LINGUISTIC ANALYSIS ON CURSIVE CHARACTERS

CHIAI **AL-A**TROSHI
Dept. of Educational Counselling and Psychology, College of Basic Education, University of Duhok, Kurdistan Region-Iraq

(Received: April 3, 2022; Accepted for Publication: October 12, 2022)

ABSTRACT

Document Analysis has major importance in Information Retrieval Systems. Dredged with vaults of paper and material documents, to protect very important information and the summaries, without losing their meaning and importance, each document need to be properly curated and processed. Ancient written documents possess many types of cursive language character sets, which are very tedious to discriminate the characters and subsequently the right meaning. To overcome the difficulties of reading the cursive language characters and prevent misunderstanding the meaning and the importance of documents, an improvised CNN [6] model to work on OCR and Tesseract API has been proposed in this work. The documents are scanned, curated and preprocessed in the forms of images. CNN are the best algorithms, hitherto in the existing AI and Deep Learning arena. CNN with OCR API could contribute to the development of efficient strategies of character recognition even with complex cursive styles. A method which is adaptable to the classification and segmentation of the text images with cursive styles is proposed I this article. Tesseract is the popular and effective OCR library with rich API that can enrich the CNN-OCR model.

***KEYWORDS*:** Convolutional Neural Network, Deep Learning, Machine Learning, Linguistic analysis, character recognition.

INTRODUCTION

Out of the languages English characters are well recognized which contain uniform cursive syllables and notes, where OCR is best applied to work on them. Even characters such as consonants, vowels and compound characters are well identified with tools built with Tesseract. The question of recognizing non-english characters worldwide with cursive characters, such as characters of Arabic, Urdu, Telugu, Tamil and other south Indian languages drives many constraints [1,2]. An effective OCR engine is believed to eliminate all the constraints in character sets and input formats as well. The challenge is further on reading the real world documents which use very common character sets, which needs a natural language parser [1,2]. Digitization of old documents of languages with cursive characters and the graphical formats of the scanned sources are the important considerations during the design of efficient recognizer.

Scanned image sources are obviously noisy, pepper noise dominates in the images of cursive language text, in order to escape from misclassification, the noise has to be eliminated or corrected [7,8,9]. Popular error margin methods are employed in the noise mitigation of very sensitive noise which is extra pixels in images of cursive language text, whether in typed fonts or handwriting styles [3]. Cursive language contains connected characters and words, words are connected with sufficient spaces or special characters and further into sentences. Extra noise is noted in random locations of the cursive language text images, which is generally spread throughout the text [12, 14]. Extra noise influences the connectivity of words and sentences, violating the chain code and sometimes dimensions of characters [3]. A "character image variation" is observed in general among the cursive language character images; this variation is symmetric thus providing the opportunity of equally divided segments.

PROBLEM STATEMENT

Given a set of images that contains cursive language text, the objective is to detect the output variables from the sets of input images as high as accurately possible. A method called "recognize and search method" is employed for images with cursive language characters, of any language. A background corpus or knowledge is

chiai@uod.ac

supplied to quickly match the patterns and identify the characters by an OCR algorithm. A suitable OCR engine is proposed that can apply the techniques of sampling and identify the character patterns and further classify and recognize the groups of characters. A multi-language tool for OCR algorithm is a challenging requirement to parse the cursive characters which draw various inferences in understanding the scanned image content of cursive language text. As a typical document processing in information retrieval and text summarization, the documents with images of cursive language text is classified and indexed based on the keywords and key-characters of the language.

METHODOLOGY

The fundamental of the experiment starts with recognition of cursive language characters as individual characters. Isolated characters from the images are recognized through segmentation. A consequent level wise segmentation allows immensely parse the text.

