A Multi-Faceted OCR Framework for Artificial Urdu News Ticker Text Recognition

Sami-Ur-Rehman*, Burhan Ul Tayyab*, Muhammad Ferjad Naeem*, Adnan Ul-Hasan† and Faisal Shafait*
*School of Electrical Engineering and Computer Sciences
National University of Science and Technology, Islamabad, Pakistan
Email: {14mseesrehman, btayyab.bese15seecs, 14beemnaeem, faisal.shafait}@seecs.edu.pk
†Email: adnan.ulhasan@gmail.com

***Abstract*—Content based information search and retrieval has allowed for easier access to data. While Latin based scripts have gained attention and support from academia and industry, there is limited support for cursive script languages, like Urdu. In this paper, we present the first instance of Urdu news ticker detection and recognition and take a micron sized step towards the goal of super intelligence. The presented solution allows for automating the transcription, indexing and captioning of Urdu news video content. We present the first comprehensive data set, to our knowledge, for Urdu news ticker recognition, collected from 41 different news channels. The data set covers both high and low quality channels, distorted and blurred news tickers, making the data set an ideal test case for any automatic Urdu News Recognition system in future. We identify and address the key challenges in Urdu News Ticker text recognition. We further propose an adjustment to the ground-truth labeling strategy focused on improving the readability of recognized output. Finally, we propose and present results from a Bi-Directional Long Short-Term Memory (BDLSTM) network architecture for news ticker text recognition. Our custom trained model outperforms Google's commercial OCR engine in two of the four experiments conducted.**

I. Introduction

Latin based scripts have received significant attention from the academia and industry over the years. This has led to impressive results for trivial problems such as recognition of machine printed text and has set new benchmarks for complex problems such as Scene Text Recognition and Handwriting Recognition[1]. The rise of deep learning and public challenges such as ICDAR Robust reading competition have further increased the knowledge pool for these problems. However, non Latin scripts such as Arabic and Indic scripts have not received the same attention.

Urdu is the second most dominant language in the South Asian Region. It is also the national language of Pakistan and has an audience of well over 150 million. Urdu is mainly written in Nastaleeq script; a script that was developed in the Persian region in the 14th and 15th century. It has 37 alphabets and the language is highly context sensitive. Urdu alphabets join together to form ligatures. A ligature is a combination of two or more alphabets joined to form a unique shape. In total Urdu language has well over 26,000 ligatures. Moreover, Urdu is written from top right to bottom left in the sense that characters overlap in a word. This makes Nastaleeq script recognition a much more complex problem than Latin scripts.

News channels are an essential medium of transmission of information in the Indian sub-continent. Internet penetration in the region has been improving over time but a vast amount of information is still distributed through traditional television. While the big production companies have accompanying newspapers and websites, smaller local companies transmit information through television only. Since the information is not completely digitized, a ticker detection and recognition system will allow for information storage, indexing and retrieval. We introduce the first data set for news ticker recognition and a recognition framework for the task. The only previous work on Urdu news videos was proposed by Raza and Siddiqi [2], in which the authors used stroke-width transform to detect the ticker text in the news videos. Recently Arabic, the foster-sister language of Urdu, has received attention from researchers [3] [4] [5] which has also benefited research for Urdu [6]. Some of the key challenges in Urdu text recognition in the context of news ticker text are explained in the succeeding subsections.

1) Character Segmentation: Unlike the latin-based languages, the cursive nature of the Nastaleeq script (used predominantly in all Urdu publications, paper and digital) makes it nearly impossible to achieve perfect character-level segmentation of text. It is written from right to left, following no standard baseline and characters stacked diagonally with-in a word, allowing it to fit more text into less space horizontally. Moreover, the shape and size of each character varies with its position in the text. Together, they allow the text to digress vertically, from its allocated space (see Fig. 2b), and therefore interfere with the text lines on top or beneath it, making line-segmentation a big challenge [7].

