Urdu Optical Character Recognition

Author

Syed Wajih Ul Hassnain Shah

Regn Number

00000206949

Supervisor **Dr. Hasan Sajid**

Department of Robotics and Artificial Intelligence

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

August, 2021

Urdu Optical Character Recognition

Author

Syed Wajih Ul Hassnain Shah

Regn Number

00000206949

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machine Engineering

Thesis Supervisor:

DR. HASAN SAJID

Thesis Supervisor's Signature: _____

DEPARTMENT

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

AUGUST, 2021

FORM TH-4

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by:(Student Name & Regn No.) Syed Wajih Ul Hassnain Shah (00000206949)
Titled: Urdu Optical Character Recognition be accepted in partial fulfillment of the requirements for the award of Masters in Robotics and Intelligent Machines Engineering degree.

Examination Committee Members

1. Name: Dr. M. Jawad Khan Signature:_____

2. Name:Dr. Karam Dad Kallu Signature:_____

Supervisor's name: Dr. Hasan Sajid Signature:_____

Date: _____

_____ _____
Head of Department Date

COUNTER SINGED

Date:_____ _____
Dean/Principal

Declaration

I certify that this research work titled "Urdu Optical Character Recognition" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Syed Wajih Ul Hassnain Shah

00000206949

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

Syed Wajih Ul Hassnain Shah

Registration Number

00000206949

Signature of Supervisor

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by Mr. Syed Wajih Ul Hassnain Shah (RegistrationNo.00000206949),of Department of Robotics and Intelligent Machines Engineering (SMME) (School/College/Institute) has been vetted by under signed, found complete in all respects as per NUST Statutes/Regulation, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MBA/MS/MPhil degree. It is further certified that necessary amendments as pointed by GEC members of the scholar have also been incorporated in the said thesis.

Signature:_____

Name of Supervisor: Dr. Hasan Sajid

Date: _____

Signature (HOD):_____

Date: _____

Signature (Dean/Principal):_____

Date:_____

Acknowledgments

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which Your setup in my mind to improve it. Indeed, I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr. Hasan Sajid's help throughout my thesis and for Computer Vision and Machine Learning courses which he has taught me. I am also thankful to Afaq Ahmed, and Omer Zeb for their support and cooperation.

Finally, I would like to express my gratitude to Affan Zia for helping me out in write-up process.

Dedicated to my exceptional Mother whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

Existing commercial software such as ABBY and Google vision API provides support for Arabic and Urdu text. Still, accuracies are low because of the writing style of a non-Latin text. OCR for Urdu started way back in 2003 when a system could recognise isolated Urdu Characters only, but with the increase of data and digitisation research interest towards Urdu OCR increased. Other motivations include the Urdu data explosion in financial and economic sectors, including printed and handwritten scanned documents. Optical Character Recognition for Urdu is challenging due to the fact that being a non-Latin script it has a cursive writing style. These challenges need to be solved in different phases of the OCR system. This thesis presents an Urdu Optical Character Recognition (OCR) system and a data generation and encoding technique that is useful to standardise data for optical character recognition. The proposed model consists of a four staged network, and the first stage normalises the image while the second stage and the third stage is used for feature extraction and sequence generation. The final stage is the prediction stage which is responsible for predicting digitised text present in an image. The proposed algorithm is compared against baseline implementations of widely adapted supervised deep learning methods.

Key Words: *Optical Character Recognition (OCR), Deep Learning, Supervised Learning, Convolutional Neural Networks (CNN), Sequential Models, Connectionist temporal classification (CTC), Spatial Transformation Network (STN)*

Table of Content

Declaration IV
Plagiarism Certificate (Turnitin Report) V
Acknowledgments viii
Abstract X
CHAPTER 1 Introduction 1
CHAPTER 2 Literature Review 4
CHAPTER 3 DATA AND DATA PREPROCESSING 7
3.1. Dataset Generation 7
3.1.1 Dataset Distribution 10
CHAPTER 4: PROPOSED APPROACH 11
4.1. Transformation Stage 11
4.1.1 Spatial Transformation Network 11
4.1.1.1 Localisation Network 11
4.1.1.2 Image Sampler 14
4.2. Feature Extraction Stage 14
4.2.1 ResNet 15
4.2.2 Convolution layer 15
4.2.3 Pooling Layer 16
4.2.4 Skip connection or Identity function 17
4.2.5 Architecture of ResNet 18
4.3. Sequence Modeling Stage 20
4.3.1 Recurrent Neural Networks 20
4.3.2 LSTM Cell 22
4.3.3 Bidirectional LSTM 24
4.4. Prediction Stage 24
4.4.1. Attention Layer Mechanism 25
4.6. Loss Function 25
Chapter 5: Results 26
..... 27
5.1. Word Error Rate: 29
5.2. Qualitative results: 31
Chapter 6: Conclusion 32
References 33

Table of Figures

(a) non-joiners (b) joiners 8
Sample dataset with different backgrounds 10
(a) Identity transformation parameters. (b) Affine transformation grid 12
Convolution process of CNN 16
Illustrates average pooling and max pooling 17
Residual block of ResNet 18
Archietectuire of ANN 21
Shows RNN architecture 21
Illustrates LSTM cell 22
LSTM archietecture 24
Train Validation Curve for Attention Network. 27
Train, Validation curve for CTC 28
Accuracy comparison CTC and ATTN 29
Performance bar graph. 30
shows qualitative results 31

Table of Tables

(a) non-joiners (b) joiners 8
Sample dataset with different backgrounds 10
(a) Identity transformation parameters. (b) Affine transformation grid 12
Convolution process of CNN 16
Illustrates average pooling and max pooling 17
Residual block of ResNet 18
Archietectuire of ANN 21
Shows RNN architecture 21
Illustrates LSTM cell 22
LSTM archietecture 24
Train Validation Curve for Attention Network. 27
Train, Validation curve for CTC 28
Accuracy comparison CTC and ATTN 29
Performance bar graph. 30
shows qualitative results 31

CHAPTER 1 Introduction

The organisation of this thesis has been organised in chapters. **Chapter 2:** contains the literature review of the Urdu optical character recognition systems. **Chapter 3:** describes the process through which data has been synthesised for training purposes. **Chapter 4:** contains the proposed methodology for solving Urdu optical character recognition. **Chapter 5:** contains the results and experiments of the system.

As with many other technologies, the Urdu OCR (Optical Character Recognition) technology, which has a long and storied history and standard procedures and types, provides many benefits while also presenting major challenges [1]. Comparatively speaking, the recognition of Urdu text has only recently been tried [2], as compared to script recognition algorithms for other cursive and non-cursive scripts]. With great popularity comes widespread use of the Urdu language, which is written and understood in many countries across the world. According to sources, however, there has been little to no progress or achievement in identifying its script from the beginning of the project. Several factors contribute to the significant research lag, including inefficiencies across various areas, including dictionaries and research funding, and the provision of necessary equipment, benchmark datasets, and other key utilities.

In this thesis, we proposed a four-stage network that consists of the transformation stage, feature extraction stage, which is responsible for extracting important features and then sequential modeling stage, which generates sequence for the prediction stage to predict data. Secondly, we proposed a method for data generation since data acquisition for optical character recognition is a tedious task. This study aims to systematically evaluate the contribution of individual modules and propose previously ignored module combinations that improve the state of art.

As old as computerised document analysis and recognition itself, optical character recognition research has been going on for decades.Most of the document recognition challenges are ultimately aimed towards a complete OCR system that can ultimately transform the large collection of current paper documents into digital form in all types of variations. Optical Character Recognition systems for the Roman script have been extensively researched during the last few decades. Commercial OCR software today offers about 100% rates of language recognition based on the Roman script. Chinese and Arabic OCR research is also well developed with an acceptable excellent rates of recognition.

Some multilingual OCRs have also been created with the goal of proposing approaches that are generic in nature [3]. Despite these

remarkable advances, OCRs for many languages throughout the world are either non-existent or in the early stages of development, Urdu being one of them and being the topic of our study. Research on Urdu and comparable cursive scripts like Pashto, Farsi etc. is still available in its early days.

In the beginning, optical character recognition was conceived as a way to assist visually challenged individuals. During the period 1870 to 1931, two well-known devices, the Tauschek's reading machine and the Fournier Optophone, were created to assist the blind in reading [3]. According to the inventor, the machine known as Gismo, which was developed in the 1950s to translate written communications into machine codes, was utilised for computer processing during that time period. These gadgets were also capable of reading aloud text from a screen or keyboard. David H. Shepard, a cryptanalyst, and Harvey Cook collaborated on the development of the gadget. The first business to sell out of these optical character recognition systems was Intelligent Machines Research Corporation.

As a result of the development of passport scanners and price tag scanners in the 1980s, OCR technology advanced tremendously. Caere Corporation, Kurzweil Computer Products Inc., and ABBYY were founded in the late 1980s and early 1990s, respectively. The OCR technology has advanced dramatically between 2000 and 2017. The web OCR technology has been established, and various smartphone applications that enable the real-time translation of foreign languages have been developed. Hewlett Packard and the University of Nevada, Las Vegas together developed Tesseract, a prominent OCR engine. Adobe and Google Drive have also made several OCR software available online for free. Latin has been the focus of most OCR research during the past decade. Cursive scripts such as Arabic and Urdu were studied decades after Latin script was discovered.