In the process of recognition and segmentation of images that belong to cursive language text, the 'begin' and 'end' of the partial or fully cursive characters is observed by tracing the full image which pixel columns of the isolated character set. While, the images of characters are segmented column-wise and the scanning for characters proceeds horizontally through sentinels of pixels. The presence of first black pixel in the column is observed as 'begin' and on subsequent scan the presence of white pixel as the 'end'. Thus, an image of a character in the images of cursive language text is identified as the image in-between the two white columns. A repository or corpus is maintained for several such small images of individual characters that represent the individual images of cursive language text. Each isolated character is said to have four void sides. The 'chain of code' is observed for each sample character with the start of a white pixel, and similar pattern as chain of code is observed all other characters of the isolated inputs of text images. Hence, the classification of the character images is eased out and the chain of code is calculated and matched with the samples of characters. This method is observed as reliable in identifying the characters. The text images with common chain of code which is identified as matched with each column reading column by column with every character in the sampled images is determined as a class, however with a considerable error margin.

For mitigating with the text of different sizes and styles in the image text, a threshold of error margin is setup to negotiate difference of properties, also to ease out the process of comparison of characters. A classifier for the characters with a considerate error margin and computed chain of code is built. The chain of code is subsumed in the matching method, if any character though looks alike is not matched by a classifier. In some cases the error count is mismatched for similar character images, which is identified to be as images of different widths. Therefore, the sampled images of cursive language text are scaled and normalized to similar width and height, with or without noise empty columns or rows are appended according to the dimensional requirements of the sampled image. Further the character in the sampled image shall be identified by classifier with a least error count and error margin.

OPTICAL CHARACTER RECOGNITION

An image processing function $\boldsymbol{X}$ is employed on the set of input images $\boldsymbol{I} = \{\boldsymbol{i}_0, \boldsymbol{i}_1, \boldsymbol{i}_2,\ldots,\boldsymbol{i}_n\}$,where $\boldsymbol{X}$: $\boldsymbol{I}\rightarrow\boldsymbol{V}$, $\boldsymbol{V} = \{V_0, V_1, V_2, \ldots V_n\}$, each of the $\boldsymbol{V}$ is the set of output variables, $\boldsymbol{i}$ is an image in $\boldsymbol{I}$ containing lines of cursive language text and $\boldsymbol{V_i}$ is the information extracted from the image $\boldsymbol{i}$. The accuracy of the algorithm depends on the three steps viz., detection, classification and recognition.

In a set of lines scanned from the image, where each line comprises of symbol images, the process of OCR should output the list of lines, where each line is a list of symbols. If $\boldsymbol{S}'$ is the output symbol, then $\boldsymbol{S}' = \{\boldsymbol{L}'_0, \boldsymbol{L}'_1, \boldsymbol{L}'_2\ldots \boldsymbol{L}'_n\}$, and each i^{th} line in the output $\boldsymbol{L}'_i = \{\boldsymbol{s}'_0, \boldsymbol{s}'_1, \boldsymbol{s}'_2\ldots \boldsymbol{s}'_n\}$. Difference between the symbol and symbol image shall be noted in output and input to the OCR engine. Thus, the function of OCR operates on input $\boldsymbol{S}$ and output $\boldsymbol{S}'$. The general composition of the cursive text characters in the corpus of input is described in the following figure Fig.1.

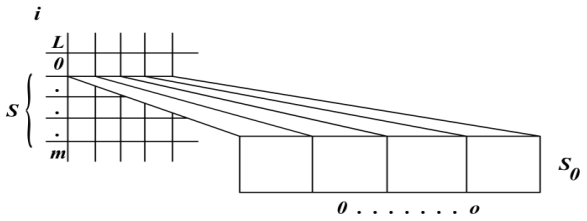

Fig.(1):- Composition of symbols and lines in images.

Noise Filters

Many filters are available and applied to images to filter noise, in order to make the OCR algorithm read characters efficiently. Most commonly, a linear filter which uses a convolutional approach is applied to eliminate the fundamental visible noise from the images.

