Segmentation-Free Character Recognition System for Historical Documents Using HMMs

Shreya Mittal, Santhwana Santhosh Kumar, Satwika Chowdary, Remya Sivan, Peeta Basa Pati*
Department of Computer Science and Engineering
Amrita School of Computing Bangalore
Amrita Vishwa Vidyapeetham, India
*ORC ID: 0000-0003-2376-4591

***Abstract*—Character recognition systems for complex scripts like Malayalam face significant challenges when dealing with historical texts, such as palmleaf documents, due to the noise present and the complicated structure and variety of styles in these scripts. The varied morphological features of these scripts are often beyond the capacity of traditional segmentation-based techniques. In order to address the issue, we propose a segmentation-free character recognition system in which the input image is first divided into blocks and further segmented into three zones. Features are subsequently extracted in each zone and modeled by HMMs, which permits reliable identification without requiring precise character segmentation. The recognised text is then translated into English to improve its readability for a non-native speaker. Our system performs well on the translation of recognised text and also ensures outstanding recognition accuracy with significant improvements over traditional methods.**

***Index Terms*—Segmentation-Free Character Recognition, Handwritten Document Digitization, Hidden Markov Model (HMM), Feature Extraction, Historical Text Recognition**

I. Introduction

Digitization of historical manuscripts, especially those handwritten in ancient scripts, is one of the most emerging fields of study in digital humanities and computational linguistics. Such documents are usually preserved on delicate materials like palm leaves and are very valuable both historically and culturally.Their handcrafted quality combined with the passage of time made it relatively hard to decipher and translate using traditional techniques. Digitalization of these manuscripts not only gives a guarantee for the safety of their content for future generations but also creates new opportunities for deeper analysis and further research.

During these years, the methods used to digitise handwritten documents have changed enormously. Earlier approaches mostly depended on segmentation-based techniques, where an image processing algorithm first extracts individual words or characters from the document before feeding them to a model. Though these methods have turned out to be successful in some applications, they can be difficult to use with complex or deteriorated scripts where segmentation is difficult. Particularly when dealing with old scripts, the step of correctly separating characters from linked or overlapping strokes frequently results in mistakes that spread throughout the recognition process, reducing the overall accuracy.

Due to the fact that segmentation-based systems have their own drawbacks, methods of segmentation-free recognition have been investigated. The following methods avoid character segmentation but focus on recognizing the whole sequences or zones of text. By doing so, they are in a better position to handle irregularities and intricacies of handwritten scripts found mainly in ancient documents. Methods avoiding segmentation are very promising in terms of contextual integrity of the text, often lost when characters are isolated.

There exist a lot of different approaches to segmentation-free recognition and each has certain strengths and challenges. To model and recognise text sequences directly from photos, methods including Hidden Markov Models (HMM), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Encoder-Decoder models have been used. These models are designed to be trained on patterns and features across different zones of a document image so that more accurate recognition of complex scripts can be obtained. The choice of the approach often depends on the unique qualities of the text and the goals of the digitization project.

In this paper, we propose a segmentation-free character recognition system specially designed for the digitization of ancient handmade documents on palm leaves. Document images will be divided into blocks, which are then further divided into three zones for the extraction of features, modeled using an HMM. This approach overcomes most of the problems due to traditional segmentation-based methods while improving the accuracy of identifying complex handwritten scripts. This system is quite a development in this domain and offers a more reliable solution for the preservation and translation of historical texts into modern languages like English.

The rest of the paper is organized as follows: Section 2 reviews related work; Section 3 details the methodology; Section 4 presents result; Section 5 details analysis; and Section 6 concludes with future directions.

II. LITERATURE SURVEY

Su et al. [1] proposed a methodology that acknowledges the problems inherent to the Chinese handwriting recognition as a domain. Another challenge that relate to this is the problem of segmentation where characters overlap or come close to each other as observed in the realistic handwriting. Altwaijry and Al-Turaiki [2] in their proposed methodology has shown the growing interest of Arabic handwriting recognition due to the unique challenges posed by the Arabic script , and that includes its cursive writing nature, context-sensitive shapes of their alphabets, and a wide range of handwriting styles. Din et al. [3] in their proposed methodology describes several techniques and methodologies used in OCR for the Urdu language. The survey provides a wide range of various methodologies and their suitability in the identification of printed Urdu text.

