Shruti Droupadkar, et. al., International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 15, Issue 3, March 2025, pp 190-195

RESEARCH ARTICLE OPEN ACCESS

Handwritten Text Recognition Using Optical Character Recognition

Shruti Droupadkar*, Amruta Gaikwad**,Prof. M.A.Chimanna***
**(Department of Electronics and Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune-411043)*
*** (Department of Electronics and Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune-411043)*
****(Department of Electronics and Computer Engineering, SCTR's Pune Institute of Computer Technology, Pune-411043)*

ABSTRACT
This paper explores various techniques for handwritten text recognition using Optical Character Recognition (OCR). Challenges such as blurred and noisy images, poor handwriting, and inconsistent stroke patterns make text detection difficult. OCR simplifies this process by automatically extracting text, eliminating the need for manual interpretation.Real-world applications include deciphering illegible medical prescriptions, recognizing students' handwriting in academic settings, and extracting text from blurred photographs. OCR has significant impact across multiple domains, including education, banking, and law enforcement, where digital text conversion is increasingly essential.We describe an image preprocessing pipeline that enhances text recognition, incorporating techniques such as resizing, affine transformations, and data augmentation. Furthermore, we utilize deep learning approaches, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Connectionist Temporal Classification (CTC) loss for sequence modeling. The model addresses challenges related to variations in text angle, lighting conditions, stroke thickness, spacing, and distortions caused by different writing instruments.
Keywords- Affine transformation, Convolutional neural networks, Data augmentation,Optical character recognition, Recurrent neural networks

---

Date of Submission: 15-03-2025 Date of acceptance: 31-03-2025

I.INTRODUCTION

Text is an essential form of data, and with the development of pen-input devices like electronic whiteboards and touch-based smartphones, handwritten text recognition (HTR) has gained popularity. Storing text in digital form not only ensures safety but also facilitates easier character recognition and retrieval.

Handwritten text recognition benefits people across various fields. The process involves detecting text in an image based on repeating stroke patterns, intersections, and shapes. Black-and-white images are particularly effective for text detection. This is followed by segmentation at different levels—first at the character level, then word level, and finally line level. Pre-existing datasets aid in predicting subsequent characters and words, improving accuracy.

Compared to isolated character recognition, HTR faces challenges in segmentation due to the cursive nature of handwriting and the varying inclination of characters.

II.LITERATURE SURVEY

Cuong Tuan Nguyen et al. (2014) presented a semi-incremental recognition method for online handwritten English text, aiming to improve recognition efficiency while maintaining accuracy. The proposed approach processed handwriting in segments, refining results as more strokes are received. This method balanced real-time processing speed and accuracy, making it suitable for interactive handwriting applications.

Bilan Zhu et al. (2009) explained the impact of improved path evaluation techniques in online handwritten Japanese text recognition. The authors enhanced the decoding process in a

recognition system by refining path evaluation, leading to improved accuracy. Their approach optimized stroke sequence matching, making recognition more robust against variations in handwriting styles.

This paper examined handwritten text recognition (HTR) using various deep learning techniques, focusing on classification performance and the computation of Connectionist Temporal Classification (CTC) loss. It compared different neural network architectures for HTR and evaluated their effectiveness in handling variations in handwriting styles. The study highlighted the role of CTC loss in improving recognition accuracy by aligning predicted sequences with ground truth labels.[3]

K. Saini et al.(2023) explored the recent advancements in handwritten text recognition, covering methods from traditional machine learning to deep learning, with a focus on data preprocessing, feature extraction, model selection, and fine-tuning. It evaluated various approaches on benchmark datasets, discussed their strengths and weaknesses, and highlighted research challenges and future directions in the field.

The paper presented a study on handwritten text recognition (HTR) using deep learning techniques, emphasizing advancements in accuracy and computational efficiency. It explored various architectures, including Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (RNNs) and Transformers for sequence modeling. The authors analyzed challenges such as irregular handwriting, varying stroke patterns, and occlusions, comparing different models based on their performance metrics. The study highlighted the role of Connectionist Temporal Classification (CTC) loss in improving text sequence alignment and discussed the potential of hybrid deep learning approaches for robust HTR systems.[8]

III. PROPOSED SYSTEM

The proposed system comprises multiple stages of image processing and text recognition. The overall architecture includes the following steps:

A. Image Preprocessing

1. Image Resizing

This ensures that the aspect ratio is maintained while fitting the image within a specified size. Neural networks require fixed-size inputs, whereas raw images have varying dimensions. Direct resizing would distort the text, making recognition less accurate. Therefore, it is crucial to resize while preserving the aspect ratio and adding padding if necessary.

$$f = \min\left(\frac{wt}{w}, \frac{ht}{h}\right) (1)$$

Here, f is the scaling factor, (wt, ht) are the target weight and height, and (w, h) are the original dimensions.
The min function ensures that the image fits within the target size while maintaining its original aspect ratio.