Tesseract

Around 1985 and 2005, Hewlett-Packard (HP) has developed Tesseract, as a open source OCR engine. Latest version of this open source engine is 4.0.0, employs a neural network based for training, testing recognition of specific characters. [3,8,9,10]

Images containing lines of cursive language text are gray scaled and binarized, further by using luminance method and decolorize methods, to enable the correct parts of the crucial curves of characters. Tesseract OCR API [3] can read such characters, which are not readable from human perceptions, while they look like complex scripts. While, Tesseract failed to recognize 10-20% of characters.

Convolution Neural Network:

Character recognition problems are well solved with neural networks and artificial intelligence. One of the advanced implementations of the learning networks is convolution neural network. Convolution reiterates reading the input data for several times in order that the network gets trained about the classified input data and then tested on the real inputs to employ the CNN [6, 11, 12] for classification. In this work in order to classify the text, the CNN reads the character image files and classifies into categories [7].

As convolution is the process of feature matching, therefore convolutional neural network is the natural model that stimulates human reading. One dimensional convolution is applied in cursive text character recognition.

PREPROCESSING

Image Binarization is the first step of preprocessing. The complete text is represented in black denoted as foreground and the background is white in binarization. A global binarization method using Max Variance method shall be used in preprocessing, to work on different kinds of samples. However, binarization of the input is not the final activity of the text recognition process for cursive text images.

Convolutional Neural Network (CNN) brought a dynamic change in neural networks, which has umpteen applications of character recognition with lowered error rates. Three essential layers compose a typical CNN, the convolutional layer, pooling layer and the fully-connected layer, which are stacked together to form a complete CNN architecture.

An activation map is produced by the convolutional layer, for each and every feature that is sampled for the experimentation, using convolving methods or sliding methods on a kernel or a filter across every pixel location of the sample, represented as pixel matrix of the input image. The objective of this task is to locate the features in the input image. Further a scalar product is derived on computations between weights and the regions of the input through the connection of neurons.

Computation of down-sampling is done along the spatial dimensions in the pooling layer, further reducing the requirements for the activation. On iterative application of pooling and convolution layer, a high-level reasoning of the input is discriminated by the fully-connected layer, where neurons connect to all the activations in this layer.

A 2-D matrix of neurons are exploited to generate based on convolution, which are referred as features. A block of input feature maps are input to the convolution neural network and the output is a block of feature maps that describe the categories of the cursive text images.

Features in the image are translated into perceptible and recognizable visual images representing the internal features, learned

chiai@uod.ac

features. Feature detectors are the filters that are applied on the image with the activation maps that are generated in the pooling layer. The following figure Fig.2 describes the discrimination of features in the given input images using a k-feature map.

To envision the features in the input:iterate through all layers

1. weights and bias values are extracted in convolution layer at each iteration

2. for all the filters between 0 and 1, the weights are normalized

3. all the filters are mapped into the convolution layer with respect to the image channels.

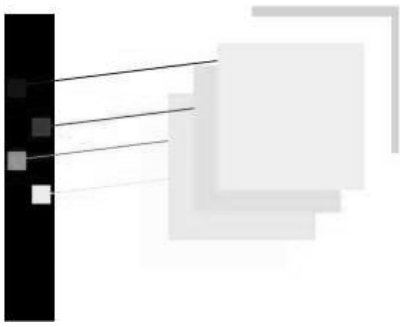

Fig.*(2)*: -k-feature map of a convolution layer.

The characters in the sample images are ambiguous, which contain similar strokes. Even the similar strokes evolve into major differences and become a challenge for discriminating the characters. Cursive strokes in the characters can be distinguished very differently as well, but most of them become misclassified. Therefore, selection, scaling, application of filters and classification are the preprocessing tasks, which are handled meticulously based on the variations in the input data. The following tables shows the comparative error rates in misclassification on the training and testing data among over 6000 images of cursive language texts prepared.

