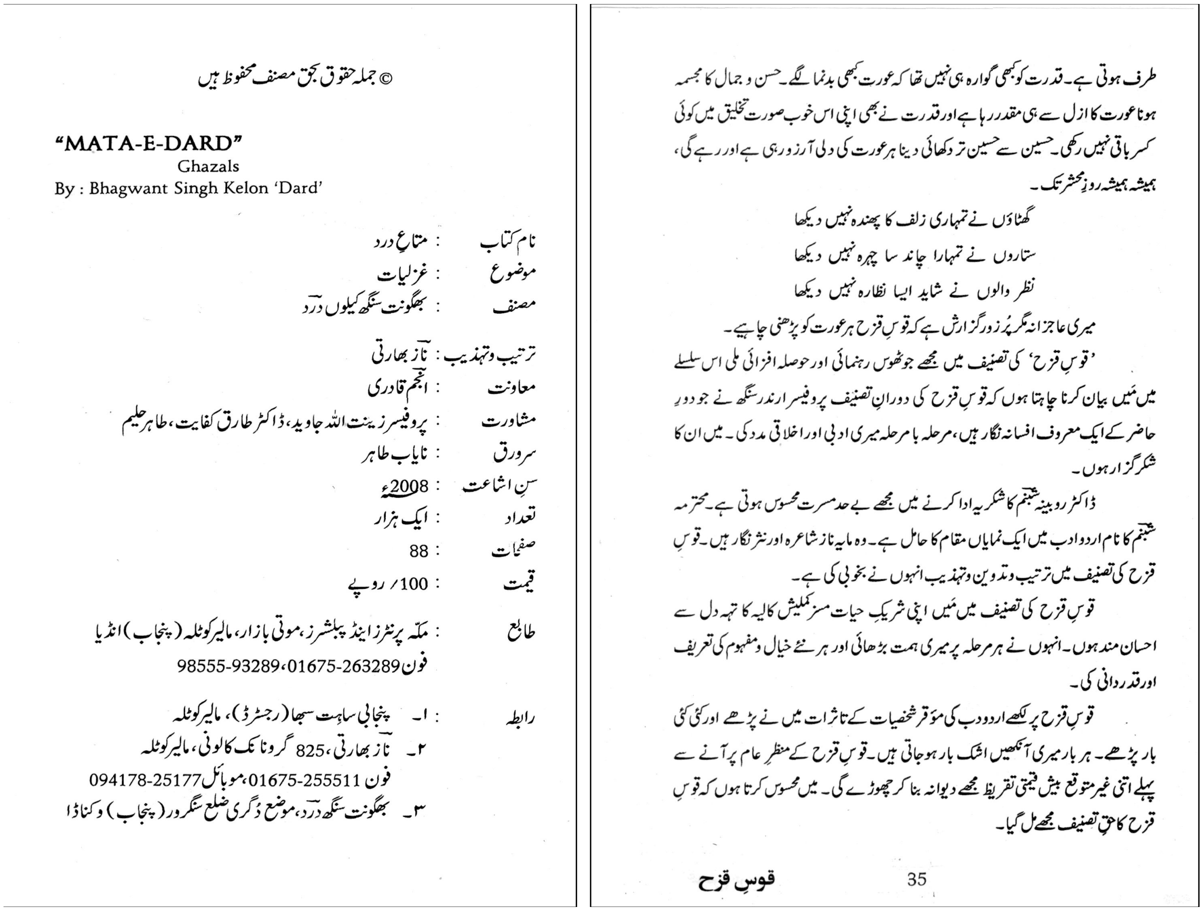
نے اپنے ایک بیان میں کہا ہے کہ انہیں اس

Figure 3.4: Sample images from the Urdu Printed Text (UPTI) Dataset. These images have been synthetically generated and annotations at ligature and line level are made available.

To better compare our transcription accuracy, we follow the data augmentation techniques used by [81]. The images are degraded using techniques described in [102] and split into 12 sets depending on the degradation parameters, namely, *elastic elongation, jitter, sensitivity* and *threshold*. Thereafter, the line images are divided into training (46%), validation (30%) and test (20%) sets by evenly distributing the clean and degraded images.

3.4.2 Consortium Project Data

In an attempt to implement an integrated platform for OCR of different Indian languages, [103] proposed a project discussing the software engineering, work-flow management and testing processes involved building a large scale system. As part of this consortium project, a dataset of Urdu printed media was collected and annotated. The dataset consists of approximately 1500 pages of scanned Urdu text annotated at the page-level by language experts. Bounding-boxes for the lines on these pages were also provided by means of manual human annotation. Post cleaning of data, a total of 29,876 Urdu text line images were obtained. Sample images from this dataset can be seen in Fig. 3.5



© جملہ حقوق بحق مصنف محفوظ ہیں

"MATA-E-DARD"
Ghazals
By : Bhagwant Singh Kelon 'Dard'

نام کتاب	:	متاعِ درد
موضوع	:	غزلیات
مصنف	:	بھگونت سنگھ کیلوں دردؔ
ترتیب وتہذیب	:	نازؔ بھارتی
معاونت	:	انجمؔ قادری
مشاورت	:	پروفیسر زینت اللہ جاوید، ڈاکٹر طارق کفایت، طاہر حلیم
سرورق	:	نایاب طاہر
سنِ اشاعت	:	2008ء
تعداد	:	ایک ہزار
صفحات	:	88
قیمت	:	100/ روپے
طابع	:	مکّہ پرنٹرز اینڈ پبلشرز، موتی بازار، مالیرکوٹلہ (پنجاب) انڈیا فون 01675-263289، 98555-93289
رابطہ	:	ا۔ پنجابی ساہت سبھا (رجسٹرڈ)، مالیرکوٹلہ ۲۔ نازؔ بھارتی، 825 گرونانک کالونی، مالیرکوٹلہ فون 01675-255511، موبائل 094178-25177 ۳۔ بھگونت سنگھ دردؔ، موضع ڈُگری ضلع سنگرور (پنجاب) وکناڈا

طرف ہوتی ہے۔ قدرت کو کبھی گوارہ ہی نہیں تھا کہ عورت کبھی بدنما لگے۔ حسن و جمال کا مجسمہ ہونا عورت کا ازل سے ہی مقدر رہا ہے اور قدرت نے بھی اپنی اس خوب صورت تخلیق میں کوئی کسر باقی نہیں رکھی۔ حسین سے حسین تر دکھائی دینا ہر عورت کی دلی آرزو رہی ہے اور رہے گی، ہمیشہ ہمیشہ روزِ محشر تک۔

گھٹاؤں نے تمہاری زلف کا پھندہ نہیں دیکھا
ستاروں نے تمہارا چاند سا چہرہ نہیں دیکھا
نظر والوں نے شاید ایسا نظارہ نہیں دیکھا

میری عاجزانہ مگر پُر زور گزارش ہے کہ قوسِ قزح ہر عورت کو پڑھنی چاہیے۔

'قوسِ قزح' کی تصنیف میں مجھے جو ٹھوس رہنمائی اور حوصلہ افزائی ملی اس سلسلے میں مَیں بیان کرنا چاہتا ہوں کہ قوسِ قزح کی دورانِ تصنیف پروفیسر اندر سنگھ نے جو دورِ حاضر کے ایک معروف افسانہ نگار ہیں، مرحلہ با مرحلہ میری ادبی اور اخلاقی مدد کی۔ میں ان کا شکر گزار ہوں۔

ڈاکٹر روبینہ شبنمؔ کا شکریہ ادا کرنے میں مجھے بے حد مسرت محسوس ہوتی ہے۔ محترمہ شبنمؔ کا نام اردو ادب میں ایک نمایاں مقام کا حامل ہے۔ وہ مایہ ناز شاعرہ اور نثر نگار ہیں۔ قوسِ قزح کی تصنیف میں ترتیب وتدوین وتہذیب انہوں نے بخوبی کی ہے۔

قوسِ قزح کی تصنیف میں مَیں اپنی شریکِ حیات مسز کملیش کالیہ کا تہہ دل سے احسان مند ہوں۔ انہوں نے ہر مرحلہ پر میری ہمت بڑھائی اور ہر نئے خیال ومفہوم کی تعریف اور قدر دانی کی۔

قوسِ قزح پر لکھے اردو دب کی مؤقر شخصیات کے تاثرات میں نے پڑھے اور کئی کئی بار پڑھے۔ ہر بار میری آنکھیں اشک بار ہوجاتی ہیں۔ قوسِ قزح کے منظرِ عام پر آنے سے پہلے اتنی غیر متوقع بیش قیمتی تقریظ مجھے دیوانہ بنا کر چھوڑے گی۔ میں محسوس کرتا ہوں کہ قوسِ قزح کا حقِ تصنیف مجھے مل گیا۔

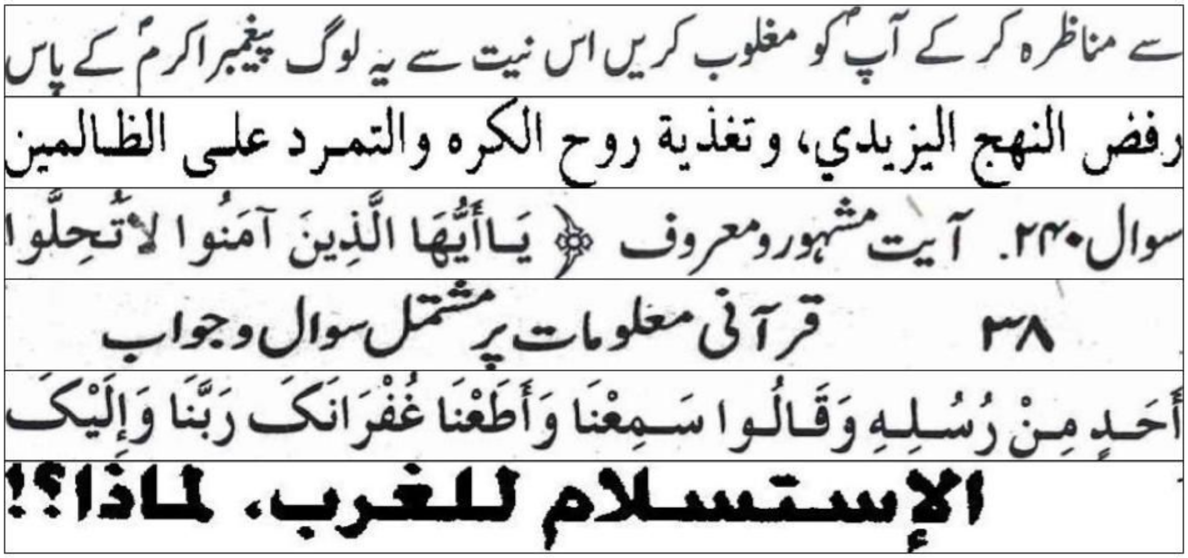
قوسِ قزح 35

Figure 3.5: Sample images of Urdu text from the consortium project.

