International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

Recognizing Handwritten Texts

Devesh Ramesh[1]
Department of Electronics and Telecommunication Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, Maharashtra, India

Mr. Gaurav Y. Tawde
Assistant Professor,
Department of Electronics and Telecommunication Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, Maharashtra, India

Aayushi Mishra[1]
Department of Electronics and Telecommunication Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, Maharashtra, India

Anushka Kulkarni[1]
Department of Electronics and Telecommunication Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, Maharashtra, India

Sudipa Majumdar[1]
Department of Electronics and Telecommunication Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, Maharashtra, India

***Abstract*—Having text in digital format is more convenient than having it written down on paper. Digital text is immune to degradation, and provides means for automating several operations. In this paper, we present a way to convert handwritten text into digital text. A deep learning approach is used for recognition. Pre-processing techniques like binarizing, using adaptive thresholding, and median filtering, are carried out. The skew of the resulting image is analyzed, and corrected if needed. The image is then segmented line wise, using a horizontal histogram projection, and each line is segmented into words, using a vertical histogram projection. Each individual word is then fed into the neural network for recognition.**

Keywords—Neural networks; deep learning; handwriting recognition; slant removal; segmentation; histogram projection.

I. INTRODUCTION

When operations are to be performed on hardcopies of text, it becomes tedious as people have to manually carry out the tasks. On the other hand, if we have softcopies of text, performing operations on it would be comparatively easier as it can be automated on a computer, and the information would not degrade over time. However, converting hardcopies of text into softcopies is not straightforward, and many people resort to manually entering all the information on a computer. This can quickly get out of hand when there are several thousands of pages of information that need to be converted.

II. PREVIOUS WORK

A. *An Overview of Feature Extraction Techniques in OCR for Indian Scripts Focused on Offline Handwriting by Gaurav Y. Tawde, Mrs. Jayashree M. Kundargi [1]*

In this paper, character recognition techniques were reviewed. The data is acquired and goes through pre-processing techniques such as binarization, noise reduction, normalization, slant removal. The image is then segmented and its features are extracted. After extracting its statistical and structural features, classifiers such as neural networks, SVM, and KNN were compared.

The paper concluded by stating that the selection of proper feature extraction techniques and classifier(s) is the key issue in OCR system because feature extraction methods best suited for one application may not produce optimal results for another application.

B. *Optical Character Recognition for Isolated Offline Handwritten Devanagari Numerals Using Wavelets by Gaurav Y. Tawde [2]*

The image is preprocessed and decomposed by a wavelet filter into a 900-element feature vector. Upon feeding this to a neural network with one hidden layer consisting of 45 neurons and a tangent sigmoid transfer function, a 10-element output vector is produced, indicating which numeral it corresponds to.

The recognition accuracy obtained using these techniques was 70%. The paper stated that further improvement in accuracy can be achieved by adding relevant features to the classifier and by making use of multiple classifiers.

C. *Handwritten Text Recognition in Historical Documents by Harald Scheidl [3]*

The proposed system is based on Artificial Neural Networks (ANN). The image is preprocessed to simplify the problem for the classifier. This includes contrast normalization as well as data augmentation to increase the size of the dataset.

The handwritten text is also set upright by an algorithm that removes the slant of the text. The classifier has Convolutional Neural Network (CNN) layers to extract features from the input image and Recurrent Neural Network (RNN) layers to The RNN outputs a matrix which contains a probability distribution over the characters at each image position. Decoding this matrix yields the final text and is done by the connectionist temporal classification (CTC) operation. A final text postprocessing accounts for spelling mistakes in the decoded text.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

III. STEPS FOR RECOGNIZING HANDWRITTEN TEXT

A. *Image Acquisition*

The image or the document containing the handwritten text is acquired through a scanner, a camera, or any other suitable digital input device [1].

The acquired images are represented in the form of a matrix. For binary images, the values of the matrix can be in the range [0, 1], where 0 corresponds to a black pixel and 1 corresponds to a white pixel. For grayscale images, the values of the matrix can be in the range [0, 255], where 0 corresponds to a black pixel, and the intensities keep increasing up to 255, which corresponds to a white pixel.

For RGB images, the values of the matrix are a sequence of 3 numbers in the form of [R, G, B], each of which range from [0, 255]. The value of R indicates of red a pixel is, that is, 0 indicates the absence of red, and 255 indicates a high intensity of red in the pixel. Likewise, the value of G corresponds to green, and the value of B corresponds to blue.

