Bridging the Gap between Handwriting and Machine: AI-based Handwritten Text Recognition

Dhruv Shah
Student - Jain University
009 Trotakacharya Kutir
Kanakpura Road Bengaluru
560082

Sai Bhargav
Student - Jain University
House No. 20 1st cross
pavamma nagar Bengaluru
560083

Dhanush B.
Student - Jain University
1580, 11th main road
Jayanagar 4th T block, Bengaluru
560041

ABSTRACT
Due to the variety and complexity of handwriting styles, handwritten text recognition (HTR) is a difficult job. In HTR, artificial intelligence (AI) has demonstrated enormous promise, opening the door to the creation of effective and precise identification systems. An overview of the HTR process using AI methods, including preprocessing of the data, feature extraction, classification, and post-processing, is provided in this paper.

To get the input data ready for feature extraction, data preprocessing includes image enhancement and segmentation. To feed the classification model with useful features, such as character shapes and patterns, pertinent features are extracted from the input picture. A neural network is frequently used in the classification model to map the extracted features to the associated characters. Spell checkers and language models are two post-processing tools that can be used to improve recognition outcomes.

We discuss the challenges and opportunities for HTR using Neural Networks, including the importance of training data, model selection, and performance evaluation. The paper also outlines the methodology, design, and architecture of the Handwriting character recognition system and the testing and results of the system development. The aim is to demonstrate the effectiveness of neural networks for Handwriting character recognition.

General Terms
Computer Vision, Pattern Recognition, Neural Networks, Artificial Intelligence, HCR, Machine Learning.

Keywords
Handwriting Text Recognition, Artificial Intelligence, Machine Learning, Artificial Neural Networks, Image Processing, Support Vector Machine, Computer Vision.

1. INTRODUCTION
This project seeks to classify an individual handwritten word so that handwritten text can be translated into a digital form. We used two main approaches to accomplish this task: classifying words directly and character segmentation. For the former, we use Convolutional Neural Network (CNN) with various architectures to train a model that can accurately classify words. For the latter, we use Long Short-Term Memory networks (LSTM) with convolution to construct bounding boxes for each character. We then pass the segmented characters to a CNN for classification and then reconstruct each word according to the results of classification and segmentation.

Thus, a lot of important knowledge gets lost or does not get reviewed because documents never get transferred to digital format. We have thus decided to tackle this problem in our project because we believe the significantly greater ease of management of digital text compared to written text will help people more effectively access, search, share, and analyze their records, while still allowing them to use their preferred writing method. The aim of this project is to further explore the task of classifying handwritten text and to convert handwritten text into the digital format. Handwritten text is a very general term, and we wanted to narrow down the scope of the project by specifying the meaning of handwritten text for our purposes. In this project, we took on the challenge of classifying the image of any handwritten word, which might be in the form of cursive or block writing. This project can be combined with algorithms that segment the word images in a given line image, which can in turn be combined with algorithms that segment the line images in a given image of a whole handwritten page.

2. EXISTING SYSTEMS
The first driving force behind handwritten text classification was for digit classification for postal mail. Jacob Rabinow's early postal readers incorporated scanning equipment and hardwired logic to recognize mono-spaced fonts [3]. Allum et. al improved this by making a sophisticated scanner which allowed for more variations in how the text was written as well as encoding the information onto a barcode that was printed directly on the letter [4]. 3.2. To the digital age the first prominent piece of OCR software was invented by Ray Kurzweil in 1974 as the software allowed for recognition of any font [5]. This software used a more developed use of the matrix method (pattern matching). Essentially, this would compare bitmaps of the template character with the bitmaps of the read character and would compare them to determine which character it most closely matched with.

The downside was this software was sensitive to variations in sizing and the distinctions between each individual's way of writing.

3. PROPOSED SYSTEM
3.1 System Workflow
The Proposed Work Involves five main steps, assuming we already have a database of images stored for training/validation purposes.

• Training data generation (word level)

• Pre-processing of the images with enhancement and filtering operations

• Word detection and segmentation

• Machine learning algorithms to train

• Implementation and validation

3.1.1 Steps In The Proposed Work

• Data capturing

• Data preprocessing

• Feature selection & extraction

• Machine Learning (training)

• Classification.

