An Analytical Survey on RNN Model for Eloquent OCR Processing with Minimum Output Errors

Jaya Darshana Singam
Center of Excellence - Artificial Intelligence Lab
Bhubaneshwar, India
coeaijds@cet.edu.in

Priti Sahoo
Center of Excellence – Artificial Intelligence Lab
Bhubaneshwar, India
coeaips@cet.edu.in

Abstract—**This Optical Character Recognition is a technique where the scanned documents or images are converted into an editable form. The process that is involved in the conversion includes many steps, and the accuracy of recognizing right context is been a challenge till today, especially when the images are the handwritten once. To recognize the handwritten material, the aspects which add-on to the whole approach generally includes, inappropriate font sizes, distance from the image center, number of strokes, ink color, background color, pixel size, etc., due to which the built setups face issue to give an appropriate result. In order to progress the context-recognition ability, lot of innovations are still going on, on NLP (Natural Language Processing) in which, RNN (Recurrent Neural Network) is an approach that is type of a DL (Deep Learning) Technique, is used for making the use of sequential information, stores pre-processed data, and are majorly used for speech and text analysis. This paper covers the analysis of the results carried out from the RNN Model that is designed using LSTM (Long Short Term Memory) architecture with various libraries.**

Keywords—RNN Model, LSTM, OCR, NN, DL, Text Accuracy, Digit Accuracy, Text and Digit Recognition

I. INTRODUCTION

In the current era of evolving technology, the information availability, information passing, communicating has become very easy through many sources available and due the smart applications. In today's current scenario, the time has become a major factor, to accomplish many small or big tasks. One of them is making documents, notes, storing data, editing text and understanding the scanned-hand written characters. To overcome these issues, the usage of OCR comes into the play where the printed or scanned papers, images, files, etc., are converted into a machine readable form. But, due to the lack of accurate results obtained by the OCR Systems in the traditional ways that are practiced before the ML (Machine Learning), AI (Artificial Intelligence) and DL (Deep Learning) techniques came to the existence in the field of technology. The common mistakes occurred by the OCR Engines for the erroneous output are:

a) Spelling mistakes or errors

b) Wrong letter or digit recognition

c) Sequence or formatting errors

d) Missing words, digits and lines

To overcome such errors, the RNN Model is used with LSTM architecture. Before directly checking the OCR Engine, the model is built using various libraries including keras, tensorflow, opencv, numpy, Pillow i.e. also called as PIL (Python Imaging Library), Scikit learn library, etc., and then trained several times with various inputs provided in the dataset for obtaining the closely-accurate results of the hand-written text documents and images.

NN (Neural Networks) which are the foundation of DL (Deep Learning) techniques which are the frameworks which represent the combination of the set of machine learning algorithms to perform a task in a more defined way. They artificially process in such a way like a human brain to replicate the works which are done by a human intelligence. A traditional NN (Neural Network) has several layers (hidden), where each layer is confined to perform specific function. The input data sequentially verves to the hidden layers and accordingly the results or the outputs are been provided and these are limited to certain amount of input data to be provided. As per the OCR functioning, the image recognition system comes into the consideration to recognize the text of the scanned documents and pictures.

RNN is also a DL Technique similar to the NN, but the advantage of the memory adds on the special requirement to use for storing the pre-trained data. Unalike the feed-forward NN, the outcome of few layers are re-entered or fed back to the initial stage i.e. as inputs of the previous layers and this process goes on. The fed-back inputs as well as their outputs, allow to train the model again and again by allowing the analysis-making options to the Engine of the sequential data. Moreover, the RNN are not limited to a specified input length and the enclosure of the links between inner-layers in the reverse direction allows to form feedback loops, which are beneficial to learn the different concepts based on context. And this trained data, is necessary to train different words and letters in various font styles or hand-written formats to get a close result as per the input.

LSTM (Long Short Term Memory) is a framework which is a type of a RNN, that is popularly used for learning the data sequentially. These are used for the complex problem solving and prediction models. The problems occurred in Long-Term dependencies of a RNN are sorted by using LSTM techniques, as the models identifies the problem from the past learnings, but it cannot learn from those mistakes or rectify them. LSTM's learn from their past learning, and due to this feature of these models, these are currently experimented in the OCR Systems, Speech Recognition Systems, etc.

II. BACKGROUND AND RELATED WORK

A. OCR systems – Tradiditional and AI Based Difference

OCR Engine consists of various steps. The steps include several stages to build the model for detecting the right characters of the text given as input image or document.

