Improving Cursive Handwriting Recognition Deep learning networks using Deslanting Normalization and Data Augmentation

Richard Louie T. Orilla
College of Computing Studies
De LaSalle University
richard_orilla@dlsu.edu.ph

Abstract: Optical character recognition is the process of conversion of images that contains typed, handwritten or printed text into a machine encoded text. Several studies have already been conducted in the creation of OCR for unconstrained cursive writing with acceptable results ranging from 70% to 85% using neural networks instead of using Hidden Markov Models. Their approach is based from the study of Yuan et. al where detection of handwriting is made using segmentation points. Currently the popular approach in Optical Character Recognition is through the use of deep-learning. In this paper, it explores the effects of specific algorithms when applied during the learning and training phase of a deep learning neural network framework, specifically the deslanting algorithm of Allesandro Vinciarelli and Juergen Luettin and random stretching of images.

Keyword(s): Handwritten word recognition, cursive handwriting, preprocessing, segmentation, optical character recognition, cursive handwriting, search, graph, lexicon matching, deep learning.

2. REVIEW OF RELATED LITERATURE

Previous researches have been made in cursive handwriting recognition. Most of these researches are directed towards utilization of Hidden Markov Models which is used in implicit segmentation of cursive handwriting characters. Explicit segmentation is considered difficult because of Sayre's Paradox. Hidden Markov Models have only been moderately successful while recurrent neural networks have delivered the best results to date [1].

2.1 STUDIES RELATED TO HIDDEN MARKOV MODEL

2.1.1 OPTICAL CHARACTER RECOGNITION FOR CURSIVE HANDWRITING

Using global parameters for estimation purposes, a segmentation method inspired from the work of Lee et al. and a Hidden Markov Model employed shape recognition. The study was able to create an offline OCR System for Cursive Handwriting. In this study it concludes that the problem in cursive handwriting is made complex by the fact that the writing is inherently ambiguous as the letters in a word are linked together including factors such as poor handwriting and even missing letters [1]. The development of this powerful segmentation algorithm which utilizes character boundaries, local maxima and minima, slant angle, upper and lower baselines and stroke height to prevent over-segmentation of characters.

2.1.2 OPTICAL CHARACTER RECOGNITION FOR HANDWRITTEN CURSIVE ENGLISH CHARACTERS

By applying a median filter during feature extraction, the study was able to improve the result of the recognition of the HMM due to errors caused by noise in the scanned image. The research mentions the Gabor filter and its drawbacks of being computationally intensive algorithm [4]. Due to performance concerns, the research favored the median filter over Gabor filter.

2.2 STUDIES RELATED TO NEURAL NETWORKS

2.2.1 HANDWRITTEN ENGLISH WORD RECOGNITION BASED ON CONVOLUTIONAL NEURAL NETWORKS

The study of Yuan et al. involves in presenting a novel segmentation technique that involves in origin segmentation and segmentation points and feed it to the lexicon-driven handwritten English recognition system. In their approach, they modified an existing rule-base methodology in paper [3] by adding additional steps after the segmentation because of the methodology's inability to segment apart adjacent characters with certain traits. This additional approach tries to detect and locate missing segmentation points during the initial segmentation and correct them in order to increase the accuracy of the algorithm.

2.2.2 FREEFORM CURSIVE HANDWRITING RECOGNITION USING A CLUSTERED NEURAL NETWORK

This study takes another approach by using a special type of feedforward neural network that converts freeform cursive handwriting to searchable text. Hidden node in the network are also present which are grouped into clusters, with each cluster being trained to recognize a unique character in the bigram. The approach in here is the use of clustered architecture of the feed-forward neural network, backed with an expanded set of observers combining image masks, modifiers, and feature characterizations and the use of overlapping bigrams as the textual working unit to assist in analysis and reconstruction.

2.2.3 Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm

Recurrent Neural Networks (RNN) output a sequence of character probabilities per sequence. Traditional approach in Hand Text recognition using RNN is to approximate the decoding output of the network which is called best path decoding. In the study of Scheild et al, they made an extension of a heuristic search algorithm, named as Word Beam Search. The algorithm's approach is a prefix-tree that contains possible word characteristics and a language model which is used to predict the upcoming words given the previous words. The language model is also able to assign probabilities to given sequence of words. The algorithm is well suited when a large amount of words to be recognized is known in advance but gives much more accurate results [4].