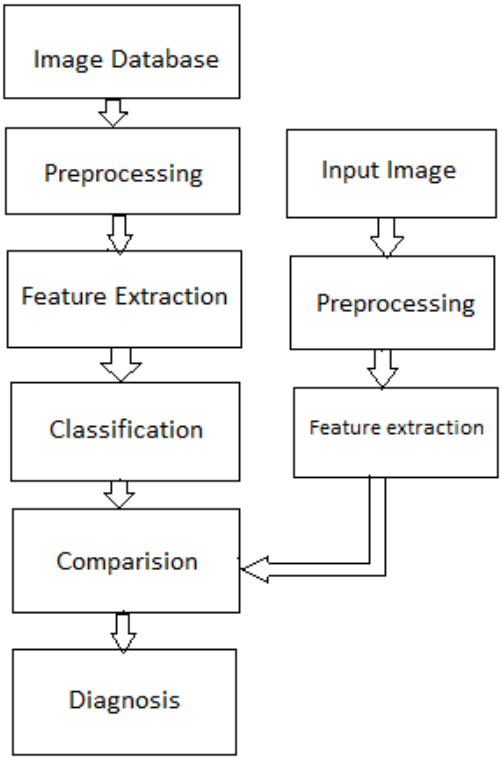

Fig 1: Proposed Methodology

3.2 Digital Image Processing

Extracting the text from the document image and processing it further for making it capable to be recognized is the task of digital image processing. Reliable character segmentation, removable of various noises, skeletonizing the character and normalizing it all come under the process of digital image processing. Before character recognition, it is necessary to isolate the individual characters from text image. Digital image processing refers processing of two-dimensional picture. Digital Images are pictures that have been converted into a computer-readable binary format consisting of logical 0's and 1's [4].

Digital image processing has a broad spectrum of applications in various fields such as remote sensing via satellites and other spacecraft, image transmission and storage for business applications, medical processing, radar, sonar, robotics and automated inspection of industrial parts [3].

The digital image processing operation can be broadly grouped into five fundamental classes, which are as follows:

3.2.1 Image Representation And Modelling

Image representation and models deal with the process of finding proper ways to mathematically represent the image. There are two ways to represent an image. The external representation deals with the boundary of the image and is used when the primary focus is on the shape characteristics. The internal representation deals with pixels within the region and is used when the primary focus is regional properties such as colour and texture. But sometimes it is necessary to use both representation techniques.

3.2.2 Image Enhancement

Image enhancement refers to the process of making a change to the features of the image such as edges, boundaries, or contrast to make a graphic display more useful for display and analysis [3]. The enhancement process does not increase the inherent information content of data but it does increase the dynamic range of the chosen features so that they can be detected easily. Image enhancement includes grey level and contrast manipulation, noise reduction, edge crispening and sharpening, filtering, interpolation and magnification, pseudo colouring and so on.

3.2.3 Image Restoration

When an image is acquired using optical, electro-optical or electronic means, there is the possibility of degradation due to the sensing environment. The different types of degradation that might occur are as geometrical degradation, illumination and colour imperfection, and blur.

Blurring is a form of bandwidth reduction of an ideal image owing to the imperfect image formation process. It can be caused by relative motion between the camera and the original scene, or by an optical system that is out of focus [4].

3.2.4 Image Analysis

Image analysis is the process of retrieving meaningful information from images, extract statistical data. The feature which is extracted during image analysis can be related to finding shapes, detecting edges, removing noise, counting objects, and measuring region and image properties of an object. For this reason, image analysis includes processes, such as segmentation, skeletonization, slant correction, and size normalization.

3.2.5 Image Data Compression

The main objective of image data compression is to represent an image signal with the smallest possible number of bits without loss of any information. Compression of data helps in speeding the transmission and minimising the requirements of storage. Visual information occupies a large storage capacity. Although the capacities of several storage media are substantial, their access speeds are usually inversely proportional to their capacity. So, the compression techniques focus on reducing the number of bits required to store or transmit images without any loss of information

3.3 Artificial Neural Networks

Artificial Neural Network (ANNs) are computer programs inspired by the functional processing of information by the human brain. ANNs consist of a number of nonlinear computational elements called neurons. ANNs gather their knowledge by detecting the patterns and relationships in data and learning through experience, not from programming. The artificial neurons also known as processing elements (PE), are connected to each other and the connection between two PEs has a coefficient, i.e., weight, which is adjustable. Each PE has weighted inputs, a transfer function, and one output. The behaviour of a neural network depends on the transfer functions, learning rules and architecture of the neurons. The neuron is activated using the weighted sum of the inputs and the activation signal is passed through a transfer function to

produce a single output of the neuron. During the training, the weights are adjusted until the error in predictions is minimized and the network reaches the specified level of accuracy. Once the network is trained and tested it can be given new input information to predict the output.