This dataset served as the *train-set* for all our experiments on Urdu OCR. The Hybrid CNN-RNN model was trained on 28,000 line images chosen randomly from the total of 29,876 images. The remaining 1,876 images were kept aside for validation purposes.

3.4.3 IIIT-Urdu OCR Dataset

We also release a new IIIT-Urdu OCR dataset consisting of 2,000 Urdu text line images along with their corresponding annotations. To incorporate maximum variance in terms of writing styles and fonts, as predominantly seen in Urdu publications, text pages from various sources were curated. High-resolution scanned copies of Urdu books and magazines were created. Sample images from this dataset can be seen in Fig. 3.6



سے مناظرہ کر کے آپؐ کو مغلوب کریں اس نیت سے یہ لوگ پیغمبر اکرمؑ کے پاس

رفض النهج اليزيدي، وتغذية روح الكره والتمرد على الظالمين

سوال۲۴۰. آیت مشہور ومعروف ﴿ يَـاأَيُّهَا الَّذِينَ آمَنُوا لاَتُحِلُّوا ا

۳۸ قرآنی معلومات پر مشتمل سوال وجواب

أَحَـدٍ مِـنْ رُسُـلِهِ وَقَـالُـوا سَـمِعْنَا وَأَطَعْنَا غُفْرَانَکَ رَبَّنَا وَإِلَيْکَ

الإستسلام للغرب، لماذا؟!

Figure 3.6: Sample images from the IIIT-Urdu OCR dataset. The dataset consists of 2,000 Urdu line images scanned at high pixel resolution and annotated at the line level.

After curating the large set of scanned Urdu pages, bounding boxes around text lines were manually made and annotations for the same were provided by language experts. The dataset has been made available publicly for future researchers to compare the performance of their solutions against our benchmark. The dataset can be downloaded from the Urdu OCR project page on the CVIT lab's website [1].

[1] IIIT-Urdu OCR Dataset : https://goo.gl/2G2oAS

3.5 Results and Analysis

In this section we discuss the efficacy of the solution architectures, discussed in Chapter. 2, in recognizing printed Urdu text images. We perform our experiments with the assumption that cropped line images are available, and not full page/text images. Simply put, we compare the performance of various networks on their text-recognition capabilities and not on their text-detection ability. We compare the results of our Hybrid CNN-RNN architecture against the previous state-of-the-art Bi-directional LSTM networks based recognition presented in [81]. The transcription accuracy is compared on the UPTI dataset and benchmark results are provided for the IIIT-Urdu OCR dataset.

The metric used to compare the performance of these solutions is CRR - *Character Recognition Rate*. In the following equation for CRR, RT and GT stand for recognized-text and ground-truth, respectively.

$$CRR = \frac{(nCharacters - \sum EditDistance(RT, GT))}{nCharacters}$$

Table 3.1 and Fig. 3.7 present accuracy comparisons of previous solutions and the variants for our solution architecture; Hybrid X-CNN-RNN and Hybrid X-CNN-RNN-FINE, where X stands for the number of convolutional layers in the model. Hybrid X-CNN-RNN-FINE model was fine-tuned on the train set of UPTI dataset, while the Hybrid X-CNN-RNN models were not fine-tuned and neither did they see any images of the UPTI dataset whilst training. These models were trained using the Consotium Project Data discussed in Section 3.4.2. It should be noted that the Hybrid CNN-RNN architectures achieve higher transcription accuracy than the previous state-of-the-art even without fine-tuning.

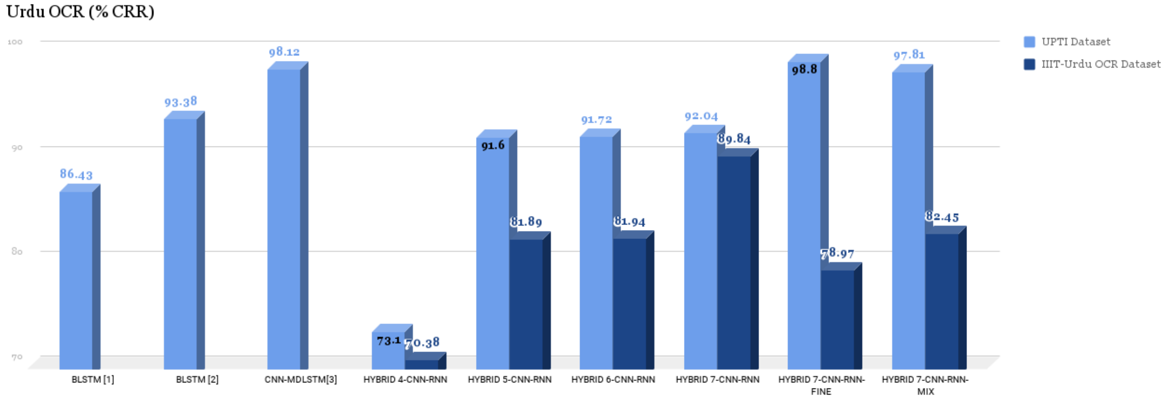


Figure 3.7: Bar chart visualization of the transcription accuracy for Urdu OCR.

We noticed that fine-tuning on the UPTI synthetic dataset reduces our models transcription accuracy on the IIIT-Urdu OCR dataset. To verify the reasons for this behaviour, we trained the Hybrid X-CNN-

Table 3.1: Transcription accuracy for Urdu OCR

URDU OCR	UPTI DATASET	IIIT-URDU OCR DATASET
	CRR (%)	CRR (%)
BLSTM [81]	86.43	-
BLSTM [104]	93.38	-
CNN-MDLSTM [105]	98.12	-
HYBRID 4-CNN-RNN	73.10	70.38
HYBRID 5-CNN-RNN	91.60	81.89
HYBRID 6-CNN-RNN	91.72	81.94
HYBRID 7-CNN-RNN	92.04	**89.84**
HYBRID 7-CNN-RNN-FINE	**98.80**	78.97
HYBRID 7-CNN-RNN-MIX	97.81	82.45

Notice that the hybrid CNN-RNN model outperforms BLSTM based method on the UPTI dataset without any fine-tuning. Also, fine-tuning on the UPTI dataset decreases transcription accuracy on IIIT-Urdu OCR dataset.

RNN-MIX model. Sampling equal number of images from the UPTI and IIIT-Urdu OCR dataset, we fine-tuned the Hybrid 7-CNN-RNN model on this sample-set. We observed that the model performance remains similar on UPTI while improves on IIIT-Urdu OCR dataset as compared to it's FINE variant. This behaviour can be attributed to the higher diversity among fonts and styles in our scanned data as compared to UPTI's synthetic images, thereby confirming the variance superiority of IIIT-Urdu OCR dataset.

The trained model scaling well to a new dataset, UPTI, demonstrates the robustness of learnt convolutional features. However, there still exist certain cases where the model fails to create accurate transcription. A qualitative analysis of the model performance can be seen in Fig. 3.8. We can see that the model fails to accurately predict the diacritics in certain cases. Also, creating transcription for images containing English numerals is erroneous.

To get further insights into the workings of convolutional layers, we visualize the filters learnt by our model (Fig. 3.9). The visualizations are created by forward-passing an image through the convolutional layer and treating each depth-channel in the output activation of the layer as a separate image. It is interesting to see that the model tried to learn innate patterns in Urdu text. The first convolutional layer learns to detect *text-edges, text-body* and *diacritics* in the text. As we go deeper in the convolutional layers, additional complex insights are brought into the detection process. The filters now differentiate

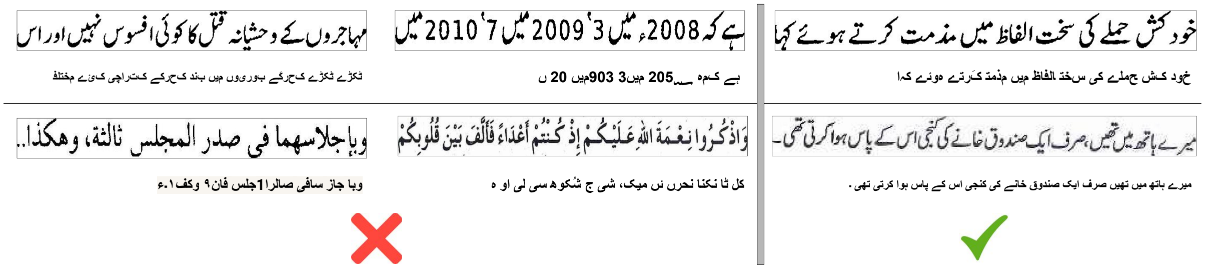

خود کش حملے کی سخت الفاظ میں مذمت کرتے ہوئے کہا
ہے کہ 2008ء میں 3، 2009 میں 7، 2010 میں
مہاجروں کے وحشیانہ قتل کا کوئی افسوس نہیں اور اس
خود کش حملے کی سخت الفاظ میں مذمت کرتے ہوئے کہا
ہے کمہ ــــ 205 میں3 903میں 20 ں
ٹکڑے ٹکڑے کرگے بوریوں میں بند کرکے کتراچی کئے مختلف
میرے ہاتھ میں تھیں، صرف ایک صندوق خانے کی کنجی اس کے پاس ہوا کرتی تھی۔
وَاذْكُرُوا نِعْمَةَ اللهِ عَلَيْكُمْ إِذْ كُنْتُمْ أَعْدَاءً فَأَلَّفَ بَيْنَ قُلُوبِكُمْ
وبإجلاسهما في صدر المجلس ثالثة، وهكذا..
میرے ہاتھ میں تھیں صرف ایک صندوق خانے کی کنجی اس کے پاس ہوا کرتی تھی ۔
کل ثا نکنا نحرں ئں میک، شی ج شکوہ سی لی او ہ
وبا جاز سافی صالرا1جلس فان۹ وکف۱.ء
✓
✗

Figure 3.8: Qualitative analysis of the Hybrid 7-CNN-RNN model on the UPTI dataset (top) and the IIIT-Urdu OCR dataset (bottom). The printed text images are enclosed in gray rectangles with the transcription made by our model given below each image.

between diacritics appearing above and below the main text-body. This behaviour adds further weight to our belief that convolutional filters trained in an end-to-end fashion are better at learning robust feature descriptors than most hand-crafted features.