Even while existing commercial software, such as ABBY and the Google vision API, provides support for Arabic and Urdu text, the accuracy of such material is low due to the writing style of a non-Latin language. OCR for Urdu was first attempted in 2003 using a system that could only identify single Urdu characters. However, as the amount of data and digitalisation rose, so did interest in Urdu OCR research. Other causes include the expansion of Urdu data in the financial and commercial sectors, which comprises printed and handwritten scanned papers as well as scanned images of documents.

There are three goals for this Thesis. They are as follows: (1) I hope that this work will serve as a learning path for readers who are interested in computer vision, pattern recognition, and artificial

intelligence to become acquainted with optical character recognition (OCR) technology and associated ideas. Ultimately, we believe that our work will have a major impact on each of these three disciplines taken together. Secondly, I believe that this thesis will provide an in-depth investigation of all of the previous and present technologies, strategies, and algorithms that may be used in various circumstances to identify Urdu text, which will be beneficial to Urdu OCR researchers in the near future. (3) I anticipate that this review will contribute to and extend the knowledge of the significance of optical character recognition technology among an even larger group of people. Also anticipated is that it will be used to identify knowledge gaps as well as possibilities for future research and development in this field.

CHAPTER 2 Literature Review

According to input acquisition (online or offline), writing style (handwritten or printed), font restrictions, script connection, and character or ligature segmentation, OCR systems can be classified into various types.

On the basis of input acquisition (online or offline), writing mode (handwritten or printed), font constraints (which may include font variations handled by a recognition system), and script connectivity (which may include a character or ligature segmentation or may completely avoid segmentation), the modes for an OCR system can be divided into several categories.

The majority of character recognition research has been done offline, using isolated and cursive analytical segmentation. For the development of Urdu OCR systems for isolated characters, some writers have chosen to employ offline recognition [10], 38]. They presented an algorithm for identifying individual Urdu characters. An Urdu font size-independent method was also developed and tested on single character ligatures in the Noori Nastalique font [11]. A similar method was presented by Nawaz et al. Basic pre-recognition techniques like picture pre-processing, segmentation (line and character), and the generation of XML files for training purposes were supplied by the system's general architecture. Urdu language text pre-processing, feature extraction, and classification were also discussed in [13]. "Soft converter" is an OCR that recognises isolated Urdu characters using a database, according to [12]. Two databases of pictures were created by Khan et al. [36] for offline recognition of Urdu text, called the "TrainDatabase" and the "TestDatabase." For offline printed Urdu characters, [14] also presented a representation and recognition scheme.

There are many variants in writing styles, which makes handwritten script identification a particularly fascinating and difficult field of pattern recognition. The identification of handwritten writing in characters like as Latin, Chinese, Arabic, and Japanese is the subject of extensive, in-depth research. As a result, much more study is needed in this area. Using numerous current state-of-the-art research, this study tries to discuss different improvements recorded over the previous few decades in the field of handwritten Indic scripts recognition For the offline recognition of handwritten Indic characters, this thorough review offers a transparent picture of several feature extraction and classification approaches. In this study, the most significant component is a systematic presentation of the reported works on handwritten Indic scripts, such as Devanagr, Bengali and Gurumukhi, as well as Kannada and Telugu. Based on what has been learned from the research, an analysis is conducted. In this paper, some

difficulties and obstacles connected to the recognition of Indic scripts are explored, which suggests some future study opportunities. An comprehensive research done in order to achieve the most accurate findings suggests the necessity for a combination of feature extraction and classification techniques. It has been suggested to build optimum convolutional neural network architecture using enhanced particle swarm optimization (PSO) method, which outperforms previous approaches.

When it comes to curly text, it may be a real challenge. As a result, numerous writers [6, [9], [15]–[18] have also worked on creating cursive analytical Urdu text OCR algorithms. Using segmentation and classification, Ahmad, et al. [15] created an OCR system that consisted of two primary components. The compound characteristics were segmented and classified. Similarly, Ahmad et al. [16] studied the features of Urdu script and developed a unique and robust technique to recognise printed Urdu script without the use of a dictionary. Similarly, [17] presented an offline recognition method for the Naskh script, which worked with segmented characters and divided them into 33 categories for recognition. According to [9], the distinctions between Naskh and Nastalique typefaces were examined, and an analytical segmentation-based methodology for pre-processing and identification of Urdu script was developed. For printed Urdu script, Pal and Sarkar [4] developed an OCR method. Multiple characteristics were used to segment and identify characters, with encouraging results. [4] and [5] are examples of similar approaches.

An algorithm's input can be reduced to a set of parameters known as features when it is too big to process. A feature vector is a grouping of features. It is necessary to extract unique features after preprocessing and segmentation, followed by classification and an optional postprocessing phase. Text features are extracted as a first priority in the feature extraction process. elements i.e. characters or words [6]. An OCR system's features are crucial since they can directly impact its efficiency and recognition rate [7]. The total number of features, quality of features, dataset along the classification method are said to contribute towards an effective OCR system [8]. [9] and [10] employ feature learning and feature engineering approaches, respectively, to extract features. Feature learning occurs when features are automatically discovered and extracted. Feature engineering refers to the process of identifying and extracting handcrafted features. We may need to minimise or choose a subset of original features after feature extraction, which is accomplished by feature selection.

By using feature learning to generate a high number of features from the data, classification algorithms may perform better overall. Specialized characteristics that cannot be developed by hand make these systems extremely desirable. To learn multi-level representations, unsupervised machine learning

methods are employed most often with feature learning. Supervised or unsupervised feature learning are both possible options. Supervised feature learning, such as supervised dictionary learning and neural networks, uses labelled input data. The labelled data enables the algorithm to learn when it fails to create the right labelled data set. However, unsupervised feature learning, such as matrix factorization, clustering [11], and auto-encoders , works with unlabeled input data.

The quality, quantity, utility, originality, and efficacy of handcrafted elements must be evaluated. In the Urdu alphabet, each letter is linked with a number of characteristics. A maximum identification rate may be achieved by employing the simplest and smallest characteristics in optical character recognition systems. Three sorts of hand-crafted characteristics can be distinguished [6], [12]

.

It has a top or bottom horizontal curve, branches, crossing points, horizontal lines, vertical lines, and a number of endpoints, among other things. [13];[14]; [15]; [16]; It needs knowledge of the character's structure or of the strokes and dots that make up the character. It is exceedingly difficult to extract structural features from Urdu characters since the form of each letter varies depending on its neighborhood.

CHAPTER 3 DATA AND DATA PREPROCESSING

3.1. Dataset Generation

The Urdu language has 58 characters in total and 40 basic characters. Urdu is derived from Arabic and Turkish and its script is cursive in nature. Urdu characters when written can come in three forms i.e isolated, initial, medial, and final because of which the same character can have different spatial shapes when written at different places. [17] [18] and [19] Different shapes of the same characters are shown in the Table 1.

Sr. No	Isolated	Initial	Middle	Ender
1	آ			
2	ا			ا
3	ب	بہ	بـ	ب
4	پ	پہ	پـ	پ
5	ت	تہ	تـ	ت
6	ٹ	ٹہ	ٹـ	ٹ
7	ث	ثہ	ثـ	ث
8	ج	جہ	جـ	ج
9	چ	چہ	چـ	چ
10	ح	حہ	حـ	ح
11	خ	خہ	خـ	خ
12	د			د
13	ڈ			ڈ
14	ذ			ذ
15	ر			ر
16	ڑ			ڑ
17	ز			ز
18	ژ			ژ
19	س	سہ	سـ	س
20	ش	شہ	شـ	ش
21	ص	صہ	صـ	ص
22	ض	ضہ	ضـ	ض
23	ط	طہ	طـ	ط
24	ظ	ظہ	ظـ	ظ

25	ع	عہ	عـ	ع
26	غ	غہ	غـ	غ
27	ف	فہ	فـ	ف
28	ق	قہ	قـ	ق
29	ک	کہ	کـ	ک
30	گ	گہ	گـ	گ
31	ل	لہ	لـ	ل
32	م	مہ	مـ	م
33	ن	نہ	نـ	ن
34	ں			
35	و			و
36	ہ	ہہ		ہ
37	ھ			
38	ء			
39	ی	یہ	یـ	ی
40	ے	یہ	یـ	ے
41	بھ			
42	پھ			
43	تھ			
44	ٹھ			
45	جھ			
46	چھ			
47	کھ			
48	گھ			

Table 1: *Urdu characters of varying shapes*

Similarly, Urdu characters are divided into two categories which are joiners and non-joiners. In the case of joiners, they can be joined from both sides i.e from left or right with other characters. [20], [21] In the case of non-joiners, they can be joined from only the right side; they would not join with other characters to their left. Joiners and non-joiner characters are given below.