There are three main components of an ANN that help on deciding which value is to be passed forward to get a better solution, Those are as follows :

(a) Weight Factor:
A neuron usually receives many simultaneous inputs and has its own weight. Weights are adaptive coefficients that determine the intensity of the input signal and are a measure of an input's connection strength.

(b) Summation Function:
The summation function adds up the product of neuron and their corresponding weight of the link e.g. ∑ xn wn. The summation function can be more complex than the simple product. There can be a selection of minimum, maximum, majority, product or several normalizing algorithms depending on the network architecture and paradigm.

(c) Transfer Function:
The final result of the summation function is transferred to the output through an algorithmic process known as a transfer function. This function can be divided into two categories.

(d) Threshold
The output of the summation function is compared to some threshold value to determine the neural output. If the sum is greater than the given threshold value, the processing element generates a signal and if less than the threshold value, no signal is generated, or vice versa. In the case of the threshold function, any values above or equal to a given threshold are converted to 1, while anything falling below it is converted to a 0 during activation [12].

(e) Sigmoid
The output varies continuously but not linearly as the input changes.

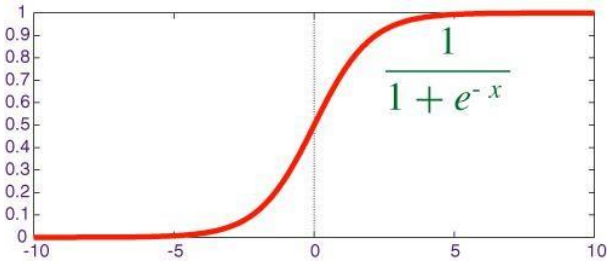

Fig 2: Sigmoid

3.4 Training Of ANN

Once the neural network has been structured, it is ready for training. Training is the process of adjusting the weights of the connection. At the start of training, random weights are assigned to the connections. There are two approaches to training, which are discussed below.

(a) Supervised Training
In supervised training, both input and output are provided. So the network has an input value, weight, and desired output value. Here the network processes the inputs and generates an output, these outputs are then compared with the desired outputs provided at the start of training. The errors acquired after the comparison are then transferred or propagated back through the system, causing the system to adjust weights, which control the network. This process is repeated continuously until the desired outputs are matched.

(b) Unsupervised, or Adaptive Training
In unsupervised training, the network is provided with the input data but not the output data. The system itself must decide what features it will use to group the input data. The system is provided with a training set, the objective is to categorize or discover features or regularities in the training data. The network internally monitors their performance and adjusts the weights accordingly. During the construction of the network, information about how to organize itself, if provided.

(c) Learning Rates
The purpose of the learning rate is to modify the connection weights on the inputs of each processing element. This process of changing the weights of the input connection to achieve some desired result can also be called the adaptation function as well as the learning mode. The higher the learning rate the convergence will be higher, resulting in a higher error in the performance.

3.5 System Design

The general methodology of HTR is demonstrated in Figure 1. This figure presents the workflow of HTR systems. It involves pre-processing, training and classification steps as shown in Figure 1. This is the traditional method of HTR system, the main drawback of this methodology is preprocessing, segmentation and classification steps. The results of OCR depend mainly on preprocessing and segmentation and feature extraction. To overcome these drawbacks this proposed method employs Deep learning method which avoids preprocessing, segmentation feature extraction process.

3.5.1 LSTM-Based Model

LSTMs and other recurrent neural networks are dominantly used for modern handwritten text line recognition models [3], [5]–[8], [16]–[18]. Our model is inspired by the CLDNN (Convolutions, LSTMs, Deep Neural Network) architecture proposed by [19]. For the convolutional layers, we use the inception [20] style architecture described in [15]. For the LSTM layers, we use between one and four stacked bidirectional LSTMs (BLSTMs).