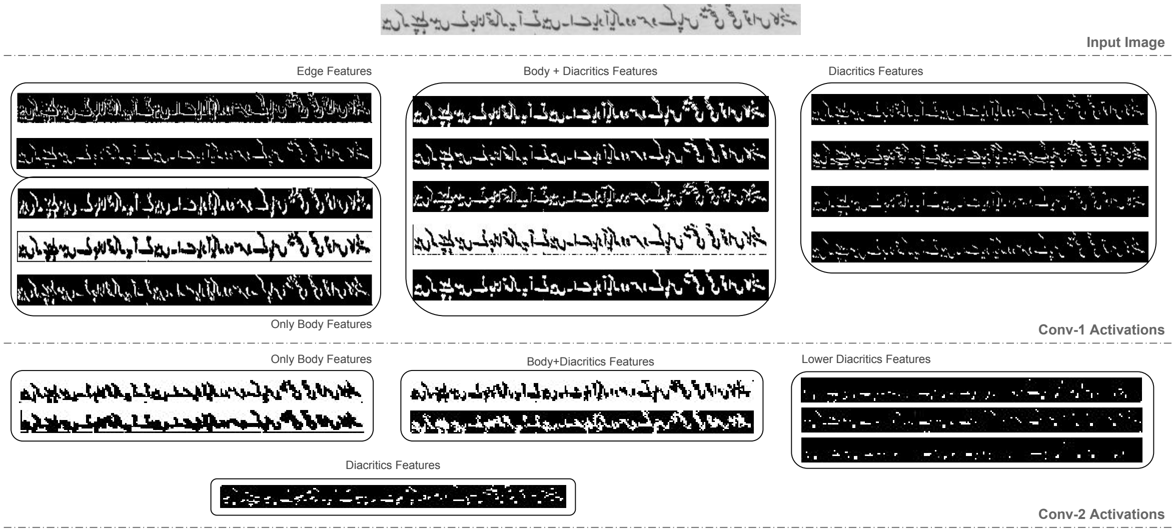


Figure 3.9: Visualization of the robust convolutional features learnt by the HYBRID 7-CNN-RNN model. The partitions show some of the activations for a given input created by the model which can be interpreted as solving a particular type of detection task (*edge, text-body, diacritics*). Notice how going deeper in convolutions brings more insights into detection - Diacritics appearing below the text-body get separate filters, adding a sense of relative positioning.

3.6 Concluding Remarks

We demonstrate the efficacy of newer script and language agnostic approaches for low resource languages like Urdu, for which traditional methods were often constrained by language specific modules. For cursive script languages, like Urdu, segmentation of individual characters/glyphs was challenging. CTC loss layer enabled segmentation-free transcription and end-to-end training of the transcription module. Furthermore, *Bi-directional* RNN's made it possible to capture contexts in both the forward and backward directions.

We show how state-of-the-art deep learning techniques can be successfully adapted to some rather challenging tasks like Urdu OCR. We outperform previous state-of-the-art solutions for Urdu OCR on existing benchmarks. Additionally, we provided insights into the robust representations learnt by the convolutional layers through visualization of activations. With the availability of better feature representations and learning algorithms, we believe that the focus of the vision community should now shift towards more complex cursive scripts which are generally also low on resources. Another possible direction to take this work forward would be the incorporation of *Attention Modelling* into text recognition, which has been proven quite effective for rather complicated tasks like object detection and captioning. We hope that the introduction of a new *IIIT-Urdu OCR dataset* and our benchmark results would instill interest among the community to take this field of research further.

Chapter 4

Arabic Video Text and Scene Text Recognition

4.1 Introduction

Creating Arabic text recognition systems which are robust is a challenging task in itself. Recognizing text embedded in video streams or occurring in natural scenes brings with it several obstacles in terms of large variance among fonts, colors, complex backgrounds, occlusions and distortions, and relatively small regions of image actually containing the text - giving rise to problems of resolution. A major hurdle in performing text recognition for inflectional scripts, like Arabic, is the lack of large quantities of annotated data. Moreover, solutions proposed for Latin and Chinese script based text recognition don't scale well to Arabic due to the cursive nature and various other intricate characteristics of its script.

For many years, the focus of the research on text recognition in Arabic has been on printed and handwritten documents [1, 2, 3, 4]. Majority of the works in this space were on individual character recognition. Since segmenting an Arabic word or line image into its sub-units is challenging, such models didn't scale well to word/line recognition tasks. In the recent years there has been a shift towards segmentation-free text recognition models, mostly based on HMMs or RNNs. Methods such as [10] and [81] used RNN based models for recognizing printed/handwritten Arabic script. Our attempt to recognize Arabic text in videos and natural scenes follows the same paradigm.

In this chapter, we demonstrate the effectiveness of our Hybrid CNN-RNN architecture in recognizing Arabic text in video frames and natural scenes. The model follows a segmentation-free, sequence-to-sequence based transcription approach. We overcome the annotated data scarcity issues by synthesizing millions of Arabic text images from a large vocabulary of Arabic words and phrases. We establish superiority of our solution on two publicly available benchmark datasets for the video text recognition task and establish the maiden benchmark for Arabic scene text recognition on a new dataset we introduce.

4.1.1 Video Text Recognition

Video text recognition refers to the task of recognizing text embedded in Video frames. It is a tougher problem than OCR since the text appears in various sizes, fonts and colors. Moreover, since the text is generally part of a larger image frame, the characters occupy only a small region of the image and have a very low resolution thereby making recognition all the more difficult in complex backgrounds. Additionally, contextual information might not be available in single video frame due to the dynamic nature of videos. For example, most News channels show live tweets and tickers running in a single line on the bottom of the screen. However, the text keeps getting updated as it moves out of the frame and new text starts to appear in its place.

With the dawn of smart phone era, there has been a sudden upsurge in sharing videos on social networking websites and content management portals. Moreover, the increasing number of TV news channels in today's world reveals videos to a be a fundamental source of information. Recognition of embedded text in these videos can greatly aid video content analysis and understanding as the text generally provides a direct and concise summary or key-point of the stories being presented in the

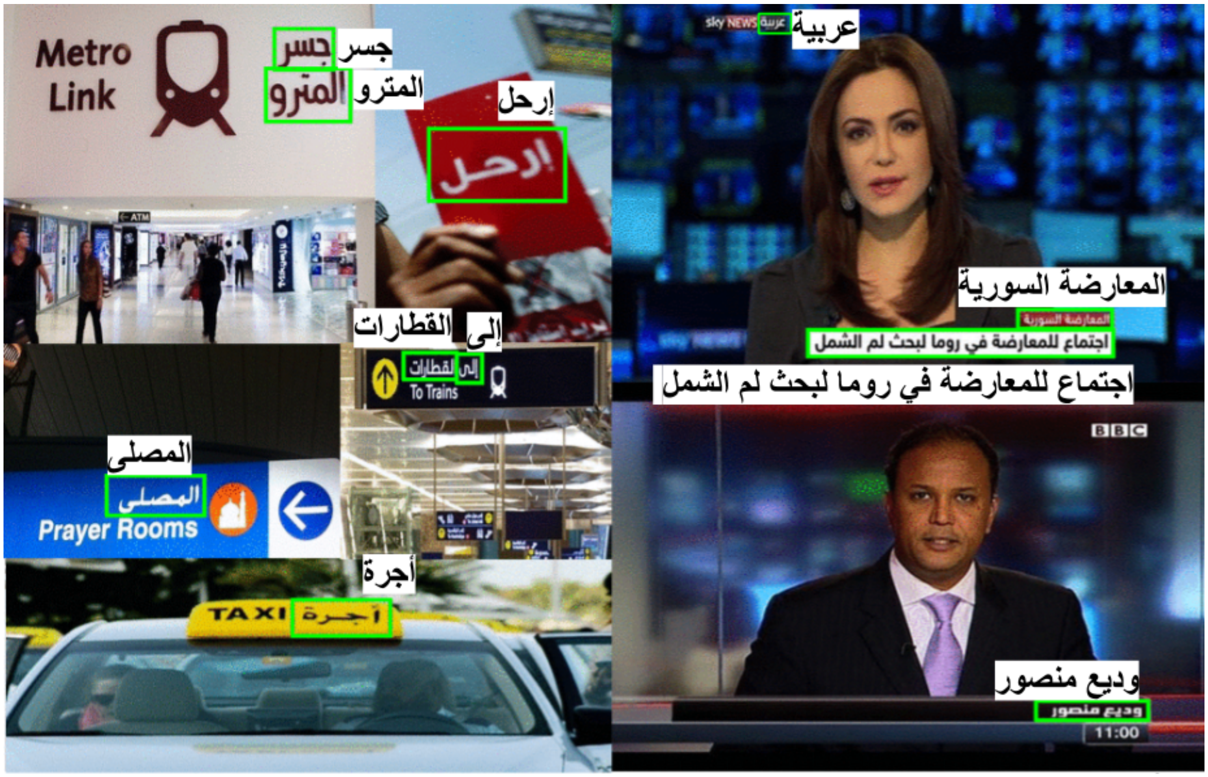


Figure 4.1: Examples of Arabic Scene Text (left) and Video Text(right) recognized by our model. Our work deals only with the recognition of cropped words/lines. The bounding boxes were provided manually.

videos. The recognized text can hence become part of the index-key in a video retrieval system. Fig. 4.1 shows an Arabic video text recognition system in action.

4.1.2 Scene Text Recognition

Scene Text Recognition refers to the problem of reading text from natural scene images. It is a tougher problem to solve than OCR as well as video text recognition. This can be attributed to the fact that OCR and video text methods generally do not generalize well to scene text recognition due to added factors like inconsistent lighting conditions, variable fonts, orientations, background noise and image distortions.