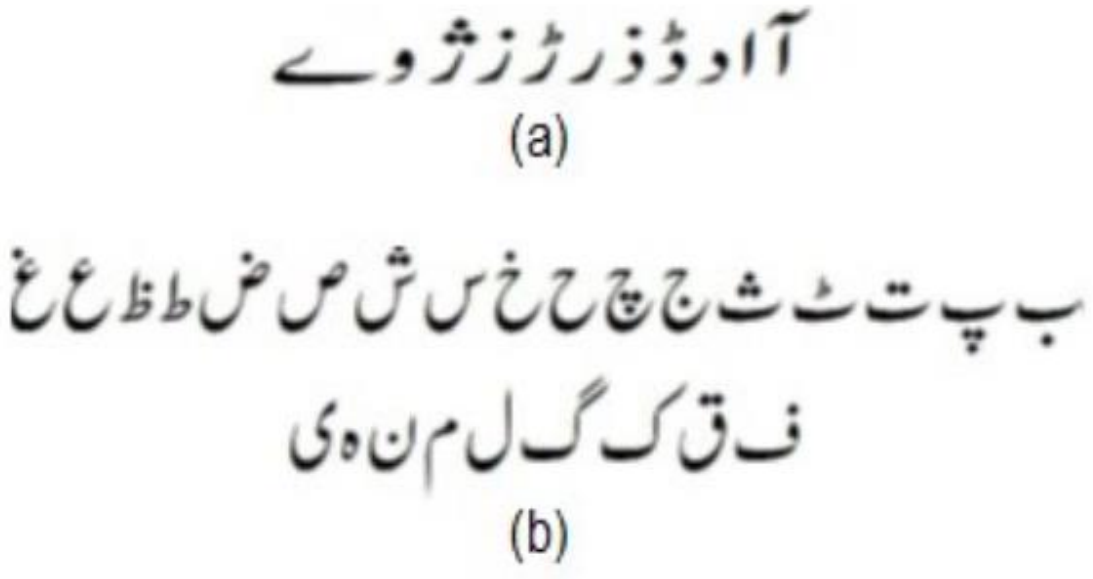


Figure 1: *(a) non-joiners (b) joiners*

As it happens in any script, Urdu script also has many fonts. They vary from each other in shape and writing style. So the data set must have some kind of representation for each font. The most commonly used font in the Urdu language is Jameel Noor Nastaliq. Different types of Urdu fonts are given below in the following table.

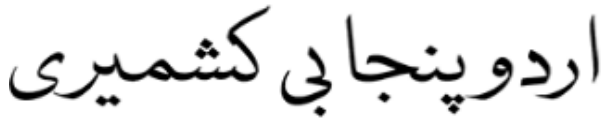
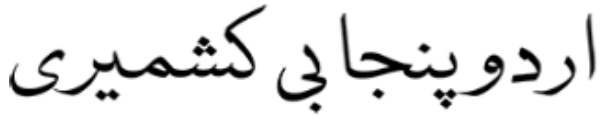
Font type	***Font Image***
Nafees Naskh	
Nafees Nastaleeq	
Nafees Pakistani Naskh	
Tahoma	

Table 2: *Types of Urdu fonts*

Since the algorithm has been developed in a supervised manner so we needed labelled data. Labelling data is a tedious and time-consuming task. So a library for synthetic text image generation has been developed which uses raqm and python PIL libraries. Raqm works in the backend because it provides support for bi-directional language scripts such as English and Urdu. Text data has been scrapped from different news sites after data scrapping unique words have been segmented out so that the balanced dataset can be synthesized. Each word has been synthesized 1000 times. To make data more representative of real-world data. Different types of distortions have been used which are given below.

- Sine Wave.
- Rotation
- Translation
- Noise addition
- Background images.

Some of the sample data has been given below.

Figure 3: *Sample dataset with different backgrounds*

3.1.1 Dataset Distribution

The distribution of the dataset has been given in the following table.

Set	**Number of Images**
Train	1000000
Validation	10000
Test	2000

Table 3: *Dataset Split*

CHAPTER 4: PROPOSED APPROACH

4.1. Transformation Stage

The transformation stage consists of a spatial transformation network whose job is to transform an input image X into the image $\tilde{X}$ which is a normalized image. Urdu text has a lot of variations and comes in diverse shapes. Sometimes an image that is fed into the network has variations in padding and font size and spatial placement. If such input images are passed into the network without any pre-processing, the feature extractor network (CNN) has to learn all these spatial invariances. To overcome this issue, a variant of the spatial transformation network (STN) has been used to reduce this burden, [5].

4.1.1 Spatial Transformation Network

In deep learning, a spatial transformer network is a separable module that conducts an orthogonal spatial transformation on a feature map. The module is conditionally applied to a single input map and produces a single output map. The resultant warping is applied to each channel of a multi-channel input signal.

An architectural building block for a spatial transformer network comprises three parts: The first of these elements is the localization network, which transmits the input feature map via hidden layers and outputs the parameters as a result of this transmission. This information is used to build a sampling grid based on the characteristics discovered from the sample grid. The converted result is created using the example grid that was previously mentioned. A grid generator handles the whole process. When a sampler is invoked, it takes in as input the sampling grid and the input function map and generates the output map by sampling points from the input feature maps in grid points. There are three main modules in space transformer networks, each of which is discussed in depth in the sections below.

4.1.1.1 Localisation Network

During the learning process of parameters that are used to create a grid, the localization network is responsible. Feature maps with width, height, and channel are provided as input to the localization network (i.e. $U \in R^{H \square W \square C}$), which returns parameters θ of the transformation T_θ that was performed to the feature map given below

$$\theta = f_{loc}(U) \qquad (4.1)$$

In certain cases, the size may change depending on the transformation type provided; it can be 6-dimensional just like affine transformation. The localization network function can have any architecture or shape, such as a fully connected or convolutional network, but it must contain a final regression layer to give the transformation parameters so that it can be used by the grid generator to generate the grid.

4.1.1.2 Parameterised Sampling Grid

When a sampling kernel is used to perform warping of the input feature map, each output pixel is computed by utilizing a sampling kernel that is located at a certain point in the input feature map. Pixel is considered as an element of a feature map that is generic, rather than as an image. The output pixels are typically defined to be located on a regular grid $G = \{G_i\}$ of pixels $G_i = (x_i^t, y_i^t)$, resulting in an output feature map $V \in R^{H' \times W' \times C}$ where H' and W' are the grid's height and width, respectively, and C is the number of channels, input and output image has the same number of channels.

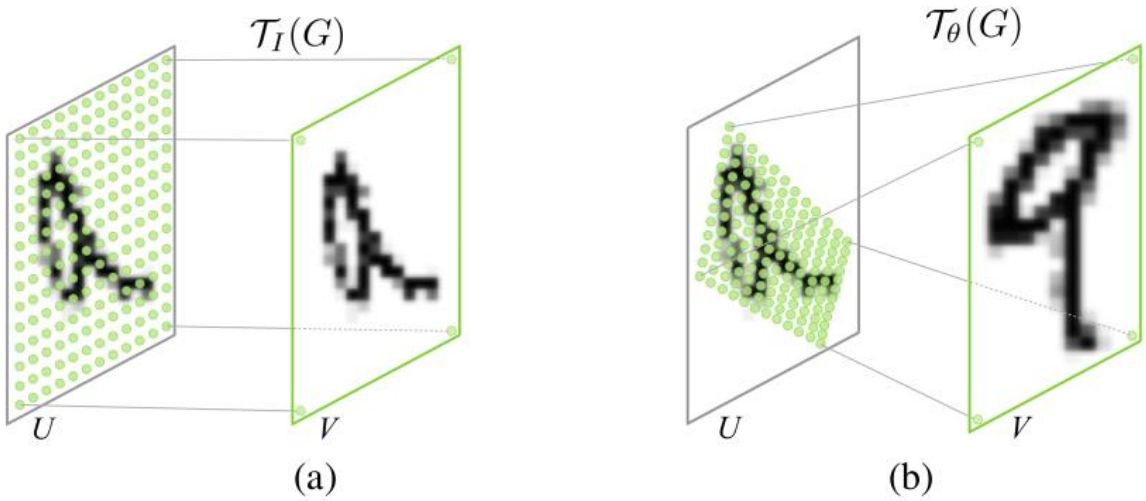


Figure 4: *(a) Identity transformation parameters. (b) Affine transformation grid*

Figure 4-1 illustrates two examples of how the sampling grid (parameterized) may be used to a visual image U to get the output V.

(a) $G = T_I(G)$ represents the sample grid, with I representing the identity transformation parameters.

(b) The sample grid warps the normal grid by applying an affine transformation T =(G) on the data.

Take it for the time being that T_θ is a 2D transformation which is affine in nature, just for the sake of simplicity and understanding The equation for affine transformation is presented below.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathrm{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \qquad (4.3)$$

where (x_i^t, y_i^t) are the target coordinates of the output feature map for the regular grid, is the source coordinates of the input feature map for the input sample points where A_θ is the affine transformation matrix.

The input coordinates are normalized coordinates, within the range $-1 \leq x_i^t, y_i^t \leq 1$ similarly output is inside the spatial boundaries, and $-1 \leq x_i^s, y_i^s \leq 1$. The transform specified allows translation, rotation, and scaling to be applied to the input feature map.

The localization network provides 6 parameters (the 6 components of A). Which are used to predict the sampling grid. The grid can be used for cropping because its area will be less than the range of (x_i^s, y_i^s). In comparison to the identity transform, the transformation performed using this grid can be seen in Fig. 4-1

T_θ may be a more limited class of transformations, such as that used for attention.

$$\mathrm{A}_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix} \qquad (4.4)$$

Instead of affine transformation other types of transformations can also be used such as projective transformation which has 8 parameters or it can be thin-plate spline transformation. As long as transformation is differentiable any type of transformation can be used, as long as backpropagation can be performed and parameters of transformation matrix can be optimized.

