Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 227 (2023) 779–784

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023)

Implementation of Handwriting Recognition and Answer Evaluation with Recurrent Neural Network

John Suryajaya Pangestu[a], Kelvin Kho[a], Dewi Suryani[a]*, Leo Andika[a]

[a]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, 11480, Indonesia*

Abstract

Many teachers still utilize exams to gauge their pupils' knowledge. However, as the teacher must comprehend each answer provided by each student individually, analyzing the exam response can take a long time. Additionally, different handwriting styles may lead to errors in scoring the response, resulting in an erroneous mark. This study suggests a web-based program that can read students' handwriting to assist teachers in assessing students' responses. The objective of this study was to develop a web-based application that can accept student responses as input and output correctness.

Keywords: Handwriting Recognition; Exam; Recurrent Neural Network

1. Introduction

Tests and exams are part of the evaluation and assessment processes used to meet academic requirements. [1]. Various types of exam questions can be answered during an examination. The most commonly used type is the essay [5]. In essay type exam question, a student needs to communicate their answer in handwritten form. Then, the teacher needs to score the student's answers by reading them one by one. However, this process heavily relies on human raters, who understand both the content and the quality of writing [11] In the other hands, responding to student papers can be a burden for teachers[8].

This paper presents an implementation of handwriting recognition for exam answer evaluation that aims to overcome this problem. The implementation will be divided into three steps. First, the system will take the exam answer as input. Then, the system will compare the answer with the answer key stored in the system to calculate the result. Lastly, the system will produce the appropriate score for the inputed answer.

The structure of the essay is as follows. The related work is presented in Section 2. Afterward, section 3 presents the algorithm used to implement the system, then section 4 presents the system testing methodology and its result. Finally, conclusions and future work are presented in section 5.

*Corresponding author.
E-mail address: dewi.suryani@binus.ac.id

10.1016/j.procs.2023.10.583

2. Related Work

Many works have been proposed to use handwriting recognition implementation in scoring an essay answer. Ramalingan et al. proposed an automated essay grading by utilizing the Linear Regression technique to train the model along with making the use of various other classifications and clustering techniques [8]. Their project mainly tested on the Automated Essay scoring with e-Rater technique where they provide the essay as an input, compare it with the essays of each set by their polarities, words used, and the content of the essay.

Without using any feature engineering, recurrent neural networks were also employed to create a method for figuring out the relationship between an essay's grade and its given grade [14]. To automating essay scoring, Taghipour et al. investigated several neural network models and carried out some analysis to gain some understanding of the models. The experiment's findings demonstrate that, without the use of feature engineering, their best system, which is built on extended short-term memory networks, beats a strong baseline by 5.6% in terms of quadratic weighted Kappa.

In 2002, Rudner et al. improved two Bayesian text classification models and used them to analyze student writings [10]. With 426 essays of 2 score points each, they calibrated both models. With their model, they were able to attain accuracy of more than 80%.

By integrating Optical Handwriting Recognition (OHR) and Automated Essay Scoring (AES) techniques [13], To give assessment results and feedback, Srihari et al. were able to create a system that could read, grade, and analyze handwritten essays from large-scale examinations. The AES system employs a set of human-scored responses to derive the scoring system parameters using a machine learning approach. It is based on the latent semantic analysis methodology. The OHR system goes through a number of pre-processing procedures, including removing forms, removing rule lines, and segmenting text lines and words.

A comprehensive convolution-based deep network architecture for recognizing cursive handwriting from line-level images was proposed by Sharma et al [12]. The architecture consists of a Connectionist Temporal Classification (CTC) output layer with 2-D and 1-D dilated non-causal convolutions. It is demonstrated that their model, using recurrent architectures, has equal performance on CER and WER metrics based on their trials with English and French handwriting. Modern models using various architectures based on recurrent neural networks (RNN) and their variations are used for the comparison. They also noted how their model is suited for low-resource and environmentally friendly deployment because it has fewer parameters and requires less training and testing time.

3. Algorithm Implementation

In this study, we suggested combining Connectionist Temporal Classification (CTC) Decoding, Bidirectional Bidirectional Long Short-Term Memory (BLSTM), and Convolutional Neural Networks (CNN) with 7 layers to decode the output of BLSTM to text. The details are explained below:

The supplied image needs to be preprocessed as the first step. Preprocessing an image is a step in the process of removing image noise. picture smoothing, edge detection, picture categorization, and other procedures are typically part of the processing process [15]. To give the algorithm noise-free images to train on, this step is necessary. We preprocessed the image in our system by making it grayscale and resizing it to 800 x 64 pixels.