Table*(1)*: -The comparative error rates in misclassification during training and testing

	Without Processing	After Binarization	After Sharpening	After Applying Linear Filters
Tesseract OCR API	97.89	97.88	97.26	97.26
CNN	94.49	94.38	94.33	94.28

EXPERIMENTATION

A. Data Collection

The benchmark data set for the experimentation is collected from MNIST. The MNIST database of handwritten fixed-size images containing training sets of 60,000 and test set of 10,000 samples developed by Corinna Cotes et. Al. [5].

Synthetic data is generated on the cursive characters from the old documents scanned and curated. Image data sets are developed scanning 500 documents of cursive character sets and curated to the sentence level.

B. Methodology

Preprocessing is curating the collected data into samples, with equal dimensions. Cropping, chunking and truncating spaces in the images do collect the images into the repository. Categorization of images, eliminating noise using the noise filters is the secondary step in the methodology while preprocessing the inputs. The data sets size being a demand for the full implementation of CNN, large data sets are built during the curation in the preprocessing of the methodology. Comparative to the MNIST data sets around 7,000 images are developed by scaling, curating the images into sentence wise images, character wise images. All the images of the cursive character text are broken into several character images with single cursive character. A 28 x 28 sized cursive character images in grayscale after eliminating the noise are considered as the prepared input. A methodology of applying CNN for classification of cursive text images is represented in the figure Fig.3.

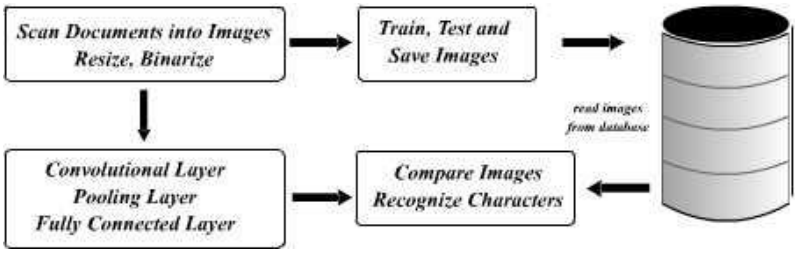

Fig.(3): -Methodology of CNN implementation

The preprocessed images are input into the convolutional layer, max-pooling layer and fully-connected layer respectively. The convolutional layers and max-pooling layers each of four are used for the enhancement of the images in the mentioned model. A 2 x 2 maximization is performed in the two hidden layers of the model and ReLU is used after each hidden layer. The following figure explains the process of enhancement in the layers on the images. During the process of convolution, the stride operations with maxpool filter is representing in the following image Fig.4, the evolution of a feature from the cursive text images with iterative stride operations.

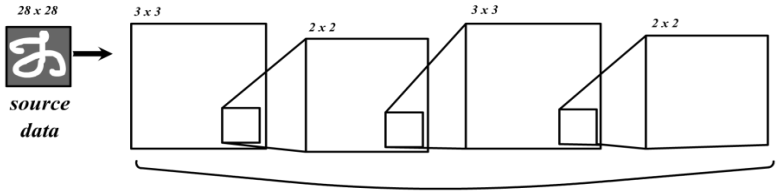

Fig.(4) :- Convolutional and Pooling layers

C. Training and Testing

In above mentioned model almost 80% of data is used for training and 20% for testing. Each character after scaling and resizing has been drawn into a partition of 200 images whereas 50 images were used for testing the dataset. A random partition attempt tests the accuracy for each attempt. The maximum of average-accuracy of the attempts is considered as the accuracy of the model.

RESULTS AND DISCUSSIONS

A classification table is drawn illustrating the predicted number of successes compared with the number of successes actually observed and the predicted number of failures compared with the number of failures actually observed. The possible outcomes in the classification table are (TP) true positives, (TN) true negatives, (FP) false positives, (FN) false negatives. The confusion matrix is extremely useful to measure the precision, recall, specificity, accuracy of the classification model that is applied in the CNN. Most importantly the Area Under Curve (AUC) in the Repeater Operating Characteristic (ROC) Curve,

