Offline Recognition and Translation of Text Using Neural Networks

R.Divya
Dept. of CSE,
SRM University,
Chennai, India

A.Varshini
Dept. of CSE,
SRM University,
Chennai, India

S.G.Dhanavarshini
Dept. of CSE,
SRM University,
Chennai, India

Dr.V.Rajasekar
Dept. of CSE,
SRM University,
Chennai, India

Abstract *- Text recognition is a prominent domain, employed over several languages. Conventional procedures allow handwritten texts to be scanned and stored as such. Nevertheless, the toughest job is to retrieve the information word by word. OCR (Optical Character Recognition) is an electronic mode that helps in the translation of a handwritten text into machine modifiable text from scanned documents and images. Although various enforcements of OCR exist, deep learning approaches such as neural networks in this regard gives a better, generic solution. In our paper, we have used CNNs(Convolutional Neural Networks), along with RNNs (Recurrent Neural Networks); here LSTM(Long Short term memory), one of it's type and finally CTC(Connectionist Temporal Classification). The implementation is carried out by PyTorch, which is a Python-based library computer vision and natural language processing to build the neural networks (CNN and LSTM RNN). The resulting output is then processed by CTC for calculating the CER(Character Error Rate).*

Once the text is recognized, it is translated into several other languages, which is enabled using Google Translation API. This API uses Google's pre-organized machine translation kits to deliver quick and accurate results. A user-friendly forum is created through a web application, which accepts an input handwritten text image or document to scan word by word and further translate to any language of the user's choice .

Keywords: Text Recognition, PyTorch, CNN, LSTM RNN, CER, Translation

I. INTRODUCTION

Even though there are plenty of technical writing tools available, many of us still prefer to take notes traditionally. However, it is robust to store as well as extract the piece of information necessary in less time. This approach is tedious and cumbersome, hence leading to a great deal of necessary information being misplaced or not peer processed. The electronic form of texts on the other hand proves to be highly efficient. This in turn allows the individuals to quickly maneuver through the data and help them explore as well as gain information inefficiently. The ultimate goal of the project is to recognize the text written and convert it to English. Such handwritten samples in English are processed through the CRNN model to be recognized in the same language but in the electronic form. Written language is a conventional term, which is narrowed down to a certain means of written language to use for our functions. For a particular input image to be identified by our system, we have classified the images from running hand to hand printed letters. The words are better recognized if they are legible and easily understood, although in some cases it becomes difficult to do so. The pre-processing step helps in modifying and make them available to operate through the model in many such cases.

The focus on making the word to be correctly identified is carried out with complete word photos as a result of PyTorch which helps to focus more on pixels from the image compared to other choices or slices of a photo. From the output thus received, by exploiting complete words, we have improved by extracting letters out of each word, later distinguishing such characters on an individual basis to reconstruct a whole phase.

The superimposed layers added in our project helps to deliver a formatted output in English which is easily visible to the user. This forms a standard medium for uploading images of texts, which gives an approval/acknowledgement of the image having been identified successfully. All of this can be easily perceived by the user through the application created for this purpose.

We have extended the scope of our project by including the translation of text. Having received the outcome in English text, one should make use of another such application to convert this text to any language. Several such applications exist, including Google Translate which helps the user to convert the

words from one language to another. We have employed the Translate API from Google in our application thus created, to help translate the text. This enables the user to easily convert as well as translate via the same application. Moreover, it is highly efficient and user-friendly, thus satisfying the needs of the user on both ends.

II. RELATED WORK

Human beings have always expressed their thoughts through letters, transcripts, etc.; to convey their thoughts to other people. However, since the introduction of technical knowledge the configuration of transcription quickly to machine-produced digital words. So individuals feel a need for such a technique that will redesign the interpretation to computerized content since it makes the procedure of such data in a matter of seconds and basic**.**

Numerous sorts of scientists attempted to progress such a framework before. Acknowledgment investigations of handwritten character images remain intently inferable from their colossal applications. Rajib et al. [5] had planned a written English text identification system that supported the Hidden Markov Model (HMM). The methodology utilizes 2 completely different characteristic extraction namely, international and local characteristic extraction. The global characteristics include several options like gradient options, projection options as well as curvature options in the numbers of 4, 6 and 4. However, the native or local characteristics are calculated by splitting the selected images into 9 identical parts. Each block's gradient characteristics are determined by disintegrating all four function vectors, rendering the entire range as thirty-six native choices. The method above resulted in fifty (local and global) options for every illustrative image. Then, to coach it, these square options values are forwarded to the HMM model. Post-process knowledge is an additional step that this methodology uses to minimize the cross-identification of various categories. This technique requires a lot of training time and has extraction. If multiple characters' square values are combined during a single image.Above all, it performs poorly only in the case of these inputs.

