Handwritten Character Recognition Using Deep-Learning

Mohsina Al Vra
Student, Department of ECE
University BDT College of Engineering
Davangere, Karnataka

Dr. Lakshman Naik R
Professor, Department of ECE
University BDT College of Engineering
Davangere, Karnataka

***Abstract*— Character recognition is a rapidly growing field within computer vision. One of its essential components is handwriting recognition, which involves the capability of a machine to understand and interpret handwritten input from various sources, such as paper documents, photographs, and touch screen devices. To facilitate this, an application has been built that can take an image of handwritten text and convert it into digital text, making it editable and searchable. The process involves training a character recognition model using Python and various libraries, including OpenCV, NumPy, Pandas, Pillow, and TensorFlow. Convolutional Neural Networks (CNN) are commonly employed as the underlying algorithm for this task, as they have proven to be highly effective in image recognition tasks.**

Keywords—HCR(Handwritten Characters Recognition), CNN(convolutional Neural Network), Deep Learning, TF(TensorFlow), Python, OpenCV

I. INTRODUCTION

In the present-day world, Artificial Intelligence (AI) is often referred to as the new Electricity, given its transformative impact on various industries. Advancements in AI and deep learning are rapidly occurring, shaping new possibilities every day. Within the realm of research, Handwriting Recognition stands as an active area, where deep neural networks play a pivotal role. Although recognizing handwriting comes naturally to humans, it poses a formidable challenge for computers. Handwriting recognition systems are broadly classified into two categories: Online and Offline methods. Both approaches explore different techniques to decipher and understand handwritten input, enabling a wide range of applications across various domains. As the field of AI continues to evolve, Handwriting Recognition holds immense promise, leading to enhanced automation, efficiency, and accessibility in numerous real-world scenarios.Character Recognition (CR) remains a vibrant and challenging research area, driven by its diverse applications. In this project, the main focus lies in offline recognition of handwritten English characters, achieved through the detection of individual characters and converting them into digital, readable text offline. Handwritten word recognition can be broadly categorized into two classes: holistic and segmentation-based approaches. The holistic approach is particularly suitable for recognizing a limited-size vocabulary. It involves extracting global features from the entire word image and using them for recognition on the other hand, segmentation-based strategies start from either the stroke or character level and progressively build towards forming meaningful words. By breaking down the word into simpler isolated characters or strokes through segmentation, this approach opens the door to recognizing words from an unlimited vocabulary. The segmentation-based strategy offers a more flexible and scalable solution, as the system can handle a wide variety of words without being limited to a predefined set of characters. As a result, this methodology paves the way for tackling more complex and real-world handwriting recognition tasks, making it a significant area of interest for researchers and developers alike.

II. RELATED WORKS

Utilizing fuzzy logic, a character recognition system has been developed [1]. They have designed their character recognition system using fuzzy logic, which can be implemented on a VLSI (Very Large-Scale Integration) structure. This innovative system exhibits robustness against distortion and shift variations. Notably, they have incorporated a Hamming neural network into their solution, further enhancing the system's accuracy and performance in recognizing characters.

A cutting-edge technique for recognizing handwritten Tamil characters using Neural Networks has been developed [2]. The researchers utilized a Kohonen Self-Organizing Map (SOM), an unsupervised neural network, as a pivotal component in their innovative system. This system is designed not only for the recognition of handwritten Tamil characters but also extends its capability to recognize other Indic languages. The results produced by their system are notably close to accurate; however, occasional errors may occur when dealing with poorly segmented handwritten characters.

One of the authors introduced a distinctive approach for person authentication based on their handwriting [3]. In their system, the author employed a Multi-layer feedforward neural network. Through this research paper, the author posited that the height and width of a handwritten alphabet are individualistic to each person, leading to a unique characteristic. Consequently, the author devised a novel method for recognizing and identifying individuals based on their handwriting.

An innovative approach to handwritten character recognition has been developed, eliminating the need for feature extraction [4]. Their system has been implemented using Matlab, featuring a feedforward neural network with backpropagation.

One of the authors has proposed a distinctive method for handwriting recognition [5]. In their system, they utilized a Self-Organizing Map (SOM) for feature extraction and a Recurrent Neural Network (RNN) for learning. The authors conducted their experiments on the recognition of Japanese characters.

III. METHODOLOGY

There are five major steps included in this project for detection process.