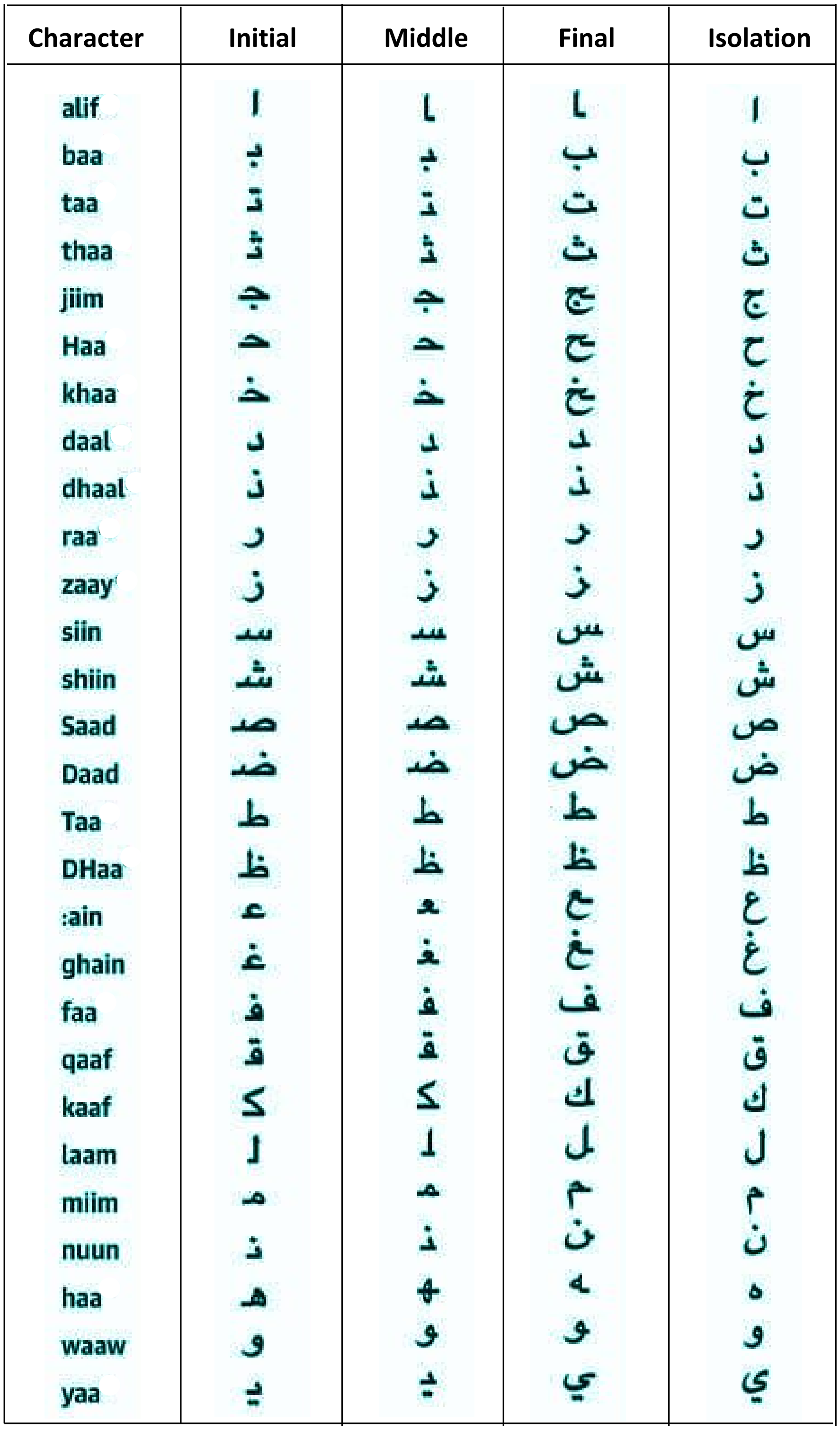
This problem has been drawing a lot of interest in the recent years, partially due to the rapid development in autonomous driving industry and wearable capturing devices , like GoPro and Google Glass, where natural scene text recognition is a major component for information capture and context gathering. What's surprising is the fact that even though Arabic is the 5th most spoken language in the world after Chinese, English, Spanish and Hindi, there had been no work in the field of word-level scene text recognition previous to our attempts. Hence, we felt that our experiments and dataset could catalyze an area ripe for research.

4.2 Intricacies of Arabic Script

Automatic recognition of Arabic is a pretty challenging task due to various intricate properties of the script. There are only 28 letters in the script and it is written in a semi-cursive fashion from right-to-left. The letters of the script are generally connected by a line at its bottom to the letters preceding and following it, except for 6 letters which can be linked only to their preceding letter. Such an instance in a word is referred to as *paw* (part-of Arabic word). Arabic script has another exceptional characteristic where the shape of a letter changes depending on whether the letter appears in the word in isolation, at the beginning, middle or at the end. In general, each letter can have one to four possible shapes which might have little or no similarity in shape whatsoever. A graphical representation of this phenomenon can be seen in Fig. 4.2.

Another common occurrence in Arabic script is that of *dots* and *diacritics*. A dot is considered as a main part of the letter and is generally used to differentiate between letters that have the same main body. Diacritics play an important role in deciding the pronunciation of a letter. However, the dots and diacritics are not related to the specific character and hence pose a problem when trying to associate the components that belong to the same letter in order to perform accurate transcription.

Arabic script falls under the category of cursive scripts, where the letters often join together in a word to form a single connected shape. This shape is often referred to as *ligature* or sub-word. An Arabic



Character	Initial	Middle	Final	Isolation
alif	ا	ـا	ـا	ا
baa	بـ	ـبـ	ـب	ب
taa	تـ	ـتـ	ـت	ت
thaa	ثـ	ـثـ	ـث	ث
jiim	جـ	ـجـ	ـج	ج
Haa	حـ	ـحـ	ـح	ح
khaa	خـ	ـخـ	ـخ	خ
daal	د	ـد	ـد	د
dhaal	ذ	ـذ	ـذ	ذ
raa	ر	ـر	ـر	ر
zaay	ز	ـز	ـز	ز
siin	سـ	ـسـ	ـس	س
shiin	شـ	ـشـ	ـش	ش
Saad	صـ	ـصـ	ـص	ص
Daad	ضـ	ـضـ	ـض	ض
Taa	طـ	ـطـ	ـط	ط
DHaa	ظـ	ـظـ	ـظ	ظ
:ain	عـ	ـعـ	ـع	ع
ghain	غـ	ـغـ	ـغ	غ
faa	فـ	ـفـ	ـف	ف
qaaf	قـ	ـقـ	ـق	ق
kaaf	كـ	ـكـ	ـك	ك
laam	لـ	ـلـ	ـل	ل
miim	مـ	ـمـ	ـم	م
nuun	نـ	ـنـ	ـن	ن
haa	هـ	ـهـ	ـه	ه
waaw	و	ـو	ـو	و
yaa	يـ	ـيـ	ـي	ي

Figure 4.2: Characters of the Arabic script. Notice how the characters can take multiple representations depending on their contextual position.

word may be composed of one or many ligatures and the ligature shape thus formed may hold little or no correspondence to the original characters incorporating the ligature. From a recognition perspective, this is a really challenging task to be able to classify ligature into its sub-character components. Moreover, the contextual sensitivity, based on position of the letter in the word as discussed above, adds manifold to the complexity of splitting ligatures to character components.

4.3 Related Work

Though there has been a lot of work done in the field of text transcription in natural scenes and videos for the English script [8, 9, 92, 106], it is still in a nascent state as far as Arabic script is considered. Previous attempts made to address similar problems in English [92, 106] first detect individual characters and then character-specific Deep Convolutional Neural Networks (DCNNs) are used to recognize these detected characters. The short-comings of such methods are that they require training a strong character detector for accurately detecting and cropping each character out from the original word. In case of Arabic, this task becomes even more difficult due to intricacies of the script, as discussed in Section 4.2.

Another approach by [92] was to treat scene text recognition as an image classification problem instead. To each image, they assign a class label from a lexicon of words spanning the English language. For practical purposes, 90k most frequently occurring words were chosen so that the lexicon has a reasonably bounded size. However, this approach is limited to the the size of the lexicon used for its possible unique transcriptions and the large number of output classes add to training complexity. Moreover, any random combination of characters cannot be detected as it might not occur in the lexicon, and hence recognizing proper nouns is erroneous. Therefore, the model is not scalable to inflectional languages like Arabic where the number of unique words and ligatures is much higher as compared to English.

Another category of solutions typically embed image and its text label in a common subspace and the recognition problem is converted to a search retrieval problem in the embedded space. The recognition/retrieval is performed on the learnt common representations. For example, [107] embed word images and text strings in a common vector-subspace and thus convert the task of word recognition into a retrieval problem. Similarly, [108] and [109] address the problem of learning word image representations; given the cropped image of a word, they try to find robust, compact and fixed-length descriptors. They use these mid-level features for recognition/retrieval tasks thereafter for scene text recognition.

The segmentation-free transcription approach was proven quite effective for Indian Scripts OCR [80, 110] where segmentation is often as complicated as Arabic. A similar approach was used in [9] for English scene text recognition. Hand crafted features derived from image gradients were used with an RNN to map the sequence of features to a sequenes of labels. Unlike the problem of OCR, scene text recognition required more robust features to yield results comparable to the transcription based solutions of OCR. A novel approach combining the robust convolutional features and transcription abilities of RNN was introduced by [84]. Our Hybrid CNN-RNN network has been inspired from this work along with a CTC loss layer to incorporate end-to-end training of the network possible.

Works on text extraction from videos have generally been in four broad categories; edge detection methods [111, 112, 113], extraction using connected components [114, 115], texture classification methods [116] and correlation based methods [117, 118]. Previous attempts at solving video text recognition for Arabic used two seperate routines; one for extracting relevant image feature and another for classifying features to script labels for obtaining the target transcription. In [119], text-segmentation and statistical feature extraction are followed by fuzzy k-nearest neighbour techniques to obtain transcriptions. Similarly, [72] experiment with two feature extraction routines; CNN based feature extraction and Deep Belief Network (DBN) based feature extraction, followed by a Bi-directional LSTM layer [88, 89].

4.4 Datasets

Arabic text datasets are mainly dedicated to printed text and handwritten documents. Moreover, the datasets available for Arabic video text recognition are only a few thousand images and hence can't be used to train any large DCNN's. Also, there no publicly available datasets for Arabic scene text recognition, to the best of our knowledge. In this section, we discuss the details of our synthetic data generation routine - used to overcome the annotated data deficiency. Thereafter, we describe the current benchmark datasets for Arabic video text and finally throw light on the new IIIT-Arabic dataset that we release publicly for Arabic scene text recognition.

4.4.1 Synthetic Data Generation

Models for both video text and scene text recognition problems are trained using synthetic images rendered from a large vocabulary using freely available Arabic Unicode fonts. For the scene text task, images were rendered from a vocabulary of a quarter million most commonly occurring words on the Arabic Wikipedia. A random word from the vocabulary is first rendered into the foreground layer of the image by varying the *font, stroke color, stroke thickness, kerning, skew* and *rotation*. Later, a random perspective projective transformation is applied to the image, which is then blended with a random crop from a natural scene image chosen at random from the Places dataset [120]. Finally, the foreground layer is alpha composed with a background layer, which again is a random crop from another random natural scene image. A visualization of this process can be seen in Fig. 4.3.

