Using CRNN to Perform OCR over Forms

Siddhesh Shinde
Information Technology (University of Mumbai)
Xavier Institute of Technology (University of Mumbai)
Mumbai, India

Tanmey Saraiya
Information Technology (University of Mumbai)
Xavier Institute of Technology (University of Mumbai)
Mumbai, India

Jayesh Jain
Information Technology (University of Mumbai)
Xavier Institute of Technology (University of Mumbai)
Mumbai, India

Prof. Chhaya Narvekar
Information Technology (University of Mumbai)
Xavier Institute of Technology (University of Mumbai)
Mumbai, India

***Abstract*:-In digitization, most of the documents, especially the forms are filled and processed online to speed up the process. But one such organization, Indian Railways, has a process of filling out offline forms for ticket booking. This process is very time-consuming as it requires an attendee to check the form and enter the details into the database. To speed up this process, Optical Character Recognition(OCR) is a very viable option for this case. This paper presents a structured process of locating input fields on the form, scanning the input data, processing the data and entering the data to the final database. We use CRNN(Convolutional Recurrent Neural Network) model to perform OCR on the user's handwritten input. This automated system aims at reducing the waiting time of the current system being used.**

Keywords—Optical Character Recognition, Deep Learning, Convolutional Neural Network, Recurrent Neural Network, Con-nectionist Temporal Classification

I. INTRODUCTION

Digitization has increased the use of the internet in many private sectors as well as in government organizations. Indian Railway introduced the IRCTC portal for the citizens of India which allowed them to book train tickets online and save themselves from the trouble of standing in long queues throughout the day to finally get their tickets. But some people are living in rural areas who lack any computational skills as well as they have almost negligible knowledge about the internet. They are left with the only option to fill out the offline form and personally go to the ticket booking window and stand there throughout the day in a long queue waiting to book railway tickets. To address this issue, this paper presents an automated upgrade to the existing railway ticket booking system, which will cut down the waiting time of the passengers for ticket booking with a large margin.

The methods used in this system is a general method of performing OCR on human handwriting. The form filled by the user is to be scanned and sent as an input to the system. The model will then detect all the user input areas on the form as the main goal is to extract the user-entered information only. Then OCR is performed on these input fields and the generated output is the data that the user entered in the handwritten format.

II. SYSTEM ARCHITECTURE

The system architecture consists of a Word Segmentation algorithm followed by a neural network architecture to perform optical character recognition on the words after segmenting them from the sentences. The input data fed to the system is the scanned image of the form filled by the user. From this scanned image, input fields are located and cropped as individual fields. These cropped images are then fed to the Word Segmentation algorithm which separated every word in these images and we get every word separated in every input field. These individual words are then passed on to the CRNN model to perform OCR. The network architecture is comprised of three elements, the convolutional layers followed by the recurrent layers and finally by the transcription layer, i.e. the CRNN(Convolutional Recurrent Neural Network) model which was originally presented by Baoguang Shi, Xiang Bai, and Cong Yao [1].

The convolutional layers are used to extract feature sequences from the input images. This output is passed on to the recurrent layers for making predictions for each frame of the feature sequence. Finally, the transcription layer or the CTC is used to translate the per-frame predictions by the recurrent layers into a label sequence.

A. Word Segmentation

To perform OCR using the CRNN model, the model should be fed by images of words as input. Hence individual words from the form need to be extracted and passed on to the model. Initially, the form is scanned and by using Haar Cascade filters, bounding boxes are formed around the input fields of the form. These images are cropped and fed as an input to the Scaled Space Technique which segregates the words from human handwriting.