Allows crop, translation, and isotropic scaling by s, t_x, and t_y. The Transformation T_θ too can also be more broad, such as a projective transformation plane with 8 parameters, parts-like affinity, or a thin platform spline. Indeed, the transformation can take any parameterized form as long as it is differentiable with respect to the parameters. This is critical since gradients can be optimized.

4.1.1.2 Image Sampler

After computing the sampling grid $T_\theta(G)$we need to map these points to generate V. So the input to the image sampler is input features of the image and a sampling grid which will produce output image which later can be passed to feature extractor to compute prominent features.

Each coordinate (x_i^s, y_i^s)which has been computed using transformation matrix and input features corresponds to the input location and that location needs to be mapped at a spatial location to generate output feature map U. a Mathematical equation to perform this task can be represented by the following equation.

$$OV_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \theta_x) k\left(y_i^s - n; \theta_y\right) \forall i \in [1 \dots H'W'] \forall c \in [1 \dots C] \qquad (4.4)$$

In the above equation, θx and represent parameters of kernel $k()$. The above equation can be used to construct the output image, and this task can be performed using any interpolation technique. For our methodology bilinear interpolation was used.

The above method gives plug and plays mechanism parameters of which can be optimized using a backpropagation algorithm i.e gradient descent. Spatial transformer networks are usually small networks that can be easily implemented on any small GPU.

4.2. Feature Extraction Stage

After normalization of data i.e output from the spatial transformation network (STN) is passed on to the Convolutional neural network which generates feature vector i.e $\{V_i\}$ where $i = 1, \dots\dots, l$ where l represents numbers of columns of the feature vector. These feature extractors are the representation of spatial information about the characters which are written in the input image. This feature represents the information along the horizontal line and is distinguishable from each other. Many architectures of convolutional neural networks have been implemented which are listed below.

- Visual Geometry Group (VGG)
- Region-Based Convolutional Neural Networks (RCNN)
- Residual Networks (ResNet)

Out of all networks which are mentioned above, ResNet performed the best because it has skip connections that solve the vanishing gradient problem.

ResNet will be described in the following section.

4.2.1 ResNet

It has been established that ResNet performed best in object recognition problems and also it serves best in the role of feature extractor. The reason behind it is that it is made of residual blocks which solve the vanishing gradient problem. Hence it was decided that ResNet will be used as a feature extractor.

The residual block of ResNet comprises the following units.

- Convolution layer.
- Pooling layer
- Skip connection or identity function

All of these functionalities of ResNet will be defined below.

4.2.2 Convolution layer

Convolution is an image processing procedure that comprises a kernel ki.e it is a window that has some height and width and usually it has an odd dimension. The whole idea of convolution is to reduce the number of features and leave out only features that are sharp and important. This helps in solving the dimensionality curse of the input data. Mathematically it is defined as follows.

$$I_{new} = \sum k.I \qquad (4.5)$$

k is the kernel of convolution I is the input image and $I_{new}$is the output of the convolution layer. The following figure shows the convolution process.

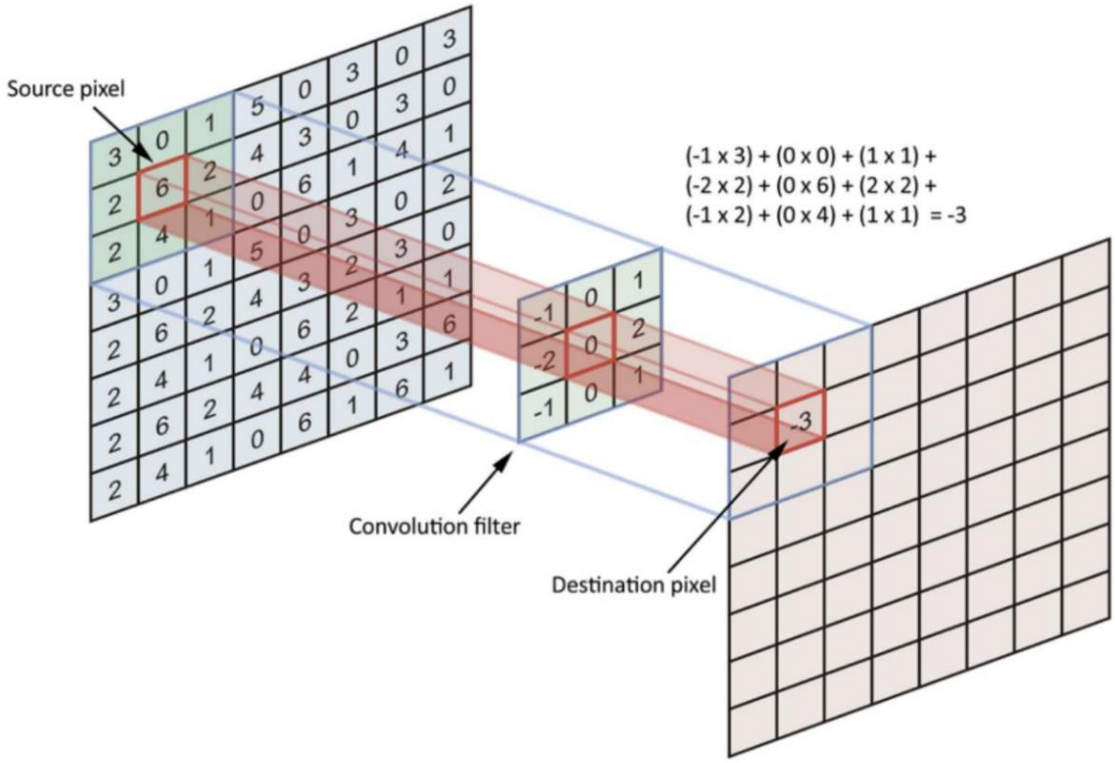


Figure 5: *Convolution process of CNN*

The output of the convolution layer is usually passed on to the max-pooling layer or activation function. The idea behind the convolution layer is that it learns high-level features leaving out low-level features. The output of convolutions is passed on to Rectified Linear activation unit. The purpose of ReLU is that it induces non-linearity in the input image.

4.2.3 Pooling Layer

The purpose of the max-pooling layer is to reduce the number of features resulting in the reduction of input features thus making the training process less computationally expensive and makes it more efficient.

There are different types of pooling. Average pooling and max-pooling functions are the most popular ones. The pooling layer also comprises of a kernel k of $N \times N$ size. In max-pooling maximum number in the window is picked and the rest are discarded while in average pooling, an average of all numbers in a window is taken and that average is passed on to the next layers. The pooling process is visually described in the following figure.

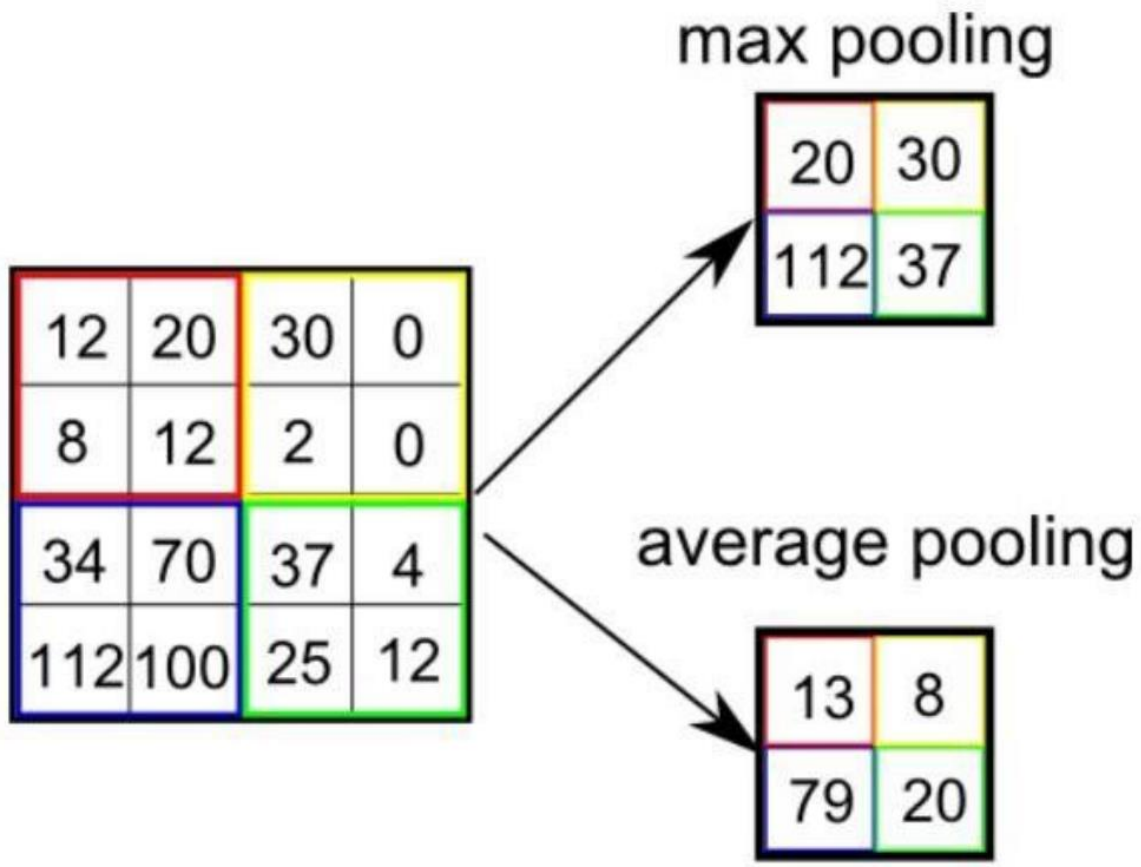


Figure 6: *Illustrates average pooling and max pooling*

The average pooling and convolution process is usually performed in an alternative manner which results in the reduction of dimension. After the final pooling or convolution layer output is flattened out and is passed on to the fully connected layer.