A relative interpretation study of many identifiers on written number identification presented in Liu et al. [8] where the recognition is improved by using gradient extraction along with the classifiers. By means of vector space embedding, graphical similarity features are used for recognition by Andreas Fischer et al. [9]. Options presenting the accuracies in character recognition tasks square measure gradient, curvature [1] options. In some text identification studies, Gabor remodels [2] and statistical/structural options [3] have been successfully used. In a recent study [4] written on Nagari script and Bangla text identification, the rippling transformations of input text images were subjected to a minimum-layer technique.

Velappa Ganapathy et al.[6] presented a technique of identification that would help multi-scale coaching of the neural network. This approach used a limited minimum value to increase the precision, which is measured using endorsed minimal distance technique. Furthermore, this approach requires the use of a graphical user interface, which can reveal the character in the scanned image. This technique combines a certain degree of preparation for an additional precision of eighty-five. This technique used images with giant resolution (20 or 28 pixels) to train with less preparation time.

Although there's as yet a need of undeniably more investigation right now, acknowledgment reads are available in several languages such as English [8,9].Fei yin [10] created an integrated framework based on neural network language models(NNLM) and hybrid neural network language models for Chinese character recognition. In this framework, two different techniques are implemented namely feedforward NN and recurrent NN to construct the Hybrid NNLM. Once the hybrid NNLM is constructed it is used for geometric context modeling to identify the characters within a specific amount of time. This framework also analyses an upper bound performance on text to calculate the word error rate.

Japanese [11], Arabic [12], and other Indian languages like Hindi, Telugu, etc. can be found in [13-20]. Nevertheless, every one of them experiences the ill effects of some assortment of disadvantages: specifically low modification speed, less accuracy, increased error recognition rate and inadmissible execution with vociferous information, and so forth. T. Som et al. [7] uses a fuzzy logic performance to boost the precision of handwritten identification systems. During the technique, handwritten pictures are resized to 20 × 10 pixels that the fuzzy technique uses in every category. The bounding box is formed around the figure to determine the vertical and horizontal projection of the text. Once the photograph is cropped to a bounding box, it is re-scaled to the scale of 10×10 pixels. The rescaled images are trimmed with the help of a trimming function. To create a sample matrix all the preprocessed pictures are fed into the matrix in sequential order. When the sample images are fed by the user it is tested for matching against the sample

matrix, this technique is comparatively quick but yields less precision.

III. EXISTING SYSTEM

A. *Architecture Description*

- The present framework presents an end to-end neural network prototype which comprises convolutional and intermittent networks to perform proficient offline HTR on pictures of contents.
- The Encoder-Decoder Framework with Attention provides an increase in inaccuracy. as compared to the standard RNN-CTC technique for HTR.

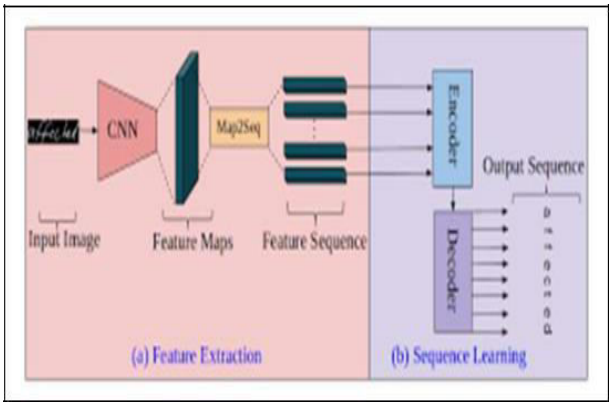

Fig 1: Model summary : (a) indicates the feature series derived from feature maps during CNN while (b)depicts how the visual feature series is matched to output characters.