B. *Image Enhancement*

The acquired image can be enhanced to make it easier for classification. This can be done by first converting the image into grayscale, whose intensity resolution would be 8 bits (256 intensity levels ranging from 0 – 255).

The grayscale image is then converted in a binary image whose intensity resolution is 1 bit (2 intensity levels ranging from 0 – 1). This can be done by setting an intensity threshold, and converting all the intensity values below the threshold to 0, and converting the rest to 1. This process is known as thresholding.

A global threshold can be used, but would fare poorly against shadows. To overcome this, we can divide the image into different regions, and use a different threshold for each region. This process is known as adaptive thresholding.

C. *Image Segmentation*

The binarized image can now be divided into individual lines, the individual lines can be divided into words, and the words can be further broken down into separate characters.

D. *Feature Extraction*

Given a character, we need to extract features that can be used to uniquely classify the character. The extracted features are represented in the form of a vector.

These features can be statistical, that is, the features are computed from the distribution of the pixels, or the features can be structural, that is, the features are computed from the topological and geometrical properties of the character [1].

E. *Recognition*

Recognition is the process of classifying a character based on its features. There are various approaches for classification such as Hidden Markov Model (HMM), Support Vector Machine (SVM), and Artificial Neural Network (ANN). Our implementation of handwritten text recognition is based on ANN.

IV. IMPLEMENTATION

A. *Image Acquisition*

The images from the IAM dataset are used for training the model. A portion of the images from the dataset were held back from training for the purpose of validation. An iterator is used to feed the images to the model as and when required instead of storing all of the images in the memory beforehand.

The model accepts an image input of size 128x32, but the images from the dataset vary in size. For resizing the image into the suitable dimensions, both the dimensions of the image were scaled down by the same factor until one of its dimensions matched the corresponding input dimension. This was done to scale the image down without causing any distortions to the written text.

For the other dimension, a blank image was attached to the right of the scaled down image to satisfy the required input dimensions. The colour of this blank image is the gray value of the input image with the highest frequency.

To increase the size of the dataset, data augmentation is used, wherein, instead of always attaching the blank image to the right of the scaled down image, a fraction of it was attached to the left, and the remainder was attached to the right. This fraction could be varied randomly each time the augmentation is performed [3].

B. *Image Enhancement*

The input image is first converted into a grayscale image, which is further converted into a binary image by performing adaptive thresholding. But this image may contain several tiny spots as a result of adaptive thresholding. To remove these spots, we use median filtering, wherein, 5x5 regions of the image are considered at a time, and are replaced with their median pixel value. The resulting image is dilated to thicken the handwritten text. The image might also contain slanting text, which would have an impact on the accuracy of the model. To remove the slant from the image, we first rotate the image by a set of pre-determined angles. For each rotation, we do the following:

- We initialize a variable and set its value to 0.
- We take a look at the number of foreground pixels in each column of the image
- If the number of foreground pixels of the column is equal to the distance between the first foreground pixel and the last foreground pixel, that is, if the stroke is continuous, the square of the number of foreground pixels is added to the variable.
- For each column, this procedure is repeated.

The value of that variable would be maximum for the angle of rotation where the image appears straight. This is how the angle of rotation is determined [3]. Consider the following figures:

Fig. 1. Straight Line

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

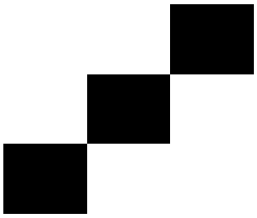

Fig. 2. Inclined Line

In Fig. 1,
The number of foreground pixels in the first column = 0
The number of foreground pixels in the second column = 3
The number of foreground pixels in the third column = 0
Therefore, value of the variable would be $3^2 = 9$

In Fig. 2,
The number of foreground pixels in the first column = 1
The number of foreground pixels in the second column = 1
The number of foreground pixels in the third column = 1
Therefore, the value of the variable would be $1^2+1^2+1^2 = 3$

C. Image Segmentation

The enhanced image is broken down into separate lines. These lines are further broken down into words, which are in turn, broken down into character segments. A segment with a width greater than a specified threshold is treated as a blank space that separates two words. This segment is used as a delimiter for words.

The segmentation of a page into lines is performed as follows:

- The summation of pixel intensities is calculated for each row.
- A histogram is plotted whose x-axis corresponds to the rows, and y-axis corresponds to the summation of pixel intensities for that particular row.
- A summation threshold is set.
- If the summation of the pixel intensities for a particular row falls below that threshold, that row is considered the as the end of a line.
- Such rows are recorded in a list.
- The list of rows is then iterated over, and the image is segmented at those particular rows.
- The resulting images are stored as individual lines.