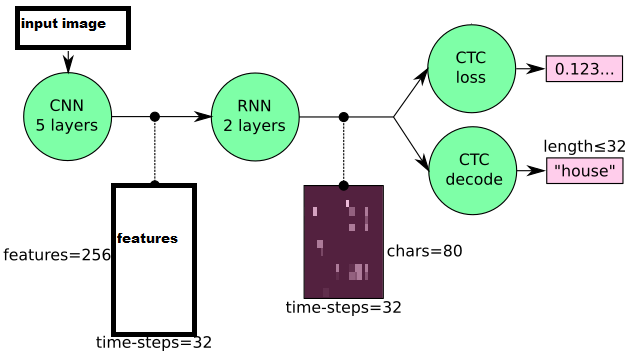

Fig 3: CNN Architecture of the proposed model

Along depth, information can be propagated over time with a large receptive field. [21] uses GRCLs as an alternative to CNNs, and the GRCLs are still followed by BLSTMs. We use GRCLs as an alternative to LSTMs, and the CNN layers are not changed. In this way, the model has only feed-forward connections. We show that the combination yields comparable accuracy with LSTM-based systems

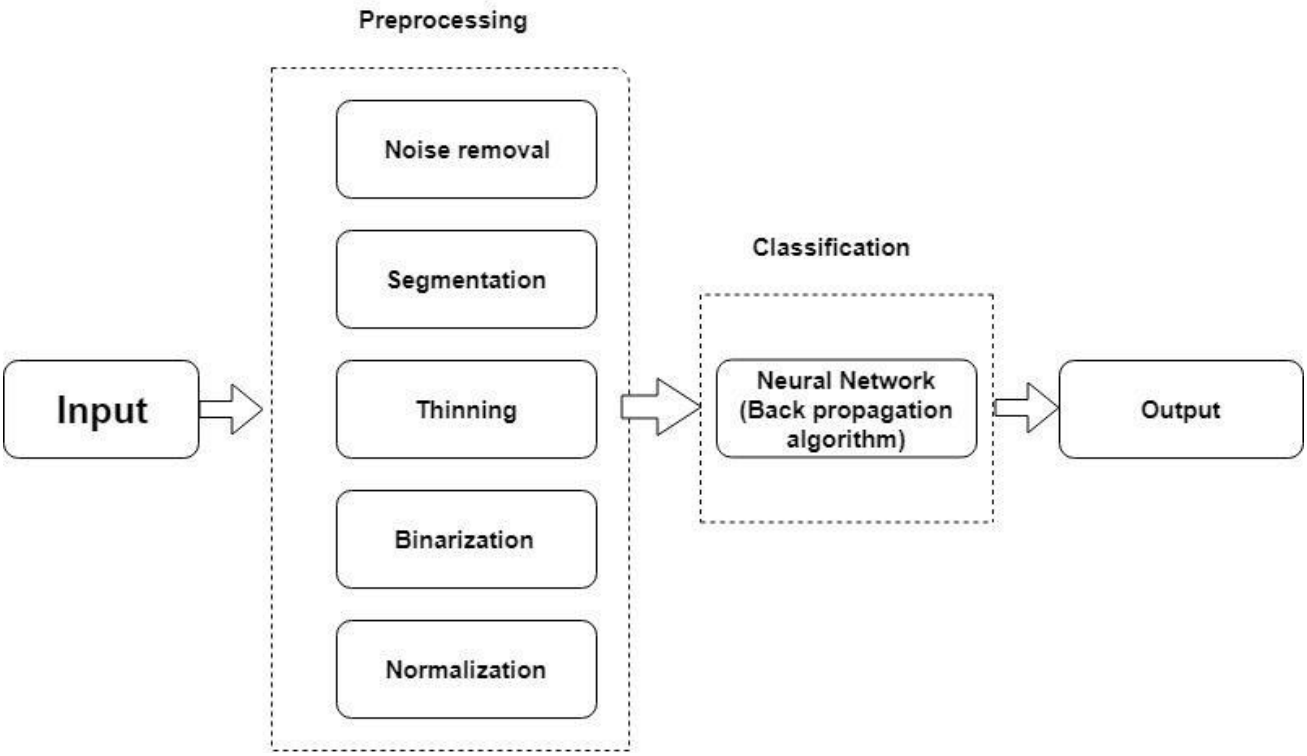

Fig 4: General Design of The System

4. TRAINING & RESULTS

4.1 Training

For the purpose of training, random weights are supplied to the connections between the nodes. The back-propagation algorithm adjusts the value of the connection depending on the error value generated at the output." Neural Network OCR" was used for the training of different characters. It was used for training both handwritten and machine-written characters.

