Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

Improved Offline Optical Handwritten Character Recognition: A Comprehensive Review using Tensorflow

Prajna Nayak (Author)
Department of Computer Science
Don Bosco Institute of TechnologyMumbai, India

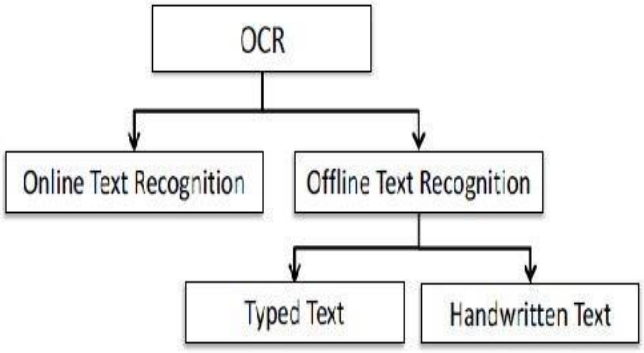
Sanjay Chandwani (Author)
Department of Information Technology
Thadomal Shahani Engineering College Mumbai,India

***Abstract*— Nowadays, Handwritten Character Recognition (HCR)is a significant and challenging topic of research in the field of image processing. In recent years, extensive research has been conducted on the recognition of handwritten English alphabets. Numerous identification approaches are now being used to recognize handwritten English alphabets (character). HCR's application domain is digital document processing, which includes data mining from data entry, checks, loan applications, credit card applications, tax, and health insurance forms, among others. We offer an overview of current research towards the recognition of handwritten English alphabets in this survey. There are no restrictions on the writing technique in a handwritten text. Handwritten alphabets are difficult to recognize due to the wide variety of human handwriting techniques, differences in letter size and shape, and angle. Offline handwritten character recognition will be accomplished in this study utilizing a convolutional neural network and Tensorflow. Offline Handwritten Text Recognition (HTR) systems convert scanned photos containing text to digital text. We will train a Neural Network (NN) using word-images extracted from the IAM dataset. Due to the fact that the input layer (and thus all subsequent layers) can be kept short for word- images, NN training is possible on the CPU (of course, a GPU would be better). This implementation satisfies the bare minimal requirements for HTR with TF. Soft Max Regression is used to assign probability to handwritten characters being one of numerous characters since it produces values between 0 and 1 that add to 1. The objective is to design software that is extremely accurate, has a small time and space footprint, and is also ideal in terms of performance.**

Keywords—Handwritten Character Recognition, Neural Network, Tensor Flow

I. INTRODUCTION

OCR is a vast area of research in Soft Computing, artificial intelligence (AI), pattern recognition (PR), and computer vision. OCR is a broad term that refers to the process of digitising handwritten texts or images of printed materials in order for them to be electronically changed, stored, and searched more effectively and accurately. Despite decades of research and advancements in this area, machines are still far from human-level analysis capabilities. The purpose of an OCR approach is to recognise manuscripts (like humans do) inside a challenging article. OCR techniques are broadly classified into two categories: online text recognition and offline text recognition. Offline OCR is classified into two subcategories: typed text and handwritten text. Offline text recognition is a technique that utilises a static version of an article to process it. Offline text recognition is divided into two subcategories: typed and handwritten articles. Both subcategories process problem is the primary constraint complexity in OCR, typically requiring language-specific approaches. On the other hand, optical character recognition (OCR) of typed articles is in high demand for practical applications such as historical article analysis, official note and article processing, and car plate recognition. OCR (optical character recognition) of written documents for English alphabets has become one of the most successful uses of pattern recognition (PR) and artificial intelligence technologies (AI). Almost majority of the current research in this area is focused on articles that are exceedingly complicated, loud, and twisted, as well as improving text recognition rates.

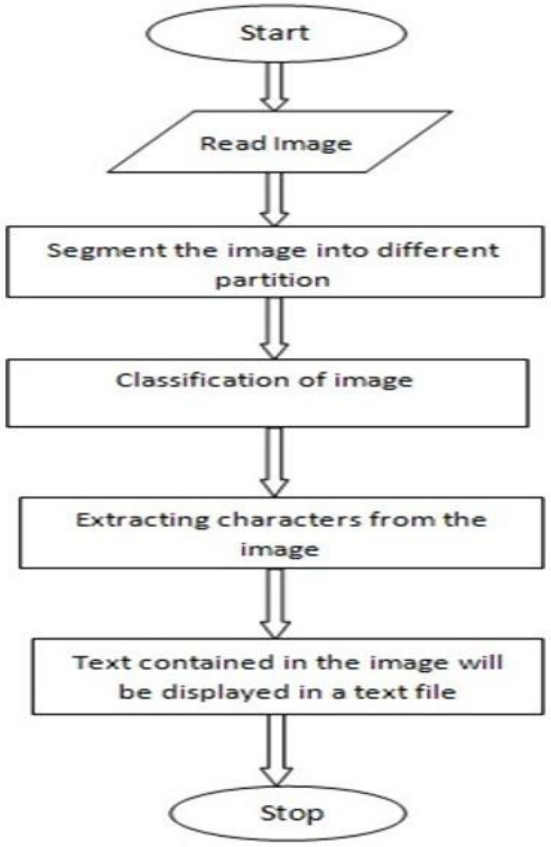


an image of the article obtained by a scanner or camera. Clearly, given the variety of handwriting styles and the un-benchmarked nature of handwritings, the offline handwriting identification

