विश्वजीवनामृतं ज्ञानम्

Atal Bihari Vajpayee Indian Institute of Information Technology and Management Gwalior

Information Technology
Mini Project ITIT-3203

Notebook Plagiarism Checker using Optical Character Recognition Technique

---

by
Praveen Kumar Sahu
(2018IMT-070)

Under the supervision of
Prof. Rajendra Sahu

1 Abstract

Nowadays, due to the pandemic, people generally avoid going to a crowded place. This pandemic has also impacted our education system as people currently have fear. So due to this, there was a tremendous increase in the use of an online examination system, a system used to evaluate the individual's knowledge. But there is a significant drawback of this system, as many individuals use online sources to answer the question asked to them. Because of this drawback, deserving candidates were suffering as despite using their mind, they are getting similar marks that candidates who are just copy-pasting. This also has an increasing burden on the course instructor as they have to face difficulty in determining whether the response given by candidate was by using own mind or by using online available sources. So reduce the burden on the course instructor a system need to be developed which will help them to check the plagiarism in the candidate response.

Notebook Plagiarism Checker tool aims to simplify the process of checking the notebook by an individual for plagiarism. This tool will be leveraging the capabilities of text detection and text recognition mechanism. This checker tool will recognise the character from the notebook's image and convert the recognised text into an editable text file. This tool is based on artificial neural networks. Prepossessing methods will enhance the images which will help neural network to capture more features from the input image. CNN layers was used to extract the feature from the prepossessed image which were the passed to RNN layers which will help to get the characters from the input image. The generated distribution over character were then decoded using connectionist temporal classification function which will generated the desired result. For model building and training, IAM handwritten dataset was used.

Contents

2 Introduction

2.1 Background Information

There is a significant need for the automated tool to check plagiarism in the handwritten notebook. This tool will give more power to notebook checker/ course instructor to check it for plagiarism. As a checker, knowing the sources of writing will be a crucial part to determine how much score any candidate will get. This project can also be used to convert the handwritten note into an editable text file. The primary motivation behind the making of this tool is the evolution of the online examination system. A significant drawback of this new system is that it is hard to check if the answer written by any candidate was by his efforts or whether they have copied it from some source. One way to check plagiarism in a handwritten notebook is by manually typing each word written by the candidate, but this process requires a lot of time. Other way to do so is by utilising the available technologies.

2.2 Literature Review

There has been a lot of improvement in the text recognition, text detection and vision tasks. Also, the majority of works were related to deep neural networks. There has been use of attention based technique for text recognition such as (9). In this project, I will be utilising the pipeline structure which involves the preprocessing steps which involve convolution neural networks to extract the features followed by a recurrent neural network that consists of LSTMs(1) which generates the output sequence. Similar approach has been taken into (2; 3). This work will use the concept introduced in the Attention OCR paper(3) with some modification in the architecture of neural networks used, in order to improve the accuracy of the system. There has been many research made and many going on handwriting recognition(8). Many attempt have been made in markup generation from handwritten mathematical expression using attention mechanism(4). There has been many proposed methods(11) for the handwriting recognition which uses Multi-Dimensional LSTMs(14; 16) and connectionist temporal classification(CTC)(10) function.

2.3 Project Objective

Exam and tests are the way to check the how much knowledge candidate has gain with regards to particular subject. Nowadays there is increase in use of online examination method. But there is major drawback of this system is that the deserving candidate is not getting benefit through this online system because of the use of wrong methods and copy pasting of answers from online sources. The aim of this tool is to check for the plagiarism in candidate response. But how can we check for plagiarism in handwritten response. So I thought of using deep learning methods to build character recognizer which can be used to convert the notebook's image to editable text file which can be used to check plagiarism. I will make document editable as this tool can also be used to prepare notes. The project objective is to combine the attention based text extraction with handwriting recognition technique to build unique solution for checking plagiarism in handwritten textual data.

3 Methodology

3.1 System Architecture

Firstly user adds the file to be converted, then each page of the file was separated and converted to image format and will be used as input to the major component involved in this proposed tool i.e. neural network, which will be trained to predict the text from the images. The proposed model is combination of convolution neural network for text detection and recurrent neural networks for predicting the sequence of characters. Then this sequence of characters were passed through transcription layer which forms label and converts the characters into meaningful words. These word are the are added to the text file which is the final output of the tool.

