A Convolution Neural Network for Optical Character Recognition and Subsequent Machine Translation

Goutam Sarker, PhD
SMIEEE, Associate Professor
Department of Computer Science and Engineering
National Institute of Technology, Durgapur

Swagata Ghosh
M. Tech. Student
Department of Computer Science and Engineering
National Institute of Technology, Durgapur

ABSTRACT
Optical character recognition has been a longstanding challenging research topic in the broad area of machine learning and pattern recognition. In the present paper, we investigate the problem of textual image recognition and translation, which is among the most daunting challenges in image-based sequence recognition. A convolutional neural network (CNN) architecture, which integrates optical character recognition and natural language translation into a unified framework, is proposed. The accuracy of both OCR output and subsequent translation is moderate and satisfactory. The proposed system for OCR and subsequent translation is an effective, efficient and most promising one.

General Terms
Pattern Recognition; Convolution Neural Network (CNN); Optical Character Recognition (OCR); Machine Transcription

Keywords
Recurrent CNN; LSTM; CNN based LSTM; Performance Evaluation; Accuracy

1. INTRODUCTION
Optical Character Recognition (OCR) refers to the process by which an electronic device accepts as input the image of some text which may be either an uploaded photograph or the scanned image of some printed text and converts it into plain text. OCR finds wide application across all fields of life like automatically detecting the license plate number of cars, recognizing the names of shops, restaurants, malls or service centres from their neon hoardings, recognizing the text from a scanned page of any book etc. The proposed system of CNN based OCR and subsequent Machine Translation will be especially useful for the foreigners who are unable to understand the local languages and find difficulty reading the signboards across the streets; in such cases the application may be used to recognize the textual characters from the input photographs of the signboards and translate them to the user's native language for better comprehension. [10,11,12,3,14,15,16,17,18]

Convolutional Neural Network (CNN) is a revolutionary and dramatic change in the field of neural networks, and it has helped computer scientists achieve astounding improvement in lowering the error rates in image classification and other applications in the field of computer vision. The Long Short Term Memory (LSTM) has the advantage of remembering the outputs of the past inputs and can therefore predict the last item in a sequence by referring to the past outputs as well as the current inputs. The LSTM reduces the disadvantages of Traditional Recurrent Neural Network (RNN) since it was not possible to store a lot of previous outputs in the RNN as it could only refer to the immediate past information along with the present data.

Machine translation is a part of machine transcription which deals with text processing and translation requires sequence modelling as humans naturally do not translate word for word, but they translate groups of words or phrases together and for this, sequence modelling is needed. [5,6,7,8,9]

The present application is implemented using CNN for training the OCR and the translator is implemented using the LSTM model for sequence modelling.

2. RELATED WORKS
In the year 2017 an architecture called as the Gated Recurrent Convolutional Neural Networks for implementing the OCR was proposed by Jianfeng Wang of Beijing University of Posts and Telecommunications, Beijing, China and Xiaolin Hu of Tsinghua National Laboratory for Information Science and Technology, Beijing, China. [1]

Another end-to-end trainable neural network for image-based sequence recognition that was applied for recognizing the musical score sheet was proposed in the year 2015 by Baoguang Shi, Xiang Bai and Cong Yao of Huazhong University of Science and Technology, Wuhan, China. [2]

3. THEORY OF OPERATION
The present section presents the details of the neural network architectures that are used to implement the application of Translator through OCR.

3.1 CNN
Ordinarily a CNN is comprised of three different layers namely convolutional layer, pooling layer and fully-connected layer. When these layers are stacked together, a complete CNN architecture is formed. [19,20,21,22,23,24]

The convolution layer is used to produce the activation map for all the features by convolving or sliding a kernel or filter across every location of the pixel matrix of the input image. This layer is essential to find out which feature exists especially in which part of the image. [24,25,26,27,28]

The convolutional layer of the CNN determines the output of neurons connected to local regions of input through the calculation of the scalar product between their weights and the regions connected to the input volume. The rectified linear unit (commonly shortened to ReLU) is used to apply an 'elementwise' activation function. The ReLU function is more advantageous than the sigmoid activation since it decreases the possibility of vanishing gradient where the weight updating process slows down significantly.