2.2 STUDIES RELATED TO CURSIVE DATA NORMALIZATION

2.3.1 A new normalization technique for cursive handwritten words

Several deslanting algorithms had been proposed in the past such as the traditional method for slope removal by Bozinovic and Srihari in 1989 that assumes that slanting information is contained in the almost vertical strokes or the used of slanted histograms by Guillevic and Suen in 1994. In the paper of Allesandro Vinciarelli and Juergen Luettin however, the new normalization technique is based on the hypothesis that the word is deslanted when the number of columns containing a continuous stroke is at maximum which when paired with Hidden Markov Models for recognition allows a much more accurate representation of the deslanted words [2].

3. THEORETICAL FRAMEWORK

The architecture design of this new way of segmenting cursive characters is separated into three components. Each of these components specializes on one specific task and each work in a linear fashion where one component passes its result to the next component as seen in the diagram below.

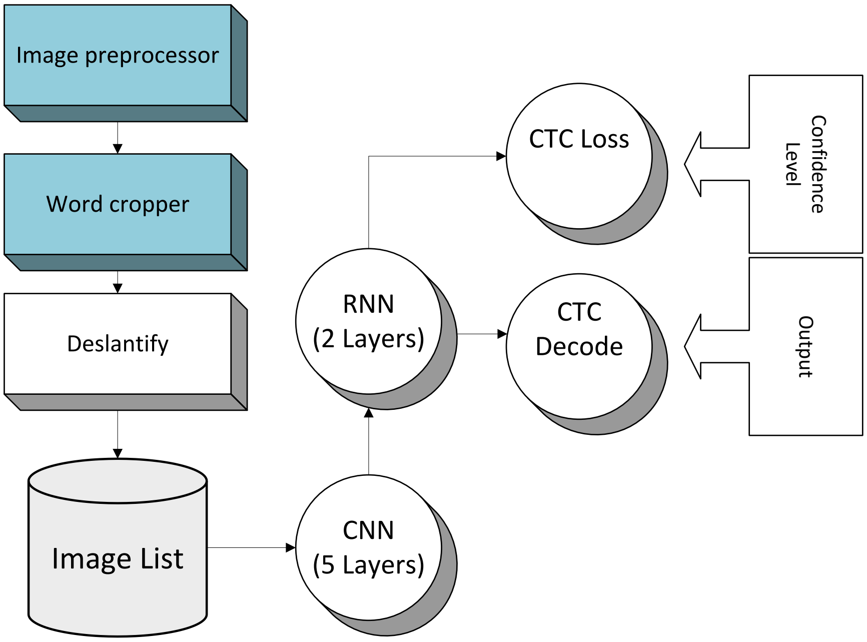


Figure 1 The conceptual model of the proposed system

3.1 Image Preprocessor

The goal of the image preprocessor component is to take the image and remove unnecessary noise in the image. This is done by first making the image grayscale and then applying a threshold to the image where any color value that is less than 160 is removed. Since there are cases of images that contain noises after thresholding as seen in the figure below, erosion is necessary. Erosion is applied once to the image.

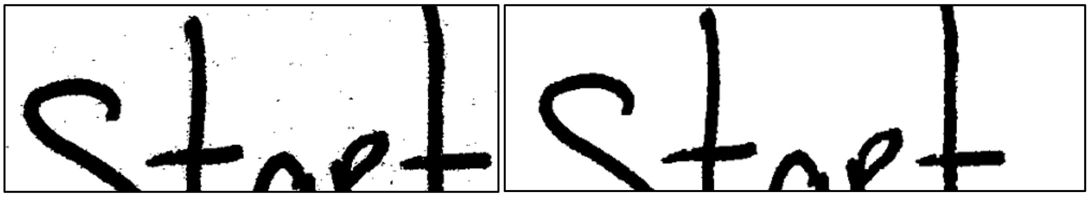


Figure 2 Image A (the one shown in the left) shows some noises in the image after thresholding is applied while Image B (shown in the right) is the effect after erosion is applied to the image

Dilation is then applied after this process in order to make it easier to determine the parts in the image that contain text. Two iterations are done first and then saved as the final image to be cropped which will be the one to be feed to the neural network, while further nine iterations are applied in order to make it easier to measure the moment on the image. Before it gets passed to the word cropped, the image is inverted first where instead of showing black as the text, it will become white instead.

3.2 Word Cropper

The analysis topological structure of binary images by border following is needed in order to determine the regions of the image that contains text after preprocessing. Several border following algorithms have already been presented but in this study the implementation of Suzuki et al. was used. In their algorithm, they presented a method to count 1-components and extracting the borders of a binary picture when it is desirable to disregard all the 1-component but the outermost 1-components which is more effective in finding image contours than other border following algorithms.