3.2 Design Diagram

The following flowchart is the generalized block diagram of the whole application:

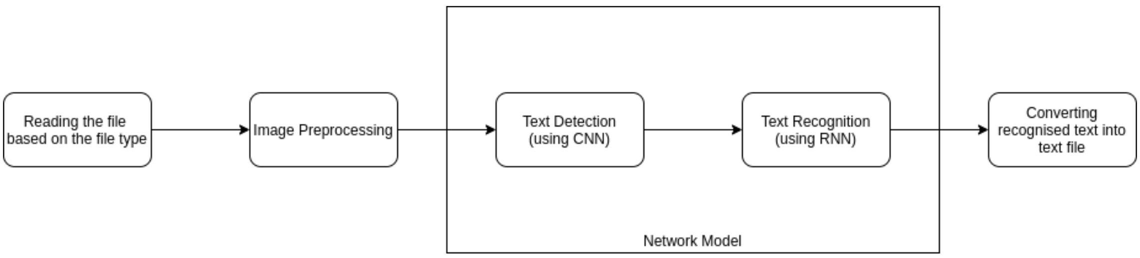
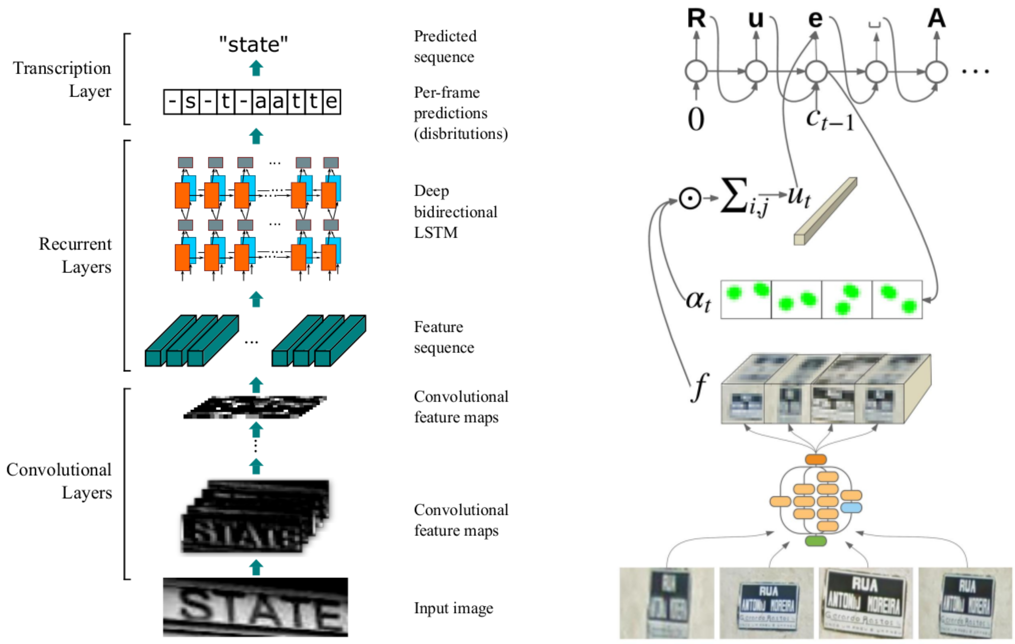


Figure 1: Flow diagram of tool

3.3 Preprocessing

At start some preprocessing tasks were performed on images to enhance the feature and details of the input image which might help in improving the accuracy and efficiency of the system. These processed input images were then passed through the convolutional layers, to extract the feature sequence from them, then this feature



, ,

(a) Network architecture
(from arXiv:1507.05717v1)

(b) Architecture of model
(from arXiv:1704.03549)

Figure 2: The structure of the model

sequence was used as input for the recurrent neural network to identify or predict the sequence or character distribution. Then the predicted sequence was passed through transcription layer which converts the determined sequence into labeled sequence and also calculates the character error rate or word error rate.

3.4 CNN-feature sequence extraction

This section is about the use of convolution neural networks for the feature extraction tasks from the scanned image. Before passing the image to the CNN some preprocessing tasks was done which include dilation, erosion, filtering for noise removal and resizing so all the input image have similar characteristics. After this preprocessed image were passed through CNN which generate feature vectors. Then sequence of feature sequence was formed using the generates feature vectors. ANN model consist of 7 CNN layers that maps the input image to feature sequence of size $100 \times 1 \times 512$ i.e. CNN will map input image onto sequence of width, length and feature 100, 1 and 512 per element respectively.

3.5 RNN-sequence labeling

As we now have the feature sequence from the image, now we need to transform this to a text string. So I used deep bidirectional Recurrent Neural Network, which is capable of capturing the useful information from the feature sequence as time t and then using the captured information to capture the new information from coming into RNN at time $t+1$. RNN uses hidden states to store the previously determined information and use this information determine the next coming sequence. Each time RNN receives any feature sequence x_t it updates it hidden state or internal state, then according to this state and input sequence output is generated.