The pooling layer of the CNN simply performs down sampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation. [26]

Finally, after several convolution and pooling layers, the fully-connected layer takes care of the high-level reasoning of the neural network by performing the same duties as in standard Artificial Neural Network (ANN), and the neurons in this layer have connections to all the activations.

3.2 Recurrent CNN (RCNN)

A recurrent convolutional neural network is a convolutional neural network that intends to capture time or sequence dependent behavior – such as natural language, stock prices, electricity demand and so on. This can be achieved by feeding back the output of a convolutional neural network layer at time t to the input of the same network layer at time t + 1.

Recurrent convolutional neural networks when "unrolled" programmatically during training and validation, gives us the model of the network at each time step, which shows that the output of every layer at a previous time t-1 is fed to the same layer at the current time step t. Thus the recurrent CNN is a feedback type neural network.

The main problem with recurrent convolutional neural network, is that it faces its limitations while attempting to model the dependencies between words or sequence values separated by a large number of other words, i.e. between words which have a significant distance between them, thus leading to the vanishing gradient problem. This is because small gradients or weights are multiplied several times over through the multiple time steps, and the gradient shrinks asymptotically to zero. This means the weights of all the previous layers won't be changed significantly and therefore the network will fail to learn the long-term dependencies. This is where the utilization of the LSTM model comes in, as the LSTM gates combined with the layers of recurrent CNN can help solve this problem of vanishing gradient to a great extent.

3.3 CNN-based LSTM

An LSTM network is similar to a recurrent neural network but it has LSTM cell blocks instead of the standard neural network layers. These cells comprise various components called the input gate, the forget gate and the output gate, the functions and applications of which are outlined in the present section.

In the architecture of a single LSTM cell, it is assumed that there is a new word/sequence value x_t at time step t that is being concatenated to the previous output from the cell h_{t-1} at time step t-1. The first step for this combined input is to go through a tanh layer. The second step is that this input is passed through an input gate. An input gate is a layer of sigmoid activated nodes whose output is multiplied by the input passed through tanh layer. These sigmoid functions at the input gate can act to eliminate any unnecessary element of the input vector. A sigmoid function gives output values that vary between 0 and 1, so the weights that connect the input vector to these nodes can be trained to produce output values close to zero to "switch off" certain input values, or even produce outputs close to 1 to "pass through" other values.

The next component in this cell which facilitates the flow of data is the internal state or the forget gate loop. An LSTM cell can have an internal state variable S_t, which when lagged one time-step gives the value of the variable S_{t-1} that is added to the input data to create an effective layer of recurrence. At this stage, the addition on operation in place of a multiplication operation, helps reduce the risk of vanishing gradients. However, this recurrence loop, controlled by a forget gate – works the same as the input gate, but instead helps the network to determine which state variables are to be "remembered" and which are to be "forgotten".

Finally, the output layer has a tanh squashing function, whose output is controlled by an output gate, that determines what values are actually permissible as output from the cell h_t.

The mathematics of the internal working of an LSTM cell is described in the following sections: [4]

Input Gate

Firstly, the input is squashed between -1 and 1 using a tanh activation function. This can be expressed by:

$$g = \tanh(b_g + x_t*U_g + h_{t-1}*V_g)$$

Where Ug and Vg are the weights for the input and previous cell output, respectively, and bg is the input bias. The output of the input gate which is a series of nodes activated by the sigmoid function, is given by:

$$i = \sigma(b_i + x_t*U_i + h_{t-1}*V_i)$$

This value of g is then multiplied element-wise by the output of the input section of the LSTM cell is then given by:

$$g \circ i$$

Where the '$\circ$' operator stands for the element-wise multiplication operation.