Before a character can be trained, data for learning should be generated and the error limit should be set.

In the following Figures, Figure 6 and Figure 7, provides the details during the training of handwritten character. The progress section in the figure gives the time taken for the training, and the number of iterations required to achieve the goal of minimum errors.

The number of iterations for achieving the minimum errors for handwritten i.e. 436000, which was still under training, was very large than the one required for machine-written characters i.e. 100. The time needs to train a machine-written character was found to be of few seconds whereas it took a couple of hours for training handwritten characters

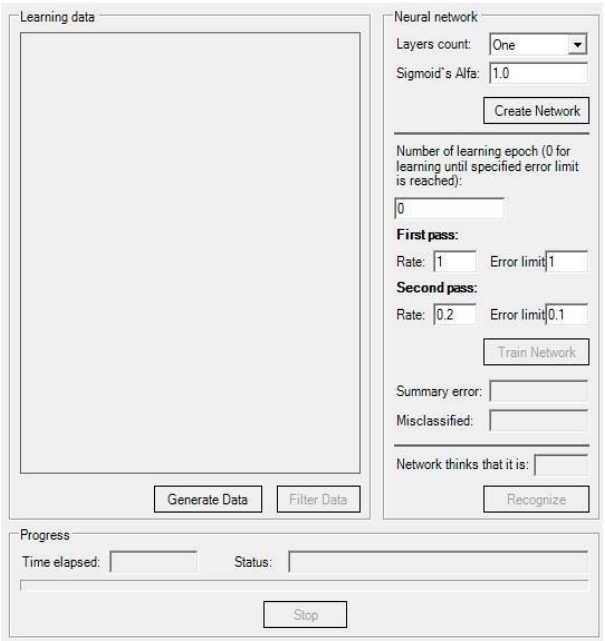

Fig 5: Training UI

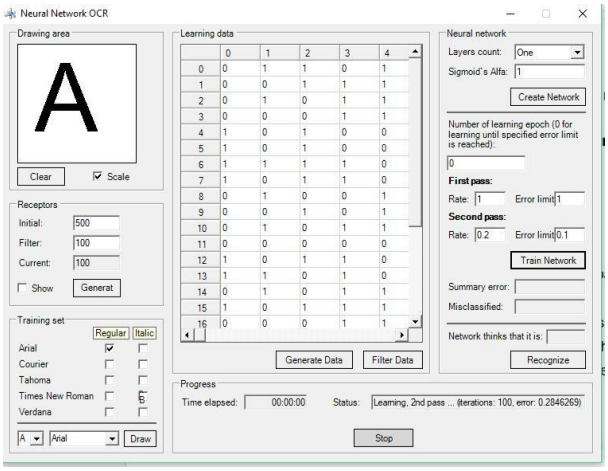

Fig 6: Training Machine Writes a Text Character

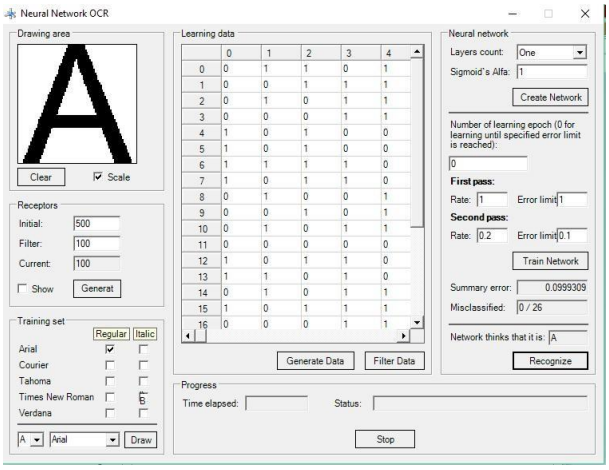

Fig 6: Training Machine Recognizes a Text Character

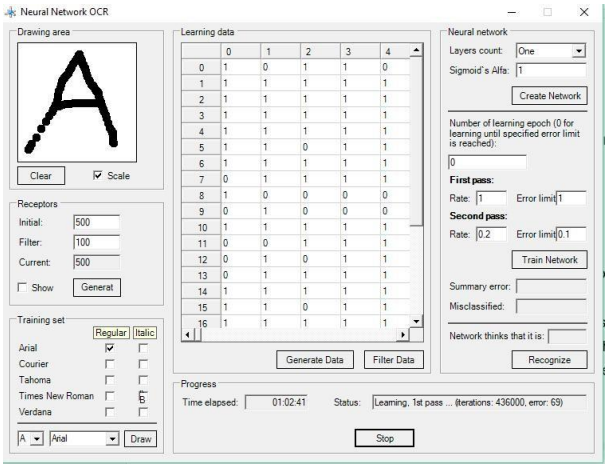

Fig 7: Training Machine Writes a 'Handwritten' Character

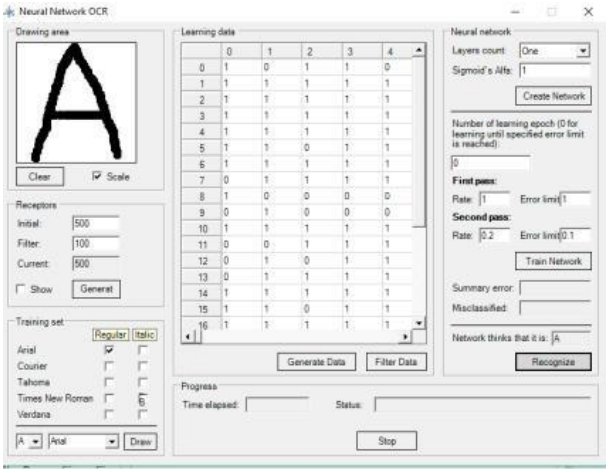

Fig 7: Training Machine Recognizes a Handwritten Character

4.1.1 Recognition Of Characters

Recognition is the final step in the HTR." Neural Network HTR" was used for the recognition of different characters. The precision of the system recognizing the character depends on the training of the characters. After training a character, the system is capable of recognizing the same character.

The Following Figures, Figure 8 and Figure 9 shows the recognition process of a character. Once the character is trained. The updated weights are stored in a weight database to be used for the recognition process. When a character is to be recognized, one should draw the character in the drawing area and then press recognize. The output for the recognized character will be displayed in the text box titled" Network thinks that it is:"

4.2 Results

Although LSTM beat the state-of-the-art at word level accuracy, 2DLSTM character level accuracy is slightly lower in comparison. It implies that the LSTM model is prone to make additional spelling mistakes in words which have already got mislabeled characters in them, but overall makes fewer spelling mistakes at the aggregate word level.