The Scaled Space Techniques, which was originally pro-posed by R. Manmatha and N. Srimal, uses the Line Seg-mentation to detect local maxima(the white space between the lines) and the local minima(the actual text) in the image. As the Line Segmentation tends to give some false local maxima and minima, a Gaussian filter is used to smooth out the projection and reduce sensitivity to noise. These lines are further explored to generate a blob of the image. A blob is regarded as a connected region in space. The blob is formed using an anisotropic Gaussian filter which is defined as:

$$G(x; y; \sigma_x; \sigma_y) = \frac{1}{2\sigma_x\sigma_y} e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)} \quad (1)$$

It is observed that the response of the anisotropic Gaussian filter is high in the range of three to five. The second-order anisotropic Gaussian differential operator is defined by:

$$L(x; y; \sigma_x; \sigma_y) = G_{xx}(x; y; \sigma_x; \sigma_y) + G_{yy}(x; y; \sigma_x; \sigma_y) \quad (2)$$

For a two dimensional image, the corresponding scaled space line image is calculated as:

$$I(x; y; \sigma_x; \sigma_y) = L(x; y; \sigma_x; \sigma_y) \ast f(x; y) \quad (3)$$

By selecting a suitable scale, the blobs of characters merge into blobs of words and then it is easier to delineate the words [7].

B. Convolutional Neural Network

The segmented words are then fed as an input to the CRNN model. The CRNN model is briefly divided into three parts: the convolutional network is present at the top followed by a recurrent network to capture sequential features from the images and lastly the Transcription layer(CTC) to map the output of the recurrent network to a labeled sequence.

The convolution network is generated by using the standard convolutional layers along with ax-pooling layers from the CNN model. A CNN model uses images as an input. With the help of weights and biases, it extracts features from those images. If a standard Deep Neural Network(DNN) model is used to extract features from an image, the pixels from the images would have to flatten which disorients the original features in the image. This would lead to little to no accuracy when the pixel dependency is very high in the input images. Whereas, filters or kernels are used in CNN to perform convolution operations on the image to capture the spatial and temporal dependencies in the image. Along with convolutional layers, max-pooling layers are used in the CRNN. Max-pooling layers are used to scale down the image which helps to reduce the computational power and focus on extracting the dominant features. It also helps to reduce the noise present in the image.

The initial layers of a CNN model are used to detect simple features such as horizontal or vertical lines. As we move deep into the model, the complexity of the features detected increases such that the features detected makes no sense to human eyes. But these features help the CNN model to become more robust and predict the images with good accuracy.

As CNN requires a fixed input shape for all images, each of the input images are scaled accordingly and passed on to the initial convolutional network. The convolutional network extracts feature vectors from these images which are passed on as input to the recurrent network.

The feature vector is generated from left to right by column. As the convolution and max-pooling functions along with activation function are performed on the local regions, a feature map is generated where each column maps the original image and these regions are in the same order to their corresponding columns in the feature map.

As the CNN model has a fixed input shape for all images, it is not appropriate for detecting objects where the sequence is to be retained. Hence, the CRNN uses recurrent networks to capture the sequence of the scanned word images which is important to predict the output [1].

C. Recurrent Neural Network

The output from CNN is fed to a bidirectional Recurrent Neural Network(RNN). The recurrent layer is used to predict a label for each frame in the feature sequence. The recurrent layer has three advantages. Firstly, it can capture the contextual information from a sequence. While predicting the label for any alphabet, it may be easier to distinguish it by combining it with neighboring alphabets rather than considering them individually. Secondly, the error can be back-propagated to the convolutional layers which help to train the entire CRNN model using a single loss function. Thirdly, the input of arbitrary lengths can be fed as an input to the RNN layer.

RNN layer has multiple RNN units between the input and output stages. These units are used to capture past context to predict the label. Although traditional RNN units help to capture the past context, they face a problem of vanishing gradients where they cannot carry forward the context for distant units.

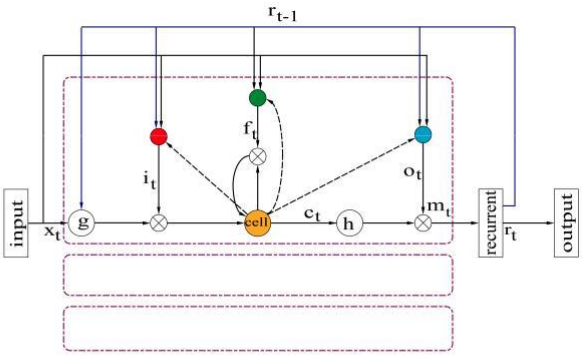

Fig. 1. LSTM memory block.