Afterward, we extract the important features on the image that have already been preprocessed. The process of feature extraction is crucial to picture categorization. It provides for the most accurate representation of image content. [7]. In this step, we used 7-layer Convolutional Neural Networks (CNN) to get important features from the images. The result from this step is 100 x 8 features of 512 that will be fed into the Bidirectional Long Short-Term Memory (BLSTM) algorithm.

The use of BLSTM based RNN facilitated using forward and backward layers to get longer range context in both directions [9]. BLSTM algorithm will then process the input from previous step and do the sequence labelling process from the features which have similar pattern. To train the RNN to recognize similar patterns out of different handwriting, since different people have different kinds of handwriting, we composed the ground truth text to calculate the loss. This step helps us to minimize the negative maximum likelihood path. Finally, we use CTC to decode the result from the previous step to produce the recognized text that will be compared to the stored dataset earlier.

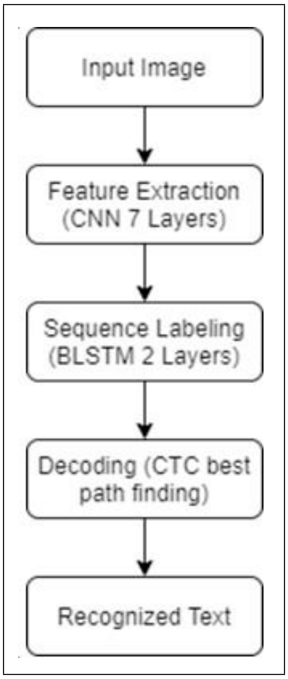

Fig. 1. Proposed Algorithm Flowchart Diagram

4. Result and Discussion

4.1. Experimental Settings

The experiment is done using a computer with Intel Core i7-7700HQ processor which have 3.8GHz maximum turbo frequency, along with 4.00 Gigabytes (GB) of RAM and 500 GB of Hard Disk Space. For the software, we used Windows 10 Operating System and PyCharm Integrated Development Environment.

To do the experiments, we used the dataset that we retrieved from IAM Dataset by Papers With Code[6] website. IAM Dataset consists of 13.353 images of handwritten lines of texts created by 657 writers.

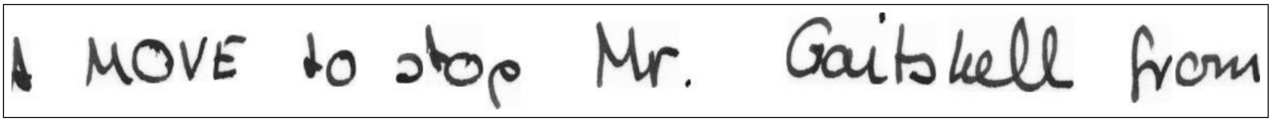

Fig. 2. Example iamge of IAM Dataset

In this study, we used the word segmentation function called contour from openCV [3]. The function helps us to divide the input image into segments based on every line of text. We combined the function with a built-in python sorting algorithm to produce the sorted word segmentation. The result is shown in figure 3

To optimize the training rate for our system, we applied RMSProp as the optimizer. RMSProp is an optimization algorithm designed for neural networks. This optimizer is primarily dependent on the current gradient and gradient moment averages [4]. RMSProp uses 0.001 as the learning rate and we use 25 as Batch Size. The bigger the batch size used, the better the loss graphic will be, and it helps us better in optimizing the process. The comparison between the smaller and bigger number of batch sizes is shown in figure 4. The vertical axis shows the CTC loss value, and the horizontal axis shows the time in seconds. The upper image shows a batch size of 5, while the lower image shows a batch size of 25.