Forget gate and state loop

The output of the forget gate is expressed as:

$$f = \sigma(b + x_t*U_f + h_{t-1}*V_f)$$

The output of the element-wise product of the previous state and the forget gate is expressed as $St-1 \circ f$. The output from the forget gate / state loop stage is:

$$S_t = S_{t-1} \circ f + g \circ i$$

Output gate

The output gate is expressed as:

$$O = \sigma(b_o + x_t*U + h_{t-1}*V_o)$$

So the final output of the cell with the tanh squashing, can be shown as:

$$h_t = \tanh(S_t) \circ O$$

LSTM word embedding layer and hidden layer size

The input xt and ht−1 are vectors of a certain length, and all the weights and bias values are matrices and vectors respectively. Since LSTM is used for sequence modelling to aid in text processing, a vector in this context represents a word. Word vectors that are quite close together in vector space, represent nothing but words appearing relatively close to the same words in sentences.

In order to convert each word in plain text into a meaningful representation in word vector, we used an embedding layer of size 500; that means that each word is represented by a vector of length or size 500, i.e. the word vector is given by

$$[X_1, X_2, \ldots, X_{500}]$$

The size of the embedding layer output is thereafter matched with the number of hidden layers in the LSTM cell. Each hidden layer in the cell is nothing but a set of nodes or neurons, that form the general architecture of the convolutional neural network. [29,30,31]

The CNN-based LSTM architecture

The input shape is given as: (a) batch size, (b) number of time steps [h_0, h_1,…, h_k], and (c) hidden layer size. That means, for each batch sample and each word in the number of time steps, there is a 500 length embedding word vector to represent the input word. These embedding vectors will be learnt during the training period of the model. The input data is then fed into several layers of neurons that constitute the hidden layer of size 500 and these are basically the layers of CNN. The layers of CNN are wrapped in a TimeDistributed Layer before passing them over to the LSTM model. The output from these CNN layers are passed through the LSTM gates to get the results.

Finally, the output layer has a softmax activation function applied to it and this output is compared to the training data for each batch. The error is thereafter back propagated to the previous layers. The training data in this context is the input words that are advanced by just one-time step. At each time step the CNN-LSTM model tries to predict the immediate next word in the sequence. Hence the output layer has the same number of time steps as the input layer.

4. OVERVIEW OF THE CURRENT APPROACH

The present section discusses the implementation of the application of OCR using CNN and subsequent translation of the resultant text set using LSTM.

4.1 CNN Architecture for OCR

The CNN model for the OCR is a sequential model consisting of two sets of a combination of convolution layers and max pooling layer, followed by a fully connected neural network layer. The weights are updated following the Back-Propagation Algorithm.

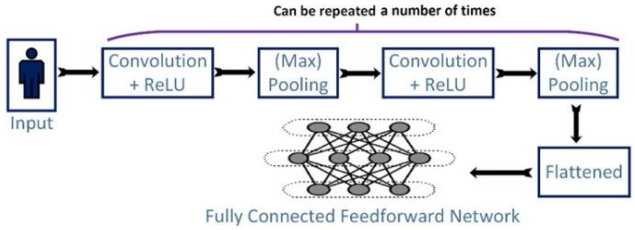


Fig. 1 – Fully connected FeedForward Network

4.2 CNN-Based LSTM Architecture for the Translator

1. The CNN model is built as a sequential model of two pairs of convolution layer and max pooling layer, and the output of these layers is flattened and passed on to a dense fully connected neural network of hidden layer size = 500 to take input word vectors of length 500.

2. The output layer of the CNN model is then wrapped in a Time Distributed Layer.

3. It is then passed on to the LSTM layer followed by a densely connected output layer.

4. The CNN-based LSTM model is thus basically a combination of LSTM gates attached to the output layer of the preceding several layers of CNN.