2) Ligature Coverage: A major challenge in the recognition of Nastaleeq Urdu is the profound variations of character shapes in different ligatures. The shape of each character varies with its relative position in the ligature. A study [8] estimates the total number of unique ligatures in the Urdu language to be in the vicinity of 26000 and in the light of the work by Naeem et al. [9] and verified by our own analysis (see Fig. 10), ligature coverage in the training set plays a vital

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.83

Fig. 1: Comb-tooth distortion

role in achieving lower error rates on test sets.

3) Static and scrolling ticker-text: News Tickers are static or scrolling based on the choice of the news channel. Both variations come with their own challenges. The static tickers have a fixed space on the video frame to fit all the text. News Channels decrease the font size to fit more text in certain cases. Some channels also perform horizontal compression of the text within the frame, and in the process disrupting the strokewidth-per-pixel standards of the script. The font size descaling is accommodated by normalizing the ticker height, but the horizontal compression, as we observed during the model training stage (discussed in IV), leads to increased training time to learn these representations.

In an event of frame-rate mismatch between the recording device and the video, the distortion observed is called motion interpolation, in which the intermediate frame interpolated by the recording device creates a blurring effect which affects recognition (referred to as motion blur). Another artifact associated with the moving portions of a video frame is known as the comb-tooth or saw-tooth distortion (Fig. 1). This distortion is characterized by the wavelength encoding scheme in interlaced TV signal transmissions [10]. In an interlaced video signal, alternate rows of pixels are sent in consecutive frames. Therefore, the difference in the positions of a moving object between two frames is what causes this ragged edges effect. The later two distortions mentioned in this subsection are beyond the scope of this paper and will be addressed in a follow-up research.

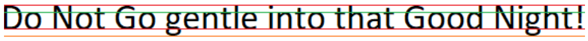
This paper is further divided into four sections. Section II discusses the data set preparation process. Section III introduces our ticker extraction technique. Section IV introduces our Urdu news ticker recognition model and section V concludes the study.

II. Data Acquisition and Data set Preparation

In this section, we discuss in detail the dataset preparation process and the rigorous measures that were taken to ensure maximum diversity throughout our dataset.

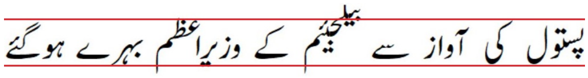
A. Collection of Data

In order to capture maximum diversity in our dataset, we collect news video recordings from three different sources: Pakistan Telecommunication Company Limited (PTCL), Digital Wireless Network (DWN) and Youtube. The first two sources are limited to publish content in the resolution specified by the service provider, Standard-Definition (SD) in both cases. Youtube offers a range of resolution options for each channel. We use Youtube to record the news content



(a) Line of sample English text[a]

[a]taken from author Dylan Thomas' [1914-1953] famous poem. First published in 1951 in an Italian Literary journal, *Botteghe Oscure*.



(b) Line of sample Urdu text[a]

[a]http://www.bbc.com/urdu/world-40101041

Fig. 2: Unlike latin-based machine printed text, the *Nastaleeq* script does not conform to any ligature positioning standards.

in High-Definition (HD) and lower resolution formats. Both service providers, PTCL and DWN, use the HDMI standard as output from the TV-box. The videos are recorded using Magewell's USB capture HDMI dongle[1]. For Youtube videos, we use a linux tool, *youtube-dl*[2], to download videos in varying resolution formats. Table I summarizes the specifications of the video resolutions recorded from the three service providers and as many as 41 unique news channels.

