In-Air Character Recognition Using Deep Learning

[1]Savla Gaurav, [2]Deshmukh Nikhil, [3]Dand Meet, Manoj Dhande

Student, Assistant Professor,
Dept. of C.S., Shah and Anchor Kutchii Engineering College,
Chembur, Mumbai-400088

ABSTRACT: A comparison of three algorithms – Logistic Regression, Shallow Neural Network and Deep Learning are used in this study to create an offline handwritten character recognition system. Recognition of similar shaped character is difficult problem and in character recognition system most errors occur due to similar shaped characters. In this article we propose a generic comparison between three algorithms which works well and the accuracy of this can be increased if the processor of computer is probably high. Our first comparison algorithm that is Logistic Regression which detects many similar characters thus giving us the less efficient success ratio. Next algorithm is Shallow Neural Networks this method is based on observation that there exists a relationship between heights and widths of alphabets written by individual which is unique and specific to him. Our last comparison algorithm is Deep learning Algorithm, we have used Convolutional Neural Network Algorithm to detect the handwritten characters. In today world it has become easier to train deep neural network due to availability of huge amount of data and various Algorithmic Innovations. Now-a-days the amount of computational power needed to train a neural network has increased due to the availability of GPU's and other cloud-based services like Google Cloud platform and Amazon Web Services which provide resources to train a Neural network on the cloud. We have designed a image segmentation based Handwritten character recognition system. We have developed this comparison algorithm system using python programming language.

KEYWORDS: Logistic Regression; Shallow Neural Networks; Deep Learning; Convolutional Neural Network.

I. INTRODUCTION

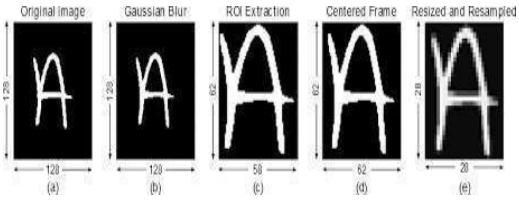
Hand written recognition is an important study issue in the realm of human-computer interaction because of its various applications in virtual reality, sign language recognition, and computer gaming. Human-computer interaction relies heavily on vision-based letter recognition (HCI). It might be used to communicate between people and machines. It varies from standard hardware-based solutions in that it enables human-computer interaction through gesture recognition. Gesture recognition, which recognizes the gesture or movement of the body or body components, determines the user intent. The human brain can easily grasp and analyst pictures. When the eye identifies a certain image, the brain may quickly segment and decode its many components. That procedure is carried out automatically by the brain, which entails not only analyzing the sights but also comparing their many properties to what it already knows in order to recognize these aspects. Image processing is a branch of computer science aimed at simulating human behavior in machines. Image processing is the study of pictures with the goal of extracting valuable information from them. This technique employs algorithms to turn hand written gestures into a digital format that computers can understand. Several areas, especially nowadays, and several software have been developed that use this concept. On them, leading in higher-quality photographs or modifications to particular traits that may be used to extract useful data. Many conditions call for image processing.

II. RELATED WORK

In [2] The Importance of good benchmark and standardized problem in computer vision and learning vision cannot be understated. The MNIST dataset has become a standard benchmark for learning, classification and computer vision system. Contributing to its widespread adoption are the understandable and institutive nature of the task, the relatively small size and storage requirements and the accessibility and ease-of-use of the database itself. Database was derived from large database known as NIST Database which contains digits, uppercase and lower-case handwritten letters. This paper introduces a variant of the full NIST dataset which we have called Extended MNIST (EMNIST), which follows the same conversion paradigm used to create the MNIST dataset. The result is a dataset that constitutes a more challenging classification task involving letters and one of that shares the same image structure and parameters as the original MNIST task, allowing for direct compatibility with all existing classifiers and systems. Benchmark results using an online ELM algorithm are presented along with a validation of the conversion process through the comparison of the classification results on NIST dataset. The mainstream of the computer vision community mainly focuses on the detection/segmentation of the characters Meanwhile, character recognition has not been well explored in the literature. Early works related to recognition are dedicated to detecting the digits. A few methods based on handcrafted low-level tackle the problem. However, to predict the correct accuracy of the character using the algorithms we need super computers or high-end performance computers. As in the current world companies like Alphabet have achieved only 75% accuracy in predicting the characters correctly.