2. Affine transformation

An affine transformation preserves lines and angles while allowing operations such as scaling, translation, rotation, and skewing of an image. It is represented by a 2×3 matrix in computer vision.

Applications:
Handwriting normalization: Removes slant in handwritten text.
Text alignment: Adjusts text to a standard position for better OCR accuracy.

3. Padding strategy

Extra pixels are filled with white (255) since most datasets contain black text on a white background. This approach prevents stretching or shrinking, which helps avoid distortions that could negatively impact the model's accuracy.

4. Segmentation

Each primitive segment is assumed to be either a character or part of a character. The segmentation process analyzes several features, including: gap length between preceding and succeeding strokes, average stroke length in the horizontal direction, overlap between two adjacent strokes, minimum

point distance between adjacent strokes, direction between the centroids of two adjacent strokes.[1]

B. Data augmentation

Handwritten text recognition faces challenges due to variations in writing instruments (e.g., pencil, pen), backgrounds (rough or smooth surfaces), lighting conditions, and stroke thickness.

1. Photometric Augmentation

Gaussian blur is a common image processing technique that smooths an image by applying a Gaussian kernel, reducing noise and fine details. The kernel assigns weights to neighboring pixels based on a Gaussian distribution and is convolved with the image, producing a weighted average for each pixel.

The degree of blurring depends on the kernel size and standard deviation of the Gaussian distribution—the larger these values, the stronger the blur.
Here is the equation for convolution using a gaussian kernel

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2-y^2}{2\sigma^2}} \quad (2)$$

where σ is the standard deviation of the Gaussian distribution.

2. Brightness and contrast adjustments

In data augmentation, adjusting brightness and contrast enhances a model's ability to generalize by simulating various lighting conditions and color variations, making it more robust in real-world scenarios.

Let $f(i,j)$ represent the original image pixels and $g(i,j)$ the transformed pixels. The transformation is given by:
$g(i,j) = \alpha \cdot f(i,j) + \beta$ (3)
where αcontrols contrast and β controls brightness.

C. Text recognition

1. Convolutional Neural networks (CNNs)

CNNs (Convolutional Neural Networks) extract spatial features from input images, capturing stroke patterns, curves, textures, and fundamental shapes that define characters. They process both grayscale and RGB images of handwritten text, converting raw images into meaningful feature representations.[2]

A CNN consists of three key layers: the convolutional layer, pooling layer, and fully connected layer, each increasing in complexity. The convolutional layer performs most of the data processing, where a kernel moves across the input image, computing the dot product between pixel values and the filter to extract features. The pooling layer reduces spatial dimensions while preserving essential information by summarizing feature regions. The fully connected layer aggregates extracted features to learn global relationships and make predictions.[3]

Mathematically, convolution is defined as

$$Y(i,j) = \sum_m \sum_n X(i-m, j-n) \cdot W(m,n) + b \quad (4)$$

where X is the input image, W is the filter, b is bias and Y is the feature map. Further the spatial dimensions should be reduced while preserving important features

$$y = \max X(i,j) \quad (5)$$

Then extracted features are passed to the recurrent network for sequence modelling.Thus CNN largely helps in extracting invariant features, reducing load for the next stage.

2. Recurrent neural networks (RNNs)

Unlike traditional feedforward neural networks, RNNs incorporate recurrent connections, allowing information to be passed across time steps.

The core principle of RNNs is to use the output of the previous time step as an input for the current time step. This enables the network to: establish dependencies between sequential data, handle variable-length sequences, capture temporal correlations and contextual relationships.[4]

The hidden state in an RNN functions as a memory unit, updating at each time step and propagating information to subsequent layers or time steps. This memory propagation allows RNNs to track previous data and adjust outputs accordingly.

The calculation formula for the internal hidden state of an RNN is as follows:

$$h_t = \tanh(w_t h_{t-1} + w_x x_t + b) \quad (6)$$

where $h_t$represents the hidden state at time step t, $w_t$the weight matrix for the hidden state, $w_x$the weight matrix for the input x_t , b the bias vector.