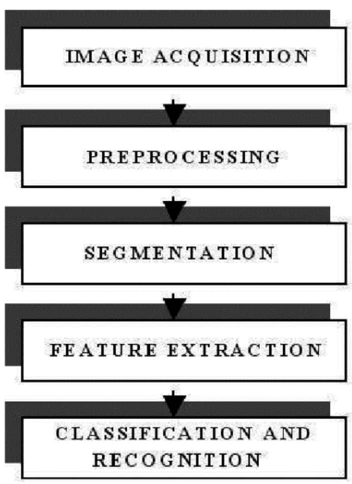


Figure 1: Block Diagram

In this section, the proposed recognition system is described. A typical handwriting recognition system consists of pre-processing, segmentation, feature extraction, classification and recognition stages. The schematic diagram of the proposed recognition system is shown in Fig.1. An information stream outline is a graphical portrayal of the "stream" of information through a data framework, displaying its procedure perspectives. Morphological processing is also included where we investigate the effect of erosion and dilate methods. Erosion is a morphological transformation that combines two sets using the vector subtraction of set elements. Dilation is a pseudo-inverse of the erosion. Instead of combining two sets using vector subtraction of set elements, it uses their addition.

IMAGE ACQUISITION: An action of retrieving image from an external source for further processing where the taken image can be through camera or some scanner . The image should have a specific format such as JPEG, PNG, BMP etc. The input captured may be in gray, color or binary form.

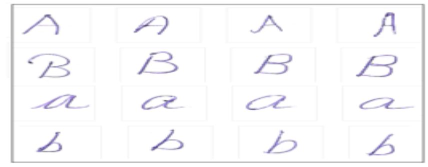


Figure 2: Sample DataSet

PRE PROCESSING: Pre-processing is a series of operations performed on the scanned input image that is used to improve the quality of image and boost them for analysis and further processing. It includes noise reduction, image resizing etc.

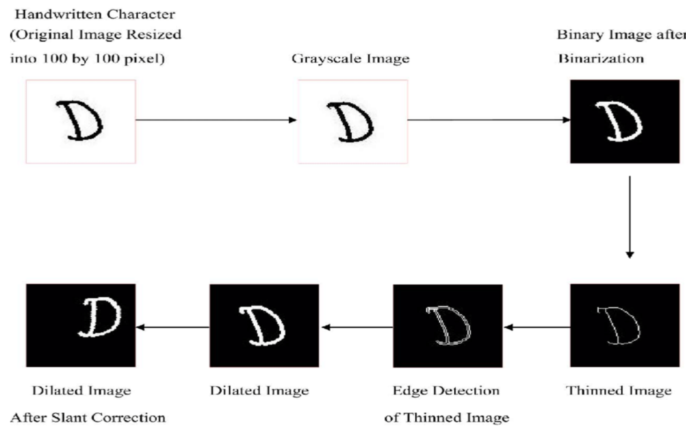


Figure 2.1: Steps of Preprocessing

SEGMENTATION: In Character Recognition techniques, the Segmentation is the most important process. Segmentation is done to make the separation between the individual characters of an image.

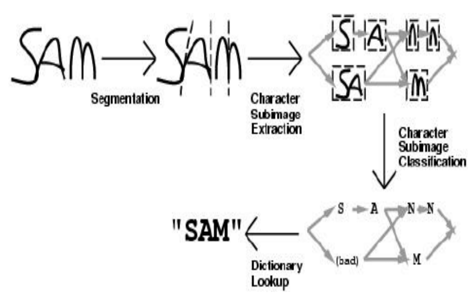


Figure 2.2: Breakdown of Characters

FEATURE EXTRACTION: Feature extraction is the process to retrieve the important data from the raw data. The important data means on the basis that the characters can be represented accurately. The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements.

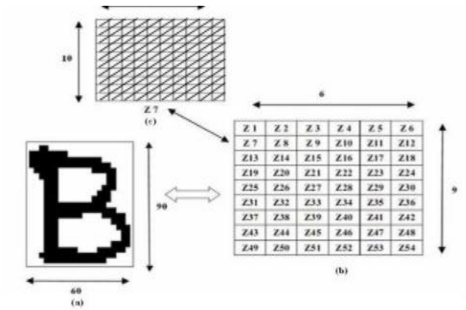


Figure 2.3: Extraction of Charaters

CLASSIFICATION AND RECOGNITION: The classification stage is the decision making part of a recognition system and it uses the features extracted in the previous stages and gives the final result.