The segmentation of a line into words also follows a similar process, except that the histogram's x-axis corresponds to the columns, and y-axis corresponds to the summation of pixel intensities for that particular column.

The segmentation of a word into characters also follows a similar process, except that in the end, a threshold is set for the width of the image, and if the width of an image exceeds this threshold, it is treated as a space, which is our delimiter in this case. The characters before each delimiter are treated as a word.

D. Feature Extraction

To extract the features from the image, 5 Convolutional Neural Network (CNN) layers are used. Each CNN layer performs the following:

1) Convolution

In image processing, a convolution operation between an image and a kernel is carried out as follows:

0	1	2	1
1	2	0	2
0	1	1	0
1	0	1	2

Fig. 3. Sample Image

1	2	3
3	2	1
0	1	0

Fig. 4. Sample Kernel

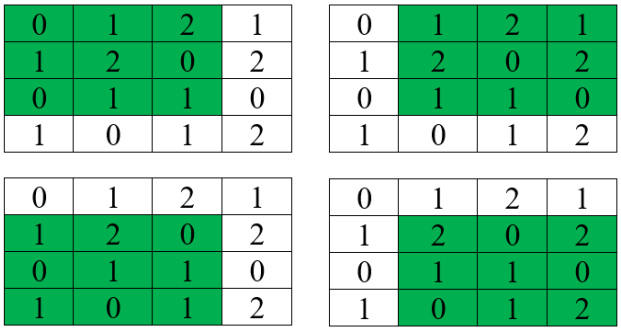

Fig. 5. Regions Covered by the Kernel

16	17
8	14

Fig. 6. Output Image

- The kernel is first placed at the top left of the image, and the product between the kernel, and the region it covers is calculated.
- In this case, the product would be (0*1) + (1*2) + (2*3) + (3*1) + (2*2) + (0*1) + (0*0) + (1*1) + (1*0) = 16.
- This region is now replaced with the product calculated.
- The kernel is then slid by a certain length to the right. This length is known as the stride length. The same process is repeated for the new region the kernel covers.
- After reaching the end of a row, the kernel is placed on the next row, and again, the process is repeated.

In a convolution layer, the convolution operation is carried out in the same manner, but instead of using a single kernel, multiple kernels are used. Each kernel is convolved with the image, and the results are concatenated together depth-wise, and are passed as the input for the next operation.

2) Activation

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

Each convolution operation is followed by an activation function. The activation function used in this case is the Rectified Linear Unit (ReLU). Its output is given by max(0, x), where x is the input. This is done to introduce non-linearity into the model.

3) Pooling

The pooling operation may be performed after a series of convolution and ReLU operations or immediately after every convolution and ReLU operation. There are different types of pooling operations. Over here, the max pooling operation is performed, wherein, NxN regions of the input are considered, and are replaced by their maximum pixel value.

E. Recognition

For recognition, Recurrent Neural Networks (RNN) are used. RNNs are used to model data with variable length. They have an internal state, which is updated each time the RNN processes an input sequence. Two Long Short-Term Memory (LSTM) layers, which is a variant of an RNN, are stacked together, and a bidirectional RNN is created from it. The input sequence would be passed from front to back, and from back to front, resulting in two output sequences. This would be then concatenated and passed on to the Connectionist Temporal Classification (CTC) layer.

The CTC layer is responsible for decoding the output of the RNN, and for calculating the loss. The RNN would give an output of multiple paths, that is, multiple possibilities of the recognized word. The score of a path is calculated by multiplying the probabilities of the characters of that path. Therefore, the loss of the output, which is composed of multiple paths, is the summation of the scores of all the individual paths. For decoding, the CTC layer calculates the word corresponding to the best path. The best path is calculated by taking the character with the highest probability for each step along the axis in which the writing happens [3].

V. RESULTS

A. Pre-Processing

Here, we have an input image which has uneven lighting. To correct the uneven lighting, we use adaptive thresholding. But this leaves us with several tiny spots on the image. To remove these spots, we use median filtering, followed by scanning the image of small objects, and removing them. After this, the letters appear to be hollow. To correct this, we use dilation.