TABLE I: Details of the channel video recordings

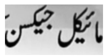
Video Quality	Resolution	# of channels	Source
Full-HD	1920 x 1080	3	Youtube
HD-Ready	1280 x 720	3	Youtube
SD	720 x 576	38	PTCL & DWN
480p	854 x 480	4	Youtube
	776 x 480	3	
	640 x 480	6	
	600 x 480	4	

In the context of news videos, the space allocated to the ticker is fixed and usually designed to fit most ligatures, but some complicated ones (as we will see later in this section) manage to escape and get truncated in the process. Diacritics (*zabr*, *zer*, *pesh* etc.) and *Nuqta* (dots associated with Urdu characters) are affected by this the most. In light of this, for the development of any Urdu news ticker recognizer system to work in the real-word, a reasonable representation of such behavior should be included in training data. Our multi-scheme ticker extraction strategy ensured this. However, as the network has to learn all the different sorts of truncated character shape variations, this leads to longer convergence times during training.

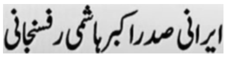
According to a study [8], 2300 of the most frequent ligatures in the language can cover up to 99% of the corpus. However, the study does not take into account the dynamic nature of the news industry. The appearance of person/place names (Fig. 3a, 3b, 3c), foreign-language phrases written in Urdu (Fig. 4a, 4b,

[1]http://www.magewell.com/usb-capture-hdmi

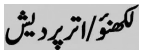
[2]https://rg3.github.io/youtube-dl/



(a) *"Michael Jackson"*

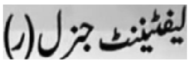


(b) *"Iranian President Akbar Hashemi Rafsanjani"*

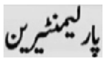


(c) *"Lucknow/ Uttar Pardesh"*

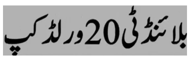
Fig. 3: Sample person/place names occuring in the Urdu text



(a) *"Lieutenant General (R)"*

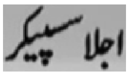


(b) *"Parlia-mentarian"*



(c) *"Blind T20 World Cup"*

Fig. 4: Sample foreign-language words written in Urdu



(a) Phrase with a typo

(b) Correct Phrase

Fig. 5: Misspelled tickers: A case of a missed white-space character between the two words is depicted, resulting in the two ligatures being joined. Such an error is crucial for cursive scripts, like *Nastaleeq*, as it disrupts the shapes of close-by ligatures as well.

4c) and non-proofread text (Fig. 5a) makes it a challenging task.

Through our extensively cautious data collection process, we have made an effort to maximize the inclusion of unique ligature shapes in our data set. To accomplish this, we devised a highly selective data collection strategy. Firstly, during the news videos recording state, we extended the data collection process to span a period of over eight months, ensuring that our data set consists of a wide variety of news content (among other variations e.g. evolution of news channels, in terms of picture quality, ticker position and color, text color, fonts used etc.). The maximum length of a single video was limited to five minutes and a delay of at least four hours was added between two recordings. Secondly, during the ticker extraction stage, in order to reduce the effect of similar news aggregating in the data set, we put a delay of 10 seconds (one in 250 frames selected) between simultaneous frame captures from an individual video. The tickers extracted from the selected frames were further exposed to a process of random selection (selection ratio = 0.3). As a result, we were able to accumulate a total of 6823 unique ligatures in our data set.

B. Ground-truth preparation

Overlapping ligature shapes is one of the major challenges in Nastaleeq script recognition. The issue arises because of the uneven lengths of the white space between the ligatures (see Fig. 6). To add to the already daunting task, InPage, the biggest source of printed Nastaleeq text, allows a non-breaking space (in addition to the normal white space) to separate the ligatures. When used correctly, the non-breaking space is reserved for the rare cases where a word contains

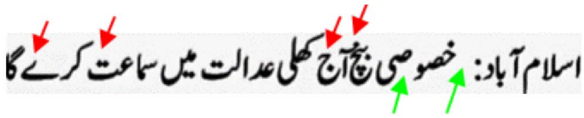


Fig. 6: Uneven spacing issues. The red arrows point towards the overlapping ligatures, while the green ones indicate the sparse holes.