II. PROPOSED METHODOLOGY

In the previous sections, we defined the goal and need for this research.Our task requires the employment of a NN. It is composed of layers of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and a final Connectionist Temporal Classification (CTC) layer.
Additionally, we may see the NN more formally as a functionthat transfers an image (or matrix) M of size WH to a character sequence (c1, c2,...) of length 0 to L. As can be seen,the text is recognized on a character-by-character basis, whichmeans that words or texts not included in the training data can be recognized as well (as long as the individual characters getcorrectly classified).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021



Operations:

A. CNN:

The CNN layers are fed the input image. These layers are trained to extract the most important information from an image. Each layer is composed of three distinct operations. To begin, the convolution operation is applied to the input, which results in the application of a filter kernel of size 5X5 in the first two layers and 3X3 in the last three layers. Following that, the non-linear RELU function is used. Finally, a pooling layer condenses image regions and produces a smaller version of the input. While the image height is reduced by two in each layer, feature maps (channels) are added, resulting in a 32 X 256-pixel output feature map (or sequence).

B. RNN:

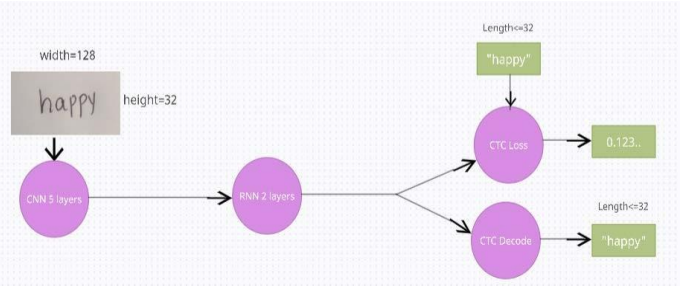
Each time step comprises 256 characteristics; the RNN propagates pertinent information via this sequence. The widely utilized Long Short-Term Memory (LSTM) implementation of RNNs is used because it is capable of propagating information over greater distances and has more robust training features than a RNN. The output sequence of the RNN is mapped to a 32-by-80 matrix. The IAM dataset has 79 distinct characters; one additional character is required for the CTC procedure (CTC blank label); thus, each of the 32 time steps contains 80 entries.

C. CTC:

While the NN is being trained, the CTC is supplied with the RNN output matrix and the ground truth text and is tasked with calculating the loss value. While inferring, the CTC is provided with only the matrix and is responsible for decoding it into the final text. Both the ground truth and recognised texts have a maximum length of 32 characters.

D. Data Input:

It is a 128X32 gray-value image. Typically, the images in the collection are not exactly this size, so we adjust them (without distorting) until they are either 128 pixels wide or 32 pixels tall. The image is then copied into a (white) target image with a resolution of 128X32. Finally, we normalise the image's grey levels, which simplifies the NN's duty. Data augmentation can be simply included by replicating the image and aligning it to the left, or by arbitrarily scaling the image.



III. RESULTS

CNN:

Create a kernel of size kk for each CNN layer to be used in the convolution procedure.

```
kernel = tf.Variable(tf.truncated_normal([k, k, chIn, chOut], stddev=0.1))
conv = tf.nn.conv2d(inputTensor, kernel, padding='SAME', strides=(1, 1, 1, 1))
```

Then, pass the convolution result to the RELU operation and then to the pooling layer with size pxpy and step size sxsy.

```
relu = tf.nn.relu(conv)
pool = tf.nn.max_pool(relu, (1, px, py, 1), (1, sx, sy, 1), 'VALID')
```

RNN:

Create and stack two RNN layers, each with 256 units.

```
cells = [tf.contrib.rnn.LSTMCell(num_units=256, state_is_tuple=True) for _ in range(2)]
stacked = tf.contrib.rnn.MultiRNNCell(cells, state_is_tuple=True)
```

Then, construct a bidirectional RNN from it, traversing the input sequence from front to back and vice versa. As a result, we obtain two 32X256-byte output sequences, fw and bw, which we later concatenate along the feature-axis to generate a 32X512-byte sequence. Finally, it is mapped to a 32X80-element output sequence (or matrix) that is supplied into the CTC layer.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

```
((fw, bw),_) = tf.nn.bidirectional_dynamic_rnn(cell_fw=stacked, cell_bw=stacked, inputs=inputTensor, dtype=inputTensor.dtype)
```

CTC:
We input the operation both the ground truth text and the matrix for loss computation. A sparse tensor is used toencode the ground truth text. Both CTC processes requirethe length of the input sequences.

```
gtTexts = tf.SparseTensor(tf.placeholder(tf.int64, shape=[None, 2]), tf.placeholder(tf.int32, [None]), tf.placeholder(tf.int64, [2]))
seqLen = tf.placeholder(tf.int32, [None])
```

