iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41156

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

Hand Witten Text Recognition using Deep Learning

Makineni Surya Tej[1], Tungala Veerendra Saradhi[2], Mallempati Spandana[3], Vundru Savya[4]
[1, 2, 3, 4]*UG Student, Department of CSE, PVPSIT-Vijayawada, Andhra Pradesh, India*

Abstract: ***The dependence on people over technology has never been higher, to the point where deep learning and machine learning algorithms can conduct anything from object detection in images to adding sound to silent movies. Similarly, handwritten text recognition is a major field of research and development with a plethora of possibilities. Using this method, the system is taught to look for similarities as well as differences between diverse handwriting samples. This program turns a picture of a handwritten transcription into digital text. Convolution Neural Network (CNN) with the help of IAM databases are used on the proposed work to recognize the handwritten text. Our main goal is to integrate the aforementioned models, with their release times, in order to create the best text recognition possible, model.***

Keywords: Handwritten Text Recognition (HTR), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Connectionist Temporal Classification (CTC)

I. INTRODUCTION

Using this method, the system is taught to look for similarities as well as differences between diverse handwriting samples. This program turns a picture of a handwritten transcription into digital text. Clearly, we used neural network models to achieve handwritten digital identification with the help of IAM databases in this article. Our main goal is to integrate the aforementioned models, with their release times, in order to create the best text recognition possible, model. One of the new features in computer recognition is character recognition. People's abilities are far more powerful than their ability to see any object or business. The system has a hard time detecting the text. During text recognition, the input picture is processed, features are extracted, and a classification schema is created, teaching the system to recognize the text.

In [1] authors implemented various pre-processing techniques. They proposed an offline character recognition based on ANN model. Although usage of several techniques did not improve the accuracy rate for handwritten text recognition i.e. we are unable to reach 100% accuracy. In [2] authors implemented a system and achieved an accuracy of 79.9% and an accuracy of 83%.8 with the hybrid-MMI approach. In [3] authors implemented their method with 89.6% accuracy for handwritten Devanagari numerals. In [4] implemented a method using a fuzzy set with an accuracy of 90%.

II. PROPOSED DESIGN

A. Proposed Text Recognition System

The proposed systems use scanned images of handwriting, as shown in the figure 1.0. Our proposed model uses word images as input. In word pictures, NN training is achievable on the CPU as the input layer (so all opposing layers) are usually kept small (Yes, the GPU could be better). The minimum requirement for HTR implementation is TF.



Figure 1: Image taken from the dataset converted into digital text

Figure 1 shows the transformation of image text to digital text.

B. Model Overview

Figure 2.0 shows the model used in the proposed work.
It consists of three layers as shown inFigure 2.1

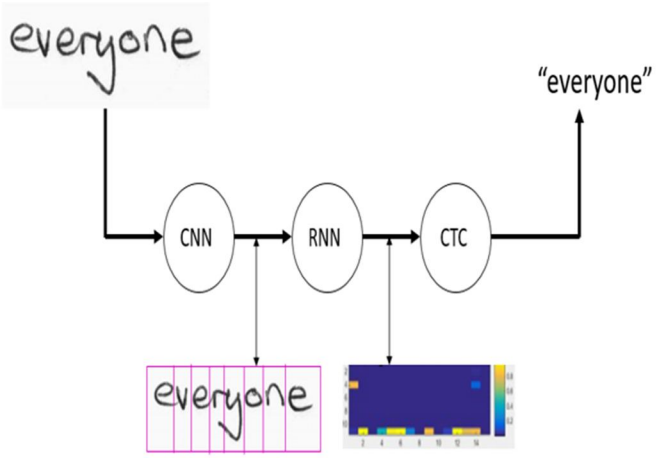


Figure 2.1: Sample Overview/Flow of the proposed model

To get started, we must first process the images to remove the noise.
NN is a function that transfers an image to a sequence c1, c2 .. as follows .

$$NN : M \rightarrow (C_1, C_2, ..., C_n)$$
$$(W \times H) \rightarrow (0 \leq n \leq L)$$

C. Operations

The CNN layers have three procedures in each tier.
A 55% filter is employed in the first two layers of the convolution operation, and a 33% filter is utilized in the last three levels of the input.
The non-linear function is used.
A pooling layer aggregates picture areas
Finally a reduced version of the input. Is produced
Input: It's a 128 x 32 pixel gray-value picture. Because the images in the collection seldom have this precise dimension, we enlarge them Figure depicts this procedure.