In ResNet max-pooling is used and output is passed on to the fully connected layer which learns a high-level representation of features that are learned through convolutions.

4.2.4 Skip connection or Identity function

As the neural network goes deeper, it induces the problem of vanishing gradient i.e skip connection passes the output from the previous layers and adds it up with succeeding ones. Which allows convolution neural networks to solve vanishing gradient problems. The following figure illustrates this process.

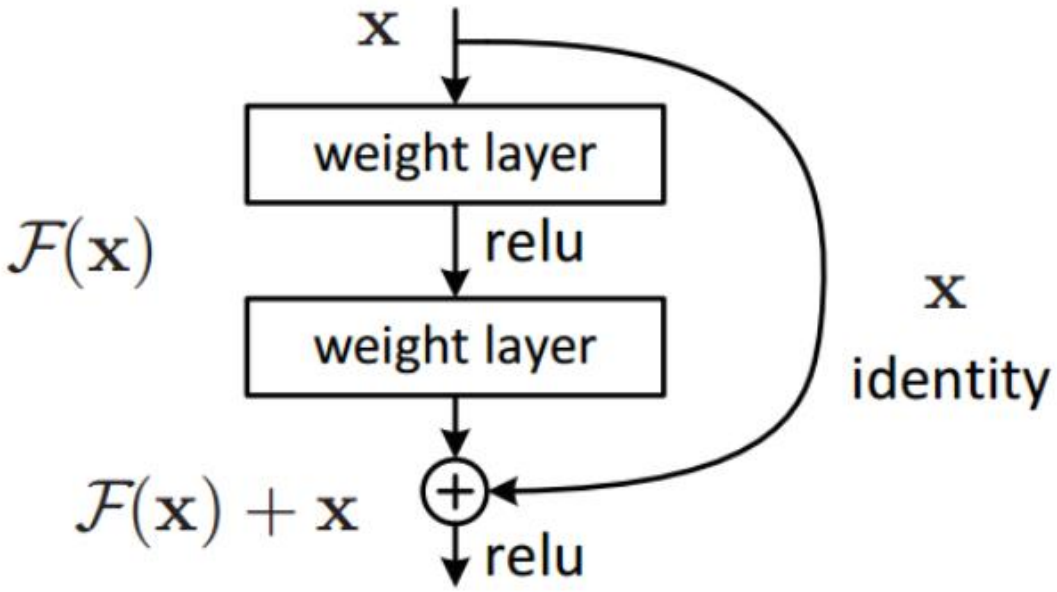


Figure 7: *Residual block of ResNet*

$$H(x) = F(x) + x \tag{4.6}$$

If

$$F(x) = 0 \tag{4.7}$$

Then

$$H(x) = x \tag{4.8}$$

The above equations show that if $F(x)$ goes to zero it does not hurt the learning process at all because it will pass on the weights of previous layers. Hence ResNets are capable of learning even when we have very deep convolutional neural networks.

4.2.5 Architecture of ResNet

The trainable layers ResNet module has been 29 in total. The dimension of the output layer is 512 X 187. The architecture of our ResNet module is shown in the following table.

Layer	**Configuration**	**Output**
Input	Gray Scale	100X32
Conv1	c: 32 k: 3 × 3	100 × 32
Conv2	c: 64 k: 3 × 3	100 × 32
Pool1	K: 2×2 s: 2×2	50 × 16
Block1	C:3, k: 128x128 C:3, k: 128x128	50 × 16
Conv3	C:3, k: 128x128	50 × 16
Pool2	K: 2×2 s: 2×2	25 × 8
Block2	C:256, k: 3 × 3 C:256, k: 3 × 3	25 × 8
Conv4	C:256, k: 3 × 3	25 × 8
Pool3	K: 2×2 S: 1×2 p: 1×0	26 × 4
Block3	C:512 k: 3 × 3 C:256, k: 3 × 3	26 × 4
Conv5	C:512, k: 3 × 3	26 × 4
Block4	C:512, k: 3 × 3 C:512, k: 3 × 3	26 × 4
Conv6	c: 512 k: 2 × 2 s: 1 × 2 p: 1 × 0	27 × 2
Conv7	c: 512 k: 2 × 2 s: 1 × 1 p: 0 × 0	187 × 1

Table 4: *Architecture of Feature Extractor*

4.3. Sequence Modeling Stage

The feature vectors which have been extracted by the feature extraction network have been passed on to Bidirectional LSTM. Before passing it on to the BiLSTM it has been reshaped to series of sequences so that BiLSTM can learn from it.

Input to BiLSTM is

$$V_i \in V \tag{4.9}$$

Where V_i is the ith feature vector i. column from the features V extracted by the feature vector. This vector is passed on to the Bidirectional LSTM which generates a sequence given below.

$$H = Seq(V) \tag{4.10}$$

The variant of LSTM which is Bidirectional has been explained in the following section.

4.3.1 Recurrent Neural Networks

Recurrent Neural networks have been designed to learn temporal information. Artificial neural networks and Recurrent Neural Networks are kind of similar but the main difference between them is that ANNs are feed-forward connections while Recurrent neural networks have backward connections pointing backwards. The architecture of ANN and RNN are given below.

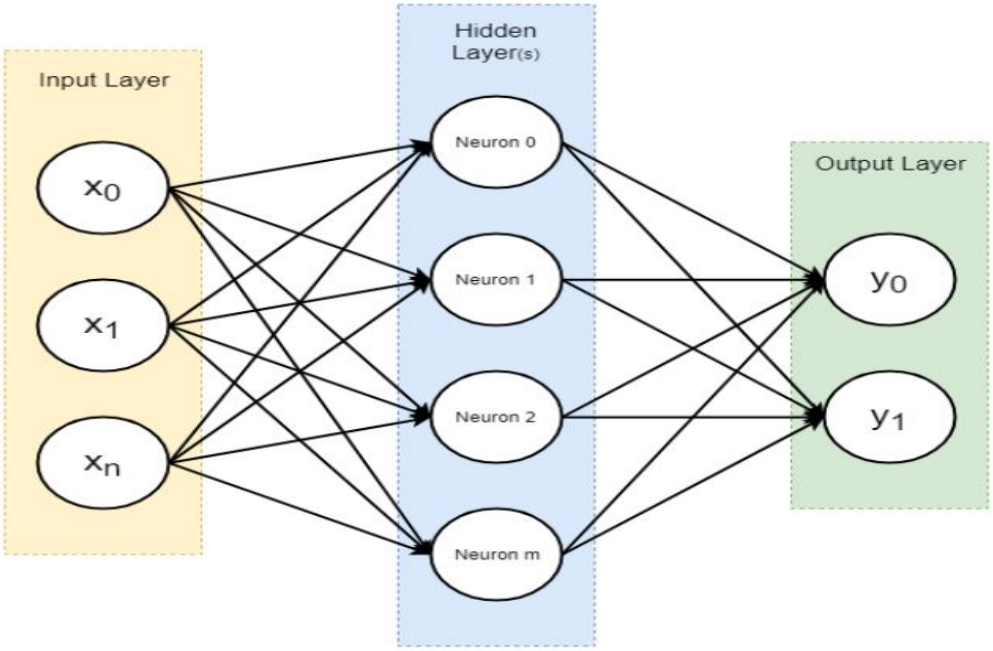


Figure 8: *Archietectuire of ANN*

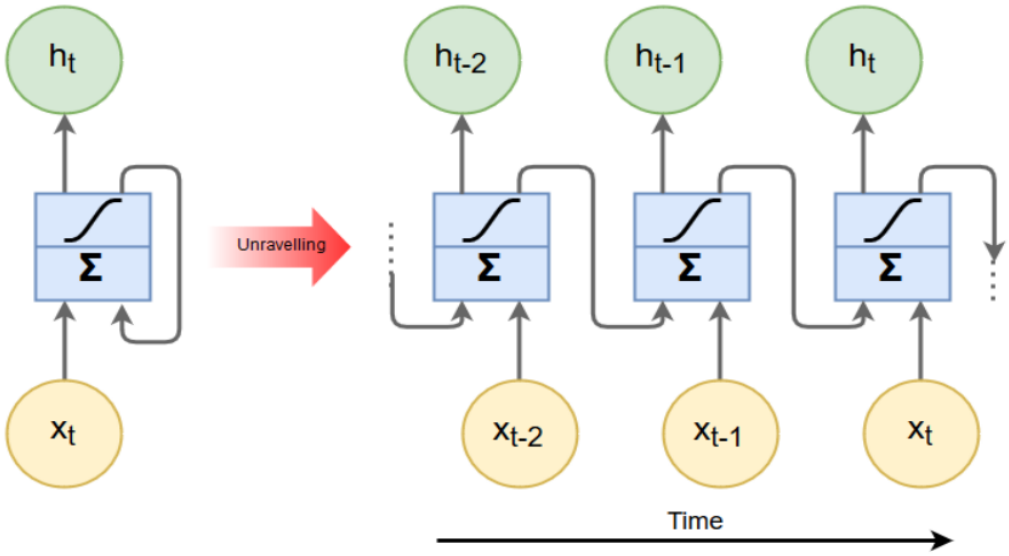


Figure 9: *Shows RNN architecture*

The above figure shows RNN cells that are stacked together are connected side by side. If we look closely each RNN cell has two inputs, one is the input data for that current time step $x_t$and the other input is the output from the previous cell from the previous time step i.e h_{t-1}. Unfortunately, RNNs have a vanishing gradient problem i.e is when sequences tend to go long there is a problem of gradients vanishing away. For that reason, instead of using RNN cell LSTM cell was used in a BiLSTM which has been explained below.