The synthetic line images for video text recognition are rendered from random text lines crawled from Arabic news websites. The rendering process is much simpler than the scene text variant, since real video embedded text usually has uniform color for the text and background. These embedded texts also lack a perspective distortion usually. A detailed report of the rendering process is made

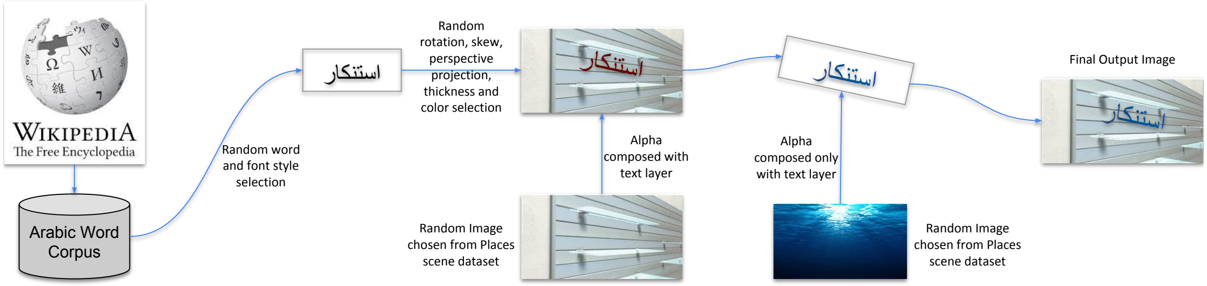

WIKIPEDIA
The Free Encyclopedia
Arabic Word Corpus
Random word and font style selection
استنكار
Random rotation, skew, perspective projection, thickness and color selection
استنكار
Alpha composed with text layer
Random Image chosen from Places scene dataset
استنكار
Alpha composed only with text layer
Random Image chosen from Places scene dataset
Final Output Image
استنكار

Figure 4.3: Visualization of the Synthetic Pipeline for Arabic scene text image generation.

available in [121]. In totality, around 2 million video text line images and 4 million scene text word images were rendered for training the respective solution models. The model for video text recognition was initially trained on the the synthetic data and then fine-tuned on the train partitions of real-world datasets, ALIF [72] and ACTIV [73]. Sample images from this rendering process can be seen in Fig. 4.4

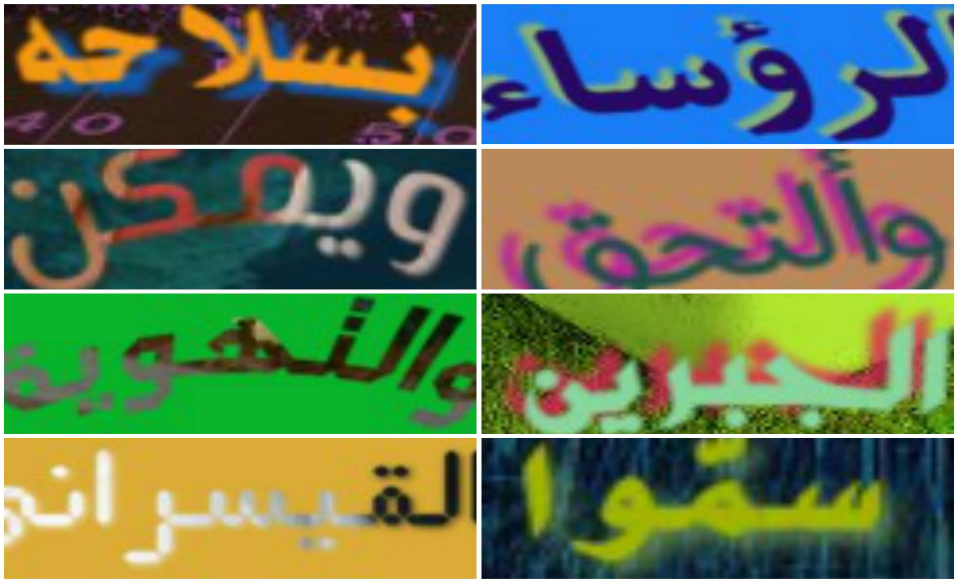


Figure 4.4: Sample images from the rendered synthetic Arabic scene text dataset. The images closely resemble real world natural scene images.

4.4.2 ALIF and AcTiV Dataset

The ALIF dataset [72] consists of 6,532 cropped text line images from 5 popular Arabic News channels. ACTIV dataset is larger than ALIF and contains 21,520 line images from popular Arabic News channels. The dataset contains video frames wherein bounding boxes of text lines are annotated. Sample images from the two datasets can be seen in Fig. 4.5.

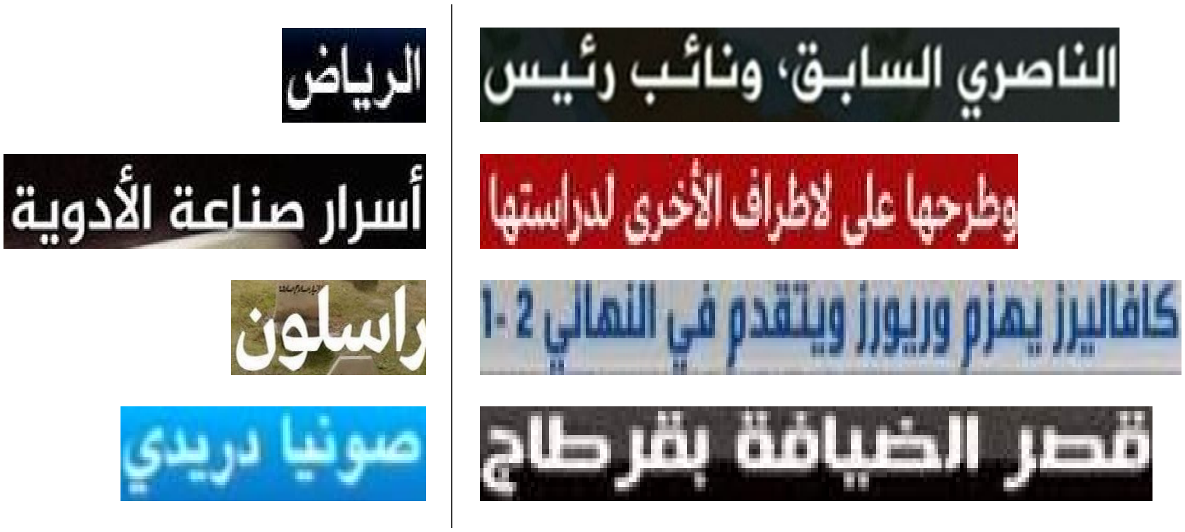

الرياض
أسرار صناعة الأدوية
راسلون
صونيا دريدي
الناصري السابق، ونائب رئيس
وطرحها على لاطراف الأخرى لدراستها
كافاليرز يهزم وريورز ويتقدم في النهائي 2-1
قصر الضيافة بقرطاج

Figure 4.5: Sample video text recognition line images from the ALIF (left) and ACTIV (right) datasets.

4.4.3 IIIT-Arabic Dataset

A new Arabic scene text dataset was curated by downloading freely available images containing Arabic script from Google images. The dataset contains 2,000 Arabic word images captured in various scenarios like local markets & shops, billboards, navigation signs, graffiti, etc. The images span a large variety of naturally occurring image-noises and distortions. The images were manually annotated by human experts of the script and the transcriptions as well as the image data is being made publicly available [1] for future research groups to utilize. Sample images from the dataset can be seen in Fig. 4.6.

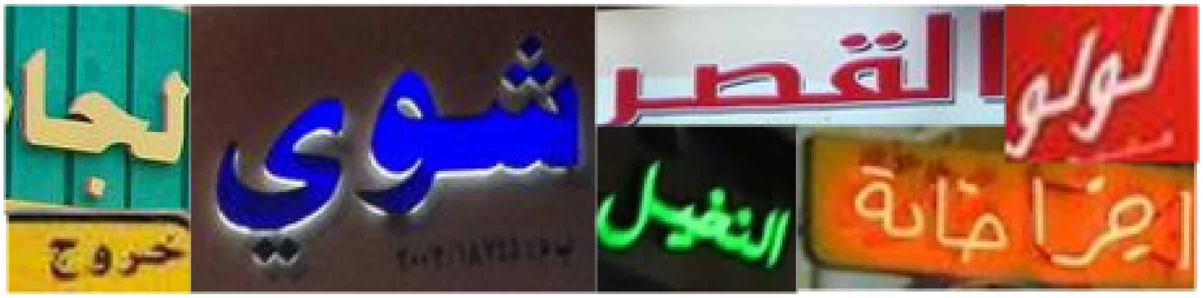


Figure 4.6: Sample images from the Arabic scene text dataset. The captured images span various scenarios like local markets & shops, billboards, navigation signs, graffiti, etc.

[1] IIIT-Arabic Dataset download : https://goo.gl/QoUKE2

4.5 Results and Analysis

In this section we showcase the performance of our Hybrid CNN-RNN architecture, discussed in Section 2.5.2, in recognizing Arabic script appearing in video frames and natural scene images. It is assumed that input images are cropped word images from natural scenes and not full scene images. Since there were no previous works in Arabic scene text recognition at word level, the baseline results are reported on the new IIIT-Arabic scene text dataset we introduce. For the video text recognition task, results are reported on two existing video text datasets - ALIF and ACTIV.

Results on video text recognition are presented in Table 4.1 and Fig. 4.7. Since there has been no work done in word-level Arabic scene text recognition, we compare the results obtained on the IIIT-Arabic dataset using a popular free OCR system - *Tesseract* [122]. The results for Arabic scene text recognition can be seen in Table 4.2 and Fig. 4.8. Similarly, Arabic video text recognition baseline comparisons are also made on the ABBYY OCR system [123]. The performance has been evaluated using the following metrics; CRR - *Character Recognition Rate*, WRR - *Word Recognition Rate* and LRR - *Line Recognition Rate*. In the below equations, RT and GT stand for recognized text and ground truth respectively.

$$CRR = \frac{(nCharacters - \sum EditDistance(RT, GT))}{nCharacters}$$

$$WRR = \frac{nWordsCorrectlyRecognized}{nWords}$$

$$LRR = \frac{nTextImagesCorrectlyRecognized}{nImages}$$

It should be noted that even though the methods compared on for video text recognition use a separate convolutional architecture for feature extraction, unlike the end-to-end trainable Hybrid CNN-RNN architecture, we obtain better character-level and line-level transcription accuracy for the Arabic video text recognition task and set new state-of-the-art.

On video text recognition task, we report the best results so far on the benchmark datasets. Scene text recognition is a much harder problem compared to video text recognition or OCR. The variability in terms of lighting, distortions and typography make the learning pretty difficult. Hence, the lower transcription accuracy in the experimentation shown in Table 4.2 is in agreement with our expectation. A qualitative analysis of the Arabic scene text recognition task can be seen in Fig. 4.9. At times, the hybrid model fails to correctly transcribe images with a high density of dots and diacritics.

Table 4.1: Accuracy for Video Text Recognition.