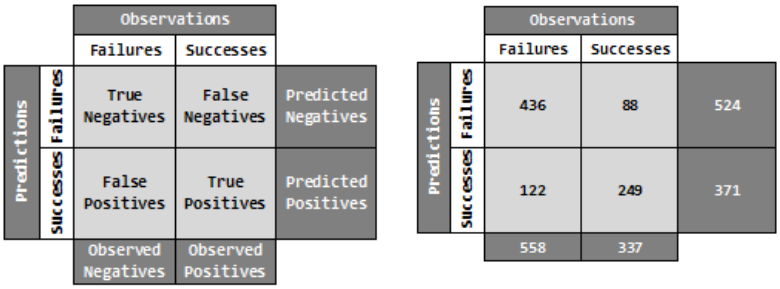

Fig.(5) :- Classification Table / Confusion Matrix for MNIST database of handwritten images

Precision = (TP) / (TP+FP); Recall = (TP) / (TP + FN) and

F-Measure = (2 * Recall * Precision) / (Recall + Precision)

The relative statistics for the above values is as follows:

True Positive Rate (Sensitivity)	**0.930051813**
True Negative Rate (Specificity)	**0.928571429**
Accuracy	**0.929078014**
False Positive Rate	**0.071428571**
Positive Predictive Value	**0.871359223**
Negative Predictive Value	**0.962290503**
Precision	**0.871359223**
Recall	**0.930051813**
F-Measure	**0.871359223**

The following data is computed on the samples of 100 images collected from the MNIST database, representing the number of elements recognized correctly and incorrectly.

Table*(2)*:- Worked table for computing the ROC on recognized characters as correct or incorrect.

	Recognized		***Cumulative***				
No. of Random Samples of Images	***Correct***	***Incorrect***	***Correct***	***Incorrect***	***FPR***	***TPR***	***AUC***
			0	*0*	*1*	*1*	*0.066038*
50	49	0	49	0	0.9339623	1	*0.132075*
100	98	0	147	0	0.8018868	1	*0.198113*
150	147	4	294	4	0.6037736	0.9896373	*0.264081*
200	198	7	492	11	0.3369272	0.9715026	*0.257933*
250	197	16	689	27	0.0714286	0.9300518	*0.055151*
300	44	96	733	123	0.0121294	0.6813472	*0.006428*
350	7	103	740	226	0.0026954	0.4145078	*0.001117*
400	2	66	742	292	0	0.2435233	*0*
450	0	56	742	348	0	0.0984456	*0*
500	0	38	742	386	0	0	*0*
	742	386					

The peak points in the ROC curve represents the better fit of the proposed CNN framework. The AUC can be inferred that, closer to 1 is maximum value i.e., better fit. The AUC value which is closed to 0.5 states that the proposed model has ability of discriminating between correct and incorrect recognition of images as characters of cursive language, which is a due chance.

While evaluating the performance of CNN on cursive handwritten characters from MNIST database, an approach with incremental characteristics is employed on classification. First a five character classes of the dataset is chosen in the experiment to compute the accuracy level in the activity of recognition. Then subsequently the number of character classes is gradually increased, consequently at

one point the level of accuracy will decrease. The successive results of CNN began to come down with a significant difference, about number of character classes reaches to a higher range. As the number of character classes increase the accuracy and success of CNN decreases.

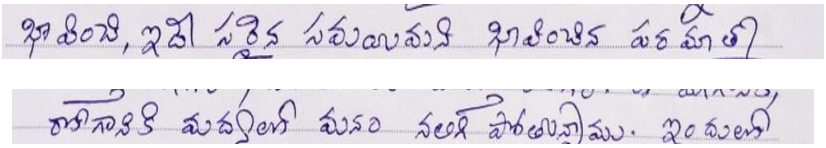

***Fig.*(*6*)*:*-** Examples of Handwritten Images that are considered as input to the experiment.

CONCLUSION

The differences between traditional Tesseract OCR Engine and the CNN for image processing have been discussed. Preprocessing the source images before actual implementation of character recognition is a mandatory, which will improve the efficiency of detection and recognition. Modern OCR [13] implementations certainly have advanced features of recognizing the characters of multiple languages. As far as OCR engine developed in the laboratory is concerned that they can recognize a particular language which is defined in the corpus as background knowledge, the attempt in this paper have been made that any language with cursive characters can be recognized with the better corpus provided with all the possibilities of representing the parts of the characters. Images of various parts cursive characters in the languages have been studied and presented the basic approach of identification of cursive characters and the differences between a traditional Tesseract OCR engine and CNN for image processing.