Long-Short Term Memory(LSTM) is a type of RNN unit which is used to deal with the problem of vanishing gradients. An LSTM consists of three gates: input, output and forget.

LSTM also has a memory cell that is used to carry forward the context till long range. The input at each cell is passed on to the gates and the output generates corresponds to whether the data in the memory cell must be kept or erased. Figure 1 represents how data flows in an LSTM cell. The gates in LSTM are sigmoid functions with pointwise multiplication which outputs a number between 0 and 1. This value indicates how much information should be passed on from the memory cell.

The modern LSTM architecture contains peephole connec-tions from its internal cells to the gates in the same cell to learn the precise timing of the outputs [3]. An LSTM network computes a mapping from an input sequence x = (x1,..., xT) to an output sequence y = (y1,. . . , yT) by calculating the network unit activations using the following equations iteratively from t = 1 to T:

$$i_t = (W_{ix}x_t + W_{im}m_{t\ 1} + W_{ic}c_{t\ 1} + b_i) \quad (4)$$

$$f_t = (W_{f\,x}x_t + W_{f\,m}m_{t\ 1} + W_{f\,c}c_{t\ 1} + b_f) \quad (5)$$

$$c_t = f_t\ \ c_{t\ 1} + i_t\ \ g(W_{cx}x_t + W_{cm}m_{t\ 1} + b_c) \quad (6)$$

$$o_t = (W_{ox}x_t + W_{om}m_{t\ 1} + W_{oc}c_t + b_o)$$

$$m_t = o_t\ \ \ h(c_t)$$

$$y_t = (W_{ym}m_t + b_y)$$

where the W terms denote weight matrices, W_{ic}, W_{fc} , W_{oc} are diagonal weight matrix for peephole connections, the b terms denote bias vectors, s is the logistic sigmoid function, and i, f, o, and c are respectively the input gate, forget gate, output gate, and cell activation vectors, all of which are the same size as the cell output activation vector m, o is the element-wise product of the vectors, g and h are the cell input and cell output activation functions [4].

D. Connectionist Temporal Classification

The output generated by RNN is a matrix containing score for every individual character at a particular time-step. This matrix with the corresponding ground-truth text(transcript of the word in the image) is passed on to CTC as input. It then uses all possible alignments of the ground-truth text in the image and sum over all the scores generated. This results in the score of that ground-truth text.

Here, the encoding of duplicate characters is resolved by using a pseudo-character denoted by "-". While decoding, all these blanks are removed. Example: the word "too" can be encoded as "—ttttooooooo–" or "–t–o–o-" or "too". The network is trained to output text in an encoded format. Later on, the best path decoding algorithm is used to decode the encoded text. When the output matrix of the RNN and the ground-truth text are passed on to the CTC, a loss value is calculated for that particular pair of a word and its ground-truth text. Figure 2 shows a matrix with two time-steps (t0 and t1) and three characters ("a", "b" and "-"). For every time-step, the character scores sum to 1. The thin line in the figure represents the text "a" whereas the dashed line represents the text "".

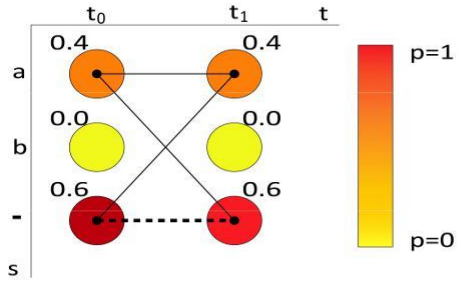

Fig. 2. An example of a matrix containing character-score.