The methodology proposed by Espan˜a-Boquera et al. [8], describes the methodology of recognizing unconstrained offline handwritten texts using hybrid Hidden Markov Model (HMM) and Artificial Neural Network (ANN) models. Michael et al [9] in their proposed methodology provide a literature review of the existing approaches to handwritten text recognition. It mainly discusses sequence-to-sequence model, and contrast it with other types of methods like Hidden Markov Model, as well as the combination of Hidden Markov Model and neural networks.

The method proposed by Kumar and Sharmaa [4] is called SEGmentation-free Writer Identification (SEG-WI) which is based on Convolutional Neural Network (CNN) free from text segmentation. Evaluation of the proposed technique in the IAM, CVL, IFN/ENIT, Kannada, and Devanagari datasets provides better results compared to existing techniques with top-1 accuracy lying consistently in the range of 87. 06% to 100%. The authors Ghosh et al. [5] aims at solving the problem of directly applying scene text recognition models for mobile applications, where the latter are constrained in terms of their size. This method puts forward a light-weight Convolutional-Recurrent Neural Network (CRNN) model that can satisfy the demands both in efficiency and accuracy. The methodology proposed by Balci et al. [12] is the Handwritten Text Recognition using Deep Learning, aims at different methods and techniques to recognize handwritten individual words and convert handwritten text into digital form in this paper. The authors utilized two primary methods: The first method for word features extraction is direct word classification based on Convolutional Neural Networks (CNN); and the second one is based on character segmentation using Long Short Term Memory (LSTM) networks connected with CNN. The methodology proposed by Ingle et al. [13] aims at improving Offline Handwritten Text Recognition (HTR) systems and discussing ways to incorporate HTR into the large-scale Multilingual OCR systems. The literature survey looks at the evolution from LSTM basic models into a more efficient Neural Network model without Recurrent Links and therefore with higher accuracy and training/imputation parallelism.

The methodology proposed by Ghena and Meng [6] explores the use of AI and machine learning for text recognition and identifies corresponding solutions to the pertaining issues that affect model performance and accuracy. The proposed methodology by Peng [7] focuses on the issues related to the offline handwritten character recognition regarding application of segmentation methods, problems linked to interleaving and touching characters in handwritten manuscripts. The methodology proposed by Wang at el. [**?**] addresses the problem of end-to-end text recognition in natural images by harnessing the power of big, deep neural networks combined with recent advances in unsupervised feature learning. The methodology proposed by Kim1 et al. [14] describes an end-to-end method of recognizing text in handwritten page images, which is conceived of as a combination of five functional modules . The methodology proposed by Do at el. [10] focuses on the issues of scene text detection and recognition, while considering the case when texts are located in complex scenes , with different fonts, sizes, styles, languages, and large amount of noises, cluttered backgrounds, etc.

III. DATASET

The dataset used in this study contains 2,250 samples of the lines extracted from ancient handwritten documents on palm leaves. The document images were preprocessed and segmented into blocks with an average of about 25 blocks per line. Then features were precisely captured from each block, to identify the fine details of the handwritten script.

There are a total of 196 different features describing a wide array of textual and structural characteristics for each sample from the segmented zones of the document. These features will be used to feed our segmentation-free Hidden Markov Model character recognition system. The dataset thoroughly depicts ancient scripts and gives freedom to the model to learn and recognize characters, therefore translating them into English with high accuracy.

IV. METHODOLOGY

The approach of the given character recognition system that is free from the segmentation process includes the following steps: Applying the pre-processing on the input image for the character recognition and feature extraction of the characters and then recognizing them by the aid of Hidden Markov Models (HMM). The aim is to bring eye approach strategy, which might be useful to address scripts as of Malayalam and the newly discovered much older scripts. It is relatively stable, and does not crumble in the face of problems that threaten traditional segmentation based approaches; it also offers greater accuracy in recognition.