4.3 Otsu's Threshold Method

In computer vision, Otsu's method is used to reduce a gray-scale image to binary image by automatically performing clustering-based image threshold method. [3]

4.4 Dilation-Erosion Method

Dilation process removes the noise that is involved within the object of the image completely but this leaves the noise in the background enlarged. The background noise is then removed by the erosion process.

Dilation followed by erosion, often referred to as "closing", thus helps in noise removal of the image. [4]

4.5 Algorithm to Train the CNN for OCR

Input:

Printed English alpha-numeral image set

Output:

Trained CNN model to recognize English alpha-numeral character set

Steps:

1. Each input image is converted to gray scale image.

2. The gray-scale image is reduced to binary image using Otsu's Threshold method.

3. Noise removal of the image is done by Dilation process followed by Erosion process.

4. The images are used to train the CNN

4.6 Algorithm to Test the CNN for OCR

Input:

(a) Image of an area of printed English text

(b) MS Word file containing plain target text

Output:

(a) MS Word file of the recognized text

(b) Accuracy of the results evaluated by comparing the output text to the actual target text

Steps:

1. The input image is pre-processed and segmented, and the kernel of the CNN slides through the receptive area of size 20x20 pixels.

2. The CNN model predicts each character in the text and writes them into a MS Word Document.

3. The number of errors are calculated by comparing the output text with the target text word for word.

4.7 Algorithm to Implement the Translator with CNN-based LSTM

Input: MS Word file containing the output of the OCR

Output: Microsoft Word document containing the translated Spanish text

Steps:

1. The input text is represented using word vectors of length 500 and the CNN-LSTM model is trained using a training dataset of 10000 bilingual sentences.

2. This input text is split into sentence pairs and all types of punctuation marks are removed.

3. The input sentence pairs are then used to train and test the CNN-based LSTM model for translator

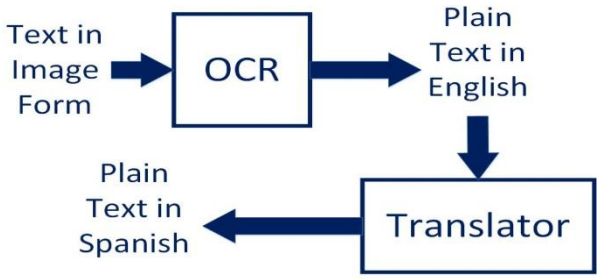


Fig. 2 – Step by Step Diagram of the Application

5. EXPERIMENTAL RESULTS

The experimental results produced the plain text as recognized from the English characters of the given input image in a Microsoft Word document, and the subsequent translated version of the Spanish text in another document**.**

The input dataset for training the CNN model for the OCR comprises the printed images of 52 English alphabets both in capitals and in small caps, 10 Arabic digits andsome common special characters like: ‘,’, ‘.’, ‘?’, ‘!’, ‘(’, ‘)’, ‘—’ etc.