(7) For the example shown in the image, the score for path "aa" can be calculated by multiplying the probabilities of text "a" at time-step t0 and t1, i.e. 0.4 * 0.4 = 0.16. Similarly,

(8) the score for all possible permutations to predict output as "a" can be calculated. Summation over all these scores will give

(9) us an overall score for the text "a" and the ground-truth text. All possible paths for the text "a" are "aa", "a-" and "-a". Summation of their scores will be (0.4 * 0.4) + (0.4 * 0.6) + (0.6 * 0.4) = 0.64. Now, if the ground-truth text is "", the loss value is calculated, i.e. the score for ""(0.6 * 0.6 = 0.36). This loss value is back-propagated to the network and the weights are updated accordingly.

For predicting the labels for unseen images, the best path decoding algorithm is used. This algorithm works in two stages. First, it calculates the best path by selecting the charac-ters with the highest probabilities at that particular time-step. Second, it decodes the generated encoded text by removing the duplicate characters as well as the blanks. After that, the output generated is the recognized text of the word in the image [6].

III. Experimental Evaluation

Experiments were carried on IAM Dataset which contains 115320 words with their transcripts [5]. These images are extracted from pages of scanned text using an automatic seg-mentation scheme. The images are of varying size depending upon the handwriting and the length of the word and are grayscale images in PNG format.

After filling the form , the form is scanned and then processed to detect all the input fields. Figure 3 shows the detection of bounding boxes around the input fields of the

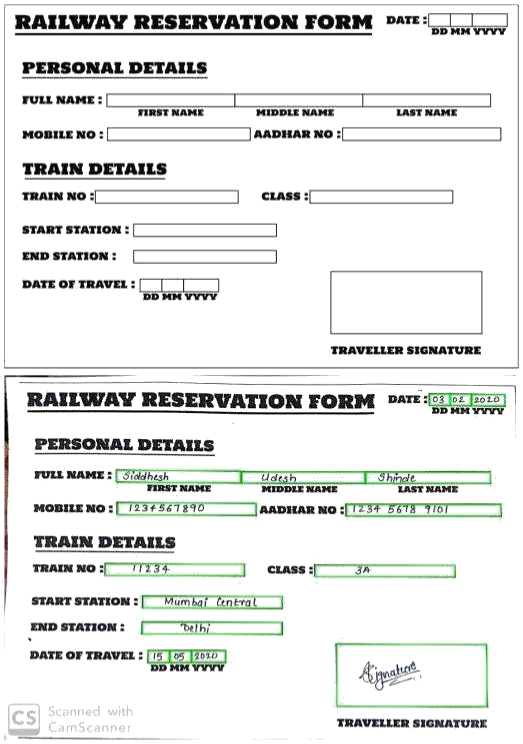

Fig. 3. Form with detected Bounding Boxes.

form. Images are cropped according to these boxes and then are further passed on to CRNN model.

This paper presents the CRNN model used to perform OCR. The input image is a gray-scale image of size 128x32. After performing convolution via 5 CNN layers, the images are transformed into size of 32x256. The RNN part consists of 2 LSTM layers containing 256 nodes each to propagate information through the sequence and map the sequence to a matrix of size 32x80. Each element in the matrix is a score for the corresponding character(80 characters) at that particular time-step(32 time-steps). The CTC layer then uses this matrix to train along with the ground truth text. While inferring, it uses the beam search decoding algorithm to predict the output. The current model gives an error rate of 10.625% on IAM word dataset. This can be reduced by using Best Path Decoding or Word Beam Search algorithm in CTC. Since typically the words encountered in railway forms are proper nouns, Vanilla Beam Search algorithm is the best choice.

IV. Results and Decision

An evaluation was done on forms filled by people having various handwriting styles. It was observed that handwriting styles matching the style of the IAM dataset, i.e. having more inter-word spaces and less intra-word spaces were segmented and recognized correctly.

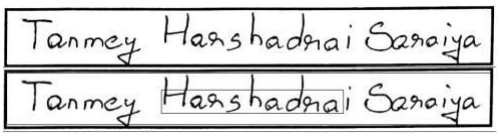

Fig. 4. Congested input and detected bounding boxes.