Fig. 7: Static ticker extraction using Efficient and Accurate Scene Text Detector (EAST) [11].

more than one connectable ligatures, but with a need to present them separately without the addition of a visible white space in between. However, in a time-sensitive usecase like broadcast media, where the content is not usually proofread, the two white spaces end up being used interchangeably. Although, this does not affect the error rates obtained during recognition, but it severely disrupts the readability of the text. For the usecase of recognizing news ticker text, the readability of the recognized text is as crucial as the error rate. We propose a ground truth labelling strategy to address this problem. Instead of separating two words with a white spaces, we propose to use a white space to separate any two ligatures in the text.

We use Google's Urdu keyboard extension to transcribe the groundtruth in UTF-8 enconded .txt files. The naming conventions used for the groundtruth are the same as used for the images:

{***channelName***}_{***ticketType***}{**tickerStyleNo**}_{**tickerNo**}

channelName is the name of the channel in lower case and without any spaces. s is used for static ***tickerType***, while d represents dynamic (scrolling) ***tickerType***. The ***tickerStyleNo*** is a one-to-two digit code assigned to indicate the within-channel variations of the ticker, and ***tickerNo*** is just the count (one-to-many digits) of that specific type of ticker in the data set.

III. Ticker Extraction

We employed two different strategies for extracting static and scrolling tickers. In order to automate the process of

(a) A sample scrolling ticker cropped using the ticker position information for that channel.

(b) By using the information of ticker scrolling speed for that channel, the number of redundant frames can be found and the non-overlapping portions of the two frames joined together to make a single news ticker.

Fig. 8: Scrolling ticker extraction

static ticker extraction, we fine-tuned an implementation[3] of the Efficient and Accurate Scene Text Detector (EAST) [11] over a mixture of AcTiV [4] and Raza et al. [2] data sets to be able to return the bounding boxes of all the text within a frame (Fig. 7).

The extraction of scrolling tickers, however, was more complex and required manual intervention. We leverage the knowledge of fixed position and scrolling speed of the ticker for each channel. Using this information, our algorithm extracts the non-overlapping parts of the ticker. This is referred to as an early-fusion recognition technique, as portions of the images are joined together before recognition to produce a complete news ticker. In contrast, the late-fusion approach performs recognition on the individual frames, as they are, and instead, joins the recognized text output to reach the same goal from a different path. While the late-fusion approach conserves memory on the run-time, it may run into time-lag issues if the recognition process is not real-time. We, however, use the early-fusion approach for an entirely different reason of generating a data set, for which tickers needed to be extracted and joined before recognition (Fig. 8).

The two algorithms are described below:

Algorithm 1: Static Ticker extraction

Input: *video*
Output: cropped text lines
#reading the video frames
frames = **read_video**(*video*)
frameHeight, *frameWidth* = *frames*.shape()
#filtering the frames (picking 1 from every 250 frames)
fltrFrames = *frames* [start=1: end: step=250]
#getting bounding boxes from EAST and cropping
WBBs = [ ]; #word bounding boxes
for *thisFrame* **in** *fltrFrames*:
    WBBs.append(**get_wbb_from_east**(*thisFrame*))
    #filtering based on height, width
    for *wbb* **in** *WBBs*:
        if width(*wbb*) $> \frac{frameWidth}{3}$ and
          height(*wbb*) $< \frac{frameHeight}{8}$:
            save(**crop**(*wbb*, *thisFrame*))
            #crop function returns a cropped block