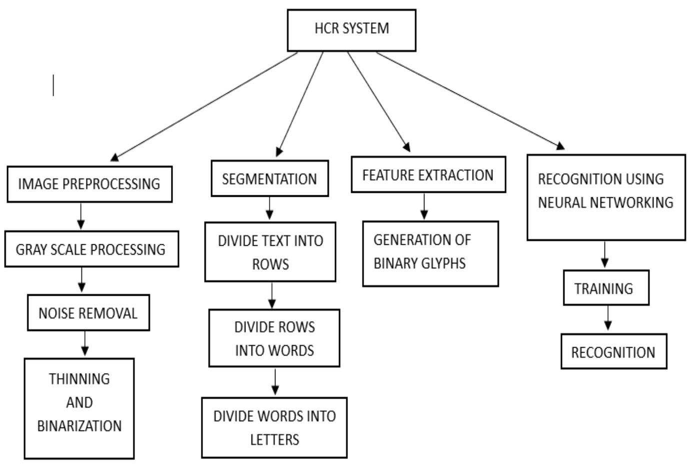


Figure 2.4: Implementation Process

Algorithm Used:

A.) CONVOLUTION NEURAL NETWORK (CNN): This has emerged as an effective tool for analyzing big data which uses complex algorithms and artificial neural networks to train machines/computers so that they can learn from experience, classify and recognize data/images just like a human brain does. Four main types of layers used to build a CNN are: Input Layer, Convolution Layer, Pooling Layer, Fully Connected Layer.

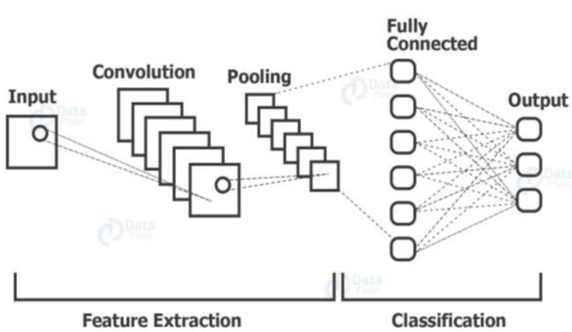


Figure 3: CNN Processing

Convolution is applied to input data to filter the information and produce a feature map. The pooling layer is used to reduced the spatial dimensions of the feature map while preserving the previous layers. Fully connected layer is where all the nodes in one layer are connected to the nodes in the next layer. The neural network architecture can be summarized as follows:

1. Input Layer: This initial layer receives and holds the raw pixel values of the input image. It serves as the entry point for visual information into the network.

2. Convolutional Layer: Positioned after the Input Layer, the Convolutional Layer receives the outputs from the previous neuron layer that is connected to the input regions. In this layer, a specified number of filters is applied to the input data. Each filter acts as a 5x5 sliding window, traversing the input data to extract relevant features. The output is obtained by selecting the pixel with the maximum intensity within each filter's receptive field.

3. Pooling Layer: Following the Convolutional Layer, the Pooling Layer performs a down-sampling operation along the spatial dimensions (width and height). This process reduces the volume of the data, retaining important features while reducing computational complexity.

4. Fully Connected Layer: The final stage of the network, the Fully Connected Layer, computes the scores for various classes based on the learned features. It identifies the class with the maximum score, determining the most likely class corresponding to the input digits. This layer is crucial for classification tasks.

Data Collection and Testing: In order to work with the MNIST dataset for handwritten digit recognition, we begin by importing and installing the necessary libraries. Next, we read the MNIST dataset files using Python, fetching the training and testing images as well as their corresponding labels. The dataset is automatically saved in a folder named "MNIST_data" within the same directory as the external image and code files. By following these steps, we can access the dataset conveniently for training and evaluating our recognition model. In order to work with the MNIST dataset for handwritten digit recognition, we begin by importing and installing the necessary libraries. Next, we read the MNIST dataset files using Python, fetching the training and testing images as well as their corresponding labels. The dataset is automatically saved in a folder named "MNIST_data" within the same directory as the external image and code files. By following these steps, we can access the dataset conveniently for training and evaluating our recognition model. For testing the handwritten text recognition model, we need to set up a dedicated test server and network environment. Additionally, a test PC setup is essential to facilitate the evaluation process. To create the test data for the test environment, we can leverage the capabilities of Visual Studio (VS) code, a versatile integrated development environment. The test server and network setup ensures that the model can be deployed and assessed under controlled conditions. This enables us to evaluate its performance and accuracy in recognizing handwritten digits. Alongside, the test PC setup provides a platform to execute the recognition model and collect relevant performance metrics. To carry out the evaluation, we generate test data in the VS code, simulating real-world scenarios of handwritten digits. This test data will be used to assess the model's ability to accurately recognize and classify the input digits. By creating a comprehensive test environment and test data, we can gain valuable insights into the effectiveness and reliability of our handwritten characters recognition system.

IV. RESULTS and DISCUSSION

In this project we have given image as an input then it predictsthe output by loading the model which is already previously created and saved.