A. Input Image Acquisition

1) Document Scanning: This is done through getting images of manuscripts belonging to the past regimes or even past civilization. These are either scanned or photographed carefully to make the scanned script as clear as possible.

2) preprocessing: This includes making the scanned image to be neat by using functions such as the denoise, the contrast and the thresholding. The goal is to produce a clean image in which all “noise” and interferences that could interfere with further processing of the image have been minimized or removed completely.

B. Image Segmentation into Blocks

1) Block Division: The received preprocessed image will be subdivided or partitioned into smaller image pieces better for convenient processing. This step is very crucial for simplification of the recognition process because every block as discussed above and seen from the context below will contain the sequence of characters which can further be treated as another element.

2) Zone Segmentation: Each block in question is further divided in to the three vertical strips namely the top strip and the bottom strip and the strip in the middle. They are based on the usual segmental formation of the Malayalam script in which segments of a character are placed in the the mid zone (consonant), or the top zone or the bottom zone which contains vowels modifying mark.

C. Feature Extraction

Zone-wise Feature Extraction: For each zone for a block certain morphological features are extracted. These features may include:

1) top Zone: In particular, characters that are displayed in a manner that would indicate they are features such as diacritics or character modifiers.

2) Middle Zone: The properties of the main characters' body and their constitution

3) Bottom Zone: Even more diacritical marks or tail strokes. Statistical and Structural Features: Edge detection, stroke width, contours shapes and pixel distribution of these features are then calculated within each zone. The features are thought to reflect the individual characteristics of the script's morphology

D. Modeling Using Hidden Markov Models (HMMs)

1) HMM Training: The extracted features from each zone are used to train Hidden Markov Models. The HMMs are specifically designed to model the sequential nature of the script, allowing for recognition even when characters are connected or overlapping

2) Sequence Recognition: During recognition, the HMMs evaluate the likelihood of different character sequences based on the observed features from the zones. The model outputs the most probable sequence of characters for each block.

E. Post-Processing and Translation

1) Text Reconstruction: The recognized characters from each block are summed up and then the sequences of the entire full texts are recognized. Any mistake made by the model is rectified with reference to the context it was generated from and with reference to the language models.

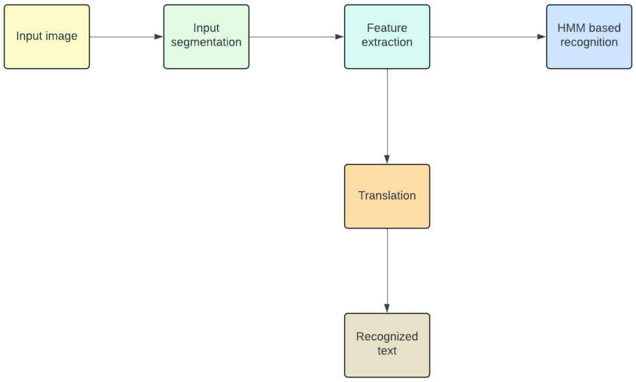

Fig. 1. Block Diagram of the Proposed Model

2) Translation into English: The recognized text is translated from Malayalam to English using a translation model.This step ensures that the text is accessible to non-native speakers, improving its readability and usability

F. Validation and Accuracy Measurement

1) Performance Evaluation: The system's performance is evaluated using a dataset of manually transcribed documents. Key metrics include recognition accuracy, translation accuracy, and processing speed.

2) Comparison with Traditional methods: The results are compared against traditional segmentation-based recognition methods to highlight the improvements in accuracy and reliability.

V. Results, Analysis and Discussion

VI. LAB 2

A. The Importance of the Rank of an Observation Matrix in the Model Building for Classification

The rank of an observation matrix is quite important in the classification because it shows how much independent information enters the learning stage. It is necessary to make a better difference in the classes using the model. This is important because higher ranks are a guarantee in the implementation of accurate character recognition by the HMMs.

