iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.42949

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 V May 2022- Available at www.ijraset.com

Handwritten Letter Recognition using Artificial Intelligence

Jeevitha D[1], Muthu Geethalakshmi S[2], Nila I[3], Santhoshi V[4]
[1, 2, 3, 4]*Department of Computer Science Engineering, Jeppiaar Engineering College, Chennai*

Abstract: Images are easily processed and analysed by the human brain. When the eye sees a particular image, the brain is able to instantly segment it and recognize its numerous aspects. This project proposes the Deep Learning conceptual models based on Convolutional Neural Network (CNN). A comparison between the algorithms reveals that the handwritten alphabets, classified based on CNNs outperforms other algorithms in terms of accuracy. In this project, different architectures of CNN algorithm are used: Manual Net, Alex Net, LeNet Architecture. These architectures contain a convolution layer, max pooling, flatten, feature selection, Rectifier Linear Unit and fully connected softmax layer respectively. The image dataset with 530 number of training images and 2756 numbers of testing images are used to experiment the proposed network. The best accuracy and loss efficient model will be deployed in the Django framework in order to create a user interface for giving the character to be identified and receiving the output result of identified character.
Keywords: Handwritten Recognition, Alphabets, Deep Learning, TensorFlow, CNN, Django.

I. INTRODUCTION

The Handwritten Recognition is the ability of computer to recognize the human handwritten text from different manual sources like images, papers, photographs, documents etc. It is easy for humans to recognize the character from the image, but the question is whether it is possible for a machine to identify it accurately. Hence, the Deep Learning concept is applied.

Convolutional Neural Network, one of the deep learning algorithms is used to recognize the alphabet and compare the architectures of Convolutional Neural Network to deploy the highest accuracy with minimal loss to get the better results. Django framework is used in deployment of the model, where the user can upload the manually written alphabetic images and get the output.

II. RELATED WORK

Deep Convolutional Generative Adversarial Network is applied, instead of multi-layer perception. Comparing CGAN (Conditioner Generative Adversarial Network) and DCGAN (Deep Convolutional Generative Adversarial Network) and highlighting the differences and similarities are done by using the hand written data sets. For segmentation of digits, used horizontal and vertical projection methods. For classification and recognition, SVM is applied. For Feature Extraction, Convex Hull Algorithm is proposed. In Digit detection from digital devices from multiple environmental conditions, two automatic segmentations are used to recognize seven segment digits with some detection methods. It mainly focused on segmentation and digit recognition on many points of view and it is still in a process to get robust solution.

III.PROPOSED SYSTEM

In this proposed system, four modules are used: Manual Architecture, LeNet Architecture, AlexNet Architecture and Implementation in Django framework. Here, Django Framework place an important role, where the user can see the output in an interactive console web application. The input in this proposal system is in raw image format, where the user gets the input their system, to get the output as its equivalent editable value. The process flow of the proposal system is: database of alphabet→ feature extraction→ features selection→ usage of deep learning algorithm's architecture→ alphabetic prediction. Epochs value is mentioned during the training of particular architecture. Testing and Training of images for every architecture is viewed by Graphs. Fig. 1 shows the process flow of the proposed system.

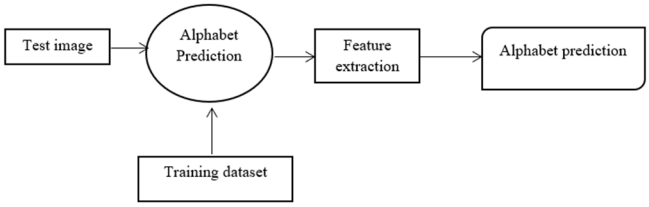


Fig. 1 Workflow of Handwritten Alphabet recognition

IV. WORKING PROCESS IN CNN MODEL

The Convolution Neural Network algorithm takes an input image, analyzes the image and distinguish between them. Compared to other classification algorithms, convolution requires less pre-processing. Convolutions can learn these filters with enough training. This architecture is inspired by the organization of the Visual Cortex, which is linked to the connectivity pattern of Neurons present in the Human Brain. The Receptive Field, known as individual neurons can only respond to stimuli in a small area of the visual field. Their network consists of four layers with input containing 1,024 units, first Hidden layer containing 256 units, second Hidden layer containing 8 units and 2 output units. Fig. 2 shows the general architecture layers of CNN.