VideoText	**ALIF Test1**		**ALIF Test2**		**AcTiV**	
	CRR(%)	*LRR(%)*	*CRR(%)*	*LRR(%)*	*CRR(%)*	*LRR(%)*
ConvNet-BLSTM [72]	94.36	55.03	90.71	44.90	–	–
DBN-BLSTM [72]	90.73	39.39	87.64	31.54	–	–
ABBYY [123]	83.26	26.91	81.51	27.03	–	–
Hybrid CNN-RNN network	**98.17**	**79.67**	**97.84**	**77.98**	**97.44**	**67.08**

The hybrid CNN-RNN architecture we employ outperforms previous video text recognition benchmarks.

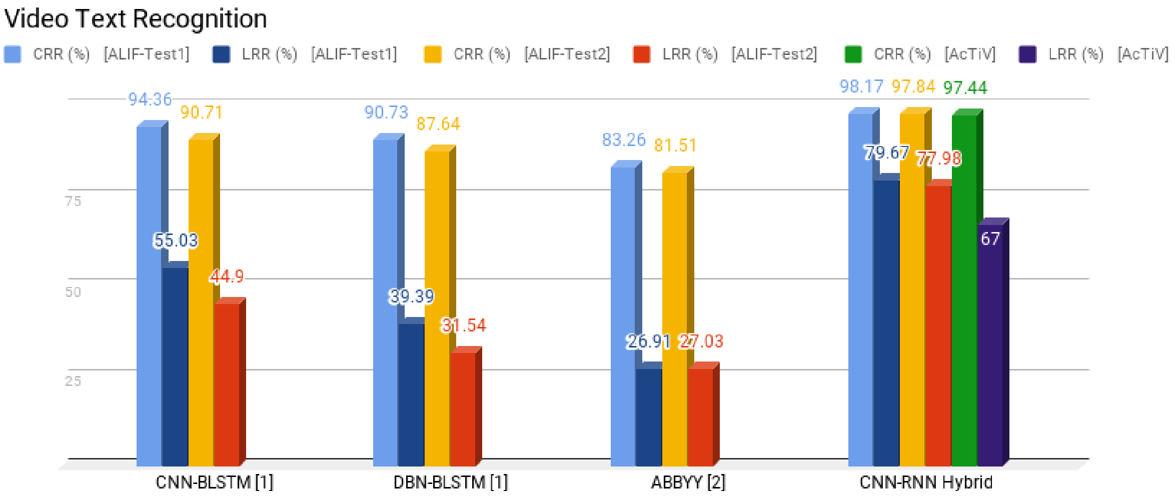


Figure 4.7: Bar chart visualization of the transcription accuracy for Arabic video text recognition.

Table 4.2: Accuracy for Scene Text Recognition.

SceneText	**IIIT-Arabic Dataset**	
	CRR(%)	*WRR(%)*
Tesseract [122]	17.07	5.92
Hybrid CNN-RNN network	**75.05**	**39.43**

Lower accuracy on scene text recognition as compared to video text recognition problem testify the inherent difficulty associated with the problem compared to printed or video text recognition.

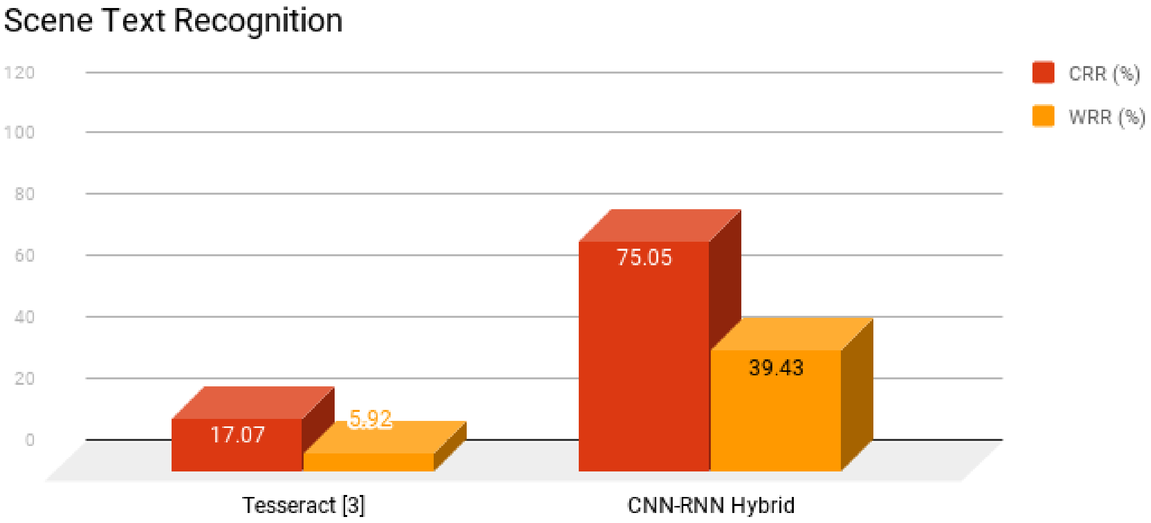


Figure 4.8: Bar chart visualization of the transcription accuracy for Arabic scene text recognition.

4.6 Concluding Remarks

We demonstrate that state-of-the-art deep learning techniques can be successfully adapted to challenging tasks like Arabic video text and natural scene text recognition. The newer script agnostic approaches are well suited for complex cursice scripts like Arabic where traditional methods generally fail due to intricacies of the script. The success of RNNs in sequence learning problems has been instrumental in the recent advances in speech recognition and image-to-text transcription problems. This came as a boon for languages like arabic where the segmentation of words into sub-word units was often troublesome. The sequence learning approach could directly transcribe images to text, while modelling the context in both forward and backward directions.

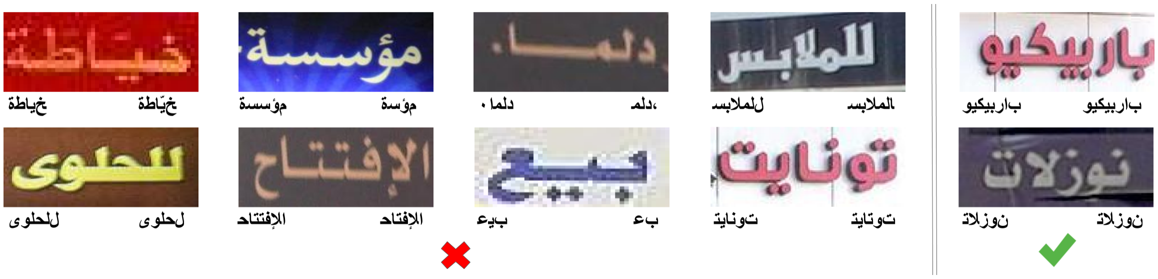


Figure 4.9: Qualitative results of Scene Text Recognition. For each image, the annotations on bottom-left and bottom-right are the label and model prediction, respectively.

We showcase the efficacy of our Hybrid CNN-RNN architecture in performing unconstrained video text and scene text recognition by setting the new state-of-the-art results for video text recognition on existing benchmark datasets. We also present a synthetic data generation pipeline to create large amounts of annotated data to improve training diversity and solve the data scarcity problem for low-resource languages. Lastly, we release a new IIIT-Arabic scene text dataset and make it available publicly.

With the availability of better learning algorithms and computation power, we feel that the focus of our vision community should now move towards solving rather difficult problems like Arabic scene text recognition. We hope that the introduction of a new dataset instills an interest among the vision community to pursue this field of research further.

Chapter 5

Conclusion and Future Work

We conclude the thesis by discussing the contributions and impact of this work and providing directions for future work that interested researchers of the field can pursue.

5.1 Discussion

In this thesis, we first analyzed the issues and challenges posed in automatic recognition of Arabic and Urdu text. Starting by giving a brief outline of the problem and the contributions of this thesis in Chapter 1, we move our focus to pose the problem from an Arabic and Urdu text transcription perspective in Chapter 2. We provide a detailed analysis of the previous solutions proposed for Arabic and Urdu text recognition in Section 2.2 and try to categorize the various abstract-classes of approaches. Next, we evaluate the existing datasets for Arabic and Urdu text recognition in various scenario in Section 2.3. Thereafter, we throw light on our solution architectures and discuss the implementation details of our Hybrid CNN-RNN Architecture in Section 2.5. We hope that this concise overview of the problem and existing solutions are beneficial to future researchers who pursue this field.

In Chapter 3, we dive into the specifics of Urdu printed text recognition, or Urdu OCR. We release the first line-level publicly available Urdu printed text dataset, IIIT-Urdu OCR in Section 2.3 and provide benchmark results on the same for future researchers to compare. We showcase the efficacy of our Hybrid CNN-RNN model for Urdu OCR in Section 3.5 and establish new *state-of-the-art* on the UPTI dataset. We also provide insights into the working of convolutional layers by creating visualizations.

In Chapter 4, we move towards the more challenging tasks of Arabic video text and scene text recognition. We release the first word-level real-world dataset for Arabic scene text recognition, IIIT-Arabic, in Section 4.4. Finally, we show how the Hybrid CNN-RNN model outperforms previous solutions on the Arabic video text recognition tasks in Section 4.5. We also establish baseline results for Arabic scene text recognition on the IIIT-Arabic dataset.

5.2 Future Work

In recent past, there has been a tremendous growth in the field of text recognition with its focus on Latin and Chinese scripts. The state-of-the-art solutions for English text recognition achieve near perfect recognition results. A few directions to explore to take this work forward can be inspired from these developments and are listed below,

- **Attention Models** : have started to receive a lot of spotlight lately. In a relevant work, [124] introduce an attention based model that automatically learns to describe the content of images and create captions for the same. A similar approach can be applied to Arabic and Urdu text recognition problems, with the attention mechanism learning to follow the text strokes of the script.

- **Spatial Transformer Networks** : try to learn a set of transformation parameters to be applied on the input image to correct spatial deformations. Such networks could be useful for our cursive scripts case which also show a slant/skew. Like, [125] show how spatial transformer networks can be integrated into an end-to-end trainable network to detect as well as recognize text.

- **Annotated Datasets** : are still rare for complex cursive scripts and we propose a synthetic data generation pipeline to overcome this difficulty. However, with larger amounts of annotated data made available, we feel the transcription accuracy would go up and hence such research would be highly beneficial for the field.

- **Handwritten & Historical Documents** : pose domain specific complexities when looked at from a text recognition perspective. Efforts can be made to see how Hybrid CNN-RNN models scale to such problems.

- **Implicit Language Models** : are grasped by the LSTM layers when performing text recognition tasks. A recent work [126] provides insights into the working of LSTM layers for OCR. They show how the implicitly learnt language model boosts the transcription accuracy for an LSTM based OCR system. A similar analysis for Arabic and Urdu could help gain deeper understanding into the working of Hybrid CNN-RNN network.