III. IMPLEMENTATION

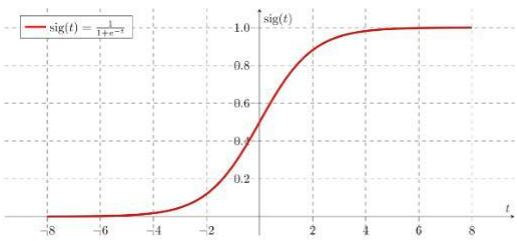
The experimental part is divided into six parts related to the methods used for assessments, hyperparameters settings, activation function, evaluation of the effects of image size, image transformations and self-attention. In this project, we have directly imported the EMNIST using MNIST loader. It comprises a total of 1,24,800 images of A-Z (randomly stored). Each of the letters in EMNIST is stored as a numbered array (28 x 28) as shown below :

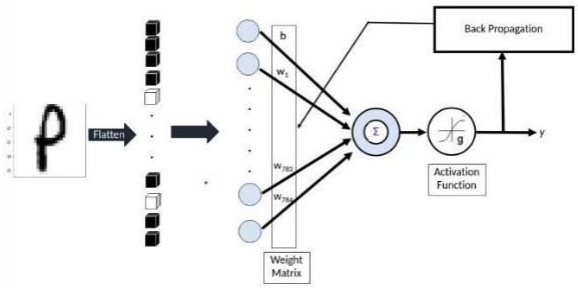


Classification of Algorithms

Logistic Regression

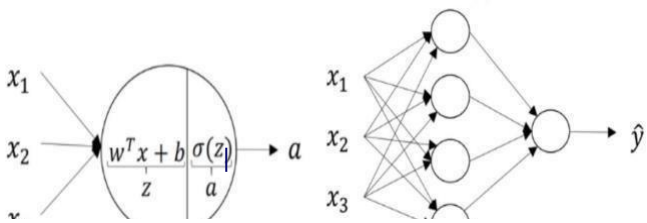
Logistic regression is a simple linear classifier. This algorithm tries to find the decision boundary by iterating over the training examples, trying to fit parameters that describe the decision boundary hyper-surface equation. During this learning process, the algorithm computes a cost function (also called error function), which represents the error measure of its hypothesis (the output value, prediction). This value is used for penalization, which updates the parameters to better fit the decision boundary. The goal of this process is to converge to parameter values that minimize the cost function. It has been proved that logistic regression is always convex, therefore the minimization process can always converge to a minimum, thus finding the best fit of the decision boundary this algorithm can provide





Shallow Neural Networks

The first part computes the output ***Z***, using the inputs and the weights



An ISO 9001:2008 Certified Journal |

Layer 1 is the input layer; layer 2 is a hidden layer; layer 3 is the output layer. x1, x2, and x3 are features fed to **the network; a1, a2, and a3 are hidden layer units; hΘ(x) is the output value (hypothesis). Each layer is fully** connected to the next one.)

The second part performs the activation on **Z** to give out the final output **A** of the neuron.

$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]},\ a_1^{[1]} = \sigma\left(z_1^{[1]}\right)$$
$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]},\ a_2^{[1]} = \sigma\left(z_2^{[1]}\right)$$
$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]},\ a_3^{[1]} = \sigma\left(z_3^{[1]}\right)$$
$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]},\ a_4^{[1]} = \sigma\left(z_4^{[1]}\right)$$

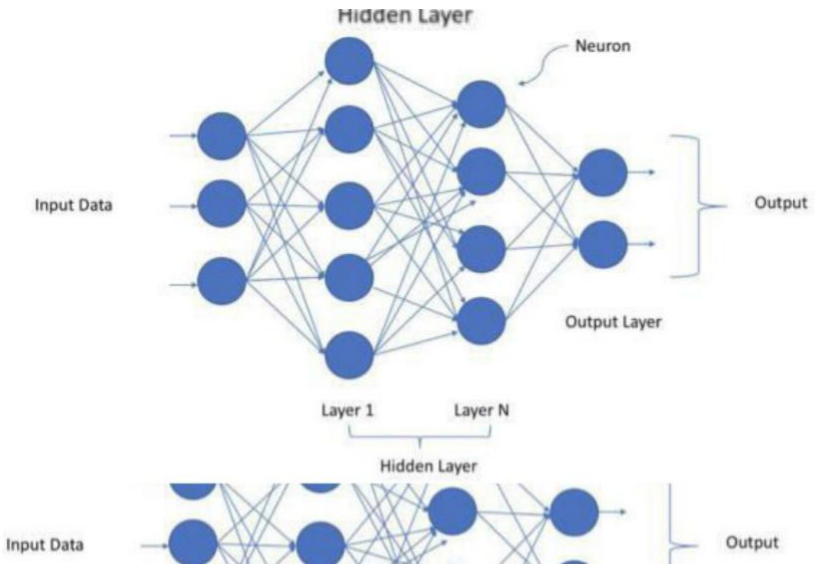
In the equations given above:

The superscript number [i] denotes the layer number and the subscript number j denotes the neuron number in a particular layer.X is the input vector consisting of 3 features' [i]j is the weight associated with neuron j present in the layer i. b[i]j is the bias associated with neuron j present in the layer i.Z[i]j is the intermediate output associated with neuron j present in the layer i. A[i]j is the final output associated with neuron j present in the layer i. Sigma is the sigmoid activation function

Deep Learning Algorithm

In deep learning, a convolutional neural network (CNN) is a class of artificial neural network, most commonly applied to analyse visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation. equivariant responses known as feature maps. They are often used in image recognition systems. Deep Neural networks can go up to 16 layers deep in search of accuracy.

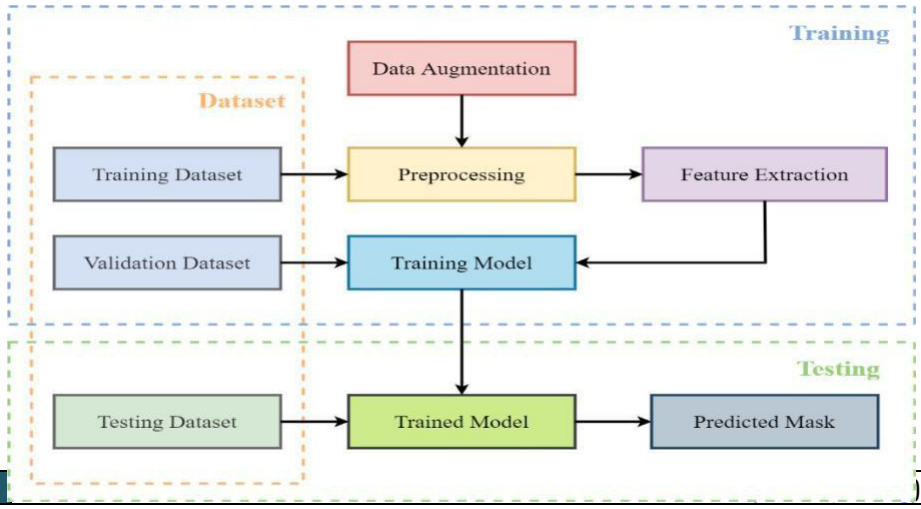
CNN consists of 3 layers namely: - Input Layer ,Hidden Layer , Output Layer It is a directed graph with at least three layers, at least one of which is a hidden layer. The more hidden layers there are, the most abstract features can be extracted from input data and the more complex the decision boundary is. Neurons are the vertices of the graph, while weighted connections are the edges. Output values of neurons are referred to as activations, and to compute the activation of a neuron in a layer, we use an activation function on the weighted sum of the connected neuron values in the previous layer. Input data are usually represented as a feature vector of real values, where the elements are fed to individual input neurons. The representation of output data (hypothesis) depends on the size of the output layer.



IV. **ALGORITHM AND METHODOLOGY**

The algorithm for training and testing process using training, validation and testing dataset are presented below:

STEP 1: Create and activate the virtual environment.
STEP 2: Import and install required packages.
STEP 3: Download the dataset.
STEP 4: Partition dataset into training, validation and testing.
STEP 5: Preprocess images and configure training pipeline.
STEP 6: Configuring the model parameters.
STEP 7: Training the model.
STEP 8: Evaluate the model.
STEP 9: Append results
STEP 10: Display Results



V. RESULTS ANALYSIS

After analysing how each parameter influences the performance of the model, we selected a configuration as shown in "Tab. IV" which in our opinion would give the most desirable results. When trained on the same exact dataset, the best model showed superior overall performance, demonstrating the im - pact of hyperparameters selected on model performance and accuracy. When compared with other algorithms, we found that Deep Learning Algorithm is the best among remaining algorithm".