4.3.2 LSTM Cell

LSTMs are special kinds of RNN that are capable of solving problems that have long-term dependencies as described in [22] It is capable of handling such problems because it can retain information for long periods of time. [23], [24] and [25] LSTM cell is illustrated in the following figure.

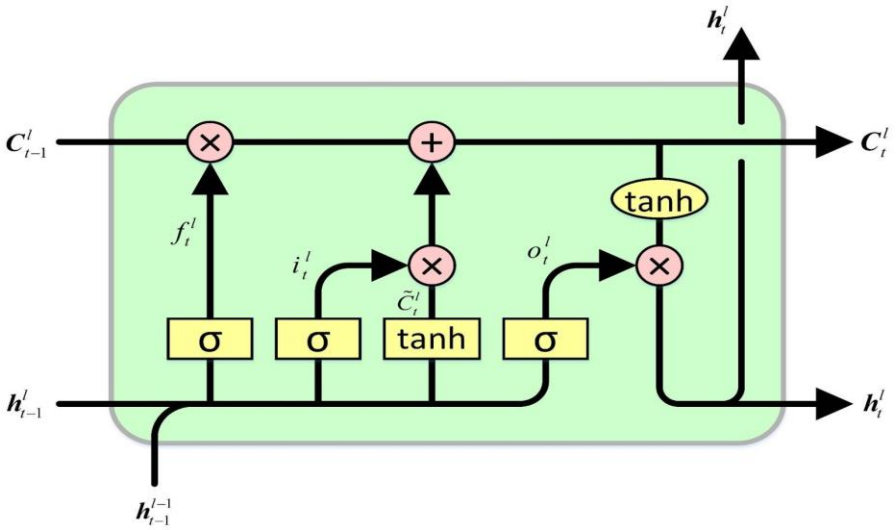


Figure 10: *Illustrates LSTM cell*

The most important module of the LSTM cell is the cell state which is capable of retaining information from the previous time step or deleting information from the previous time step. It performs this task using a series of linear operations with different kinds of gates. These gates have activation functions that are called sigmoids. Sigmoid function has output value between 0 and 1 [26]. 0 means do not pass any information forward while 1 means pass all the information. These gates will be discussed below. The first gate is called forget gate which decides whether to retain previous cell information or to pass it on. The equation of which is given below.

$$f_t = \sigma \tag{4.11}$$

Where f_t is called output of a forget gate for a given time step t. This is multiplied by the cell state of the previous time step C_{t-1}.

In the next step, it is decided which information will be stored in the cell state next. This operation will be performed in two parts. The first part is called the sigmoid layer called the input gate, and the second part is called the tanh layer. These two operations will be multiplied together. Equations are given below.

Input gate

$$i_t = \sigma \tag{4.12}$$

Tanh layer.

$$\check{C}_t = tanh \tag{4.13}$$

The new cell state will be calculated by the following equation which is composed of multiplication and addition.

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \tag{4.14}$$

The next step is to calculate the output for the next LSTM cell. Which will be calculated by the following equation.

$$o_t = \sigma \tag{4.15}$$

$$h_t = o_t * tanh(C_t) \tag{4.16}$$

Now we have a cell state and output for the next LSTM cell.

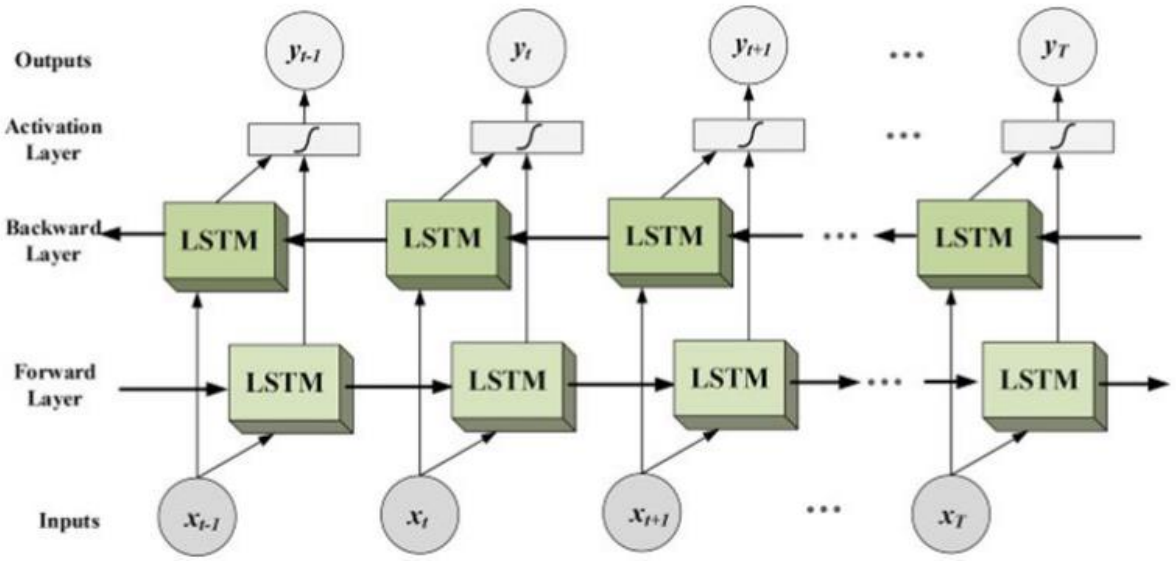


Figure 11: *LSTM archietecture*

4.3.3 Bidirectional LSTM

Bidirectional LSTM is a variant of an LSTM which performs better in sequence learning problems. Each BiLSTM has two hidden layers comprising LSTM cells. The first layer learns the sequence as it is while the second learn on the mirror copy of the original sequence. BiLSTM architecture has been displayed in the Figure 11.

In this thesis, we implemented a Bidirectional LSTM network which has two layers with two 256 hidden states. Each layer has two hidden layers that are the forward layer and the backward layer.

4.4. Prediction Stage

This module is responsible for predicting the output of the Optical Character Recognition system. And for this module, an Attention layer is written to perform this task. Input to the attention layer is a vector H which has been predicted by the BiLSTM module and it outputs the sequence of characters that is Y. The speciality of the attention module is that it is capable of learning character-level language models. This module has been explained in the following section.

4.4.1. Attention Layer Mechanism

This module is responsible for predicting the output of the Optical Character Recognition system. The equation is given below for a given time step t.

$$y_t = softmax(W_o s_t + b_o) \quad (4.17)$$

Trainable parameters in the above equations are W_o and b_o of an attention layer and s_t is a hidden state of LSTM.

$$s_t = LSTM(y_{t-1}, c_t, s_{t-1}) \quad (4.18)$$

In the above equation, c_t is a context vector that can be calculated by the following equation.

$$c_t = \sum_{i=1}^{I} \alpha_{ti} h \quad (4.19)$$

The final layer dimension of LSTM was 256.

4.6. Loss Function

The training process is performed by minimizing the loss function [27]. The loss function which was chosen for this system is log-likelihood. Equation of which is given below.

$$O = -\sum_{X_i, Y_i \in TD} \log p(Y_i \mid X_i) \quad (4.20)$$

The loss is calculated by this function using the Input image and its label. The advantage of this approach is that the system is end-to-end trainable.

Chapter 5: Results

Results have been displayed in the form of training curve, evaluation curve and Test accuracies. Another metric has been used called word error rate. These metrics have been calculated for two networks that were trained out of many. One of them is a Spatial transformer network which calculates input for ResNet and then a BiDirectional LSTM which takes feature maps from the feature extraction network. And the last layer is the decoding layer which is different in two networks. One network has a CTC loss and decoding layer. And the other network has an attention mechanism. Figure 11 shows training and validation loss for the STN_ResNet_BiLSTM_attn network while Figure 5.2 shows training and validation loss for the STN_ResNet_BiLSTM_ctc network. Similarly, Figure Figure 5.3 shows the accuracy of these two networks.