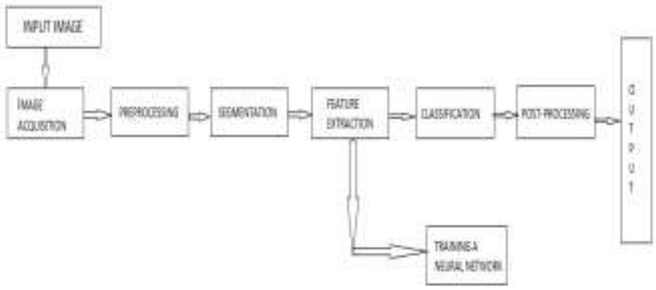


Fig 1: Steps involved in OCR processing

Fig 1. Represents the steps involved in OCR working. Each step has its own step of processing scheme.

a) Image Acquision: This section stores the input file or document or image in a acceptable format (.jpg, .jpej, .png, pdf ..etc.), captures it and replaces each pixel of the input with a black & white pixel which is also known as image segmentation.

b) Preprocessing: This step is used to clear the noise from the input, straigten the curved and non-alligned lines, and page splitting if required (dual page format).

c) Segmentation: The process that includes the sectioning and grouping of the different types of charecters or formats in the traditional OCR Engines which includes lines or paragraphs from a given input image.

d) Feature Extraction: The new technique which is also called as Document & Layout Analysis, with the help of NN and RNN, it groups the different forms of data in an input image or document including lines, paragraphs, numerical values, barcodes, images, etc. This uses the python libraries to run a DL Algorithm, whoes work is to identify the pattern and structure of the page, as well as locate them. It is an important step, for the output to be in a same format as per the input. This is done by the training done several times by giving many datasets. The traditional Feature extraction meant to classify only the lines, words or charecters.

e) Classification: The font, language, print types, all are specified in this section.And the charecters are recognized as per the training data set of the RNN, and ready to process the data to convert it into an editable format.

f) Post Processing: Last stage of the process, where the accuracy is been cross-checked before the output analysis, for the refinement of the results. Due to the proper training, it is the possibility using RNN, to achieve the accuracy to about 60-90% of the hand-written text depending on the font of the written text and about 90-100% of the printed images or documents.

B. LSTM Framework

To overcome the problem of re-gaining the past information (Long-Term Dependencies) through the identification of a single term provided as input (text in the form of image, document), is raising a problem to fill the relevant data that is actually required as per the user's query. RNN are used to retrieve the past stored-data and process it again to obtain an output for a desired input, which gives a large-unwanted big data about a term that is mentioned. To shorten this process and obtain an exact brief information on the input data, the LSTM (Long Short Term Memory) Framework is been practiced.

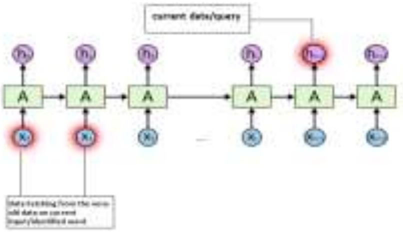


Fig 2: RNN Problem of Long-Term Dependencies

LSTM approaches a technique where the previous data is been used that is trained or fed within few outputs as well as input queries. Due to which, there is a chance of getting an accurate and a required result. Unalike RNN Models for Long-Dependencies processing, the LSTM have a simple structure with a single tanh neural layer. This layer contains a repeating module in which there are again four internal interacting layers.

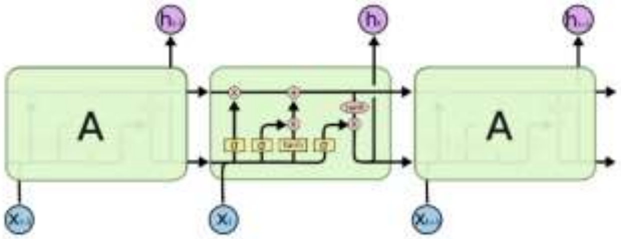


Fig 3: The LSTM Framework with the interacting layers.

The interacting layers are responsible for vector transfer, pointwise operation, concatenating (strings, inputs/words) and copying.

For an OCR to process with very less inaccuracy results for the hand-written character recognition, the training of RNN Model built for the OCR is been trained with LSTM framework, to overcome the limitations of the Long-Term dependencies and provide appropriate output.