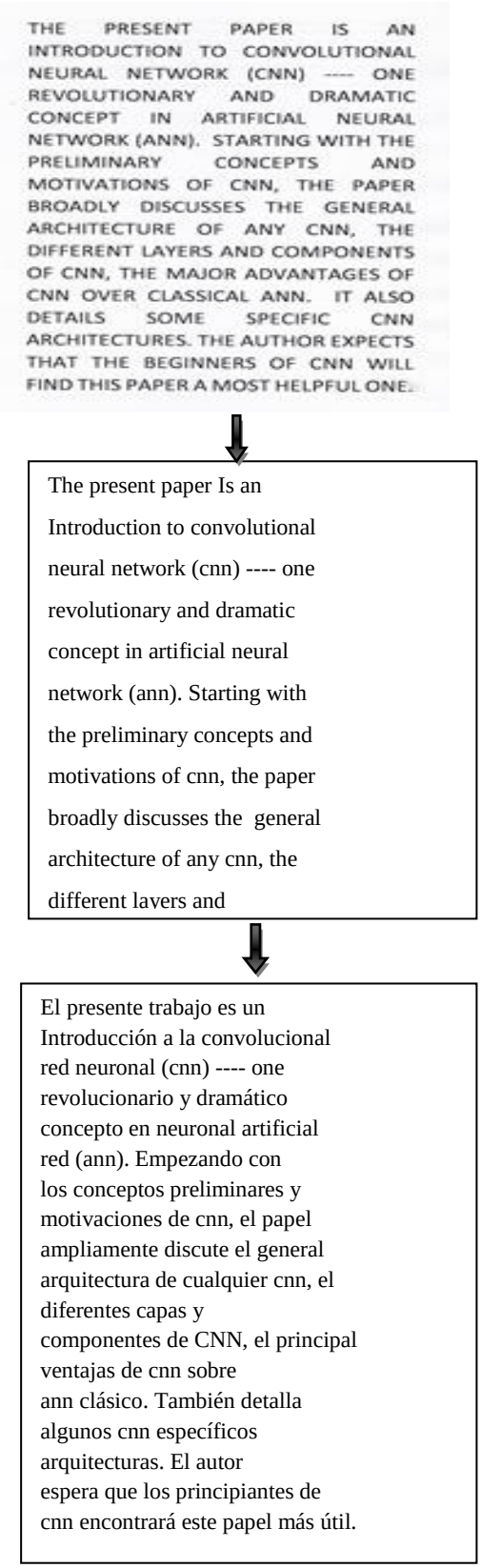
The input dataset for the translator consists of a set of 10,000 bilingual sentence (English-Spanish) pairs collected from www.tatoeba.org.

The experiments are performed in a system having a dualcore processor of type Intel(R) Core(TM) i3-5005U CPU with a clock rate of 2.00GHz and equipped with 4GB RAM and 1TB of Hard Disk space, that runs on Microsoft Windows 10 operating system.

Accuracy of the results is calculated by the following formula:
Accuracy = (1 - (number of errors/total word count)) *100

Table 1 - Experimental Results

Sl. No.	Input and Output File List			Performance	
	Input Image	**Output of OCR**	**Translation**	**OCR Accuracy**	**Translator Accuracy**
1	img1.jpg	Output1.docx	Spn1.docx	93.85	87.54
2	img2.jpg	Output2.docx	Spn2.docx	94.65	77.89
3	img3.jpg	Output3.docx	Spn3.docx	92.17	89.65
4	img4.jpg	Output4.docx	Spn4.docx	96.43	92.43
5	img5.jpg	Output5.docx	Spn5.docx	96.22	90.07



6. CONCLUSION

In the present paper, we have designed and developed one CNN-based optical character recognition system for different English text set and subsequent translation into Spanish language. The performance evaluation of the translator system

through OCR in terms of BLEU-score measure is satisfactory. The integrated system of OCR text recognition and subsequent translation is effective, efficient and most promising for future uses.

7. REFERENCES

[1] J. Wang and X. Hu, "Gated Recurrent Convolution Neural Network for OCR", 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

[2] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition", Wuhan, China, 2015.

[3] Otsu N., "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics (Volume: 9 , Issue: 1 , Jan. 1979), 1979.

[4] Andy Thomas, "Keras LSTM tutorial -- How to easily build a powerful deep learning language model", Adventures in Machine Learning, www.adventuresinmachinelearning.com

[5] Ian Goodfellow, "Deep Learning", MIT Press

[6] Sarker G., "A Treatise on Artificial Intelligence", ISBN : 978-93-5321-793-8 ; 1, 325, 2018

[7] Sarker G., "Some Studies on Convolutional Neural Networks", International Journal of Computer Applications – Vol. 182, DOI 10.5120/ijca201891, October 2018.