B. *Methodology Used*

a) The Feature Extraction module takes as input an image of a series of characters to extract visual features. A standard CNN (without the fully-connected layers) changes the loaded picture to a bulk stack of characteristics maps.

b) The Sequence Learning module maps the optical features to a series of characters. In the Encoder-Decoder framework, the model consists of two recurrent networks, one of which presents a compact result based on its understanding of the input series while the other uses the identical classification to generate the respective output series.

C. *Problems Faced*

A comparative study of word error rate showed better accuracy than the existing systems. The character error rate was marginally lower, which indicates that the model is vulnerable to make mistakes in misspelled words, yet, the total number of errors at the word level is less.

1. NVIDIA Tesla K40 GPU Accelerator, an expensive product yet effective.
2. Bottlenecks of information occur while using RNN Encoder/Decoder plus Attention for long sequences.
3. More Computation Power in order to decrease the error rates.

IV. OUR CONTRIBUTION

In our project, we propose a system which helps in text recognition from the input images and translates the same to different languages. The system's front-end uses a web application that connects to the pre-trained deep learning model as soon as an input image is provided by the user. This model is created by Convolutional Neural Networks (CNN), Long short-term memory(LSTM RNN) with PyTorch. The Virtual Machine returns the English text once the input is recognized. Then, it calls the Google Translation API to translate this English text to any language supported and suggested by the user(more than100 languages). The web application essentially then displays the output to the user. Through a large number of examinations, trials, assessment, and model usage work, we have demonstrated that the proposed arrangement is doable and is equipped for delivering a lesser character flaw rate.

A. *Architecture Diagram*

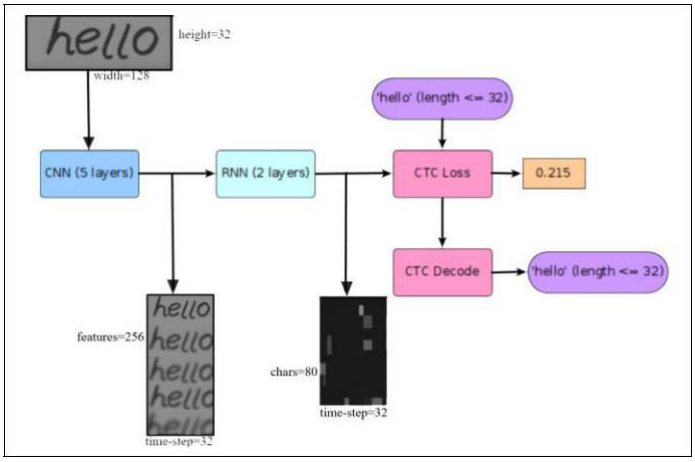

Fig 2 : Model Summary: Flow of CRNN operations through the layers with the CTC loss and decode.

An overview of our work is to essentially build a system that involves english handwritten words for input, run them through a neural network using CRNN(CNN & RNN), recognize the words, and obtain the error.Once this is performed, the word is translated into any language reinforced. Further, the two parts involved in text recognition is mentioned below:

The training process in the proposed model involves, the uploading of dataset(IAM), preprocessing of the input, deduce feature maps from it, generate a feature and test vectors,train PyTorch and save the same for testing.

The testing part involves additional processing because the characters from the image uploaded have to be found out. It uses the PyTorch from above to perform.

B. *Learning Algorithms*

i. *CNN*: The input image is fed to the CNN layer. Features are extracted from the input image by the referred layers. The deeper the network goes, the filters become more sophisticated. First, the convolution operation is performed. Kernel filter is applied of size 5*5 in the initial layers and the remaining three layers will be of size 3*3. Later nonlinear RELU function is applied. And lastly, the pooling layer summarizes the image parts and gives a reduced form of the input as its output. The feature maps are added meanwhile the height of the image is reduced by 2, so that the size of the output feature map is 32*256.

ii. *RNN***:** There are 256 features per time-step in the feature map. Two such stacks are created. The input can be traversed from front to back and vice versa since we are using bidirectional RNN. LSTM propagates data through longer distance and more vigorous training-attributes are provided in both the directions. After the feature axis is added, the size now becomes 32×512. It is reduced to form the RNN output sequence, which is a matrix of size 32×80(79 from IAM + CTC blank label).

iii. *CTC*: To compute the loss, output from the RNN matrix and the ground truth text is given to the CTC. The ground truth text will be in encoded form. For both the CTC operations, the length of the input sequence must be passed. Now we have all the necessary information to perform the loss and decoding operation. The actual, as well as the decoded text, can be of maximum 32 characters long.