III. PROPOSED METHOD

A. NN and RNN Model with LSTM Architecture for OCR

The basic difference between a Neural Network and a Recurrent Neural Network is the storage, analysing power of sequential data, limited input length, and learning from the previous data.

The Neural Network works in a feed-forward data processing.

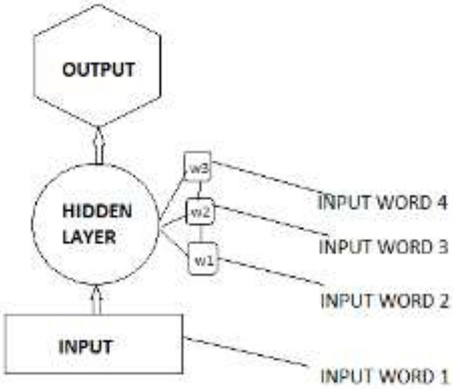


Fig 4: NN process-flow for an OCR

The RNN works with the inputs taken from the past outputs, as it allows the loops to feedback the data.

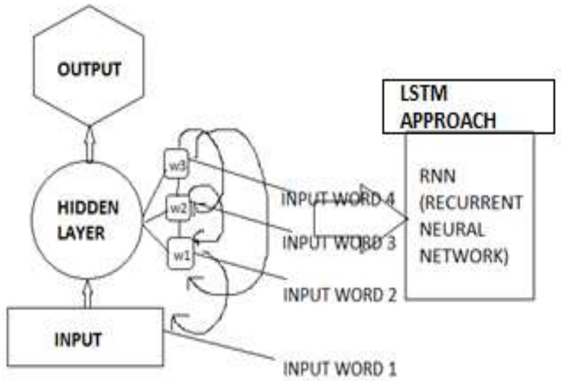


Fig 5: RNN with LSTM process-flow for an OCR

Fig 4 & 5, represents the process-flow of an OCR, which clearly shows a difference in both the network systems. According to their architectures the two above figures show the working of the OCR with two different approaches.

B. Difference between 2 Approaches

Approach 1: The data (hand-written words and text) that is given as input is carried forward through the layers of NN, and certainly all the inputs are been taken and trained and accordingly, the results are obtained. For every new input data to train the NN model, it considers it as the new set of data, trains it and gives the output. There is no space to save the trained data for better identification of text and digits.

Approach 2: The data (hand-written words and text) in this RNN model with LSTM approach, is carried out through the internal intersectional layers within a single tanh layer, and each output from a single layer is been stored and fed-back to the previous layer and with this on-going process, the output is been obtained. The process included in the intersectional layers are copying, storing, concatenating and operating. Due to this re-input of data several times, the data is trained well than that of a NN. The According to Approach 2 (Fig: 5), the mathematical representation of a RNN is:

$$h_t = f\ (h_{t-1}\ ,\ x_t) \tag{1}$$

where, h_t is the new state,
h_{t-1} is the previous state, and
x_t is the current input.

So, according to the Equation (1), the handwritten text is well-recognized irrespective of whatever font, ink colour, size or sequence it might be. The training of RNN is done by providing various datasets with different handwritings and fonts, shapes and sizes. The training can be done with the MNIST (Modified National Institute of Standards and Technology) dataset.

C. Text Classification System of Printed & Hand-written text

In the work-flow of an OCR, the step where the classification is done (according to Fig: 1), the main step of identifying the pattern of the text is done here, weather it Is hand-written one or the printed one.

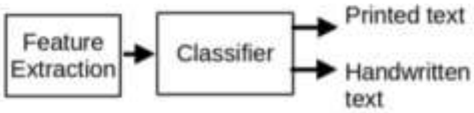


Fig 6: Classification of Printed and Handwritten Text

At the time of testing the trained data of both the printed and hand-written images and documents, there are few difficulties faced, which lead to the further improvement of the code to detect the characters of the hand-written once in a more accurate way. The basic differences that are observed while testing the data of the two text forms are tabulated the below Table: 1.

Table 1: Observed Differences between Printed Text & Observed Text while testing an OCR Engine

Sl. No	Printed Text	Hand-Written Text
1	Intensity of the Pixels (Black & White) of all the characters of the	The intensity of pixels vary from letter to letter or words and the

	provided input are similar.	text as per the written form.
2	The size, shape and alignment (straightness) of all the characters are equivalent	The size, shape and alignment of all the characters are not equivalent.
3	Due to a constant font style, the shapes are all similar to a particular once.	Due to the hand-written font, the shapes of each character vary.