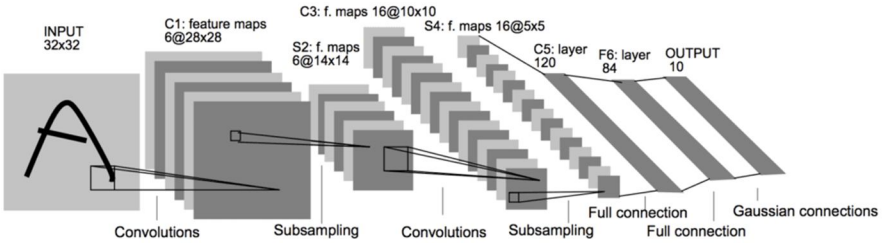


Fig. 2 Convolution Neural Network architecture with layers

A. Input Layer

Input layer in CNN contains image data. Image data is represented by three dimensional matrices. The image datas are reshape it into one column.

B. Convolutional Layer

Convolutional layer is employed for extracting the features of the image. A component of the image is connected to Convo layer to perform convolution operation, calculating the scalar product between receptive field and the filter.

Result will be a single integer of the output volume. Then, the filter over the following receptive field of the same input image by a Stride and do the same operation again. It will repeat the identical process again and again until it goes through the whole image. The output from this layer will be the input for the subsequent layer.

C. Pooling Layer

The pooling layer is employed after convolution, so as to reduce the spatial volume of the input image. It is used between two convolution layers. The max pooling is a way to reduce the spatial volume of input image, making it computationally cost efficient. The max pooling is applied in single depth slice with Stride of 2. Here, the 4 x 4D image input is reduced to 2 x 2D image.

D. Fully Connected Layer

Fully connected layer involves weights, biases, and neurons. Here, the neurons in one layer are connected to neurons in another layer. It is used in training classified images between different categories.

E. Softmax/Logistic Layer

It is the last layer of CNN which resides at the end of FC layer. Binary classification is done by Logistic and multi-classification is done by softmax.

F. Output Layer

Output layer contains the label which is within the type of one value data processed in an image.

V. HANDWRITTEN ALPHABETS IDENTIFICATION

The input Image is converted into an array value using array function package. The already classified Alphabet image dataset classifies what are the given alphabet is. Then, the Alphabet is identified using the prediction function.

The Alphabet Recognition method is a two-channel based architecture that is able to recognize the Alphabet characters. The Alphabet images are used as the input into the inception layer of the CNN. The Training phase involves the feature extraction and classification using Convolution Neural Network.

A. Libraries Used

TensorFlow- To use the tensor board to compare the loss and Adam curve of the result data or obtained log; Keras- To prepare the image dataset; Matplotlib- To display the result of the predictive outcome; OS- Gives access to the file system to read the image from the training and testing directory of the device.

B. Architectures Used

1) *Manual architecture*: Generally, CNN architecture consists of Three basic layers- an input layer, an output layer and hidden layers. Convolutional layers, ReLU layers, Pooling layers and Fully Linked layers are accommodated in the hidden layers.

Algorithm 1 For Manual CNN Architecture
1: Conv2D- (None, 126, 126, 32)
2: MaxPooling2D- (None, 63, 63, 32)
3: Flatten- (None, 127008)
4: Dense-(None, 38)
5: Dense-(None, 52)

2) *AlexNet:* AlexNet is the first Convolutional Neural Network to use a GPU to improve performance. AlexNet has 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers and 1 softmax layer in its design. Each convolutional layer consists of convolutional filters and a non-linear activation function, ReLU. Maximum pooling is achieved using the pooling layers.