C. *Pre-processing*

It is a gray scale image which is about 128×32 in size. The images from the dataset vary in size. So it is resized to a dimension of 128 with a height of thirty two is reached. Then, it is duplicated into an image in white, the target of size 128×32 as shown below. At last, gray scale images are tempered such that our model (CRNN) is simplified. Information augmentation will simply be combined by repeating the image in different arrangements rather than positioning to the left side or by arbitrarily modifying the size of the image.

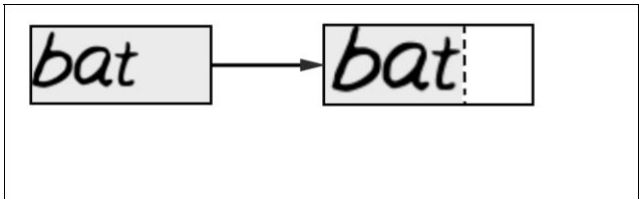

Fig 3: An example of preprocessing

The process includes the elimination of unnecessary and undesired patterns(uniform/non-uniform). This is known as noise removing. Then binarization is performed where translation to grey-scale pictures is done from all the typewritten characters. After translating the grayscale images into the binary matrix, every appearance of the character will be caught vertically. In normalization, the image is resized to the standard format as required by the model.

Mainly sizing along with skewing normalizations are performed. Size alters the picture image into the predefined rigid size. And as for scanning, we can use skew, when the text is swerved from the baseline. In order to find these, backpropagation is required.

D: *Feature Extraction*

Characteristics in the form of a vector are clustered from feature maps called as feature vectors into bitmaps. The bitmap version preserves the most important options of the image in a short area/

information range. This decreases the time passed on training the neural network while making sure that the accuracy does not get affected in character recognition. As explained above in CNN, feature maps of size 32×256 are formed.

E: CTC Loss

The matrix from above and the ground truth value or the actual input value along with the size of the sequence is sent to CTC to calculate the loss. In general, it is the negative log of the probability of GT text. This is first decoded by removing any duplicate characters and chooses the most likely path taken by the character.

F. Post-processing

Post-processing is the ultimate phase of character recognition. By using natural language, it can rectify the misclassified output. After the shape has been recognized, it will process the output. The accuracy can be enhanced if the shape is recognized purely along with the knowledge of the language. The shape recognizer's behavior varies for various handwriting inputs. It leads to each character of string for a few which includes a certain number of choices and for others a measure of confidence in every such alternative.

G. Dataset Used

IAM Dataset version 3.0 (English) [22] contains 1539 pages of handwritten text contributed by several writers (around 700). It is divided into separate sets of training, validation and test data for each writer of 61,619,401,861 separated lines. These picture elements/pixels have a mean height and width of about 124 and 1751 respectively. There are 79 characters of different forms plus white space.

H. Error Rate

The error rate is defined as the quantity of text not recognized accurately by the HTR model used. Conventionally, two types of error rates exist, CER and WER. WER or Word Error Rate is comparatively high compared to CER. This is because the length of the words is never constant. Hence, CER or Character Error Rate is a better choice. Here, it is calculated as : sum of falsely recognized characters / total characters in the given text.

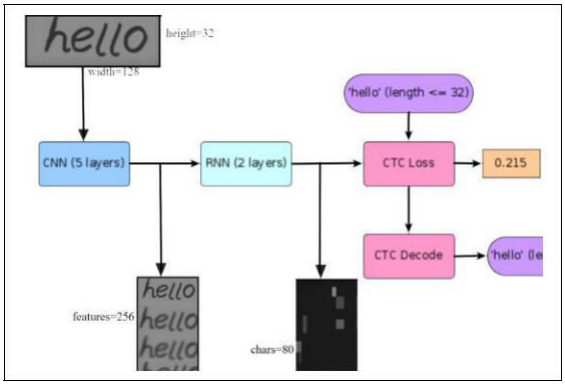

Fig 4: Overview of the actual process in the model

IV. RESULTS

The HTR model was enforced on various input handwritten images. The results turned out to be promising. The device executes pre-processing on the data. Extraction of features is done from the image representation of the bitmap and the inputs bidirectional traversal helped provide finer classification. The table below shows a relative analysis of model performance with respect to CER.