[8] Sarker, G., Dhua, S., Besra, M. (2015), "An Optimal Clustering for Fuzzy Categorization of Cursive Handwritten Text with Weight Learning in Textual Attributes", 2015.

[9] Sarker, G., Dhua, S., Besra, M., "A Learning Based Handwritten Text Categorization", 2015.

[10] Sarker, G., Dhua, S., Besra, M., "A Programming based Handwritten Text Identification", 2015.

[11] Sarker, G., "A Weight LearningTechnique for Cursive Handwritten Text Categorization with Fuzzy Confusion Matrix", 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC), 2016.

[12] Sarker G., "A New Technique for Extraction based Text Summarization", 31st Indian Engineering Congress, Kolkata, 2016.

[13] Sarker G., Roy K., "A modified RBF Network with Optimal Clustering for Face Identification and Localization", 2013.

[14] Sarker G., Roy K., "An RBF Network with Optimal Clustering for Face Identification", 2013.

[15] Sarker G., Kundu S., "A modified Radial Basis Function Network for Fingerprint Identification and Localization", 2013.

[16] Bhakta D., Sarker G., "A Radial Basis Function Network for Face Identification and Subsequent Localization", 2013.

[17] Roy K., Sarker G., "A Location Invariant Face Identification and Localization with Modified RBF Network", 2013.

[18] Bhakta D., Sarker G., "A Rotation and Location Invariant Face Identification and Localization with or without Occlusion using modified RBFN", 2013.

[19] Sarker G., Sharma S., "A Heuristic Based RBFN for Location and Rotation Invariant Clear and Occluded Face Identification", 2014.

[20] Kundu S., Sarker G., "A Modified RBFN Based on Heuristic Based Clustering for Location Invariant Fingerprint Rcognition and Localization with or without Occlusion", 2014.

[21] Kundu S., Sarker G., "A Modified BP Network using Malsburg Learning for Rotation and Localization Invariant Fingerprint Recognition and Localization with or without Occlusion", 2014..

[22] Bhakta D., Sarker G., "A Boosting Based Multiple Classifier System with Modified RBFN for Facial Expression Identification", 2014.

[23] Bhakta D., Sarker G., "A Method of Learning Based Boosting in Multiple Classifier for Clear and Occluded Face Identification", 2015.

[24] Bhakta D., Sarker G., "A New Learning Based Boosting in Multiple Classifier System for Colour Facial Expression Identification", 2015.

[25] Kundu S., Sarker G., "A Programming Based Boosting in Super Classifier for Fingerprint Recognition", 2015.

[26] Sarker, G.,Bhakta, D., "An Unsupervised OCA based RBFN for Clear and Occluded Face Identification, Intelligent Computing and Applications", Advances in Intelligent Systems and Computing, 2015.

[27] Kundu, S., Sarker, G., "A Modified SOM Based RBFN for Rotation Invariant Clear and Occluded Fingerprint Recognition", Intelligent Computing and Applications, Advances in Intelligent Systems and Computing, 2015.

[28] Kundu, S., Sarker, G., "An Efficient Integrator Based on Template Matching Technique for Person Authentication using Different Biometrics", Indian Journal of Science and Technology, 2016.

[29] Kundu, S., Sarker, G., "A Multi-level Integrator with Programming Based Boosting for Person Authentication using Different Biometrics", 2016.

[30] Kundu, S., Sarker, G., "A Person Authentication System using a Biometric Based Efficient Multi-Level Integrator", International Journal of Control Theory and Applications (IJCTA, 2017.

[31] Sarker, G. Bhakta, D. A Mega Super Classifier with Fuzzy Categorization in Face and Facial Expression Identification - Int. Journal of Engineering Trends and Technology, 2017.

[32] Kundu, S. Sarker G., "An Efficient Multi Classifier Based on Fast RBFN for Biometric Identification" (2014), Advanced Computing, Networking and Informatics - Vol. 1.