Algorithm 2: Scrolling Ticker extraction

Input: *video*, *scrollSpeed*, *offset*, *tickerBB*
#ticker *scrollSpeed* is fixed for a given channel
#*offset* is the position of the ticker delimiter in the zeroth frame
#*tickerBB* is the position coordinates of the ticker for a given channel, i.e. x, y, width, height
Output: cropped text lines
frames = **read_video**(*video*)
frameHeight, *frameWidth* = *frames*.shape()
nextFrame = *offset*
skip = $\frac{tickerBB['width']}{scrollSpeed}$
nonOverlappingFrames = *frames* [1: end: step=skip]
fullTicker = [ ];
for *f* **in** *nonOverlappingFrames*:
    fullTicker.append(**crop**(*tickerBB*, *f*))
ticker = [ ]
spaceTh = $\frac{frameWidth}{50}$
for *i*, *col* **in** **enumerate**(*fullTicker*.columns()):
    ticker.append(*fullTicker* [:, *col*])
    if **detect_white_columns**(*fullticker* [:, *i*:*i*+*spaceTh*]):
        save(*ticker*)
        ticker = [ ]

def ***detect_white_columns***(*ticker*):
    thresh = 70 to 120
    #*thresh* is chosen in consideration (direct proportion) with
    #the grayscale value of the ticker text color
    for *column* **in** *ticker*:
        if **all**([True **for** *pixel* **in** *column* **if** *pixel* $>$ *thresh*]):
            continue
        else:
            return False
    return True

In total, we have collected 20,097 tickers (Static = 16,638; Scrolling = 3,459) from a total of 41 news channels. We identified four different *Nastaleeq* fonts being used in the industry, and sorted our data set accordingly (Font1 = 12,352

[3]https://github.com/argman/EAST

tickers, Font2 = 672 tickers, Font3 = 6,029 tickers, Font4 = 1,044 tickers). In terms of the variations in the ticker types, our data set includes 118 different (based on position, texture, background color, text color, etc.) types of news tickers.

IV. Model Training and Results

Recurrent Neural Networks have proven to be an excellent model for sequence labelling problems. However, Recurrent Neural Networks used to suffer from an exploding and vanishing gradients problem, which was resolved by the introduction of a forget gate in the form of Long Short-Term Memory cell networks [12]. Moreover, sequence labelling through RNN requires explicit labelling of ground truth with the input image. This problem was solved by Graves et al. by the introduction of Connectionist Temporal Classification (CTC) layer [13] through which the LSTM output is aligned automatically without explicit marking. Since then, LSTMs have been used in various OCR problems for printed and handwritten text [14] [15] [16].

Moreover, Convolutional Neural Networks (CNN) have proven to be an excellent model for extracting features for complex problems such as Object localization, detection and segmentation. They have also found application in complex OCR problems such as scene text [17] and handwriting recognition [18]. Generally, a hybrid model based on a CNN followed by an LSTM and CTC layer (also known as a CRNN, Convolutional Recurrent Neural Network) has proven to give better results than a standalone LSTM and CTC model [19]. However, CNN based models are computationally very demanding due to the training of millions of parameters. News ticker text recognition is a relatively simpler problem than scene-text recognition, as the text is artificially over layed on top of a contrasting background. In light of its high-accuracy low-latency advantages [20], we opt for a 1-dimensional Bi-Directional LSTM (BDLSTM) model with a CTC layer. The model is implemented as part of TLSTM introduced by Naeem et al. [9] based on Tensorflow [21].

The model was trained with 256 LSTM cells in each of the forward and the backward layer. A learning rate of $1\text{x}10^{-4}$ was used and the images were normalized to a height of 48 pixels. The system was allowed to train until convergence in all the experiments. Font style-2 wasn't included in training in any of the experiments because of a lack of substantial representation in the data set (less than 1500 tickers). We segregate the dataset in terms of font styles, keeping 80% tickers from each font style for training and leaving 10% each for validation and testing. We evaluate the performance of our custom trained model on the test set against the commercially available Google Cloud Vision OCR. The outputs from the two sources were normalized (all white-spaces removed, for a fair comparison) and character-level percentage error rates (%CER) computed for each experiment. The results are summarized in Table II. As shown, our model outperforms the commercial OCR engine in two of the four experiments conducted. In Ex#1, our model suffers from high bias. As the number of training tickers are increased (in Ex#2 and Ex#3), the bias drops and our model is able to generalize better and, therefore, perform better than the commercial OCR engine. In Ex#4, the lack of representation of font style-2 in the training set, the limitation we aim to overcome by adding more relevant tickers into the dataset over time, is responsible for a higher error rate.