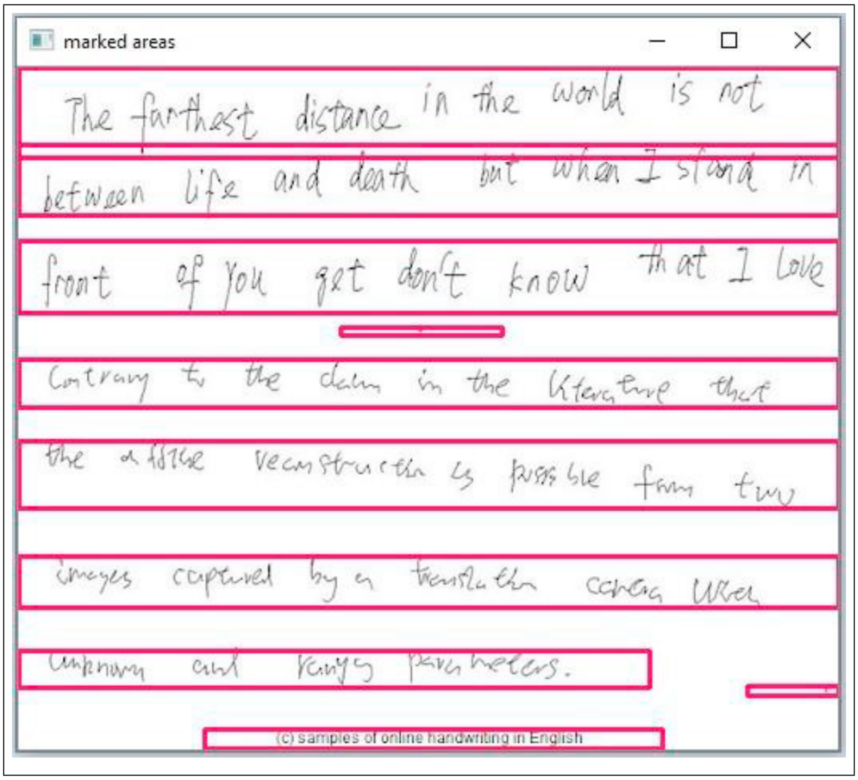

Fig. 3. Word Segmentation Result using contour function

4.2. Experiment and Result

We implemented 2 models of Artificial Intelligence (AI) to test our design. In the first model, we apply 8-layers of CNN and the combination of 7-layers of CNN and 2-layers of CNN-Long Short Term Memory (LSTM) algorithm. The purpose of using 2 models of AI is to test whether the model is producing higher accuracy for the same inputs. We use the same learning rate for each model which is 0.001. For the testing process, we used 10.400 datasets and 2.600 datasets for the validation process. The results of our experiments can be seen in table 1.

Table 1. Experiment results with different AI models

Method	Learning Rate	Training Time	Validation Accuracy	Testing Accuracy	Testing Time
CNN	0.001	±5:00:00.00	37.05%	35.11%	±0:00:30.00
CNN-LSTM	0.001	±10:00:00.00	50.54%	50.21%	±0:00:60.00

From table 1, it is shown that the CNN-LSTM method gives better results in terms of validation accuracy and testing accuracy. In addition, with the CNN-LSTM method, we achieved the Character Error Rate (CER) of 8.654728%. This result was obtained by continuing the training process until the CER is stable when it was more than 25 epochs. For the learning rate, we used 0.001 as starting value and will be adjusted by the models during the training process.

We used Best-path Decoding for the CTC Decoding step in the mentioned experiment. Best-path decoding was chosen because it is the fastest decoding algorithm where we only have to choose the character with the maximum probability at every time-step [2]. Other options is to use Beam Search Decoding algorithm which explore more solution with the cost of longer running time.