From Table 1, due to the different issues occurring in detecting the characters of the hand-written text, the RNN Model with the LSTM Approach is been performed to train the model with various input images of different hand-written pages, images and documents. For recognizing the text, to identifying the no of characters provided to specifying the accuracy and to convert the image and the document format to an editable format, the OCR Engine is been made to accomplish the whole process.

IV. Results and Discussions

According to the testing done, the results from an OCR Engine are been practically observed, examined and further been analyzed for further future application areas.

A. *Test Results of an Input Image*

The image is provided to the LSTM-RNN Model, to first store it and process the pre-processing phase of a hand-written text.

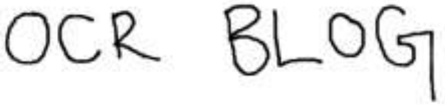


Fig 7: Input Image

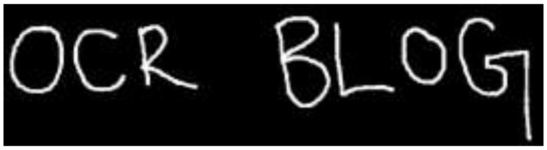


Fig 8: Tilt detection of the Input image

B. *Steps included in Identifing the Charecters of the Input*

To identify the input image after pre-processing stage, the dilation step, line segmentation and the word segmentation steps are performed to know how well the characters are been identified. As per the fed-image, first the line segmentation is been done.

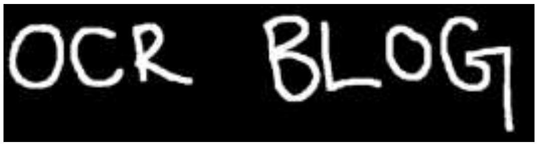


Fig 9: Dilated output of the Input image

Fig 9, represents the result of the dilated processing of the image given to process for an OCR. Dilation process adds more pixels to the boundaries of the characters for proper structuring of the image. And it lets to adjusts the pixels and suit the required quantity for accurate detection process.

Later, the Line Segmenting process is done to the input image.

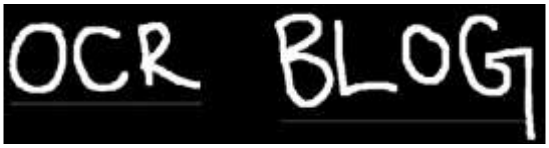


Fig 10: Line segmentation of the Input image

Fig 10, is the output of the input image (Fig 7) from a code to identify the allignment of the words and lines individually, so that there is no overlapping of the letters or digits. The thin lines which are obtained (Fig 10), describe that the words are detected properly as they are seperated with two different lines for the two given words.

So, from this step it is clear that there would be no chance of overlap of words in the output.

The next step is, the word segmentation and Identification.

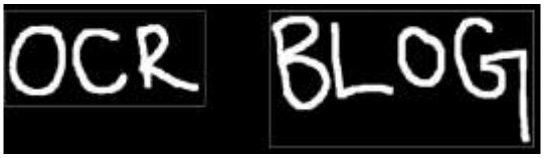


Fig 11: Word Segmentation of the Input Image

Fig 11, represents that the words are correctly identified as they are grouped seperately by the set of codes in the RNN – LSTM approached algorithm to perform this task. This step proves that the words will not overlap in the output as they are correctly identified.

The next step, is the character segmentation from the words detected.

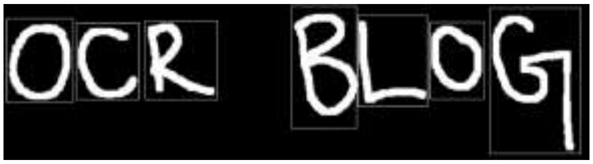
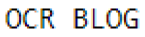


Fig 12: Character Segmentation of the Input Image

Fig 12, represents the correctness of character recognising if each word. This specifies that there is almost an accurate detection of the hand-written text with the approach of this RNN – LSTM Model. The training of this RNN Model of OCR performs all these processes for identifying the right words for an input image and document.

The final step includes the conversion of the input-text image into an editable format with this OCR Engine.