B. Regression vs. Classification Tasks

One major difference is that regression predicts continuous values—things like stock prices—while classification gives discreet labels, such as character recognition. The former aims to minimize the prediction error, while the latter tries to maximize accuracy in placing inputs into categories.

C. Suggestions for Building a Predictive System for Stock Data

Utilize time series analytics, combined with relevant features—that is, historical prices and volumes—to make sound predictions on stock prices and changes. The developed machine learning ARIMA/LSTM work shall be good enough to

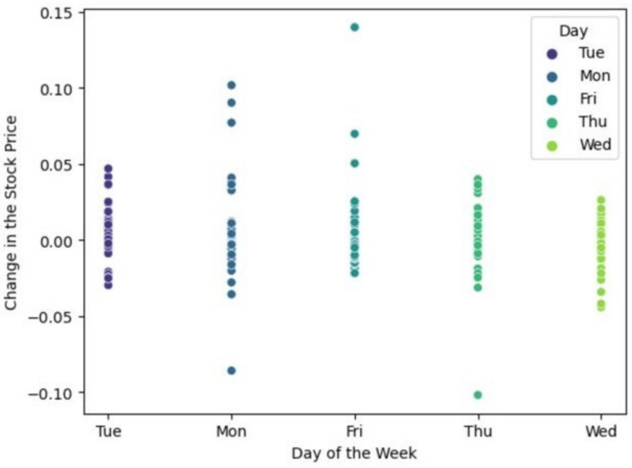

Fig. 2. Scatterplot between day of the week and change of the stock price

capture the trends and present a prediction closer to the targets through careful feature engineering and validation.

VII. Lab 3

A. Class Separation

The classes in the dataset features are not perfectly separated and hence overlapping. This can be due to the complexity involved in the hand-written scripts and probably their degradation due to the passage of time. It is observable that the classifier gets confused while differentiating a class from another, leading to misclassification

B. Behaviour of kNN with increasing k

As k increases in the kNN classifier, the model becomes less sensitive to noise, reducing the risk of overfitting. However, large values of k have the risk of having a model that underfits and smoothes out important differences between classes. Optimal performance was observed with moderate values of k and extremes gave overfitting or underfitting.

C. kNN Classifier Performance

kNN classifier performed reasonably well, but not outstandingly. Frequently it was correct with the right k, though its strong reliance on proximity in feature space can be limiting for data sets with overlapping class boundaries. Seeing metrics like accuracy and F1-score, this classifier is not necessarily among the best options against this dataset

D. Model Fit Analysis

The model exhibited a regular fit, with moderate values of k, where train and test set performance are alike. The lower the values of k, the greater would be the overfitting to data; that is, accuracy would increase on the training set compared to the test set. Similarly, for higher values of k, the model would underfit.

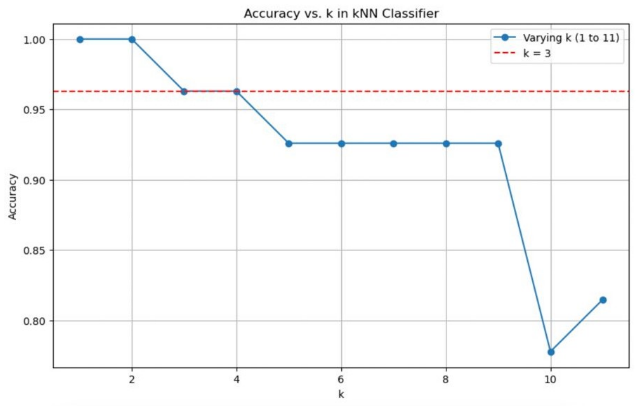

Fig. 3. Accuracy vs k in kNN classifier

E. Overfitting in kNN

Overfitting occurred at very low k values, such as k = 1, where the model became too sensitive to training data, including noise. This resulted in high accuracy during training but poor generalization on the test set. Increasing k reduced overfitting and achieved a better balance in model performance.