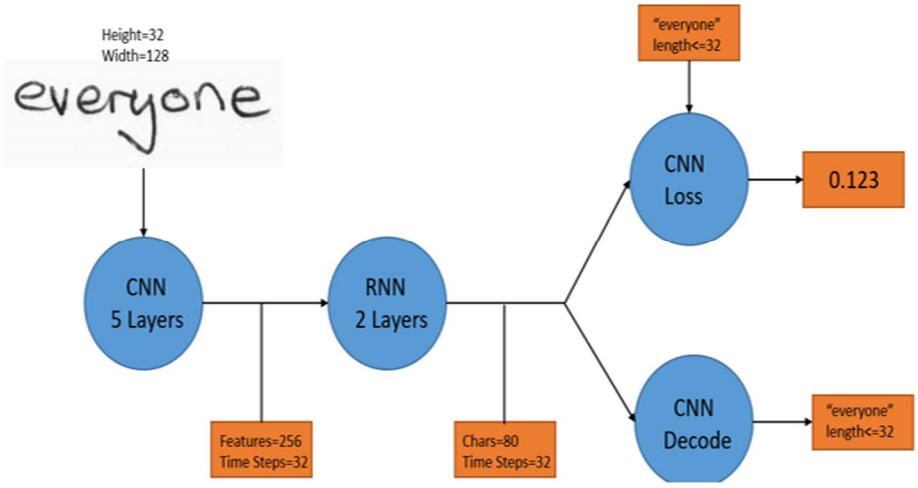


Figure 2.2: Circles specify NN functionality and rectangles specify data flow via NN

In Figure 3 the image with the maximum size is measured to fit the target 128x32 size and the remaining space is filled in white.

D. Implementation

1) *DataLoader.py:* This python module's main outcome is that it loads the IAM dataset and has other features namely splitting the dataset (95% - training set and 5% - validation set), getting the iterator information, checking and getting the element, and getting the image. In its implementation there are several functions and some of them are __init__(), validate_set(), train_set(), get_img(), get_iterator() etc.
2) *Preprocessor.py:* This python module's main outcome is that it resizes the input image to target size, by applying data augmentation. Create an image of a text line by pasting multiple word images into an image. In its implementation there are several functions and some of them are __init__(), truncate_label(), simulate_text(), process_text() etc.
3) *Model.py:* This python module's main outcome is that we are creating a Neural Network model by using three different algorithms namely Convolutional Neural Network (CNN), Recurrent Neural Network, and Connectionist Temporal Classification. Some other features which are provided are the initialization of the TensorFlow, saving the model, dumping of the neural network, etc. In its implementation there are several functions and some of them are __init__(), setup_cnn() setup_rnn(), setup_ctc(), setup_tf() etc.
4) *Main.py:* This python module's main outcome is that it takes command-line arguments which act as input for all the above three modules, that is we create and train our neural network model and test it with sample scanned text images. This module act as the main interface for everything. In its implementation there are several functions and some of them are __init__(), set_summary() train(), validate(), infer() etc.

a) *CNN:* Created a kernel of size k x k for each CNN layer and applied the convolution results using the RELU operation and repeated for all levels.
b) *RNN:* We built a two layers each consisting of 256 units. Upon which we built a bidirectional layer. The output layer consists sequences of size 32 x 256 forward and backward. After concatenating we obtained a 32 x 512 sequence. The final output is sent to the CTC layer.
c) *CTC:* The CTC is provided with an RNN exit matrix and a basic truth text while training NN, and calculates the loss rate. The CTC is simply assigned a matrix and divides it into final text while directing. The true text is stored as a sparse tensor at the bottom. Both CTC procedures must be supplied the length of the input sequences.

E. Enhancing the Model

To improve the accuracy we increase the size of the database and remove the compilation style images in the input image. At the end we increased the input size. No academic titles or descriptions of academic positions should be included in the addresses.
The affiliations should consist of the author's institution, town/city, and country.

III. THE PROPOSED NETWORK'S ARCHITECTURE

The proposed network comprises several levels, as seen in the diagram below:

A. Convolutional Neural Network Layer

CNN uses highly efficient structures to interpret 2D pictures in order to mimic human visual processing. It can also successfully learn how to extract and abstract 2D characteristics. In particular, CNN's max-pooling layer is quite good at absorbing shape fluctuations.