<u>**TEST CASE 1:**</u> **Detection conducted using CNN layering:** Includes image classifier which carries out image recognition and classification with probability rate of detection using emnist data set.

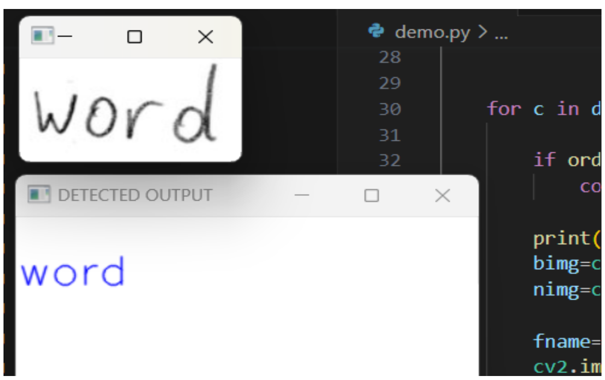


Figure 4: Input and Detected Output Windows

Figure 4.1: Segmented Images of Recognized Characters

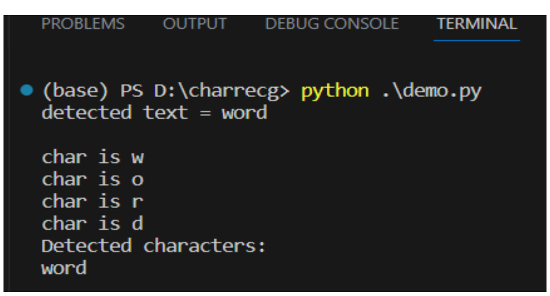


Figure 4.2: Result displayed on python terminal window

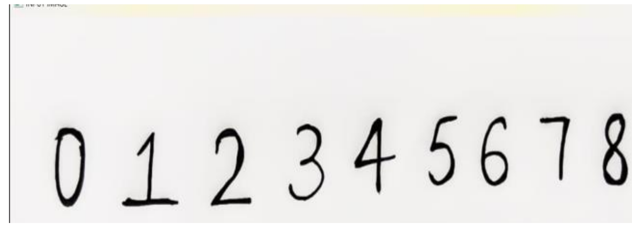


Figure 4.3: Input Digit Image

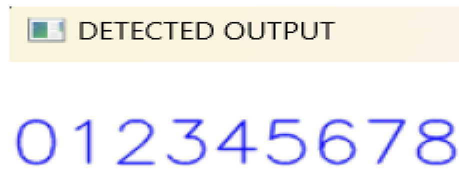


Figure 4.4: Detected output digits

<u>**TEST CASE 2:**</u> This includes the neural network which is built using the data set collected from Kaggle. It uses GPU integrated notebooks. In terminal window we have to type python along with file name where code is saved with an extension .py and run it.

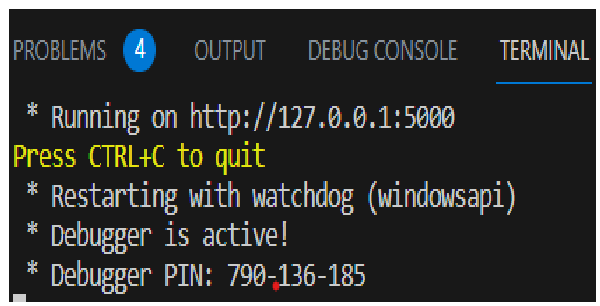
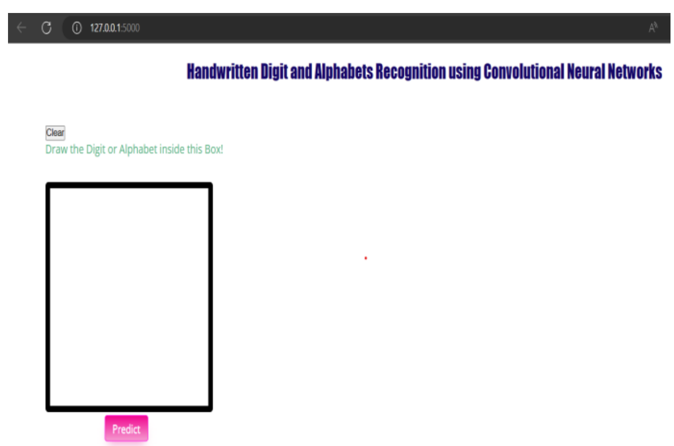


Figure 5: Link for directing external window

Click on the http link from which you will be directed to the web page server where we can write digitally and that text can be recognized.