TABLE II: The 1-D BDLSTM RNN model trained on our custom dataset was able to outperform the commercial Google Vision OCR engine in two of the four experiments.

	Ex#1	Ex#2	Ex#3	Ex#4
fonts included in training	1	1 & 3	1 & 3 & 4	1 & 3
fonts included in testing	1	1 & 3	1 & 3 & 4	2
# train lines	7,300	11,700	15,840	11700
# validation lines	910	1,435	1,992	1435
# test lines	910	1,435	1,992	1326
Our approach CER	8.27%	**5.71%**	**6.98%**	13.19%
GV CER	**8.18%**	8.13%	8.07%	**8.40%**

Finally, we share our analysis on the effect of higher representation in the training set, in terms of tickers-per-channel coverage (Fig. 9) and ligature counts coverage in the training set (Fig. 10), resulting in lower error rates on the test set, as depicted in our experiments.

V. Conclusion

In this paper, we make a major contribution by publishing, to our knowledge, the first publicly available dataset for Urdu news ticker text recognition. We present our study on Urdu news ticker recognition and propose a ticker extraction and labelling scheme. Moreover, we also propose an efficient recognition system for Urdu news and conclude that a rigorously prepared dataset, with strong emphasis on diversity and ligature coverage, can play a major role in a strive to achieve lower error rates.

References

[1] Joan A. Sánchez, Verónica Romero, Alejandro H. Toselli, and Enrique Vidal. Icfhr2016 competition on handwritten text recognition on the read dataset. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 630–635. IEEE, 2016.

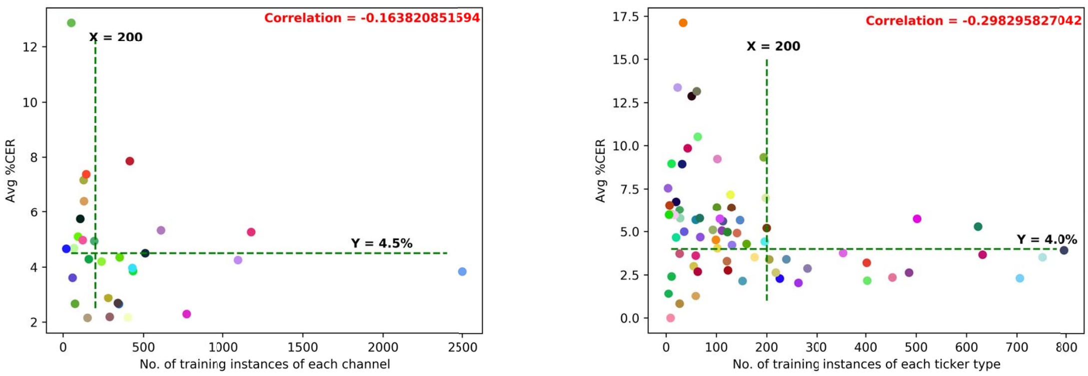
[2] Imran Siddiqi and Ahsen Raza. A database of artificial urdu text in video images with semi-automatic text line labeling scheme. In *The Fourth International Conferences on Advances in Multimedia (MMEDIA'12)*, 2012.

[3] Oussama Zayene, Sameh M. Touj, Jean Hennebert, Rolf Ingold, and Najoua E. B. Amara. Semi-automatic news video annotation framework for arabic text. In *Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on*, pages 1–6. IEEE, 2014.

[4] Oussama Zayene, Jean Hennebert, Sameh M. Touj, Rolf Ingold, and Najoua 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*, pages 996–1000. IEEE, 2015.