3. Long short term memory(LSTM)

Long Short-Term Memory (LSTM) networks enhance traditional RNNs by incorporating a memory cell and three types of gates—input, forget, and output—to regulate the flow of information. These gates allow the network to selectively retain or discard information, enabling LSTMs to learn and remember long-term dependencies in sequential data, which is crucial for understanding text meaning. The primary advantage of LSTMs is their ability to mitigate the vanishing gradient problem, a limitation in standard RNNs that hinders learning over long sequences. This allows LSTMs to effectively process long-range dependencies in data.

$$f_t = \sigma(\omega_f x_t + U_f h_{t-1} + b_f) \quad (7)$$

$$i_t = \sigma(\omega_i x_t + U_f h_{t-1} + b_i) \quad (8)$$

$$c_t = \tanh(\omega_c x_t + U_c h_{t-1} + b) \quad (9)$$

$$c_t = f_t c_{t-1} + i_t c_t \quad (10)$$

$$o_t = \sigma(\omega_o x_t + U_o h_{t-1} + b_o) \quad (11)$$

$$h_t = o_t \tanh(c_t) \quad (12)$$

Forget gate f_t decides what information to discard. Input gate i_t decides what new information to store. Output gate o_t controls what information to pass forward.
Here c_t is the cell state at time step t. U_f , U_i , U_o , U_c are weight matrices associated with hidden states. b_f , b_i , b_o , $b_c$are bias terms to add flexibility to the transformation.

An important extension of LSTMs is Bidirectional LSTM (BiLSTM), which processes sequences in both forward and backward directions. This enables the model to capture context from both past and future time steps, making it highly effective for tasks where understanding surrounding context is crucial.

BiLSTM is particularly beneficial in cursive handwriting recognition, where characters are connected, and the meaning of a character may depend on both preceding and succeeding characters. By leveraging information from both directions, BiLSTMs significantly improve recognition accuracy.

4. Connectionist temporal classification(CTC)

After RNN outputs the probability distribution over characters at each step,the CTC decodes it to the most likely sequence.
In text recognition, alignment refers to the process of sequence-to-sequence mapping, where the input sequence X= [x1,x2,…,xn] (handwritten image features) is mapped to the corresponding positions in the output sequence Y= [y1,y2,…,ym] (text transcription). Since the input and output sequences often differ in length, determining the correct character-to-image feature correspondence becomes challenging.

The Connectionist Temporal Classification (CTC) loss function eliminates the need for explicit alignment between the input and output. Instead, it computes the probabilities of all possible alignments and assigns the highest probability to the most likely transcription.[3]

Forward variable $\alpha_{s,t}$ gives the total probability of the sequence $seq[1:s]$ up to a particular timestep t while the backward variable $\beta_{s,t}$ gives the probability of the remaining sequence $seq[s:S]$ at timestep t.

The joint probability $\gamma_{s,t}$ of the sequence at every timestep is computed by:

$$\gamma_{s,t} = \alpha_{s,t}\beta_{s,t} \quad (13)$$

The total probability of all paths through a token $seq[s]$ at timestep t is:

$$P_{(seq_t,t)} = \sum_{s=0}^{S} \frac{\alpha_{s,t}\beta_{s,t}}{y_{s,t}} \quad (14)$$

Total loss is then

$$l = \sum_{t=0}^{T-1} \log P_{(seq_t,t)} \quad (15)$$

Decoding methods of CTC

1. Best path decoding,which selects the most probable character sequence without considering future context.
2. Beam search decoding,this explores multiple paths and selects the most likely sequence.
3. Lexicon-based decoding, uses dictionaries to improve recognition accuracy.

5. Batch recognition

To obtain a high recognition rate,it is best to recognize text after the whole text is completed,then we would have full information. Batch recognition is based on the fact that users are writing while thinking.So users do not need recognition results when writing and only need recognized text when they break writing.

IV.RESULTS

Here are some results after applying the concepts discussed above:

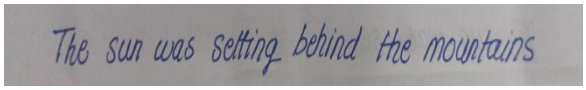

Fig.1 ground text: The sun was setting behind the mountains

Fig.2 result for Fig.1

Different results were seen in the following test cases.