Table I. Comparison of CER on the IAM Dataset used in the existing and the proposed system.

MODEL	***CER***
Baseline + LN + Focal Loss + Beam Search	8.1%
CRNN + CTC (validation set)	6

The above table shows an improved CER (Character Error Rate) of about 6%. This difference in accuracy makes the study imperative with the CRNN model used(CNN and RNN) compared to the existing system [21] on the IAM Dataset. The implementation was carried out in Nvidia GTX 1080 TI GPU.

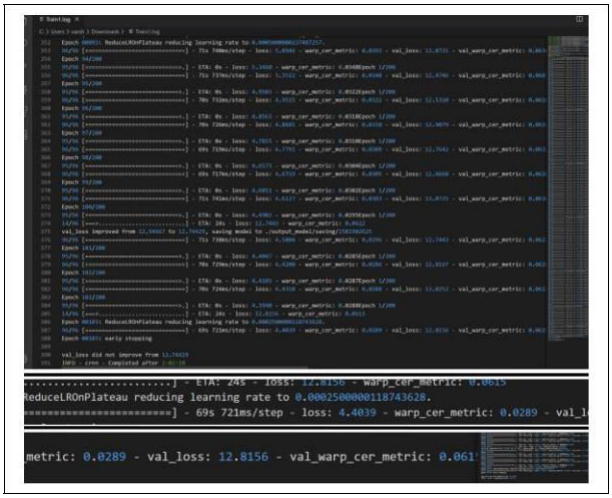

Fig 5: Shows the screenshot obtained for each epoch for training and test.

The image above displays the loss and character error rate for the training and the validation/test sets during several epochs. The loss and warp_cer_metric corresponds to the training set while val_loss and val_loss warp_cer_metric for the test set respectively.The proposed system uses less number of features for training the neural network, that produces faster convergence thereby taking less amount of time for training.

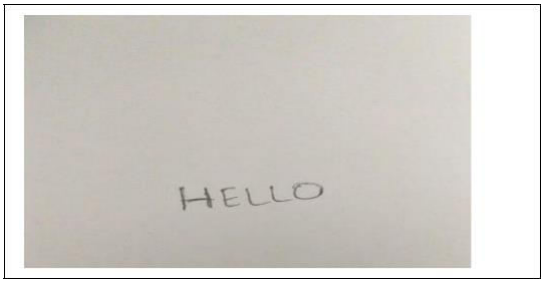

Fig 6: Input Sample Image

Fig 7: Output Sample image : The web app screen with input(bottom left) and translated output in Spanish (bottom right).

The above images show the relative results. Fig 2 shows a sample input uploaded to the web app which recognizes the word in English.

Fig 3 shows the screenshot of sample output for the input image, after converting the English word to the selected language. The above images are obtained with the help of the Google Translation API called by the virtual machine after uploading the input image.

V. CONCLUSION

The result obtained from our proposed work shows that most of the input text on the character level was recognized effectively. Our model was tested on both the validation and the training set. Two robust standard algorithms have been used namely, CNN and RNN. The feature maps from the Convolutional Neural Network(CNN) and the bidirectional Long Short Term Memory Recurrent Neural Network(LSTM RNN) helped enhance the accuracy. Different models for HTRsystems are available, but our work primarily focuses on two objectives. The first was to improve the standard CRNN model with better accuracy. The second was aimed at the user interaction, to provide a forum for translation after recognition.

Essentially, our web application satisfies both objectives. It reduces the time taken by the user to help recognize and translate the text easily without having to search for another application to translate. Besides, it proved to be highly efficient on the IAM Dataset. On a broader scope, our work can be extended to run on other datasets, with different styles of handwriting and other languages. Furthermore, the model can be extended to find the word level error rate for a better margin of comparison with other contemporary works.

VI. REFERENCES

[1] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura, "Handwritten Numeral Recognition Using Gradient and Curvature of Gray Scale Image, Pattern Recognition", vol. 35(10), pp. 2051-2059, 2002.