Algorithm 2 For AlexNet Architecture
1: Conv2D- (None, 54, 54, 96)
2: Activation- (None, 54, 54, 96)
3: MaxPooling2D- (None, 27, 27, 96)
4: Conv2D- (None, 17, 17, 256)
5: Activation- (None, 17, 17, 256)
6: MaxPooling2D- (None, 8, 8, 256)
7: Conv2D- (None, 6, 6, 384)
8: Activation- (None, 6, 6, 384)
9: Conv2D- (None, 4, 4, 384)
10: Activation- (None, 4, 4, 384)
11: Conv2D- (None, 2, 2, 256)
12: Activation- (None, 2, 2, 256)
13: MaxPooling2D- (None, 1, 1, 256)
14: Flatten- (None, 256)
15: Dense- (None, 4096)
16: Activation- (None, 4096)
17: Dropout- (None, 4096)
18: Dense- (None, 4096)
19: Activation- (None, 4096)
20: Dropout- (None, 4096)
21: Dense- (None, 1000)
22: Activation- (None, 1000)
23: Dropout- (None, 1000)
24: Dense- (None, 52)
25: Activation- (None, 52)

3) *LeNet*: LeNet is one of the first Convolutional Neural Networks to promote Deep Learning. This architecture has seven layers. The layer composition consists of 3 convolutional layers, 2 sub-sampling layers and 2 fully connected layers.

Algorithm 3 For LeNet Architecture
1: Conv2D- (None, 75, 75, 32)
2: MaxPooling2D- (None, 37, 37, 32)
3: Conv2D- (None, 12, 12, 128)
4: MaxPooling2D- (None, 6, 6, 128)
5: Flatten- (None, 4608)
6: Dense- (None, 256)
7: Dense- (None, 52)

VI. RESULT AND DISCUSSIONS

The training of the Datasets was successfully done by the architectures of CNN- Manual architecture, AlexNet and LeNet. The following graph shows the conclusion output from training the architectures.

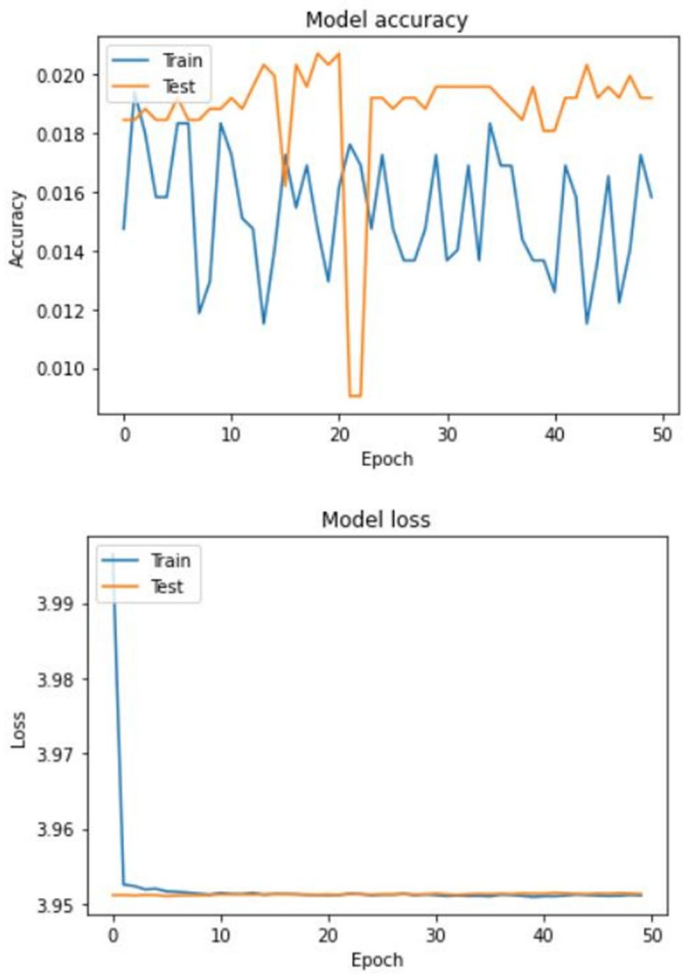


Fig. 3 Model Accuracy and Model Loss Vs Epoch for Manual architecture

Fig. 3 shows the trained efficiency of the Manual architecture, which depicts both the model loss and accuracy values from being trained with an Epoch value of 50. There is no steady loss nor steady accuracy rate curve and sudden increasing or decreasing of Accuracy and loss may Over-fit the values, resulting in exact identification of images.