Table 1. Table captions should be placed above the table

Methods	CER	WER
2DLSTM	8.2	27.5
CNN-1DLSTM-CTC	6.2	20.5

5. CONCLUSION & FUTURE WORK

Pattern recognition done using neural network can tolerate noise and trained properly, can be used to recognize unknown patterns. Neural networks if constructed using proper architecture and trained correctly can be used for different scientific and commercial applications. They can be used for data entry, text entry, data automation etc.

Various image processing algorithms were studied and tested and the algorithm that supplied the best result is included in the architecture and discussed. Segmentation is one of the difficult processes in image processing. Isolating a single character from

a bunch of characters can be achieved using a classical approach, recognition based or holistic approach.

A neural network using a back-propagation algorithm is one of the most popular algorithms for training. It is a time-consuming algorithm for training networks with a large number of nodes. Adjusting the size of the input, error margin and addition of hidden node will give a better result.

To test the application, software created by different developers were used. Multiple tests were performed to find the best option for the system. For the preprocessing phase, different algorithms of digital image processing were used. The median filter was found to be the best one to remove the noise from the image. Hilditch's Algorithm produced the best skeleton of the character.

For the test of the classification phase, training was performed using machine-written and handwritten characters. During the training, it was found that the time taken and the number of iterations to reach the desired error limit for handwritten characters was very much greater than the one required for machine-written characters. The system was trained to recognize different handwritten characters as well as machine-written characters.