Test Case ID	Objective	Steps / Description	Input	Expected Output	Actual Output	Result	Remark
1	To detect the character 'a'	User need to type a in front of webcam	Action of drawing character 'a'	The Displayed Character should be 'a'	Identifies similar characters like 'a'	PARTIAL PASS	Image is identified with less accuracy
2	To detect the character 'h'	User need to type character h in front of webcam	Action of drawing character 'h'	The Displayed Character should be 'h'	Identified correctly	PASS	Image is identified with high accuracy
3	To detect the character 'g'	User need to type character g in front of webcam	Action of drawing character 'g'	The Displayed Character should be 'g'	Identified Correctly	PASS	Image is identified with high accuracy
4	To detect the character 'p'	User need to type p in front of webcam	Action of drawing character 'p'	The Displayed Character should be 'p'	Identified Correctly	PASS	Image is identified with high accuracy

(TEST RESULT CASE)

Sr. No.	Experiments	Default Model	Best Model
1	Alpha = 0.009 in LR	Logistic Regression Neural Network Deep Learning	Deep Learning (72%)
2	Alpha = 0.5 in DL	Logistic Regression Neural Network Deep Learning	Deep Learning (72.6 %)
3	Alpha = 0.05 in NN	Logistic Regression Neural Network Deep Learning	Deep Learning (69%)
4	Alpha = 1.0 in LR	Logistic Regression Neural Network Deep Learning	Error

(PROPOSED CONFIGURATION)

VI. CONCLUSION AND FUTURE WORK

Character Recognition is a very broad field concerned with turning an image or a scanned document containing a set of characters into an encoded text that could be read by machines. In this project, we have attempted to build a recognizer for handwritten digits using the MNIST dataset. The challenge of this project was to be able to come up with some basic image correlation techniques, instead of some sophisticated algorithms, and see to what extent we can make this mechanism accurate. We have tried several versions and kept trying to improve each one in order to reach a higher performance rate. The last version has reached a rate of 57% accuracy. Unfortunately, we could not compare the performance of the mechanism we have built to some others that have already been designed and/or implemented before because we did not find any academic paper that tackles this method. The performance we have reached is far less than that of machine learning, which reaches a performance rate of 51.3%; however, it could be further improved and made into a better one. The goal of this project was to explore the field of OCR and try to come up with some techniques that could be used without going into deep computations, and even if the final result is not very reliable, it still provides an accuracy way better than random. The future steps that to go for would be having a closer look at the results of all the versions in order to find new rules. By extracting and implementing them, we will be able to enhance the performance of these versions. Moreover, it would be good if we could make some modifications to both the reference set and the rules in order to make our program more general and able to identify both typed and handwritten characters.

VII. REFERENCES

1. Y. J. Kim and S. W. Lee, ”*Offline recognition of unconstrained handwritten digits using multilayer backprop agation neural network combined with genetic algorithm*,” in Proc. 6th workshop. Image processing understanding‘, 1994, pp. 186-

2. E. Cohen, J. J. Hull, and S. N. Srihari, “understanding handwritten text in a structured environment: Determining ZIP codes from addresses, “Int. J. Pattern Recognition Artificial Intelligence, vol. 5, nos.and 2, pp. 221-264, 1991.
3. C. Y. Suen, C. Nadal, R. Legaut, T. A. Mai, and L. Lam, “Computer recognition of unconstrained

handwritten numerals,” Proc. IBEB, vol. 80, pp. 1162-1180, 1992.

4. Wei Lu, Zhijian Li,Bingxue Shi . ”Handwritten Digits Recognition with Neural Networks and Fuzzy Logic” in IEEE International Conference on Neural Networks, 1995. Proceedings.

5. N*IST Special Database 19*” https://www.nist.gov, [Online]. Available: https://www.nist.gov/srd/nist- special-database-19. [Accessed 05 March 2018].

6. www.researchgate.net/publication/264165284_RealTime_Hand_Gesture_Recognition_Using_Finger_Segmen tation.

7. www.aui.ma/sse-capstone-repository/pdf/spring-2018/HANDWRITTEN DIGITS RECOGNITION.pdf

8. www.researchgate.net/publication/264165284_RealTime_Hand_Gesture_Recognition_Using_Finger_Segmen tation.

9. www.researchgate.net/post/To_estimate_suitable_petrophysical_properties_would_it_will_be_useful_to_use_ CNN_or_GRU_algorithms_alone .

10. www.researchgate.net/publication/313844701_EMNIST_an_extension_of_MNIST_to_handwritten_letters

11. www.scribd.com/document/260297102/Character-Recognition-Using-ANN