REFERENCES

J. Mariyathas, V. Shanmuganathan and B. Kuhaneswaran, "Sinhala Handwritten Character Recognition using Convolutional Neural Network," 2020 5th International Conference on Information Technology Research (ICITR), 2020, pp. 1-6, doi: 10.1109/ICITR51448.2020.9310914.

Benaddy, Mohamed, Othmane El Meslouhi, Youssef Es-saady, and Mustapha Kardouchi. "Handwritten Tifinagh characters recognition using deep convolutional neural networks." Sensing and Imaging 20, no. 1 (2019): 1-17.

Wei, Tan, UsmanUllah Sheikh and Ab Al-HadiAbRahman. "Improved optical character recognition with deep neural network." 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA) (2018): 245-249.

Anil, R., Manjusha, K., Kumar, S.S., Soman, K.P. (2015). Convolutional Neural Networks for the Recognition of Malayalam Characters. In: Satapathy, S., Biswal, B., Udgata, S., Mandal, J. (eds) Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014. Advances in Intelligent Systems and Computing, vol 328. Springer, Cham. https://doi.org/10.1007/978-3-319-12012-6_54

Cortez, Corinna; Burges, Christopher C.J.; LeCun, Yann, "The MNIST Handwritten Digit Database". YannLeCun's Website yann.lecun.com. Retrieved 30 April 2020.

Sarker, Goutam, and Swagata Ghosh. "A convolution neural network for optical character recognition and subsequent machine translation." Int. Journal of Computer Application 182, no. 30 (2018): 23-27.

Ko, Daegun, Suhan Song, Kimin Kang, Seongwook Han, and Juneho Yi. "Optical Character Recognition performance comparison of Convolution Neural Network and Tesseract." IEICE Proceedings Series 61, no. P1-11 (2016).

Zhao, Haifeng, Yong Hu, and Jinxia Zhang. "Character recognition via a compact convolutional neural network." In 2017 International conference on digital image

computing: techniques and applications (DICTA), pp. 1-6. IEEE, 2017.

Sarker, Goutam. "A survey on convolution neural networks." In 2020 IEEE REGION 10 CONFERENCE (TENCON), pp. 923-928. IEEE, 2020.

Kim, Jinho. "Character Level and Word Level English License Plate Recognition Using Deep-learning Neural Networks." Journal of Korea Society of Digital Industry and Information Management 16, no. 4 (2020): 19-28.

Mai, VinhDu, Duoqian Miao, and Ruizhi Wang. "Vietnam license plate recognition system based on edge detection and neural networks." Journal of Information and Computing Science 8, no. 1 (2013): 27-40.

Raj, Aman, Devanshu Dubey, Abhishek Mishra, Nikhil Chopda, Nishant M. Borkar, and Vipul S. Lande. "Convolution neural network based automatic license plate recognition system." International Journal of Computer Sciences and Engineering 7, no. 4 (2019): 199-205.

Rawls, Stephen, Huaigu Cao, Senthil Kumar, and Prem Natarajan. "Combining convolutional neural networks and lstms for segmentation-free ocr." In 2017 14th IAPR international conference on document analysis and recognition (ICDAR), vol. 1, pp. 155-160. IEEE, 2017.

Sharma, Arnab Sen, Maruf Ahmed Mridul, Marium-E. Jannat, and Md Saiful Islam. "A Deep CNN Model for Student Learning Pedagogy Detection Data Collection Using OCR." In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1-6. IEEE, 2018.