From this graph, we can conclude that this model will not be efficient to Identify the Handwritten Alphabet characters.

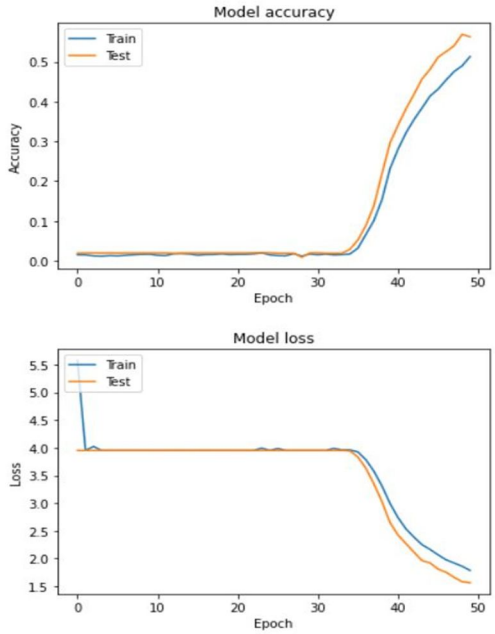


Fig. 4 A line graph depicting Model accuracy and Model loss Vs Epoch for AlexNet Architecture

Fig. 4 shows the trained efficiency of the AlexNet architecture, which depicts both the model loss and accuracy values from being trained with an Epoch value of 50. The trained values and the testing values are high but, there is no steady loss and steady accuracy rate curve in this graph. There may be an Under-fit of values in this model since half of the epoch process has the lowest Accuracy and highest Loss values.

From this graph, we can conclude that this model also will not be efficient enough to Identify the Handwritten Alphabet characters.

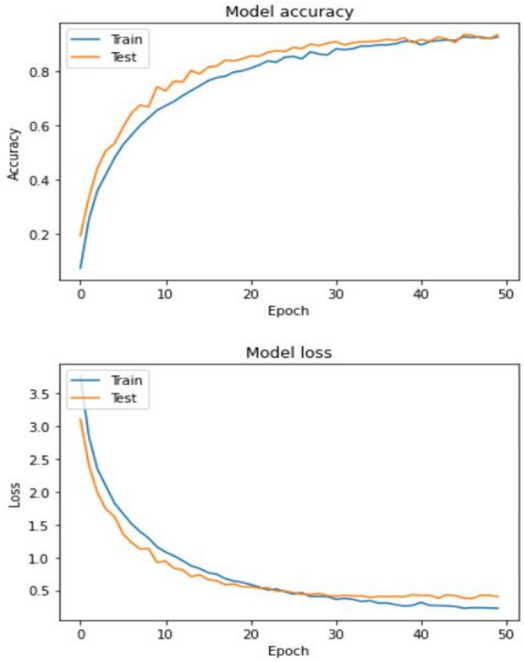


Fig. 5 A line graph depicting Model accuracy and Model loss Vs Epoch for LeNet Architecture

Fig. 5 shows the trained efficiency of the LeNet architecture, which depicts both the model loss and accuracy values from being trained with an Epoch value of 50. There is a steady loss and steady accuracy rate curve in this graph. The trained and tested values are almost coincided, which is the best model for similar image recognition.

From this graph, we can conclude that this model will be efficient enough to Identify the Handwritten Alphabet characters.

Table I
Architecture Efficiency Comparison

Sl. No	Efficiency		
	Architecture used	Loss %	Accuracy %
1.	Manual	395.13	2.00
2.	AlexNet	156.30	56.25
3.	LeNet	10.41	96.42

Table I gives the efficiency comparison between architectures. It is clearly noticeable that LeNet gets the highest efficiency of 96.42% of accuracy and 10.41% of loss. AlexNet gets its efficiency slightly lower than LeNet by 56.25% of accuracy and 156.30% of loss, followed by Manual architecture of 2% accuracy and a loss of 395.13%. From this Table, LeNet Architecture is identified to be suitable for recognizing the handwritten alphabets with a high efficiency.