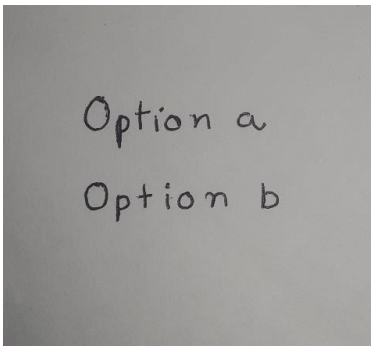

Fig. 7. Input Image for Pre-Processing

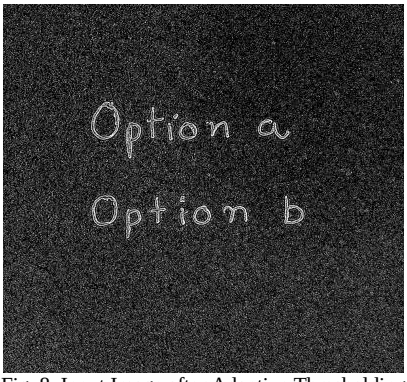

Fig. 8. Input Image after Adaptive Thresholding

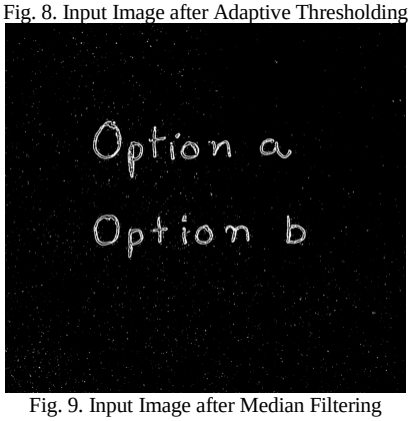

Fig. 9. Input Image after Median Filtering

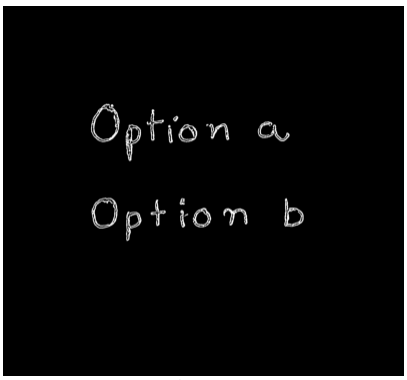

Fig. 10. Input Image after Removing Small Objects

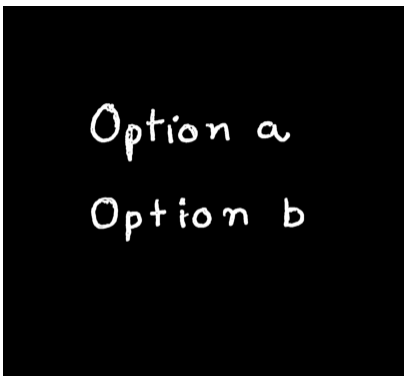

Fig. 11. Input Image after Dilation

B. Segmentation

The output of pre-processing is passed as the input for segmentation. As we can see, the horizontal histogram projection is calculated first for segmenting the image into lines. Then, the vertical histogram projection is calculated for segmenting each line into individual words.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