[2] Singh, Sukhpreet, Ashutosh Aggarwal, and Renu Dhir, "Use of Gabor Filters for recognition of Handwritten Gurmukhi character" International Journal of Advanced Research in Computer Science and Software Engineering 2, no. 5 (2012).

[3] Wu X-Q, Wang K-Q, Zhang D, "Wavelet energy feature extraction and matching for palmprint recognition" J Comput Sci Technol 20(3):411–418, (2005)

[4] Bhattacharya, U.and B. B. Chaudhuri, "Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed

Numerals", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31(3), pp. 444- 457, 2009.

[5] Rajib Lochan Das, Binod Kumar Prasad, Goutam Sanyal, "HMM based Offline Handwritten Writer Independent English Character Recognition using Global and Local Feature Extraction", International Journal of Computer Applications (0975 – 8887), Volume 46– No.10, pp. 45-50, May 2012.

[6] Velappa Ganapathy, Kok Leong Liew, "Handwritten Character Recognition Using Multiscale Neural Network Training Technique", World Academy of Science, Engineering and Technology, pp. 32-37, 2008

[7] T.Som, Sumit Saha, "Handwritten Character Recognition Using Fuzzy Membership Function", International Journal of Emerging Technologies in Sciences and Engineering, Vol.5, No.2, pp. 11-15, Dec 20

[8] H. Liu and X. Ding, "Handwritten Character Recognition using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes", Proc. 8th Int. Conf. on Document Analysis and Recognition, pp. 19-25, 2005

[9] Fischer, Andreas, Ching Y. Suen, Volkmar Frinken, Kaspar Riesen, and Horst Bunke. "A fast matching algorithm for graph-based handwriting recognition." Graph-Based Representations in Pattern Recognition, pp. 194-203. Springer Berlin Heidelberg, 2013.

[10]Yi-ChaoWu, Fei yin and Cheng-Lin Liu "Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models."Pattern Recognition Volume 35 (2017): 251- 264.

[11] Das, Soumendu, and Sreeparna Banerjee, "An Algorithm for Japanese Character Recognition." International Journal of Image, Graphics and Signal Processing (IJIGSP) 7, no. 1 (2014): 9.

[12] Parvez, Mohammad Tanvir, and Sabri A. Mahmoud. "Arabic handwriting recognition using structural and syntactic pattern attributes." Pattern Recognition 46, no. 1 (2013): 141-154.

[13] P. Sujatha, D. Lalitha Bhaskari, "Telugu and Hindi Script Recognition using Deep learning Techniques", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, Sep 2019.

[14] Anupama Thakur, Amrit Kaur , "Devanagari Handwritten Character Recognition Using Neural Network", International Journal Of Scientific & Technology Research Vol 8, Issue 10, Oct 2019.

[15] S. D. Prasad and Y. Kanduri, "Telugu handwritten character recognition using adaptive and static zoning methods," 2016 IEEE Students' Technology Symposium (TechSym), Kharagpur, 2016, pp. 299-304.

[16] Karishma Verma and Manjeet Singh ,"Hindi Handwritten Character Recognition using Convolutional Neural Network", International Journal of Computer Sciences and Engineering , pp. 909-914, June 2018.

[17] A. Indian and K. Bhatia, "A survey of offline handwritten Hindi character recognition," 2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall), Dehradun, 2017, pp. 1-6.

[18] Kannan, R.J., Prabhakar, R. and Suresh, R.M., 2008, Off-line cursive handwritten Tamil character Identification. International Conference on Security Technology, 2008(SECTECH'08), pp. 159-164.

[19] R. Dineshkumar, J. Suganthi, "Sanskrit character recognition system using neural network", *Indian Journal of Science and Technology*, vol. 8, no. 1, pp. 65, 2015.

[20] Lakshmi, C. V., Jain, R., & Patvardhan.C. OCR of printed Telugu text with high recognition accuracies . In Computer Vision, Graphics and Image Processing (pp. 786-795). Springer, Berlin, Heidelberg, 2006.

[21] A Chowdhury, L Vig "An efficient end-to-end neural model for handwritten text recognition", arXiv:1807.07965 [cs.CL] 26 Jul 2018.

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