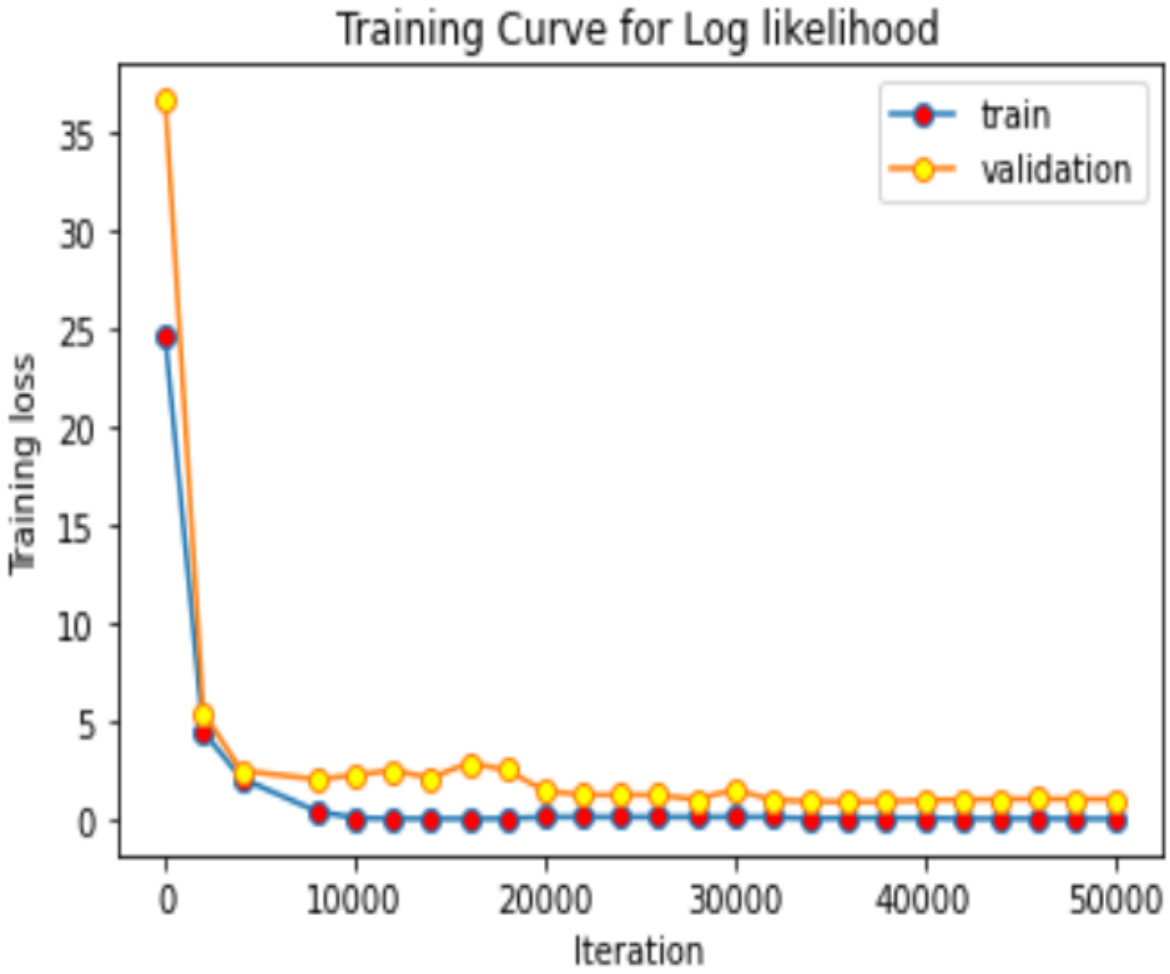


Figure 12: *Train Validation Curve for Attention Network.*

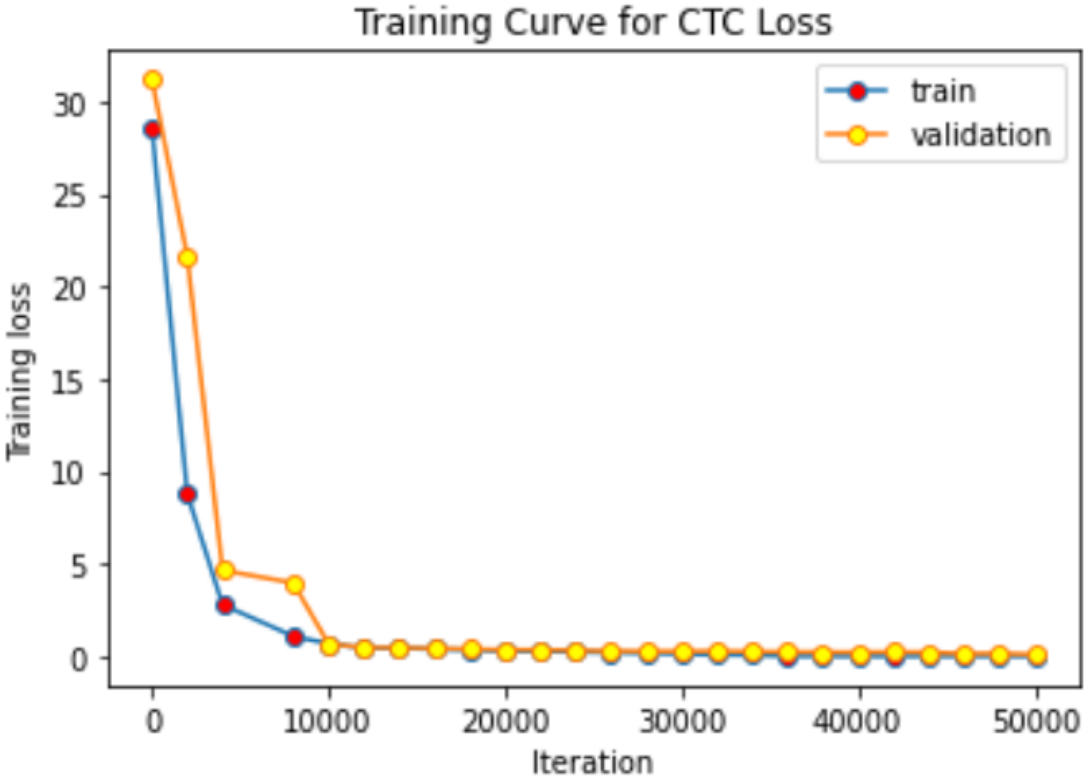


Figure 13: *Train, Validation curve for CTC*

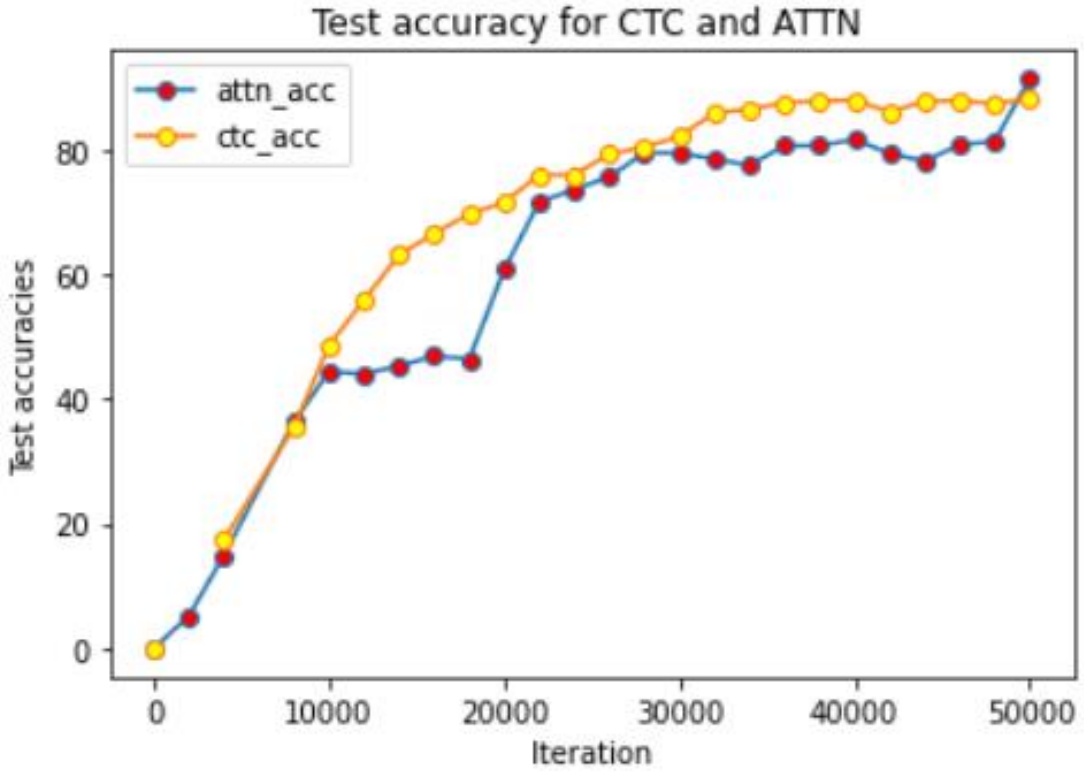


Figure 14: *Accuracy comparison CTC and ATTN*

5.1. Word Error Rate:

Word Error rate is given by the following formula.

$$Word\ error\ rate = \frac{S+D+I}{N} \tag{5.1}$$

Where

- S stands for substitutions.
- I, Stands for insertions.
- D, Stands for deletions
- N, Stands for a number of words.

Figure 14 shows all the performance metrics including word error rate.