This LSTM layer with 512 hidden map the feature sequence to a matrix of size $100 \times 1 \times 512$

3.6 Transcripting frame sequence using CTC decoder

This process was done to convert the prediction of the RNN to more meaningful sequence. So to convert the output of RNN into meaningful word CTC decoding(10) algorithm was used which was capable of generating the desired text from the output matrix of the RNN. CTC function can also be used to compute the loss value of the model provided with the ground truth text. But CTC layer needs a 1D sequence so the output from the RNN was unfolded into 1D, using simple reduction operation along the one if the dimension (usually vertical dimension was used) here height dimension, i.e.

$$P_i = \sum_{j=1}^{H} F(I_{i,j}) \tag{1}$$

After mapping, CTC decoding layer can gives at most 100 character long outputs.

3.7 Training

Training of model will be done using CTC loss function with RMSProp optimization algorithm(12) which control the learning rate and works well with non-stationary probability very well. RMSProp uses default learning rate of 0.001 and decaying rate of 0.9. While training model, batch normalisation technique was used to make the input normalised re-centering and re-scaling it. Individual training of CNN and RNN will not be required as IAM dataset(13) contains image of handwritten document and their corresponding true ground text. The speed and accuracy of the model can be varied by depth of the neural network. As the depth of CNN increases accuracy increases but speed of prediction decreases. So hyper parameter tuning will be an important steps in training model.

Type	**Description**	**Output Size**
Input	grayscale image	$800 \times 64 \times 1$
Conv + Pool	kernel 5×5, pool 2×2	$400 \times 32 \times 64$
Conv + Pool	kernel 5×5, pool 1×2	$400 \times 16 \times 128$
Conv + Pool + BN	kernel 3×3, pool 2×2	$200 \times 8 \times 128$
Conv	kernel 3×3	$200 \times 8 \times 256$
Conv + Pool	kernel 3×3, pool 2×2	$100 \times 4 \times 256$
Conv + Pool + BN	kernel 3×3, pool 1×2	$100 \times 2 \times 512$
Conv + Pool	kernel 3×3, pool 1×2	$100 \times 1 \times 512$
MDLSTM	bidir. LSTM with 512 hidden cells	$100 \times 1 \times 512$
Collapse	remove dimension	100×512
Project	project onto C classes	$100 \times C$
CTC	decode and loss	less than 100 characters

Table 1: Architectural details of neural network backbone. This table abstracts the details of each layers.

3.8 Datasets

3.8.1 IAM Handwriting Database

I'll be using this dataset to train, validate and test my model to recognise the character from the real world object. This datasets(13) contains total of 1539 pages of scanned text or 13,353 isolated and labeled text lines from 657 writers.

3.8.2 OCR dataset by Rob Kassel

I'll be using this dataset to fine tune my model and make it more capable to recognise the handwritten character. This dataset contains handwritten words dataset collected by Rob Kassel at MIT Spoken Language Systems Group(6).

4 Results and Discussion

This fully trained deep learning model will be capable of identifying the handwritten character from the given file and convert into into digital text format. The whole neural model used comprise of CNN and RNN and CTC loss and decode functions, on which training was done with image dataset with ground truth text as label. So CER will be used to determine the performance of the tool or neural network model.

Epoch	CER (%)	Epoch	CER (%)
1	92.94	11	40.55
2	91.74	12	7.39
3	91.70	13	5.99
4	71.94	14	4.891
5	72.44	15	2.81
6	53.24	16	22.00
7	51.28	17	8.41
8	35.03	18	6.46
9	22.27	19	4.29
10	13.23	20	3.24

Table 2: Epoch v/s Character error rate(CER)