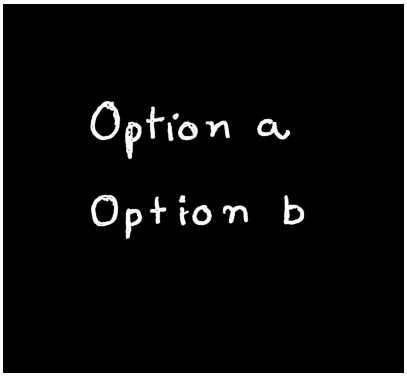

Fig. 12. Input Image

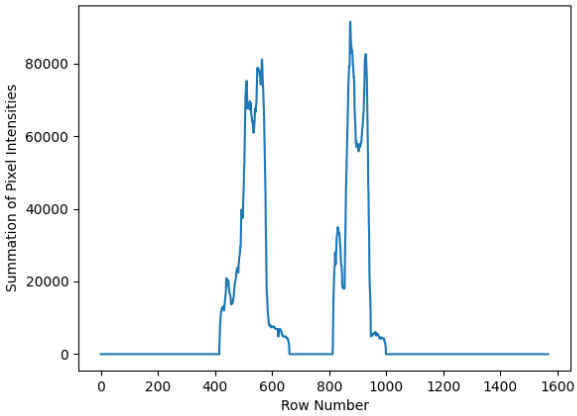

Fig. 13. Histogram for Line Segmentation

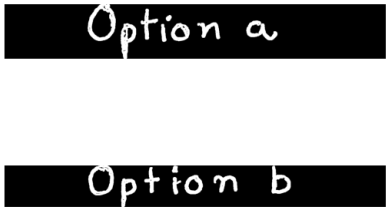

Fig. 14. Image Segmented Line Wise

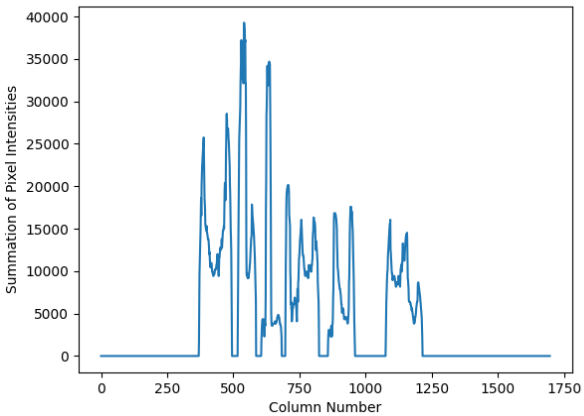

Fig. 15. Histrogram for Word Segemntation of Line 1

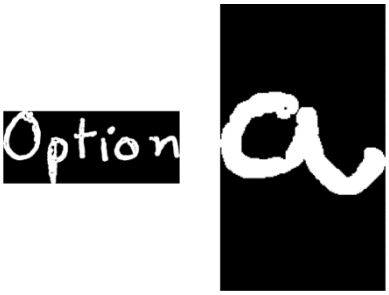

Fig. 16. Line 1 Segmented Word Wise

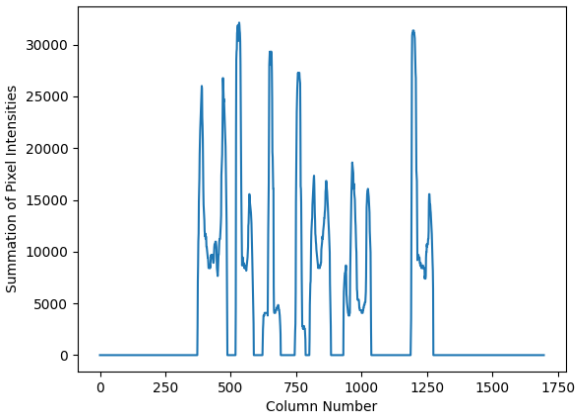

Fig. 17. Histogram for Word Segmentation of Line 2

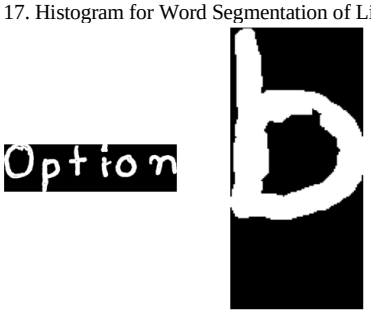

Fig. 18. Line 2 Segmented Word Wise

C. Regcogntion

For evaluating the accuracy of the model, we use the Character Error Rate (CER). The CER for a prediction is defined as the edit distance between the prediction and the ground truth divided by the number of characters in the ground truth. Therefore, the CER of the validation set would be the mean of all the individual CERs. On evaluating the validation set on the model, the CER was 9.33%. However, on passing the predicted text to a spell checker, the CER was reduced to 9.25%.

Fig. 19, shows the sample inputs to the model, and Fig. 20, shows the predicted outputs.

Fig. 19. Recognition Input

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

Fig. 20. Recognition Output

VI. CONCLUSION

Upon analyzing the results, we can see that the pre-processing, and the segmentation algorithms are performing as expected. The performance of the model is also accurate, giving a Character Error Rate of 9.33%, which was further improved to 9.25% after using a spell checker. Further improvements can be made by looking at the previous words in a sentence to determine what word would follow.

REFERENCES

[1] Gaurav Y. Tawde, Mrs. Jayashree M. Kundargi, “An Overview of Feature Extraction Techniques in OCR for Indian Scripts Focused on Offline Handwriting,” International Journal of Engineering Research and Applications, vol. 3, no. 1, pp. 919-926, February 2013.

[2] Gaurav Y. Tawde, “Optical Character Recognition for Isolated Offline Handwritten Devanagari Numerals Using Wavelets,” International Journal of Engineering Research and Applications, vol. 4, no. 2, pp. 605-611, February 2014.

[3] H. Scheidl, “Handwritten Text Recognition in Historical Documents,” Diploma Thesis, Dept. Visual Computing, Technische Universität Wien, Vienna, Austria, 2018.