Figure 4 shows the result of the bounding box detected incorrectly when the user wrote words that were congested. This affects the Word Segmentation algorithms as the distance between the words is similar to the distance between the letters.

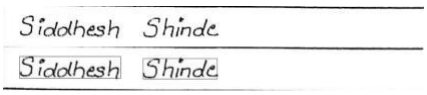

Fig. 5. Correct input and detected bounding boxes.

On the other hand, Figure 5 shows the result of the bounding box detected when the user wrote words that were spaced out. This input has more spacing between the words as compared to the spacing between the alphabets individually. It increases the probability of segregating words correctly from the input.

The accuracy of the system was found to be 72.22% after testing it over 100 random words. Figure 6 shows the words which were recognized incorrectly by the system. Recognizing the alphabet ”l” as ”d” and the alphabet ”a” with ”o” shows that the system still lacks the accuracy in determining closely resembling words. This may also happen due to users having different handwriting styles.

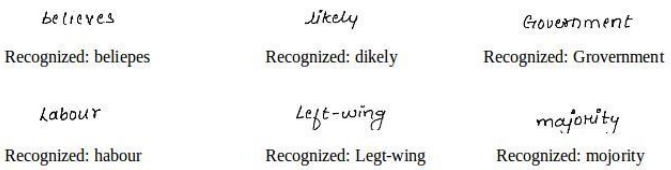

Fig. 6. Words recognized incorrectly.

Having the model trained on more vivid handwriting styles other than the IAM dataset will help in increasing the accuracy of segregating the words properly and also detecting the output of those words. Also, increasing the count of words in the training dataset which closely relates to the problem statement of railway tickets, i.e. including the station names, the sequence of numbers corresponding to mobile numbers,

etc. will help the model recognize the words with much more probability.

V. Conclusion

This paper focuses on reducing the time consumed by the organizations which are still dealing with paper-back forms for gathering their customer's data. Against this traditional approach, the system presented in this paper uses OCR to recognize user handwriting and enter the processed output to the database. The system uses the Scaled Space Technique along with a neural network architecture that integrates the spatial and temporal feature extraction advantage of Convolu-tional Neural Networks and sequential feature extraction ad-vantage of Recurrent Neural Network. The model avoids fully connected layers in CNN which reduces the computational power required to train the model. All these properties make the network an excellent approach for image-based sequence recognition.

The proposed system is a general framework, it can be integrated with the OCR for region-specific scripts which are used by the forms. The overall approach of OCR itself can be applied to other domains and problems that involve sequence prediction in images. To make the model more accurate, it can be further trained on a dataset with different handwriting styles or datasets containing different vocabulary. Increasing the layers in the convolutional network or stacking up multiple bidirectional LSTM layers can result in predicting complete sentences rather than individual words. Specifically for our problem statement, accuracy can be improved by training the model on vocabulary mostly used in railway forms(station names or train numbers).

References

[1] 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," in IEEE Transactions on Pattern Analysis and Ma-chine Intelligence, vol. 39, no. 11, pp. 2298-2304, 1 Nov. 2017.

[2] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[3] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," Journal of Machine Learning Research, vol. 3, pp. 115–143, Mar. 2003.

[4] Has¸im Sak, Andrew Senior, Franc¸oise Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," INTERSPEECH, 14- 18 September 2014, Singa-pore.

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

[6] Manmatha R., Srimal N. (1999) Scale Space Technique for Word Segmentation in Handwritten Documents. In: Nielsen M., Johansen P., Olsen O.F., Weickert J. (eds) Scale-Space Theories in Computer Vision. Scale-Space 1999. Lecture Notes in Computer Science, vol 1682. Springer, Berlin, Heidelberg.

[7] Siddhesh Shinde, Tanmey Saraiya, Jayesh Jain and Chhaya Narvekar, "Automatic Data Collection from Forms Using Optical Character Recog-nition", IRJET, Volume: 06 Issue: 10, Oct 2019.