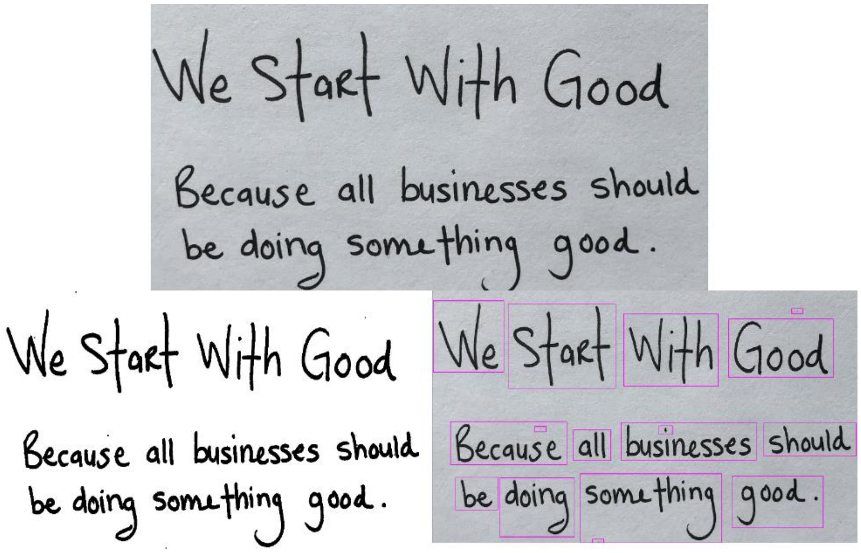


Figure 3 (a.) The image on the top is the original image (b.) image on the bottom left is the output of the image preprocessor (c.) image on the bottom right is the output of the word cropper

3.3 Deslantification model

Cropped words are normalized for further improvement in the recognition. Normalization through deslantification is used. The implementation of the deslantification algorithm follows the hypothesis of Allesandro Vinciarelli and Juergen Luettin. Depending on the handwritten style, the deslantification is successful however there are cases where the deslantification occurs on non-slanted words as seen in the figure below.

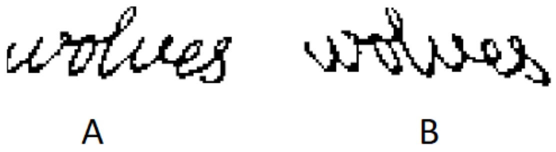


Figure 4 Image A shows the original text, Image B shows the de-slanted text

3.4 Neural Network module

The neural network consists of 7 layers in total. There are 5 CNN layers present and 2 RNN layers are used. The input image is first fed into the CNN layers where each layers are trained to extract relevant features from the image. Each of these layers contain three operations. The first convolutional operation has a 5x5 filter kernel size in the first two layers and 3x3 in the three remaining layers to the input. The next step is a Non-linear RELU function and finally, a pooling layer that summarizes image regions and outputs a downsized version of the input. Feature maps are added while the image height is downsized by 2 in each layer. The total output sequence has a size of 32x256.

The CNN accepts a gray-value image size of 128x32. Since not all images have exactly this size, it is resized either to have a width of 128 or height of 32 and then copy the source image to an empty white canvas that has the size of 128x32 as seen in the figure below.

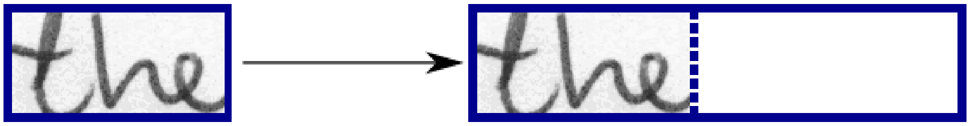


Figure 5 Image A shows the image that is originally from the IAM data set. Image B shows the same image being modified using the operations described above

The output of the CNN layers is a sequence with a length of 32. Each entry contains 256 features. Each of these features are further preprocessed by the RNN layers, however, some features already show a high correlation with certain high-level properties of the input image: there are features which have a high correlation with characters (e.g. “e”), or with duplicate characters (e.g. “tt”), or with character-properties such as loops (as contained in handwritten “l”s or “e”s).

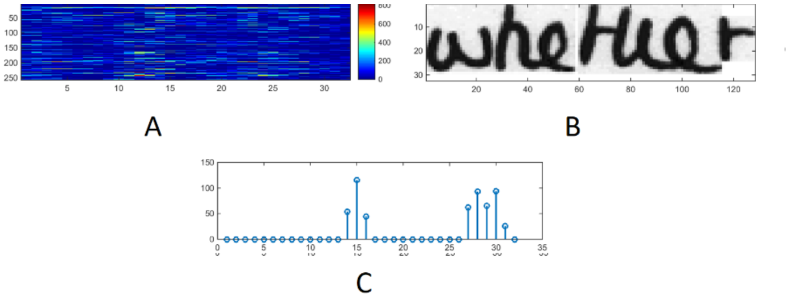


Figure 6 Image A: 256 feature per time-step are computed by the CNN layers. Image B: input image. Image C: plot of the 32nd feature, which has a high correlation with the occurrence of the character “e” in the image.