Related Publications

- **Mohit Jain**, Minesh Mathew and C.V. Jawahar, "*Unconstrained Scene Text and Video Text Recognition for Arabic Script*", in 1st International Workshop on Arabic Script Analysis and Recognition, Nancy, France, 2017. [Best Paper Award]

- **Mohit Jain**, Minesh Mathew and C.V. Jawahar, "*Unconstrained OCR for Urdu using Deep CNN-RNN Hybrid Networks*", in 4th Asian Conference on Pattern Recognition, Nanjing, China, 2017. [Student Travel Award]

Other Publications

- Minesh Mathew, **Mohit Jain** and C.V. Jawahar, "*Benchmarking Scene Text Recognition in Devanagiri, Telugu and Malayalam*", in 6th International Workshop on Multilingual OCR, Kyoto, Japan, 2017.

Bibliography

[1] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, “Improvements in hidden markov model based arabic ocr,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008. 1, 2, 33

[2] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, “Arabic handwriting recognition using baseline dependant features and hidden markov modeling,” in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pp. 893–897, IEEE, 2005. 1, 6, 8, 33

[3] A. Amin, H. Al-Sadoun, and S. Fischer, “Hand-printed arabic character recognition system using an artificial network,” *Pattern recognition*, vol. 29, no. 4, pp. 663–675, 1996. 1, 12, 33

[4] A. Amin, “Off-line arabic character recognition: the state of the art,” *Pattern recognition*, vol. 31, no. 5, pp. 517–530, 1998. 1, 33

[5] A. Lawgali, “A survey on arabic character recognition,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 2, pp. 401–426, 2015. 1

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012. 1, 17

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014. 1, 17

[8] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009. 2, 37

[9] B. Su and S. Lu, “Accurate scene text recognition based on recurrent neural network,” in *Asian Conference on Computer Vision*, pp. 35–48, Springer, 2014. 2, 12, 37

[10] M. R. Yousefi, M. R. Soheili, T. M. Breuel, and D. Stricker, “A comparison of 1d and 2d lstm architectures for the recognition of handwritten arabic.,” in *DRR*, p. 94020H, 2015. 2, 33

[11] "Abstract of speakers' strength of languages and mother tongues - 2001." https://web.archive.org/web/20080404093518/http://www.censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/Statement1.htm. 2

[12] I. A. Jannoud, "Automatic arabic hand written text recognition system," *American Journal of Applied Sciences*, vol. 4, no. 11, pp. 857–864, 2007. 3, 10, 12

[13] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban, and S. M. Golzan, "A comprehensive isolated farsi/arabic character database for handwritten ocr research," in *Tenth International Workshop on Frontiers in Handwriting Recognition*, Suvisoft, 2006. 3, 10

[14] J. H. Y. AlKhateeb, *Word based off-line handwritten Arabic classification and recognition. Design of automatic recognition system for large vocabulary offline handwritten Arabic words using machine learning approaches.* PhD thesis, University of Bradford, 2010. 3, 7, 10

[15] K. Jumari and M. A. Ali, "A survey and comparative evaluation of selected off-line arabic handwritten character recognition systems," *Jurnal Teknologi*, vol. 36, no. 1-18, 2002. 6

[16] P. Ahmed and Y. Al-Ohali, "Arabic character recognition: Progress and challenges," *Journal of King Saud University-Computer and Information Sciences*, vol. 12, pp. 85–116, 2000. 6

[17] R. Halavati, M. Jamzad, and M. Soleymani, "A novel approach to persian online hand writing recognition," *Transactions on Engineering, Computing and Technology*, vol. 6, pp. 232–236, 2005. 6

[18] N. Mezghani, A. Mitiche, and M. Cheriet, "On-line recognition of handwritten arabic characters using a kohonen neural network," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pp. 490–495, IEEE, 2002. 6

[19] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63–84, 2000. 6

[20] J. H. AlKhateeb, J. Ren, J. Jiang, S. S. Ipson, and H. El Abed, "Word-based handwritten arabic scripts recognition using dct features and neural network classifier," in *Systems, Signals and Devices, 2008. IEEE SSD 2008. 5th International Multi-Conference on*, pp. 1–5, IEEE, 2008. 6, 10

[21] H. El Abed and V. Margner, "Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten arabicwords," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2, pp. 974–978, IEEE, 2007. 7, 11

[22] H. Al-Rashaideh, "Preprocessing phase for arabic word handwritten recognition," , vol. 6, no. 1, 2006. 7, 8

[23] F. Farooq, V. Govindaraju, and M. Perrone, "Pre-processing methods for handwritten arabic documents," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pp. 267–271, IEEE, 2005. 7, 8

[24] A. Mowlaei, K. Faez, and A. T. Haghighat, "Feature extraction with wavelet transform for recognition of isolated handwritten farsi/arabic characters and numerals," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2, pp. 923–926, IEEE, 2002. 7

[25] A. Cheung, M. Bennamoun, and N. Bergmann, "A recognition-based arabic optical character recognition system," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 5, pp. 4189–4194, IEEE, 1998. 7

[26] S. N. Nawaz, M. Sarfraz, A. Zidouri, and W. G. Al-Khatib, "An approach to offline arabic character recognition using neural networks," in *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on*, vol. 3, pp. 1328–1331, IEEE, 2003. 7, 8, 10

[27] M. M. Altuwaijri and M. A. Bayoumi, "Arabic text recognition using neural networks," in *Circuits and Systems, 1994. ISCAS'94., 1994 IEEE International Symposium on*, vol. 6, pp. 415–418, IEEE, 1994. 7

[28] A. Mishra, K. Alahari, and C. Jawahar, "An mrf model for binarization of natural scene text," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 11–16, IEEE, 2011. 7

[29] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, no. 2, pp. 216–233, 2001. 8, 10

[30] R. Legault and C. Y. Suen, "Optimal local weighted averaging methods in contour smoothing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 801–817, 1997. 8

[31] W.-L. Lee and K.-C. Fan, "Document image preprocessing based on optimal boolean filters," *Signal processing*, vol. 80, no. 1, pp. 45–55, 2000. 8

[32] A. A. Atici and F. T. Yarman-Vural, "A heuristic algorithm for optical character recognition of arabic script," *Signal processing*, vol. 62, no. 1, pp. 87–99, 1997. 8

[33] J. Serra, "Morphological filtering: an overview," *Signal processing*, vol. 38, no. 1, pp. 3–11, 1994. 8

[34] M. S. Khorsheed, "Off-line arabic character recognition–a review," *Pattern analysis & applications*, vol. 5, no. 1, pp. 31–45, 2002. 8, 9

[35] A.-S. Atallah and K. Omar, "A comparative study between methods of arabic baseline detection," in *Electrical Engineering and Informatics, 2009. ICEEI'09. International Conference on*, vol. 1, pp. 73–77, IEEE, 2009. 8

[36] M. Pechwitz and V. Margner, "Baseline estimation for arabic handwritten words," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pp. 479–484, IEEE, 2002. 8, 10

[37] B. Parhami and M. Taraghi, "Automatic recognition of printed farsi texts," *Pattern Recognition*, vol. 14, no. 1-6, pp. 395–403, 1981. 8

[38] P. Burrow, "Arabic handwriting recognition," *Report of Master of Science School of Informatics, University of Edinburgh*, 2004. 8

[39] Z.-Q. Liu, J.-H. Cai, and R. Buse, *Handwriting recognition: soft computing and probabilistic approaches*, vol. 133. Springer, 2012. 8

[40] B. A. Yanikoglu and P. A. Sandon, "Recognizing off-line cursive handwriting," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 397–397, INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 1994. 8

[41] J.-x. Dong, P. Dominique, A. Krzyyzak, and C. Y. Suen, "Cursive word skew/slant corrections based on radon transform," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pp. 478–483, IEEE, 2005. 8

[42] L. Lorigo and V. Govindaraju, "Segmentation and pre-recognition of arabic handwriting," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pp. 605–609, IEEE, 2005. 8

[43] L. Zheng, A. H. Hassin, and X. Tang, "A new algorithm for machine printed arabic character segmentation," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1723–1729, 2004. 8

[44] M. Altuwaijri and M. Bayoumi, "A new thinning algorithm for arabic characters using self-organizing neural network," in *Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on*, vol. 3, pp. 1824–1827, IEEE, 1995. 8

[45] M. Tellache, M. Sid-Ahmed, and B. Abaza, "Thinning algorithms for arabic ocr," in *Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on*, vol. 1, pp. 248–251, IEEE, 1993. 8

[46] J. Cowell and F. Hussain, "Thinning arabic characters for feature extraction," in *Information Visualisation, 2001. Proceedings. Fifth International Conference on*, pp. 181–185, IEEE, 2001. 8

[47] T. Sari, L. Souici, and M. Sellami, “Off-line handwritten arabic character segmentation algorithm: Acsa,” in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pp. 452–457, IEEE, 2002. 9

[48] D. Motawa, A. Amin, and R. Sabourin, “Segmentation of arabic cursive script,” in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, vol. 2, pp. 625–628, IEEE, 1997. 9

[49] A. M. Zeki, “The segmentation problem in arabic character recognition the state of the art,” in *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on*, pp. 11–26, IEEE, 2005. 9

[50] A. Hamid and R. Haraty, “A neuro-heuristic approach for segmenting handwritten arabic text,” in *Computer Systems and Applications, ACS/IEEE International Conference on. 2001*, pp. 110–113, IEEE, 2001. 9

[51] S. A. Al-Ma’adeed, *Recognition of off-line handwritten Arabic words*. PhD thesis, University of Nottingham, 2004. 9

[52] V. Govindan and A. Shivaprasad, “Character recognitiona review,” *Pattern recognition*, vol. 23, no. 7, pp. 671–683, 1990. 9

[53] A. Amin and J. F. Mari, “Machine recognition and correction of printed arabic text,” *IEEE Transactions on systems, man, and cybernetics*, vol. 19, no. 5, pp. 1300–1306, 1989. 9

[54] M. S. Khorsheed and W. F. Clocksin, “Structural features of cursive arabic script.,” in *BMVC*, pp. 1–10, 1999. 9

[55] G. Srinivasan and G. Shobha, “Statistical texture analysis,” in *Proceedings of world academy of science, engineering and technology*, vol. 36, pp. 1264–1269, 2008. 9

[56] K. Mohiuddin and J. Mao, “A comparative study of different classifiers for handprinted character recognition,” *Pattern Recognition in Practice IV*, pp. 437–448, 2014. 9

[57] M. S. Khorsheed and W. F. Clocksin, “Multi-font arabic word recognition using spectral features,” in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4, pp. 543–546, IEEE, 2000. 10

[58] N. B. Amor and N. E. B. Amara, “Multifont arabic characters recognition using houghtransform and hmm/ann classification.,” *Journal of multimedia*, vol. 1, no. 2, pp. 50–54, 2006. 10, 12