VIII. Lab 4

A. Classification Metrics and Model Performance

Despite this, we calculated our performance measures namely accuracy, precision, recall and F1-score for both modes; training and tests. While the results clearly reveal that the model earned higher score on the training set than the testing set, they also feature mild sign of over-learning. For instance, the accuracy was higher in the training scenario but the same performed poorly in the test scenarios. This disparity suggests that the model could have captured the information within the training data set but failed to generalize on new set of data.

Overfitting vs Underfitting: From the results obtained from the training and the test set, it may be deduced that the model was slightly over adjusted. This is clear from the fact that the precision, and recall of the training data were higher than that of a test data, another sign of over fitting. It might be that during training the model learnt not the necessary patterns but the noise inherent in the training data.

B. Regression Metrics (Price Prediction)

For the price prediction task we evaluated the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and the R-squared. The R-squared value, which signals the levels of variance brought by the model, was rather high; this meant that the model could to some extent estimate prices. Nevertheless, the numbers were somewhat large to some extent because RMSE and MAPE depicted that there was some significant error level in the predictions made. This indicate that, the current model will suffice but can still be optimized, for instance by adjusting the model's parameters or applying a more elaborate model to increase its accuracy.

C. Scatter Plot Visualization and kNN Classification

A scatter plot was created for 20 training samples, where two columns of features are available: X and Y; as well as class 0 and class 1. From the visualization it is clear that the been classified points are relatively well separated. Since the kNN classifier was chosen with k=3 for classification, when the test data was classified the scatter plot of the predicted classes given below equally brought out the decision boundary line of the two classes. The points which were grouped as class 0 (blue) were mostly on the left hand side while the points grouped as class 1 which is red occupied the right hand side.

Effect of Changing k: Increasing the value of k caused down sampling of the image and overall the decision boundary was smoothed. As k increased the model was even more overfitting and the class boundary lines less influenced by the local distribution of objects in the training set. But raising k too high overwhelmed the algorithm and the resultant model under fit the data leading to several issues with grasping the details of the data. This confirms the fact that adjusting the value of k is very essential in order to avoid overfitting and underfitting.

D. Hyper-Parameter Tuning

For selecting the right value of k of the standard deviation, we used hyper-parameter tuning with the help of the GridSearchCV structure. Hence the study concluded that the value of k for our data was around 5. The main advantage is that this value offered significant performance with some low complexity, which helps to avoid over-learning of the models. As shown above, lower values of k caused overfitting while higher values of k forced underfitting.

IX. LAB 5

A. Linear Regression Performance

The linear regression model was trained on one of the features of the two attribute dataset. In order to evaluate the performance of the models, we computed different error measures, entailing MSE, RMSE, MAPE as well as R-squared. The given model seemed to have a higher degree of accuracy for the training data while it manifested comparatively high errors while testing on test datasets. This may imply that the model has over-fitted the training data – a scenario in which the model is trained so well on the data availed that it performs very well only on the data used in training.

B. k-Means Clustering

We computed K-Means clustering with k=2 which in other words means that we clustered the data into two groups. We relied on Sample Silhouette Score, Calinski Harabasz Index and Davies Bouldin's index in order to assess the quality of the clustering. I got a pretty good silhouette score, which means that the clusters were rather distinct from one another. In the same regard, the CH score indicated a high value thereby supporting the quality of clustering while the DB index further indicated a low value which pointed to the high level of distinctness of the clusters.

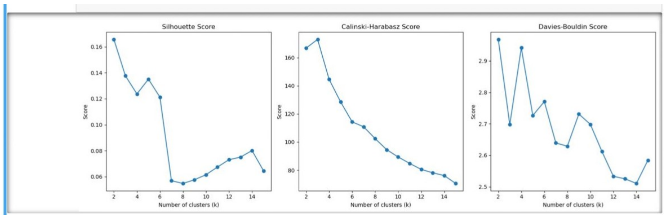

Fig. 4. Silhouette Score, Alinski-Harabasz Score, Davies-Bouldin Score

C. Choosing the Best k (Number of Clusters)

In the process of determining the optimal number of clusters, steps based on different values of k were undertaken and the following consideration elicited that the most appropriate value of k was 2, because rising the number of clusters did not significantly enhance the result.