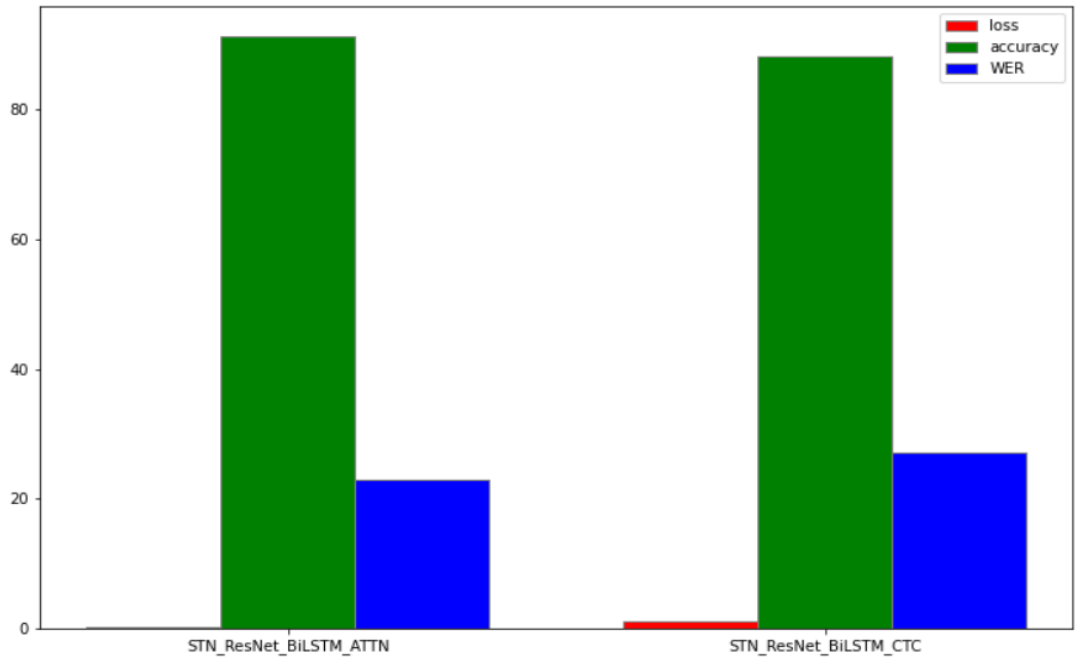


Figure 15: *Performance bar graph.*

Performance metrics have been shown in the tabular form in the following table 5-1.

	Train loss	**Validation loss**	**Test accuracy**	**WER**
STN_ResNet_BiLSTM_attn	0.00009	0.8985	91.363	37
STN_ResNet_BiLSTM_ctc	o.00095	0.10450	88.165	43
CRNN	1.73	4.25	51.6	58

Table 5: *Performance metrics in the tabular form*

5.2. Qualitative results:

The table below shows qualitative results. Data has been cropped from the original national identity cards. The model did pretty well on the unseen data.

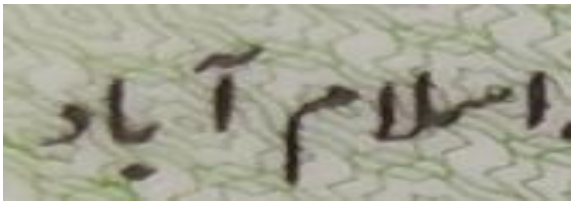
Image	Prediction
	ضرار
	فضہ
	اسلام آباد
	کیاآپ کا
	حکومت پاکستان

Figure 16: *shows qualitative results*

Chapter 6: Conclusion

Optical Character recognition is a data-centric problem that requires a lot of labelled data. In this thesis we were able to propose an approach through which we were able to generate synthetic data and using supervised learning we developed a deep learning model including a sequential model and we were able to get encouraging results. We performed inference on the real-world data and the model performed well on that. So we conclude that even with the scarcity of data we can develop an optical character recognition algorithm.

There is a huge prospect of research in the field of data segmentation of optical character recognition scenes. The performance of OCR is highly dependent on the segmentation algorithm. So further research is needed in this domain.

References

[1] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proc. IEEE*, vol. 80, no. 7, pp. 1029–1058, Jul. 1992, doi: 10.1109/5.156468.

[2] S. Naz, K. Hayat, M. Imran Razzak, M. Waqas Anwar, S. A. Madani, and S. U. Khan, "The optical character recognition of Urdu-like cursive scripts," *Pattern Recognit.*, vol. 47, no. 3, pp. 1229–1248, Mar. 2014, doi: 10.1016/j.patcog.2013.09.037.

[3] P. Romulus, Y. Maraden, P. D. Purnamasari, and A. A. P. Ratna, "An analysis of optical character recognition implementation for ancient Batak characters using K-nearest neighbors principle," in *2015 International Conference on Quality in Research (QiR)*, Aug. 2015, pp. 47–50. doi: 10.1109/QiR.2015.7374893.

[4] U. Pal and A. Sarkar, "Recognition of printed Urdu script," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, Aug. 2003, pp. 1183–1187. doi: 10.1109/ICDAR.2003.1227844.

[5] Q. Akram, S. Hussain, and Z. Habib, "Font Size Independent OCR for Noori Nastaleeq," 2009. https://www.semanticscholar.org/paper/Font-Size-Independent-OCR-for-Noori-Nastaleeq-Akram-Hussain/d82014e3f67e35f27d29bef6530a9e326b2a286d (accessed Aug. 13, 2021).

[6] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, "Handwritten Urdu Character Recognition using 1-Dimensional BLSTM Classifier," p. 10.

[7] "MQDF Discriminative Learning Based Offline Handwritten Chinese Character Recognition | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/6065480 (accessed Aug. 13, 2021).

[8] I. Ansari, "Automatic Recognition of Offline Handwritten Urdu Digits In Unconstrained Environment Using Daubechies Wavelet Transforms," *IOSR J. Eng.*, vol. 03, pp. 50–56, Sep. 2013, doi: 10.9790/3021-03925056.

[9] N. Das *et al.*, "Recognition of Handwritten Bangla Basic Characters and Digits using Convex Hull based Feature Set," *ArXiv14100478 Cs*, 2009, doi: 10.13140/2.1.3689.4089.

[10] "An Efficient Recognition Method for Handwritten Arabic Numerals Using CNN with Data Augmentation and Dropout," *springerprofessional.de*. https://www.springerprofessional.de/en/an-efficient-recognition-method-for-handwritten-arabic-numerals-/17595078 (accessed Aug. 13, 2021).

[11] "A Survey of Handwritten Character Recognition with MNIST and EMNIST | Semantic Scholar." https://www.semanticscholar.org/paper/A-Survey-of-Handwritten-Character-Recognition-with-Baldominos-S%C3%A1ez/c94145d960d8f77cbf820a4cf814d33ec486a420 (accessed Aug. 13, 2021).

[12] "Convolutional Neural Network Committees for Handwritten Character Classification | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/6065487 (accessed Aug. 13, 2021).

[13] H. E. Abed and V. Märgner, "Arabic text recognition systems - state of the art and future trends," 2008. doi: 10.1109/INNOVATIONS.2008.4781781.

[14] H. Almuallim and S. Yamaguchi, "A Method of Recognition of Arabic Cursive Handwriting," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1987, doi: 10.1109/TPAMI.1987.4767970.

[15] L. D. Harmon, "Automatic recognition of print and script," 1972. doi: 10.1109/PROC.1972.8878.

[16] W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," 1959. doi: 10.1145/1460299.1460326.

[17] "[PDF] Urdu Optical Character Recognition System | Semantic Scholar." https://www.semanticscholar.org/paper/Urdu-Optical-Character-Recognition-System-Muaz-Hussain/e657954517a794818c3447d96bc9f58413e60812 (accessed Aug. 13, 2021).

[18] A. Wali and S. Hussain, "Context Sensitive Shape-Substitution in Nastaliq Writing System: Analysis and Formulation," 2007. doi: 10.1007/978-1-4020-6268-1_10.

[19] F. Shafait, J. V. Beusekom, D. Keysers, and T. Breuel, "Page Frame Detection for Marginal Noise Removal from Scanned Documents," 2007. doi: 10.1007/978-3-540-73040-8_66.

[20] K. Kise, A. Sato, and M. Iwata, "Segmentation of Page Images Using the Area Voronoi Diagram," *Comput Vis Image Underst*, 1998, doi: 10.1006/cviu.1998.0684.

[21] Y. Ishiyama, F. Kubo, H. Takahashi, and F. Tomita, "Labeling board based on boundary tracking," *Syst. Comput. Jpn.*, 1995, doi: 10.1002/scj.4690261406.

[22] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," *ArXiv190909586 Cs*, Sep. 2019, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/1909.09586

[23] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks," *ArXiv170102720 Cs Stat*, Jan. 2017, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/1701.02720

[24] "LSTM Explained | Papers With Code." https://paperswithcode.com/method/lstm (accessed Aug. 13, 2021).

[25] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[26] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *ArXiv181103378 Cs*, Nov. 2018, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/1811.03378

[27] K. Janocha and W. M. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," *ArXiv170205659 Cs*, Feb. 2017, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/1702.05659

Urdu Optical character recognition

ORIGINALITY REPORT

10%	5%	9%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Naila Habib Khan, Awais Adnan. "Urdu Optical Character Recognition Systems: Present Contributions and Future Directions", IEEE Access, 2018 Publication	4%
2	export.arxiv.org Internet Source	2%
3	papers.nips.cc Internet Source	2%
4	Reya Sharma, Baijnath Kaushik. "Offline recognition of handwritten Indic scripts: A state-of-the-art survey and future perspectives", Computer Science Review, 2020 Publication	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		