[5] Seiya Iwata, Wataru Ohyama, Tetsushi Wakabayashi, and Fumitaka Kimura. Recognition and connection of moving captions in arabic tv news. In *Arabic Script Analysis and Recognition (ASAR), 2017 1st International Workshop on*, pages 163–167. IEEE, 2017.

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



(a) Each colored dot represents the error rate and counts of one of the 30 channels included in the training set

(b) Each colored dot represents the error rate and counts of one of the 71 ticker types included in the training set

Fig. 9: Avg %CER plotted against the counts of each type of ticker included in the training set. As the trend indicates, the error rate drops with increase in the number of tickers of each type/channel in the training set

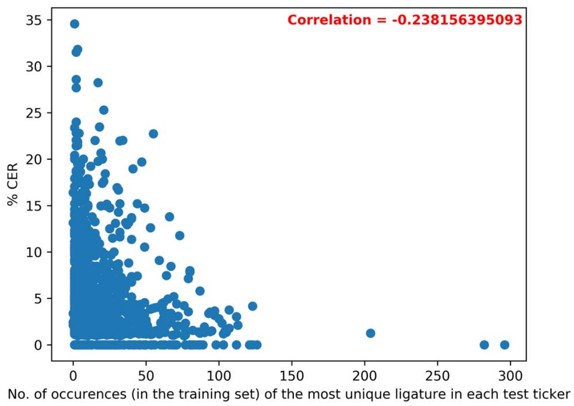


Fig. 10: Avg %CER plotted against the count of the most unique ligature (in the training set) for that ticker. This figure highlights the effect of ligature coverage in the training set resulting in lower error rates on the test set. Each blue dot represents a ticker in the test set. The trend in the plot verifies the hypothesis that ligature coverage lies at the core of solving the OCR problem for cursive scripts.

[7] Ibrar Ahmad, Xiaojie Wang, Ruifan Li, Manzoor Ahmed, and Rahat Ullah. Line and ligature segmentation of urdu nastaleeq text. *IEEE Access*, 2017.

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

[9] Muhammad F. Naeem, Noor U. Sehr, Aqsa A. Awan, Adnan Ul-Hasan, and Faisal Shafait. Impact of ligature coverage on training practical urdu ocr systems. In *2017 14th International Conference on Document Analysis and Recognition (ICDAR)*, 2017.

[10] Tetsuro Kuge. Wavelet picture coding and its several problems of the application to the interlace hdtv and the ultra-high definition images. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages III–III. IEEE, 2002.

[11] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. *arXiv preprint arXiv:1704.03155*, 2017.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling unsegmented sequence data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[14] Thomas M. Breuel, Adnan Ul-Hasan, Mayce A. Azawi, and Faisal Shafait. High performance OCR for printed English and Fraktur using LSTM networks. In *ICDAR*, Washington D.C. USA, aug 2013.

[15] Marcus Liwicki, Alex Graves, Horst Bunke, and Jürgen Schmidhuber. A novel approach to On-line handwriting recognition based on Bidirectional Long Short-Term Memory networks. In *Proc. 9th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 367–371, 2007.

[16] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pages 1764–1772, 2014.

[17] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[18] Zecheng Xie, Zenghui Sun, Lianwen Jin, Ziyong Feng, and Shuye Zhang. Fully convolutional recurrent network for handwritten chinese text recognition. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 4011–4016. IEEE, 2016.

[19] Saeeda Naz, Arif I. Umar, Riaz Ahmad, Imran Siddiqi, Saad B. Ahmed, Muhammad I. Razzak, and Faisal Shafait. Urdu Nastaliq recognition using Convolutional–Recursive Deep Learning. *Neurocomputing*, 2017.

[20] Adnan Ul-Hasan, Saad B. Ahmed, Faisal Rashid, Faisal Shafait, and Thomas M. Breuel. Offline printed Urdu Nastaleeq script recognition with Bidirectional LSTM networks. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1061–1065. IEEE, 2013.

[21] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale Machine Learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.