B. Recurrent Neural Network Layer

The RNN reduces the severity of ascending parameters and memorizing each previous output by sending each output as a subsequent hidden layer.
The formula for calculating current state:

$$h_t = f(h_{t-1}, x_t)$$

The formula for applying activation function:

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

Formula for calculating output:

$$y_t = W_{hy} h_t$$

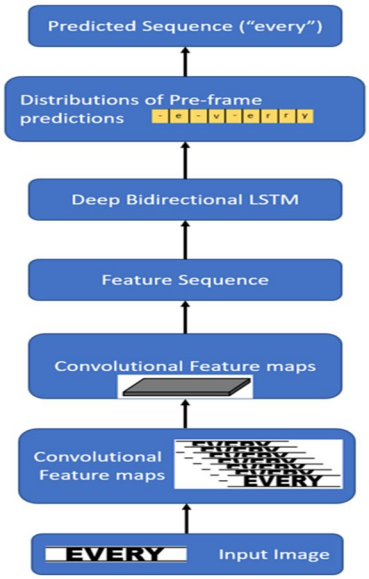


Figure 4: Architecture of the proposed model

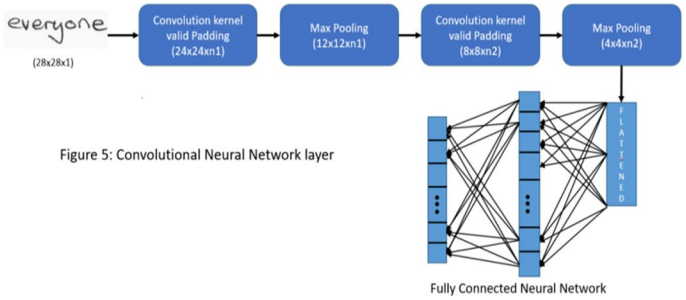


Figure 5: Convolutional Neural Network layer

Figure 5: Convolutional Neural Network Layer

C. Connectionist Temporal Classification Layer

CTC operates by adding up the chances of all conceivable alignments between the input and the label.

The blank token is a means to get around this. It has no meaning and is simply eliminated before the final word output is generated. Each input step can now be assigned a personality or a blank token by the algorithm. Initially for each input sequence, the CTC assigns each character with a probability and repeats with no blank token between them are merged and finally, the blank token is discarded. The probability label may then be assigned to the input by the CTC network. This algorithm do not need input and output alignment. It calculates the likelihood of an output given an input by adding the probabilities of all conceivable alignments. We figured out how the loss function is generated finally. Consider a basic way to motivating the particular sort of CTC alignments. Let's have a look at an example

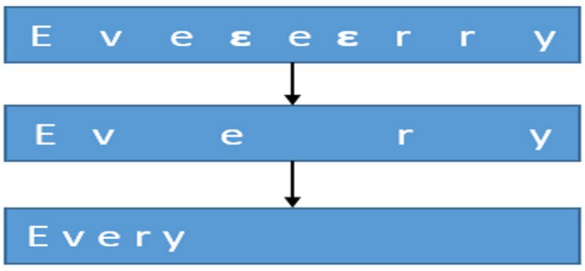


Figure 6: Collapsing of the repeats

D. Summary of Dataset

Non-printed manuscript forms were filmed at a resolution of 300dpi and saved as PNG images with 256 gray levels on the website. IAM data set is used in the proposed work. The information in the tables above pertains to one training, one testing, and two validation sets. There are several photos of the same type with a certain dimension in the data collection along with a label. It includes the image as well as the text that accompanies it. It starts with an image and then moves on to the specific text inside it.

IV. RESULTS

We gave a picture as an input as show in figure 1.0 and and the model predicts the output as shown in figure 7.4 using the trained knowledge.



Figure 7.1: Sample Input Image 1

The input supplied to the neural network to anticipate the solution is shown in figure 7.1.