[59] J. H. AlKhateeb, F. Khelifi, J. Jiang, and S. S. Ipson, “A new approach for off-line handwritten arabic word recognition using knn classifier,” in *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*, pp. 191–194, IEEE, 2009. 10

[60] A. Dehghani, F. Shabini, and P. Nava, “Off-line recognition of isolated persian handwritten characters using multiple hidden markov models,” in *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*, pp. 506–510, IEEE, 2001. 10

[61] S. Alma’adeed, C. Higgens, and D. Elliman, “Recognition of off-line handwritten arabic words using hidden markov model approach,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, pp. 481–484, IEEE, 2002. 10

[62] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji, “Recognition of off-line printed arabic text using hidden markov models,” *Signal processing*, vol. 88, no. 12, pp. 2902–2912, 2008. 10, 12

[63] S. Alma’adeed, “Recognition of off-line handwritten arabic words using neural network,” in *Geometric Modeling and Imaging–New Trends, 2006*, pp. 141–144, IEEE, 1993. 10

[64] K. Khatatneh *et al.*, “Probabilistic artificial neural network for recognizing the arabic. hand written characters,” in *Journal of Computer Science*, Citeseer, 2006. 10

[65] A. Asiri and M. S. Khorsheed, “Automatic processing of handwritten arabic forms using neural networks.,” in *IEC (Prague)*, pp. 313–317, 2005. 10, 11

[66] S. Al-Ma’adeed, D. Elliman, and C. A. Higgins, “A data base for arabic handwritten text recognition research,” in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pp. 485–489, IEEE, 2002. 10

[67] Y. Al-Ohali, M. Cheriet, and C. Suen, “Databases for recognition of handwritten arabic cheques,” *Pattern Recognition*, vol. 36, no. 1, pp. 111–121, 2003. 10

[68] Y. LeCun, “The mnist database of handwritten digits,” *http://yann. lecun. com/exdb/mnist/*, 1998. 11

[69] S. M. Awaidah and S. A. Mahmoud, “A multiple feature/resolution scheme to arabic (indian) numerals recognition using hidden markov models,” *Signal Processing*, vol. 89, no. 6, pp. 1176–1184, 2009. 11

[70] A. Lawgali, M. Angelova, and A. Bouridane, “Hacdb: Handwritten arabic characters database for automatic character recognition,” in *Visual Information Processing (EUVIP), 2013 4th European Workshop on*, pp. 255–259, IEEE, 2013. 11

[71] N. Sabbour and F. Shafait, “A segmentation-free approach to arabic and urdu ocr.,” in *DRR*, p. 86580N, 2013. 11, 25, 26

[72] S. Yousfi, S.-A. Berrani, and C. Garcia, “Alif: A dataset for arabic embedded text recognition in tv broadcast,” in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 1221–1225, IEEE, 2015. 11, 38, 39, 42

[73] O. Zayene, J. Hennebert, S. M. Touj, R. Ingold, and N. E. B. Amara, “A dataset for arabic text detection, tracking and recognition in news videos-activ,” in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 996–1000, IEEE, 2015. 11, 39

[74] P. Sankar K, C. Jawahar, and R. Manmatha, “Nearest neighbor based collection ocr,” in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 207–214, ACM, 2010. 11

[75] B. K. Jang and R. T. Chin, “One-pass parallel thinning: analysis, properties, and quantitative evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1129–1140, 1992. 12

[76] A. Amin and W. Mansoor, “Recognition of printed arabic text using neural networks,” in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, vol. 2, pp. 612–615, IEEE, 1997. 12

[77] B. Bushofa and M. Spann, “Segmentation and recognition of arabic characters by structural classification,” *Image and Vision Computing*, vol. 15, no. 3, pp. 167–179, 1997. 12

[78] B. Al-Badr and R. M. Haralick, “Segmentation-free word recognition with application to arabic,” in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1, pp. 355–359, IEEE, 1995. 12

[79] P. S. Natarajan, E. MacRostie, and M. Decerbo, “The bbn byblos hindi ocr system,” in *Document Recognition and Retrieval XII*, vol. 5676, pp. 10–17, International Society for Optics and Photonics, 2005. 12

[80] M. Mathew, A. K. Singh, and C. Jawahar, “Multilingual ocr for indic scripts,” in *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*, pp. 186–191, IEEE, 2016. 12, 37

[81] A. Ul-Hasan, S. B. Ahmed, F. Rashid, F. Shafait, and T. M. Breuel, “Offline printed urdu nastaleeq script recognition with bidirectional lstm networks,” in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pp. 1061–1065, IEEE, 2013. 12, 27, 29, 30, 33

[82] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, ACM, 2006. 12, 13, 16, 20

[83] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, 2015. 12

[84] B. Shi, X. Bai, and C. 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*, vol. 39, no. 11, pp. 2298–2304, 2017. 12, 16, 38

[85] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, “Robust scene text recognition with automatic rectification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4168–4176, 2016. 13

[86] P. Natarajan, E. MacRostie, and M. Decerbo, “The bbn byblos hindi ocr system,” in *Guide to OCR for Indic Scripts*, pp. 173–180, Springer, 2009. 14

[87] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, Mar 1994. 15

[88] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 16, 38

[89] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with lstm recurrent networks,” *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002. 16, 38

[90] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649, IEEE, 2013. 16

[91] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014. 16, 19

[92] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Reading text in the wild with convolutional neural networks,” *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016. 17, 37

[93] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, pp. 448–456, 2015. 20

[94] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990. 20

[95] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012. 20

[96] M. Nagata, “Japanese ocr error correction using character shape similarity and statistical language model,” in *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pp. 922–928, Association for Computational Linguistics, 1998. 24

[97] F. Camastra, “A svm-based cursive character recognizer,” *Pattern Recognition*, vol. 40, no. 12, pp. 3721–3727, 2007. 24

[98] T. Nawaz, S. Naqvi, H. ur Rehman, and A. Faiz, "Optical character recognition system for urdu (naskh font) using pattern matching technique," *International Journal of Image Processing (IJIP)*, vol. 3, no. 3, p. 92, 2009. 25

[99] S. Sardar and A. Wahab, "Optical character recognition system for urdu," in *Information and Emerging Technologies (ICIET), 2010 International Conference on*, pp. 1–5, IEEE, 2010. 25

[100] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, "Segmentation free nastalique urdu ocr," *World Academy of Science, Engineering and Technology*, vol. 46, pp. 456–461, 2010. 25

[101] M. Akram and S. Hussain, "Word segmentation for urdu ocr system," in *Proceedings of the 8th Workshop on Asian Language Resources, Beijing, China*, pp. 88–94, 2010. 26

[102] H. S. Baird, "Document image defect models," in *Structured Document Image Analysis*, pp. 546–556, Springer, 1992. 27

[103] D. Arya, C. Jawahar, C. Bhagvati, T. Patnaik, B. Chaudhuri, G. Lehal, S. Chaudhury, and A. Ramakrishna, "Experiences of integration and performance testing of multilingual ocr for printed indian scripts," in *Proceedings of the 2011 joint workshop on multilingual OCR and analytics for noisy unstructured text data*, p. 9, ACM, 2011. 27

[104] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, "Zoning features and 2dlstm for urdu text-line recognition," *Procedia Computer Science*, vol. 96, pp. 16–22, 2016. 30

[105] S. Naz, A. I. Umar, R. Ahmad, I. Siddiqi, S. B. Ahmed, M. I. Razzak, and F. Shafait, "Urdu nastaliq recognition using convolutional–recursive deep learning," *Neurocomputing*, vol. 243, pp. 80–87, 2017. 30

[106] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 3304–3308, IEEE, 2012. 37

[107] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014. 37

[108] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4042–4049, 2014. 37

[109] A. Gordo, "Supervised mid-level features for word image representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2956–2964, 2015. 37

[110] N. Sankaran and C. Jawahar, “Recognition of printed devanagari text using blstm neural network,” in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 322–325, IEEE, 2012. 37

[111] L. Agnihotri and N. Dimitrova, “Text detection for video analysis,” in *Content-Based Access of Image and Video Libraries, 1999.(CBAIVL’99) Proceedings. IEEE Workshop on*, pp. 109–113, IEEE, 1999. 38

[112] L. Agnihotri, N. Dimitrova, and M. Soletic, “Multi-layered videotext extraction method,” in *Multimedia and Expo, 2002. ICME’02. Proceedings. 2002 IEEE International Conference on*, vol. 2, pp. 213–216, IEEE, 2002. 38

[113] X.-S. Hua, X.-R. Chen, L. Wenyin, and H.-J. Zhang, “Automatic location of text in video frames,” in *Proceedings of the 2001 ACM workshops on Multimedia: multimedia information retrieval*, pp. 24–27, ACM, 2001. 38

[114] A. K. Jain and B. Yu, “Automatic text location in images and video frames,” *Pattern recognition*, vol. 31, no. 12, pp. 2055–2076, 1998. 38

[115] R. W. Lienhart and F. Stuber, “Automatic text recognition in digital videos,” in *Image and Video Processing IV*, vol. 2666, pp. 180–189, International Society for Optics and Photonics, 1996. 38

[116] H. Karray, M. Ellouze, and A. Alimi, “Indexing video summaries for quick video browsing,” in *Pervasive computing*, pp. 77–95, Springer, 2009. 38

[117] H. Karray and A. Alimi, “Detection and extraction of the text in a video sequence,” in *Electronics, Circuits and Systems, 2005. ICECS 2005. 12th IEEE International Conference on*, pp. 1–4, IEEE, 2005. 38

[118] M. Kherallah, H. Karray, M. Ellouze, and A. M. Alimi, “Toward an interactive device for quick news story browsing,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008. 38

[119] M. B. Halima, H. Karray, and A. M. Alimi, “Arabic text recognition in video sequences,” *arXiv preprint arXiv:1308.3243*, 2013. 38

[120] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE transactions on pattern analysis and machine intelligence*, 2017. 38

[121] P. Krishnan and C. Jawahar, “Generating synthetic data for text recognition,” *arXiv preprint arXiv:1608.04224*, 2016. 39

[122] “Tesseract ocr.” `https://github.com/tesseract-ocr/tesseract`. 41, 42

[123] "Abbyy ocr." https://www.abbyy.com/en-eu/. 41, 42

[124] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, pp. 2048–2057, 2015. 46

[125] C. Bartz, H. Yang, and C. Meinel, "Stn-ocr: A single neural network for text detection and text recognition," *arXiv preprint arXiv:1707.08831*, 2017. 46

[126] E. Sabir, S. Rawls, and P. Natarajan, "Implicit language model in lstm for ocr," *14th International Conference on Document Analysis and Recognition*, 2017. 46