OCR BLOG

Fig 13 (a): Outpur of OCR in an Editable format

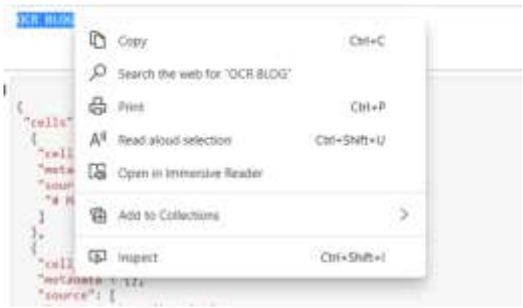


Fig 13 (b): Outpur of OCR in an Editable format

Fig 13(a) & 13(b) represents how the OCR Engine is providing an output of a hand-written text from an image to an editable form format, so that it can be easily copy-pasted in whatever document type required for further changes, storing purposes.

C. Challenges faced during building the OCR Engine

The main challenges that occurred during the whole process are:

a) From the output of the first training model made by the ANN, did nor recognize the hand-written charecters well due to the only feed-farward loops.

b) The CNN (Convolutional Neural Network) model testing, gave much better result than the ANN, but the detecting of the charecters are not done accurately.

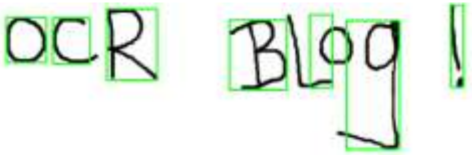
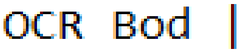


Fig 14(a): OCR test results of the CNN built model



OCR Bod |

Fig 14(b): Output result of the CNN Model

The OCR Engine detects the wrong output in case of the CNN network.

c) RNN model training before applying LSTM approach was done correctly, but the accuracy % was certainly less compared to the results that are obtained after approaching the LSTM Architecture in the RNN model for the OCR Engine.

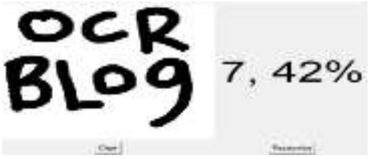


Fig 15: one of the RNN Model training phase results

V. CONCLUSION

According to the training and testing the RNN Model with the LSTM Architecture, the results of the hand-written text images
and documents are well-recognized. One of the tested results is discussed in the Results and Discussion Section.

Due to this approach, it overcomes the issues faced in the other OCR Models that are built previously with the ANN (Artificial Neural Network), KNN (K- nearest neighbor algorithm), and CNN (Convolutional Neural Netwrk) Algorithms. The accuracy of all these three network models i.e. ANN, KNN and CNN are comparatively less due to the limitations of the models. Even the RNN has few drawbacks, but with the architecture of LSTM of RNN to the OCR Engine training and building the model is done, the steps involved to provide the results of the input image and document are precisely done with a proper process. This whole procedure involved, gave an appropriate output from an OCR input files. From all the data that is tested and in use now for the conversion, the results are up to 70-95 % accurate.

VI. SCOPE FOR FUTURE

According to the analysis observed through-out the whole process, these RNN Models with LSTM architectures in OCR Models can be used in various other applications like:

a) Various language OCR Engines can be built

b) Data-sets can be made for different languages and be trained for detecting the charecters of specific languages.

c) This method can also be used for applications for language-translation systems.

REFERENCES

[1] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty and Jean-Philippe Moreux, "ICDAR2017 Competition on Post-OCR Text Correction", *Document Analysis and Recognition (ICDAR) 2017 14th IAPR International Conference on*, vol. 1, pp. 1423-1428, 2017

[2] Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, Muriel Visani and Jean-Philippe Moreux, "Impact of OCR errors on the use of digital libraries: towards a better access to information", *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*, pp. 249-252, 2017.

[3] Haithem Afli, Loïc Barrault and Holger Schwenk, "OCR Error Correction Using Statistical Machine Translation", *Int. J. Comput. Linguistics Appl.*, vol. 7, no. 1, pp. 175-191, 2016.

[4] John Evershed and Kent Fitch, "Correcting noisy OCR: Context beats confusion", *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pp. 45-51, 2014.

[5] Okan Kolak, William Byrne and Philip Resnik, "A generative probabilistic OCR model for NLP applications", *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. 55-62, 2003.

[6] Rafael Llobet, Jose-Ramon Cerdan-Navarro, Juan-Carlos Perez-Cortes and Joaquim Arlandis, "OCR post-processing using weighted finite-state transducers", *2010 International Conference on Pattern Recognition*, pp. 2021-2024, 2010.

[7] Michael Piotrowski, "Natural language processing for historical texts", *Synthesis lectures on human language technologies*, vol. 5, no. 2, pp. 1-157, 2012.

[8] Paul Hagon, "Trove crowdsourcing behaviour", *Australian Library & Information Association Information Online 2013 Proceedings*, 2013.