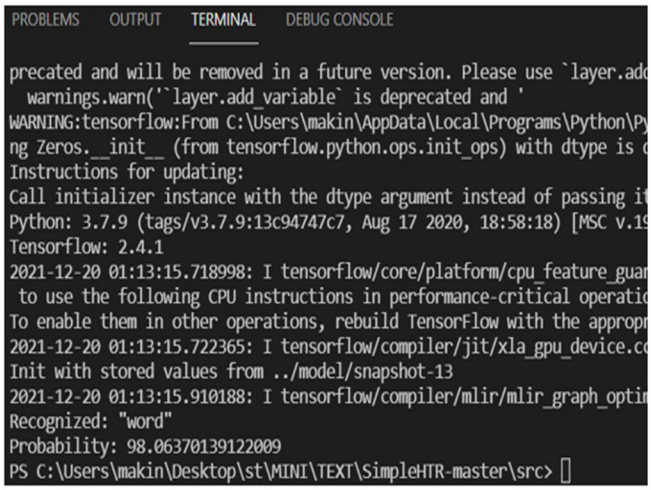


Figure 7.2: Sample Output for the input image 1

Let us test it with another image which is figure 7.3



Figure 7.3: Sample Input Image 2

For the image in figure 7.3 the output generated is shown in figure 7.4

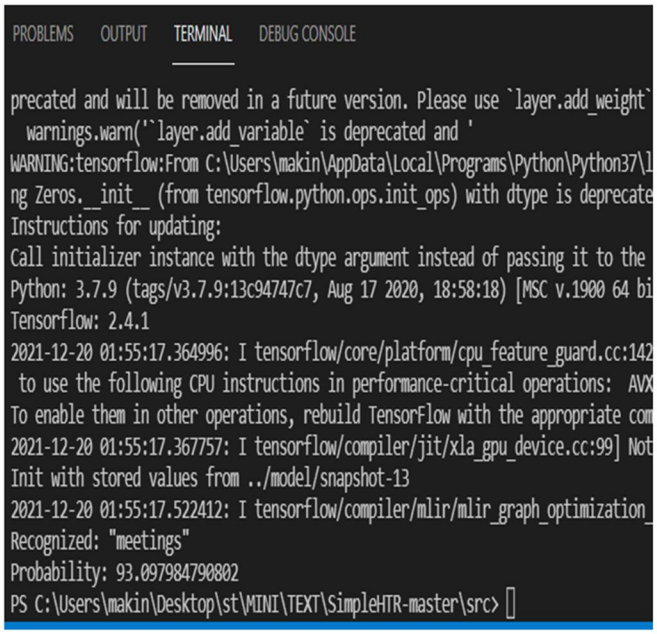


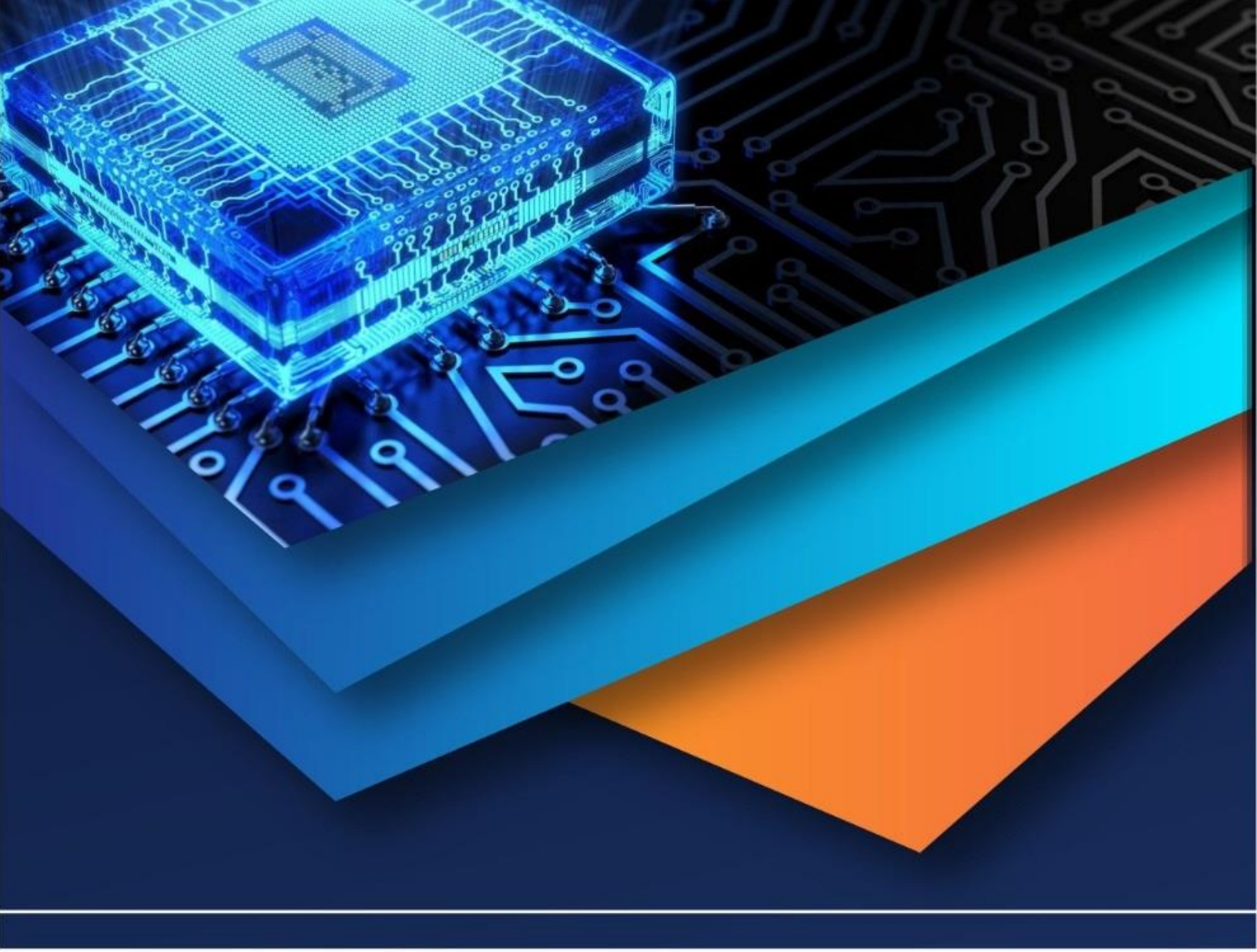
Figure 7.4: Sample output for the input image 2