X. LAB 6

A. AND Gate Perceptron

The perceptron trained on the AND gate logic was able to converge at some certain epoch. The error values went down with changes in weight and the epoch against error diagram sloped down to the required level of SSE.

B. Comparison of Activation Functions

It has been observed that the number of epochs fixed for convergence was different for different activation functions. The Step function on average took a fewer number of epochs than Sigmoid and ReLU function, despite its high shrinkage rate it was evident that Sigmoid had smoother curves of convergence.

C. Learning Rate Analysis

On the other hand, as we increased , the number of epochs required to converge actually went up until a threshold. But, with very high learning rates the model did not converge and lessons were being overshooted which meant that there do exist an optimal learning rate.

D. XOR Gate Perceptron

Since the XOR, for example, is non-linearly separable and therefore needed to be more carefully treated by not so easily converging using a simple perceptron. The takeaways from the results based on such values — additional layers or an entirely different of architecture is needed

TABLE I
EPOCHS TO CONVERGE FOR AND GATE USING DIFFERENT ACTIVATION FUNCTIONS

Activation Function	Epochs to Converge	Final Error
Step Function	Converges(94)	0.000000
Bipolar Step	Did not converge(1000)	3.000000
Sigmoid	Did not converge(1000)	0.013875
ReLU	Converges(167)	0.001996

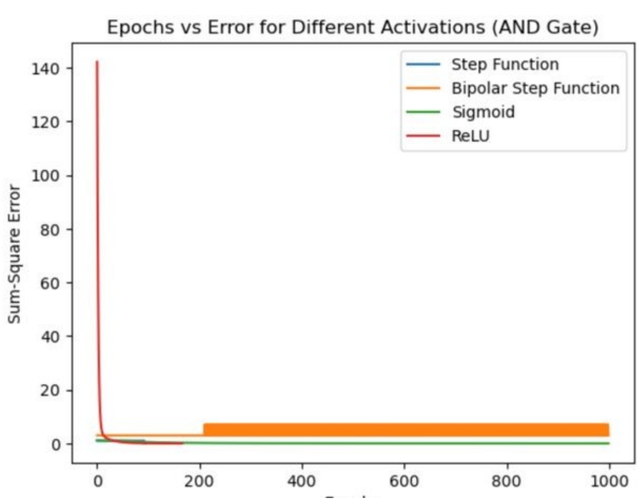

Fig. 5. Epochs vs Errors for Different Activations (AND Gate)

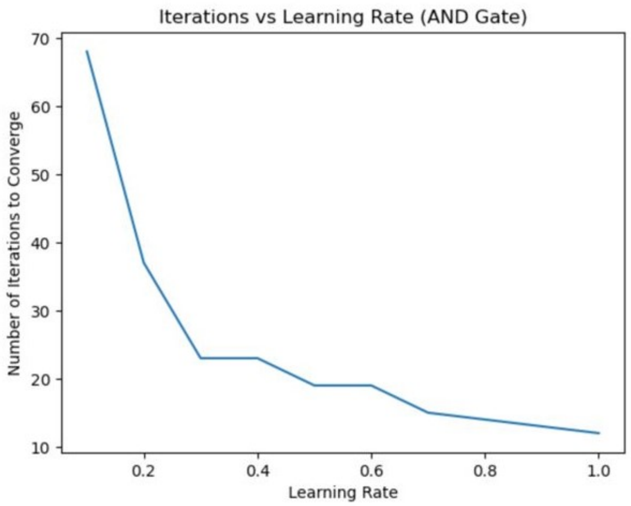

Fig. 6. Iterations vs Learning Rate (AND Gate)

References

[1] De-Jun, Hu-Jie Huang, Guan, Tian-Wen Zhang, Tong-Hua Su, , *"Offline recognition of realistic Chinese handwriting using segmentation-free strategy"* School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, PR China

[2] Isra Al-Turaiki, Najwa Altwaijry· *"Arabic handwriting recognition system using convolutional neural network"*, eceived: 11 October 2019 / Accepted: 3 June 2020 / Published online: 28 June 2020