The RNN propagates relevant information through this sequence using the Long Short-Term Memory (LSTM) RNN. The RNN output sequence is mapped to a matrix of size 32x80 which will then be used by the RNN decoder module.

In the figure below, is a visualization of the RNN output matrix for an image that contains the word **“little”**. The other matrix-entries, from top to bottom, correspond to the ASCII character set which contains letters from (A-z with numeric digits and special characters). Base on the third image (Bottom). Most of the time, the characters are predicted exactly at the position they appear in the image. Only the last character “e” is not aligned. If we are to trace this using best path decoding, it will have an incorrect output, producing **leittl** as describe in Scheild et al [4].

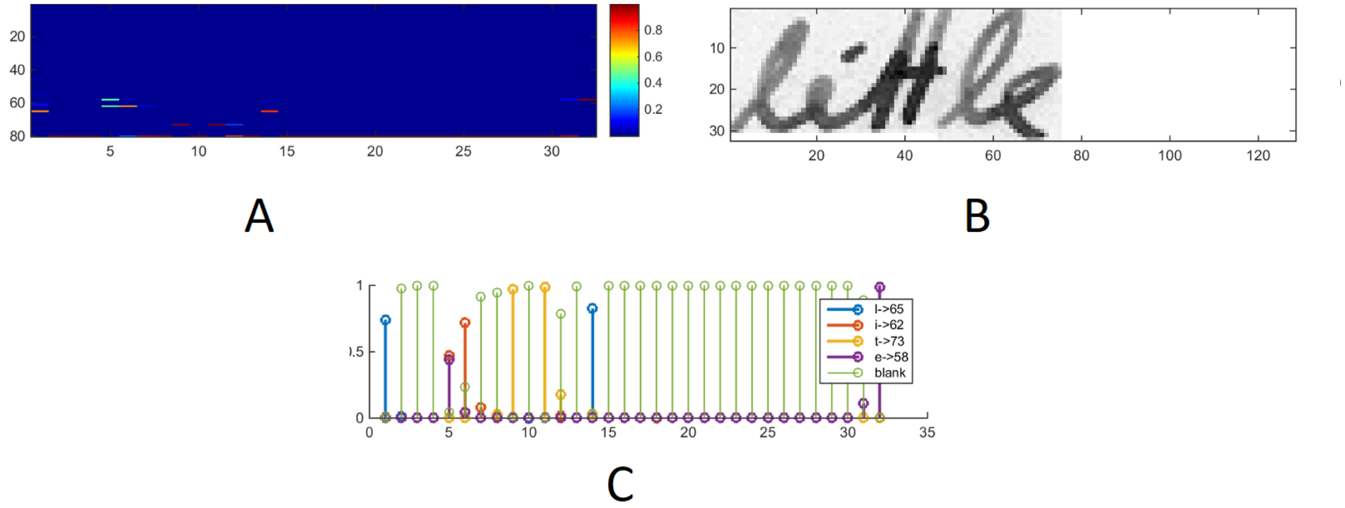
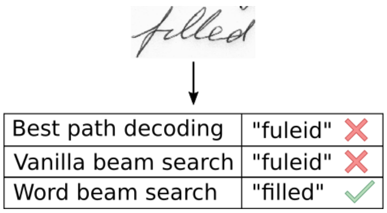


Figure 7 Image A: Output matrix of the RNN layers. Image B: is the input image. Image C: Probabilities for the characters “l”, “i”, “t”, “e”

3.5 RNN decoder module

The RNN docoder module is the Word Beam Search of the study of Harald Scheild et.al [4]. The decoder module will read the output sequence of the RNN and determine the possible word or character that is inside the image. The word beam search is used instead of best path decoding or beam search because it produces significantly better results as seen in the figure below.