V. CONCLUSION AND IMPLICATIONS FOR THE FUTURE

The proposed work obtained highest level of accuracy for the text with the little noise. The dataset has a huge impact on the accuracy. We intend to expand this research in the applying the de-noising in future to enhance the character recognition.

REFERENCES

[1] 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.
[2] Brakensiek, A., Rottland, J., Kosmala, A., & Rigoll, G. (2004). OFF-LINE HANDWRITING RECOGNITION USING VARIOUS HYBRID MODELING TECHNIQUES AND CHARACTER N-GRAMS. In EPRINTS-BOOK-TITLE s.n..
[3] Bajaj, R., Dey, L. & Chaudhury, S. Devnagari numeral recognition by combining decision of multiple connectionist classifiers. Sadhana 27, 59–72 (2002)
[4] M. Hanmandlu, O.V. Ramana Murthy - Fuzzy model based recognition of handwritten numerals. 2007 Pattern Recognition
[5] Sri. Yugandhar Manchala , Jayaram Kinthali , Kowshik Kotha , Kanithi Santosh Kumar, Jagilinki Jayalaxmi Handwritten Text Recognition using deep learning,IJERTV9IS050534, Volume 09, Issue 05 (May 2020).
[6] Fischer, A., Frinken, V., Bunke, H.: Hidden Markov models for off-line cursive handwriting recognition, in C.R. Rao (ed.): Handbook of Statistics 31, 421 – 442, Elsevier, 2013
[7] Frinken, V., Bunke, H.: Continuous handwritten script recognition, in Doermann, D., Tombre, K. (eds.): Handbook of Document Image Processing and Recognition, Springer Verlag, 2014
[8] S. Günter and H. Bunke. A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In Proc. 16th Int. Conf. on Pattern Recognition, volume 2, pages 332–337. IEEE, 2002.
[9] H. Bunke and U.-V. Marti In a system for general writer independent handwriting recognition, text line segmentation and word recognition are used. Pages 159– 163 in Proceedings of the 6th International Conference on Document Analysis and Recognition.
[10] M. Liwicki and H. Bunke, "Iam-ondb - an on-line English sentence database acquired from the handwritten text on a whiteboard," in ICDAR, 2005
[11] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in Advances in neural information processing systems, 2009, pp. 545–552.
[12] Handwritten English Character Recognition Using Neural Network, Anita Pal and Davashankar Singh, International Journal of Computer Science and Communication
[13] Sandhya Arora, "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.
[14] Om Prakash Sharma, M. K. Ghose, Krishna Bikram Shah, "An Improved Zone-Based Hybrid Feature
[15] Extraction Model for Handwritten Alphabets Recognition Using Euler Number", International Journal of Soft Computing and Engineering (ISSN: 2231 - 2307)
[16] N. Venkata Rao and Dr. A.S.C.S.Sastry - —Optical CharacterRecognition Technique Algorithms‖-2016 Journal of Theoretical and Applied Information Technology.
[17] J.Pradeep, E.Srinivasan, and S.Himavathi –—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
[18] 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



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)