[3] Imran Siddiqi, Israr Ud Din, Shehzad Khalid and Tahir Azam *"Segmentation-free optical character recognition for printed Urdu text"*

[4] Ambalika Sharmaa, Parveen Kumar, "Segmentation-free writer identification based on convolutional neural network", a Department of Electrical Engineering, Indian Institute of Technology Roorkee, India b Department of CSE, National Institute of Technology Uttarakhand, India

[5] Anjan Kumar Talukdar · Jyoti Ghosh · Kandarpa Kumar Sarma *"A lightweight natural scene text detection and recognition system"* Received: 20 September 2021 / Revised: 15 January 2022 / Accepted: 23 April 2023 / Published online: 13 June 2023

[6] Fanfei Meng* , Branden Ghena *"Research on Text Recognition Methods Based on Artificial Intelligence and Machine Learning"* , . Advances in Computer and Communication, 4(5), 340- 344. DOI: 10.26855/acc.2023.10.014

[7] Ge Peng *"Segmentation-Free Recognition Algorithm Based on Deep Learning for Handwritten Text Image"*, School of Big Data, Baoshan University, Baoshan, Yunnan 678000, China (Received 22 October 2023; Revised 15 February 2024; Accepted 15 February 2024; Published online 05 March 2024)

[8] Francisco Zamora-Martinez, Jorge Gorbe-Moya, Maria Jose Castro-Bleda, Salvador Espan˜a-Boquera, and *"Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models"* , IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 4, APRIL 2011

[9] Jochen Z ¨ ollner, Johannes Michael, Roger Labahn ,Tobias Gruning, *" Evaluating Sequence-to-Sequence Models for Handwritten Text Recognition "* Computational Intelligence Technology Lab University of Rostock 18057 Rostock, Germany {johannes.michael,roger.labahn}@uni-rostock.de , , PLANET artificial intelligence GmbH Warnowufer 60 18057 Rostock, Germany {tobias.gruening,jochen.zoellner}@planet.de

[10] THUYEN TRAN, THUA NGUYEN, DUY-DINH LE , TIEN DO, (Member, IEEE), AND THANH DUC NGO *"SignboardText: Text Detection and Recognition in In-the-Wild Signboard Images"* , University of Information Technology, Ho Chi Minh City, Vietnam Vietnam National University, Ho Chi Minh City, Vietnam Corresponding author: Thanh Duc Ngo (thanhnd@uit.edu.vn) This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2021-26-02.

[11] Tao Wang David J. Wu Adam Coates Andrew Y. Ng , *"End-to-End Text Recognition with Convolutional Neural Networks"* , Stanford University, 353 Serra Mall, Stanford, CA 94305 {twangcat, dwu4, acoates, ang}@cs.stanford.edu, 21st International Conference on Pattern Recognition (ICPR 2012) November 11-15, 2012. Tsukuba, Japan

[12] Batuhan Balci , Dan Saadati , Dan Shiferaw,*"Handwritten Text Recognition using Deep Learning"* bbalci@stanford.edu dans2@stanford.edu shiferaw@stanford.edu

[13] Ashok C. Popat, R. Reeve Ingle, Thomas Deselaers, Jonathan Baccash, Yasuhisa Fujii, *"A Scalable Handwritten Text Recognition System"*, Google Research Mountain View, CA 94043, USA Email: {reeveingle,yasuhisaf,deselaers,jbaccash,popat}@google.com , 2019 International Conference on Document Analysis and Recognition (ICDAR)

[14] Gyeonghwan Kim1, Venu Govindaraju2, Sargur N. Srihari2 *"An architecture for handwritten text recognition systems"* 1 Department of Electronic Engineering, Sogang University, CPO Box 1142, Seoul 100-611, Korea; e-mail: gkim@ccs.sogang.ac.kr 2 CEDAR, State University of New York at Buffalo, 520 Lee Entrance, Amherst, NY 14228–2567, USA Received October 30, 1998 / Revised January 15, 1999