Best path decoding	"fuleid" ✗
Vanilla beam search	"fuleid" ✗
Word beam search	"filled" ✓

Figure 8 Performance of the Word beam search compared with other RNN decoding algorithms

4. RESULTS AND DISCUSSION

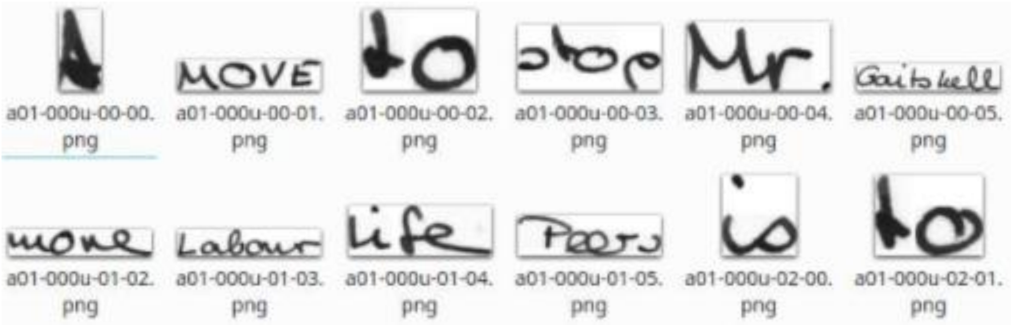
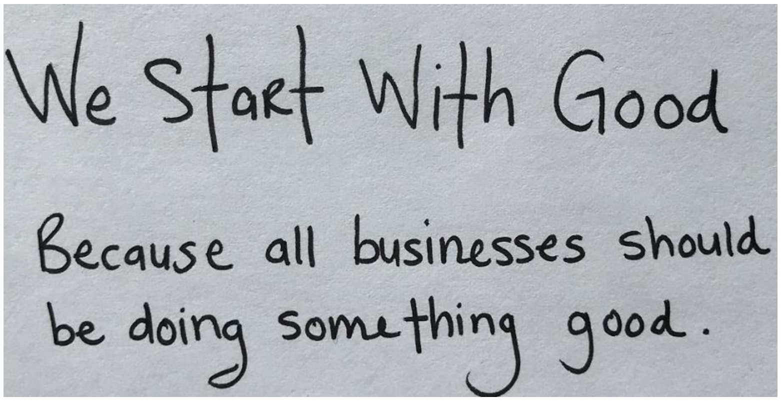


Figure 9 The first few images of the IAM dataset

The IAM dataset is used for the training and validation of the neural network. Using the regular best path decoding, it was able to achieved 68% word-accuracy with 13% character-error rate. The proposed study was able to achieve 79% word-accuracy with 12% character-error rate.

Even with a 79% word-accuracy rate, it still struggles producing correct recognition of words in a paragraph.



We Start With Good

Because all businesses should be doing something good.

Figure 10 The appearance of the test image (Sample7.jpg)

Table 1 Results of the test image

File Name	CTC	Predicted Result	Error Rate
Sample7.jpg	Best path decoding	; stret wiith good because all businesses should doing something good .	41%
Sample7.jpg	Word Beam Search	We stret With food Because all businesses should 'be doing something good .	33%

Error rate is computed by counting all the total words in the ground truth and then counting the incorrect predictions it has made in the inference. It the test, incorrect capitalization is also considered as incorrect thus increasing the error rate.

In the actual implementation, several words are jumbled up due to un-optimized word cropper and filter. Improving the word cropper and filter will improve the actual inference of the result. The same issue also appears when the word cropper module encounters cursive handwriting as of the moment, it incorrectly crops most of the text and thus a better word cropper would yield to better results.

5. CONCLUSION

Improvements in the accuracy of the recognition of handwritten text can be achieved by limiting the external factors that causes an error in a neural network. Through the use of word beam search to dramatically reduce the chance of the network picking a wrong character and deslantification of a cursive text.

It is also considered that majority of the incorrect inferences of the network is due to the un-optimized word cropper and filter. The current implementation does not remove paper lines that can normally be seen in a yellow-paper and notebook, a better filter will remove the result.

6. FUTURE WORK

As a future work could be grading of hand-written paper-based essays using the implementation of handwritten recognition system and a built-in essay grader for Filipino students. Since the work of Scheild et al [4] using a grammar for English words, it will be necessary to create another grammar to recognize both English words and Filipino words.

Another approach would be the transcription of Mangyan digitized documents. For this, it is necessary to create a training set for Mangyan text. Since it is an ancient writing system, there would be a difficulty in producing plenty of training set for deep learning, however a small training set is possible using data augmentation.

7. REFERENCES

[1] Bristow - freeform cursive handwriting recognition using a clustered neural network

[2] Vinciarelli and Luettin - A new normalization technique for cursive handwritten words

[3] Yuan, Bai, Yang, Guo and Zhao - Handwritten English Word Recognition based on Convolutional Neural Networks

[4] Schield, Fiel and Sablatnig - Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm

[5] Suzuki, S. and Abe, K., Topological Structural Analysis of Digitized Binary Images by Border Following