We now have all of the input data necessary to create the lossand decoding operations.

```
loss = tf.nn.ctc_loss(labels=gtTexts, inputs=inputTensor, sequence_length=seqLen, ctc_merge_repeated=True)
decoder = tf.nn.ctc_greedy_decoder(inputs=inputTensor, sequence_length=seqLen)
```

Traning:
The mean of the batch element's loss values is used to trainthe NN: it is passed via an optimizer such as RMSProp.

```
optimizer = tf.train.RMSPropOptimizer(0.001).minimize(loss)
```

IV. IMPROVING THE MODEL

If you want to feed the NN whole text lines rather than wordpictures, you must increase the NN's input size.

If you want to increase the accuracy of recognition, you canuse one of the following hints:

1. Data augmentation: increase the size of the dataset by performing additional (random) transformations on the inputimages.
2. Eliminate cursive writing from the input images.
3. Increase the size of the input (if input of NN is large enough,complete text-lines can be used)
4. Increase the number of CNN layers
5. Substitute a 2D-LSTM decoder for the LSTM decoder: limitthe output to dictionary words using token passing or word beam search decoding.
6. Text correction: if the identified term is not in a dictionary,look up the closest match.

V. CONCLUSION

After reading the articles, it was decided that feature extraction techniques are far more accurate than manyof the standard vertical and horizontal methods. Additionally, employing a Neural network with tried and true layers provides the benefit of a better noise tolerance, resulting in more accurate findings. This survey article analyses the primary methodologies used in the domain of handwritten English alphabet recognition during the last decade. Numerous strategies for pre-processing, segmentation, feature extraction, and classification arecovered in detail. Although several strategies for dealing with the complexity of handwritten English alphabets have evolved over the years, much research is still required before a realistic software solution can be made accessible. The existing HCR for handwritten text is extremely imprecise. We demand a proficient remedy to this issue in order to boost overall performance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

VI. REFERENCES

[1] Singh, Sameer, Mark Hewitt,"Cursive Digit And Character Recognition on Cedar Database", Pattern Recognition, 2000.Proceedings. 15th international conference on. Vol. 2. IEEE 2000.

[2] K. Gaurav and Bhatia P. K., "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition", 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[3] Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[4] U. Pal, T. Wakabayashi and F. Kimura, "Handwritten numeral recognition of six popular scripts," Ninth International conference on Document Analysis and Recognition ICDAR 07, Vol.2, pp.749-753, 2007.

[5] Reena Bajaj, Lipika Dey, and S. Chaudhury, "Devnagarinumeral recognition by combining decision of multiple connectionist classifiers", Sadhana, Vol.27, part. 1, pp.-59-72, 2002.

[6] Sandhya Arora, "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.

[7] Nafiz Arica, and Fatos T. Yarman-Vural, ―Optical Character Recognition for Cursive Handwriting,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113, June 2002.

[8] K. H. Aparna,, Vidhya Subramanian, M. Kasirajan, G. Vijay Prakash, V. S. Chakravarthy, Sriganesh Madhvanath, "Online Handwriting Recognition for Tamil", IWFHR, 2004,Proceedings. Ninth International Workshop on Frontiers in Handwriting Recognition, Proceedings. Ninth International Workshop on Frontiers in Handwriting Recognition 2004, pp. 438-443, doi:10.1109/IWFHR.2004.

[9] Surya Nath R S M, Afseena "Handwritten Recognition-A review", International Journal of Scientific and Research Publications, Volume 5, Issue 3, March 2015 1 ISSN 2250-3153

[10] Anita Pal and Davashankar Singh,"Handwritten English Character Recognition Using Neural Network", International Journal of Computer Science and Communication, pp: 141-144, 2011.

[11] J.Pradeep1 , E.Srinivasan2 and S.Himavathi3, "DIAGONAL BASED FEATURE EXTRACTION FOR HANDWRITTEN ALPHABETS RECOGNITION SYSTEM USING NEURAL NETWORK" International Journal of Computer Science & Information Technology(IJCSIT), Vol 3, No 1, Feb 2011

[12] Theingi Htike and Yadana Thein "Handwritten Character Recognition Using Competitive Neural Trees " IACSIT International Journal of Engineering and Technology, Vol. 5, No. 3, June 2013

[13] Ayush Purohit #1, Shardul Singh Chauhan#2 "A Literature Survey on Handwritten Character Recognition" Ayush Purohit et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7(1) ,2016, 1-5

[14] Manoj Sonkusare and Narendra Sahu "A SURVEY ON HANDWRITTEN CHARACTER RECOGNITION (HCR) TECHNIQUES FOR ENGLISH ALPHABETS" Advances in Vision Computing: An International Journal (AVC) Vol.3, No.1, March 2016

[15] Hallale, Sumedha B., and Geeta D. Salunke. "Twelve Directional Feature Extraction for Handwritten English Character Recognition." International Journal of Recent Technology and Engineering 2, no. 2 (2013)