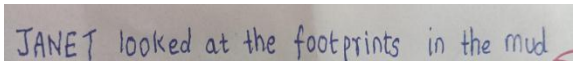

Fig.3 ground text: JANET looked at the footprints in the mud

Fig.4 result for Fig.3

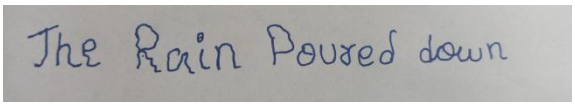

Fig.5 ground text: The Rain Poured down

Fig.6 result for Fig.5

Fig.7 ground text: Laughter filled the room

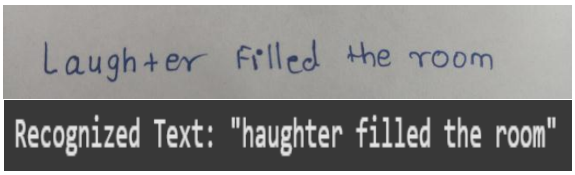

Fig.8 result for Fig.7.

Metric	Existing model	Current model
CER	19.08%	15.52%
WER	24.56%	20.91%
Word accuracy	74.67%	82.33%
Average processing time	0.2s	0.4s

here, CER: character error rate
WER: word error rate

V.CONCLUSION

This paper explored various methods for handwritten text recognition, including preprocessing techniques and neural network architectures like CNNs, RNNs, and CTC-based decoding. These approaches enhance text segmentation and recognition accuracy.

However, challenges remain, such as difficulty with multi-line text and varying

handwriting styles. Future improvements, like transformer-based models, could address these issues.

This study provides a foundation for OCR research, with potential applications in digitizing historical texts, real-time document processing, and deciphering lost scripts.

REFERENCES

[1] C. T. Nguyen, B. Zhu and M. Nakagawa, "A Semi-incremental Recognition Method for On Line Handwritten English Text," 2014 14th International Conference on Frontiers in Handwriting Recognition, Hersonissos, Greece, 2014, pp. 234-239, doi: 10.1109/ICFHR.2014.47.

[2] B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa, "Effect of Improved Path Evaluation for On-line Handwritten Japanese Text Recognition," International Conference on Document Analysis and Recognition, pp. 516–520, Jul. 2009, doi: 10.1109/ICDAR.2009.215.

[3] Ratnam Dodda, S Balakrishna Reddy, Azmera Chandu Naik, Venugopal Gaddam, "A study on handwritten text recognition classification using diverse deep learning techniques and computation of ctc loss," CVR Journal of Science and Technology,Volume 26,June 2024,E-ISSN 2581-7957, P-ISSN 2277-3916.

[4] W. Li and K. L. E. Law, "Deep Learning Models for Time Series Forecasting: A Review," in IEEE Access, vol. 12, pp. 92306-92327, 2024, doi: 10.1109/ACCESS.2024.3422528.

[5] Utsav Poudel, Aayush Man Regmi, Zoran Stamenkovic, "Applicability of ocr engines for text recognition in vehicle number plates,receipts and handwriting,"Journal of Circuits Systems and Computers, November 2023,DOI: 10.1142/S0218126623503218.

[6] K. Saini, K. Sharma, A. Agarwal, K. Jayan and D. Dev, "Handwritten Text Recognition Using Machine Learning," 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), Ghaziabad, India, 2023, pp. 121-124, doi: 10.1109/ICSEIET58677.2023.10303304.

[7] Rafael C.Gonzalez, Richard E.Woods, Digital image processing (fourth edition) (Pearson,330 Hudson Street, New York, NY 10013)

[8] L. Navya, M. F. Ali, K. P. Sai, K. Shyam and A. Ramesh, "Handwritten Text Recognition Using Deep Learning Techniques," 2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS), Kanjirapally,India, 2023, pp. 1-5, doi: 10.1109/AICERA/ICIS59538.2023.10420040.

[9] R. R. Chand, M. Farik and N. A. Sharma, "Digital Image Processing Using Noise Removal Technique: A Non-Linear Approach," 2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2022, pp. 1-5, doi: 10.1109/CSDE56538.2022.10089258.

[10] B. A. Kumar, S. Tamilarasan, A. Kiran, B. Tejaswi, M. G. Sree and M. Dasarla, "Optical Character Recognition Technology using Machine Learning," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-4, doi: 10.1109/ICCCI56745.2023.10128201.