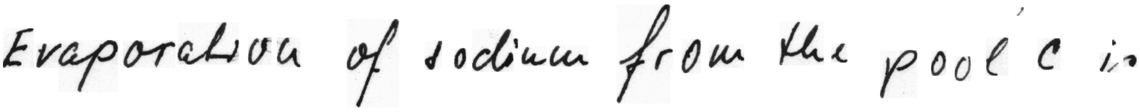


Figure 3: Predicted text: Evaporation of sodium from the pool C is

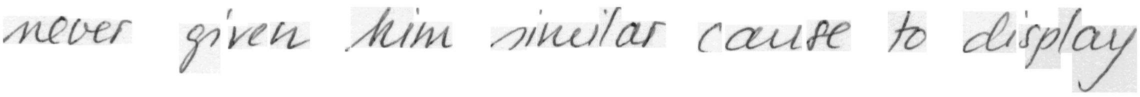


Figure 4: Predicted text: never given him similar cause to display

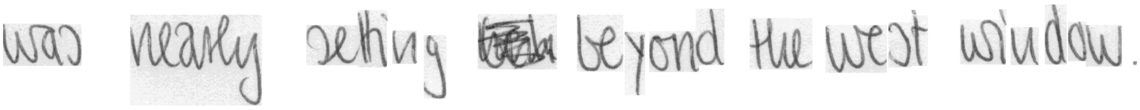


Figure 5: Predicted text: was nearly setting beyond the west window

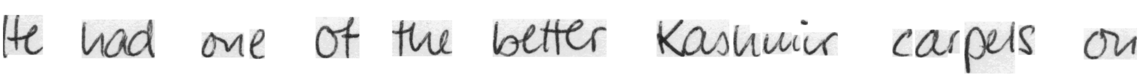


Figure 6: Predicted text: He had one of the better Kashmir carpets on

5 Conclusion and Future Scope

5.1 Advantages

1. In this paper, I have presented an implementation of neural network which can be used to convert the image with text to a editable text file which can be used to check plagiarism in it. I will utilise the deep learning approach and will use the attention mechanisms for the text detection and text recognition, which will be trained on IAM handwriting dataset which was further fine tuned using the dataset of handwritten character.

2. This tool will be used to reduce the typing effort of the individual as this allow them to just scan the file and get notes in ready to edit text format.

5.2 Limitations

Currently the presented model in this report was only able to detected the characters from one line at a time. So it requires segmentation of the multi line handwritten document. For detection of whole form or page or multi line document we need to train this model using greater input size along with some tweaks on the subsequent layers. Another major limitation of the was, as CTC was used in the model so if there are massive number of possible alignment of the character then CTC can be very costly to compute. We can improve the tool's accuracy by using more dataset or adding extra layers in neural networks we used on CNN and RNN and using text correction algorithm to reduce the character error rate.

5.3 Future Work

In the future this tool could be improved to generate the flowcharts, tables and diagrams from the hand-drawn drawing, which can be very useful in field of graphic design where hand drawn character can be directly converted into editable file. Improved version of this tool can be used in business and government institution to convert the old files and records in digital format. This tool can also be improved to convert multi line document to text file using the methods described by (14; 16). Text correction algorithm can be used to correct the spelling of the detected text.

References

[1] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, Jürgen Schmidhuber. LSTM: A Search Space Odyssey. *arXiv preprint arXiv:1503.04069*

[2] Baoguang Shi, Xiang Bai and Cong Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *arXiv:1507.05717*

[3] Zbigniew Wojna, Alex Gorban, Dar-Shyang Lee, Kevin Murphy, Qian Yu, Yeqing Li, Julian Ibarz. Attention-based Extraction of Structured Information from Street View Imagery. *arXiv:1704.0354*

[4] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, Alexander M. Rush. Image-to-Markup Generation with Coarse-to-Fine Attention. *arXiv:1609.04938*

[5] Smith, Raymond, et al. "End-to-End Interpretation of the French Street Name Signs Dataset." European Conference on Computer Vision. Springer International Publishing, 2016.

[6] Rob Kassel at MIT Spoken Language Systems Group. *https://ai.stanford.edu/ btaskar/ocr/*

[7] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4168–4176.

[8] Ciresan, Dan Claudiu, Meier, Ueli, Gambardella, Luca Maria, and Schmidhuber, Jürgen. "Deep, big, simple neural nets for handwritten digit recognition." Neural computation, 22(12):3207–3220, 2010.

[9] Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhudinov, Ruslan, Zemel, Rich, and Bengio, Yoshua. "Show, attend and tell: Neural image caption generation with visual attention". *In Proceed- ings of The 32nd International Conference on Machine Learning*, pp. 2048–2057, 2015.

[10] A Graves, S Fernández, F Gomez, and J Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In International Conference on Machine learning, pages 369–376, 2006.

[11] A. Tong, M. Przybocki, V. Maergner, and H. El Abed. NIST 2013 Open Handwriting Recognition and Translation (OpenHaRT13) Evaluation. In 11th IAPR Workshop on Document Analysis Systems (DAS2014), 2014.

[12] Sebastian Ruder, "An overview of gradient descent optimization algorithms". *arXiv:1609.04747*

[13] U. Marti and H. Bunke. "The IAM-database: An English Sentence Database for Off-line Handwriting Recognition". *Int. Journal on Document Analysis and Recognition, Volume 5, pages 39 - 46, 2002.*

[14] T. Bluche, J. Louradour, and R. Messina. Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. In *2017 14th IAPR International Confer- ence on Document Analysis and Recognition (ICDAR)*, vol- ume 1, pages 1050–1055. IEEE, 2017.

[15] Harald Scheidl, Handwritten Text Recognition in Historical Documents, *http://hdl.handle.net/20.500.12708/5409*

[16] Mohamed Yousef, Tom E. Bishop, OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by learning to unfold, *arXiv:2006.07491*