The system was designed with few specific algorithms. The performance and the processing time can be further improved using the different algorithms for preprocessing and the classification phase. This system is capable of recognizing the English alphabet but can be further trained to recognize different character sets from other languages as well

6. REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connection- ist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[3] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *NIPS*, 2009.

[4] A. Graves, M. Liwicki, S. Fernndez, R. Bertolami, H. Bunke, and

J. Schmidhuber, "A novel connectionist system for unconstrained hand- writing recognition," *PAMI*, vol. 31, no. 5, pp. 855–868, 2009.

[5] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *ICFHR*, 2014.

[6] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *ICFHR*, 2016.

[7] T. Bluche and R. Messina, "Gated convolutional recurrent neural networks for multilingual handwriting recognition," in *ICDAR*, vol. 01, 2017.

[8] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *ICDAR*, 2017.

[9] D. Castro, B. L. D. Bezerra, and M. Valena, "Boosting the deep multi-dimensional long-short-term memory network for handwritten recognition systems," in *ICFHR*, 2018.

[10] D. Keysers, T. Deselaers, H. A. Rowley, L. Wang, and V. Carbune, "Multi-language online handwriting recognition," *PAMI*, vol. 39, no. 6, pp. 1180–1194, 2017.

[11] V. Carbune, P. Gonnet, T. Deselaers, H. A. Rowley, A. Daryin, M. Calvo, L.-L. Wang, D. Keysers, S. Feuz, and P. Gervais, "Fast multi-language lstm-based online handwriting recognition," *ArXiV*, 2019.

[12] J. Walker, Y. Fujii, and A. C. Popat, "A web-based ocr service for documents," in *DAS*, Apr 2018, pp. 21–22.

[13] S. Ghosh and A. Joshi, "Text entry in Indian languages on mobile: User perspectives," in *India HCI*, 2014.

[14] T. M. Breuel, "Tutorial on ocr and layout analysis," in *DAS*, 2018.

[15] Y. Fujii, K. Driesen, J. Baccash, A. Hurst, and A. C. Popat, "Sequence-to-label script identification for multilingual ocr," in *ICDAR*, 2017.

[16] M. Kozielski, P. Doetsch, and H. Ney, "Improvements in rwth's system for off-line handwriting recognition," in *ICDAR*, 2013.

[17] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," in *ICFHR*, 2014.

[18] P. Voigtlaender, P. Doetsch, S. Wiesler, R. Schlter, and H. Ney, "Sequence-discriminative training of recurrent neural networks," in *ICASSP*, 2015.

[19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *ICASSP*, 2015.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan,

V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[21] J. Wang and X. Hu, "Gated recurrent convolution neural network for ocr," in *NIPS*, 2017.

[22] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *CVPR*, 2015, pp. 3367–3375.

[23] J. Sánchez, A. Toselli, V. Romero, and E. Vidal, "ICDAR 2015 Com- petition HTRtS: Handwritten Text Recognition on the tranScriptorium Dataset," in *ICDAR*, 2015.

[24] A. Toselli, V. Romero, M. Villegas, E. Vidal, and J. Sánchez, "ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset," in *ICFHR*, 2016.

[25] J. Sánchez, V. Romero, A. Toselli, M. Villegas, and E. Vidal, "IC- DAR2017 Competition on Handwritten Text Recognition on the READ Dataset," in *ICDAR*, 2017.

[26] Folger Shakespeare Library, "Early Modern Manuscripts Online (EMMO)." [Online]. Available: https://emmo.folger.edu

[27] K. Chen, L. Tian, H. Ding, M. Cai, L. Sun, S. Liang, and Q. Huo, "A compact cnn-dblstm based character model for online handwritten chinese text recognition," in *ICDAR*, 2017.

[28] A. Graves, "Generating sequences with recurrent neural networks," *ArXiV*, 2013.

[29] U. Marti and H. Bunke, "The IAM-database: An English sentence database for off-line handwriting recognition," *IJDAR*, vol. 5, pp. 39–46, 2002.

[30] M. Liwicki and H. Bunke, "Iam-ondb - an on-line English sentence database acquired from handwritten text on a whiteboard," in ICDAR, 2005.