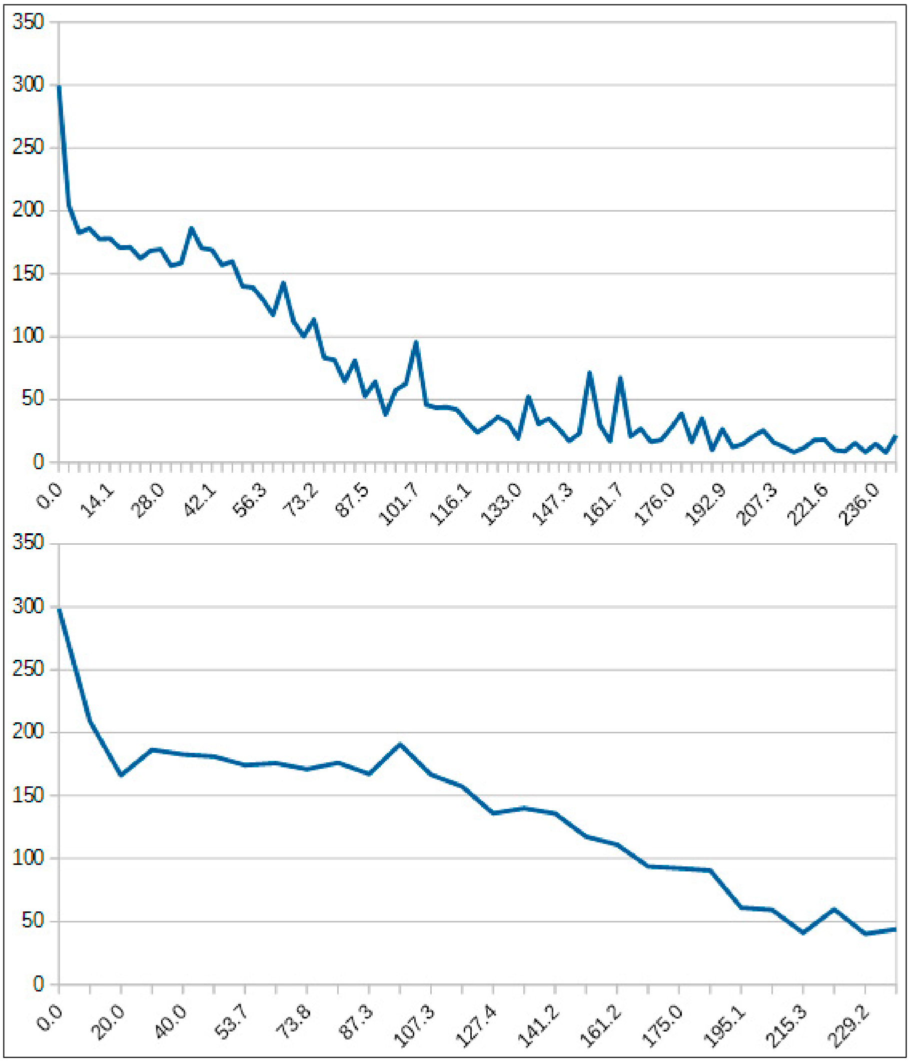

Fig. 4. Results from different Batch Size

The consideration of using Best-path decoding over Beam Search decoding is based on the shorter running time needed to run best-path decoding algorithm since CNN-LSTM cosumes a lot of time based on table 1. Therefore, best-path decoding is more suitable in our research.

5. Conclusion and Future Work

This paper has presented the development of a web application to help teachers in scoring students' essay answers with high consistency and accuracy by implementing the Recurrent Neural Network. The importance of this work is that the application can be used to reduce the scoring time human errors, thus provide better results. In addition, the Deep Learning algorithm is proven to give better performance and accuracy in the process.

In future work, we will focus on improving the line segmentation process to reduce the mistakes in the segmentation step. In addition, we will implement Natural Language Processing (NLP) to allow the application in extracting information from more complex essay answers. Finally, we will improve the user interface to increase usability and attract more users in using this application.

References

[1] Amir Mohamed Talib, Fahad Omar Alomary, H.F.A., 2018. Assessment of student performance for course examination using rasch measurement model: A case study of information technology fundamentals course. Education Research International 2018.
[2] Bansal, S., 2019. Decoding connectionist temporal classification. https://sid2697.github.io/Blog_Sid/algorithm/2019/11/04/Beam-search.html.
[3] CV, O., 2021. Contours : Getting started. https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html.
[4] Dauphin, Y., Vries, H., Chung, J., Bengio, Y., 2015. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. arXiv 35.
[5] Kurdi, M., 2015. Essay type questions their improvements short answer questions.
[6] Marti, U.V., Bunke, H., 2002. The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition 5, 39–46.
[7] Medjahed, S.A., 2015. A comparative study of feature extraction methods in images classification. International Journal of Image, Graphics and Signal Processing 7, 16–23. doi:10.5815/ijigsp.2015.03.03.
[8] Ramalingam, V., Pandian, A., Chetry, P., Nigam, H., 2018. Automated essay grading using machine learning algorithm, in: Journal of Physics: Conference Series, IOP Publishing. p. 012030.
[9] Ray, A., Rajeswar, S., Chaudhury, S., 2015. Text recognition using deep blstm networks. doi:10.1109/ICAPR.2015.7050699.
[10] Rudner, L.M., Liang, T., 2002. Automated essay scoring using bayes' theorem. The Journal of Technology, Learning and Assessment 1.
[11] Sevcikova, B.L., 2018. Human versus automated essay scoring: A critical review. Arab World English Journal (AWEJ) 9, 157–174.
[12] Sharma, A., Jayagopi, D.B., 2021. Towards efficient unconstrained handwriting recognition using dilated temporal convolution network. Expert Systems with Applications 164, 114004.
[13] Srihari, S., Collins, J., Srihari, R., Babu, P., Srinivasan, H., 2006. Automated scoring of handwritten essays based on latent semantic analysis, in: Bunke, H., Spitz, A.L. (Eds.), Document Analysis Systems VII, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 71–83.
[14] Taghipour, K., Ng, H.T., 2016. A neural approach to automated essay scoring, in: Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 1882–1891.
[15] Zhou, W., Ma, X., Zhang, Y., 2020. Research on image preprocessing algorithm and deep learning of iris recognition. Journal of Physics: Conference Series 1621, 012008. doi:10.1088/1742-6596/1621/1/012008.