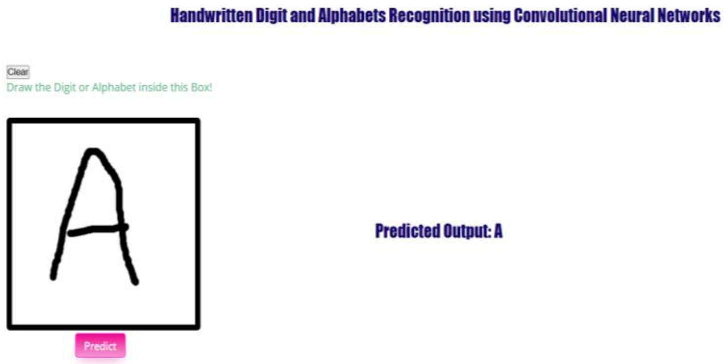


Figure 5.1: Web Page for Detection of Characters

From above results we can say that a model trained using EMNIST dataset is more convenient which gives an accuracy rate upto 95% of exact detection of characters obtained from the input image. Whereas the neural network built using the dataset collected from Kaggle is less convenient with an accuracy rate of 75% of characters recognition when compaired to emnist dataset. But both the applications are good to for detection of handwritten characters digitally and offline.

V. CONCLUSION AND FUTURE SCOPE

In the proposed project we have worked on handwritten recognition of English characters. Use of some statistical features and geometric features through neural network will provided better recognition result of English characters. This work will be helpful to the researchers for the work towards other script. The accurate recognition is directly depending on the nature of the material to be read and by its quality. Current research is not directly concern to the characters, but also words and phrases. Managing our algorithms with proper training, evaluation other step wise process leads to successful output of system with better efficiency. The algorithm employed in this project offers a perfect blend of efficiency and effectiveness, ensuring reliable results for character recognition. The system demonstrates its best accuracy when processing text with minimal noise. It is important to note that the accuracy of the system is directly influenced by the quality and size of the dataset used for training. By increasing the dataset size and diversity, we can further enhance the accuracy and robustness of the recognition system, catering to a wider range of handwritten inputs with improved precision.

Many regional languages throughout world have different writing styles which can be recognized with HCR systems using proper algorithm and strategies. In the future, we envision expanding this study to a larger scale, encompassing diverse embedding models applied to a wide variety of datasets. As technology continues to advance, the traditional paper and pen writing methods are likely to be replaced, with touch pads becoming a common medium for writing. In this scenario, we aim to develop sophisticated inbuilt software capable of automatically detecting and recognizing handwritten text on touch pads, efficiently converting it into digital text. This transformation will simplify the process of searching and understanding content, opening up new possibilities for seamless communication and data analysis in the digital age.

VI. REFERENCES

1. Wei Lu, Zhijian Li, Bingxue Shi. "Handwritten Digits Recognition with Neural Networks and Fuzzy Logic." In IEEE International Conference on Neural Networks, 1995.

2. P. Banumathi, Dr. G. M. Nasira. "Handwritten Tamil Character Recognition using Artificial Neural Networks." In 2011 International Conference on Process Automation, Control and Computing.

3. B. V. S. Murthy. "Handwriting Recognition Using Supervised Neural Networks." In International Joint Conference on Neural Networks, 1999. IJCNN '99.

4. J. Pradeep, E. Srinivasan, S. Himavathi. "Neural Network based Handwritten Character Recognition system without feature extraction." In International Conference on Computer, Communication and Electrical Technology ICCCET 2011.

5. Shun Nishide, Hiroshi G. Okuno, Tetsuya Ogata, Jun Tani. "Handwriting Prediction Based Character Recognition using Recurrent Neural Network." In 2011 IEEE International Conference on Systems, Man, and Cybernetics.

6. T. Kohonen. Self-Organization and Associative Memory. 2nd Ed. New York, Springer, 1988.

7. Y. Yamashita and J. Tani. Emergence of Functional Hierarchy in a Multiple Timescales Recurrent Neural Network Model: A Humanoid Robot Experiment. PLoS Computational Biology, Vol. 4, e1000220, 2008.

8. Lei Wang, Lei Zhang, Yanqing Ma. "Gstreamer Accomplish Video capture and coding with PyGI in Python language." In 2017 24th Asia-Pacific Software Engineering Conference (APSEC).

9. Rahul R. Palekar, Sushant U. Parab, Dhrumil P. Parikh, Prof. Vijaya N. Kamble. "Real Time License Plate Detection Using OpenCV and Tesseract." In International Conference on Communication and Signal Processing.

10. "OpenCV." https://en.wikipedia.org, [Online] Available: https://en.wikipedia.org/wiki/OpenCV/. [Accessed 05 March 2018].