The LeNet architecture is then deployed by using the Django framework. The following are the screenshots of deployment, coded in PyCharm.

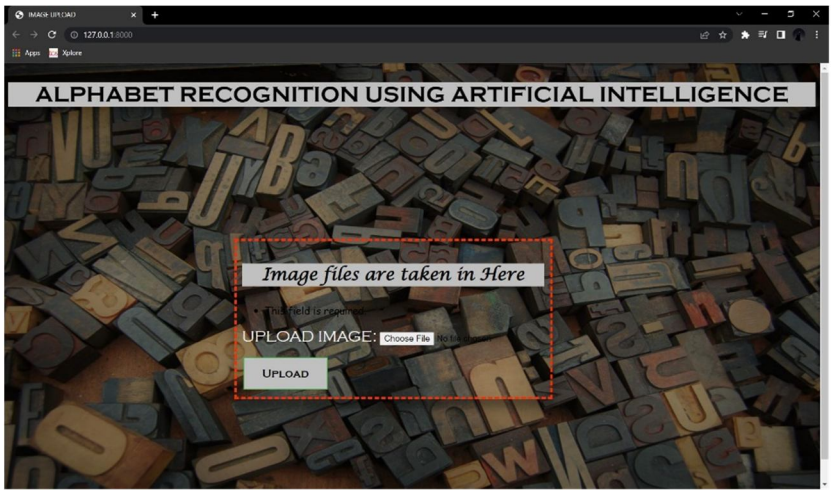


Fig. 6 Page asking for an input image to identify the Handwritten image

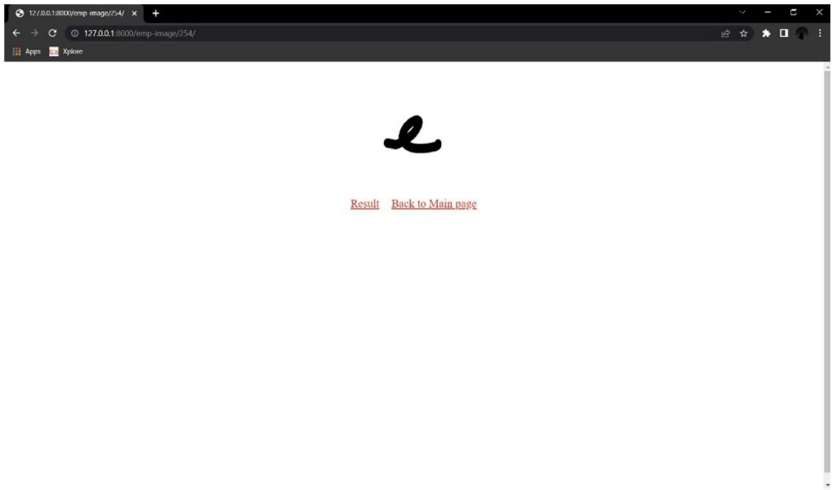


Fig. 7 Page showing the uploaded input image

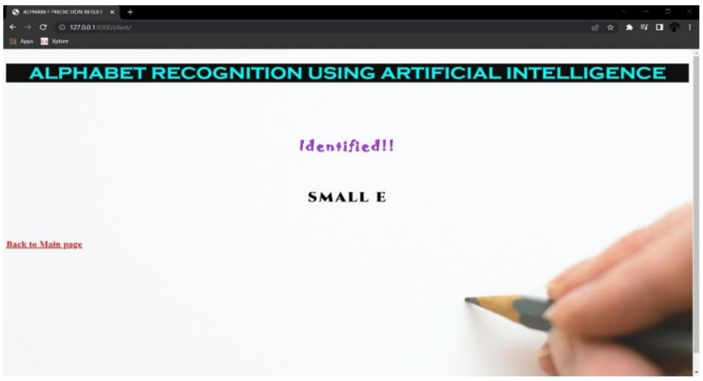


Fig. 8 Identification of the Handwritten Alphabet

VII. CONCLUSION

It highlights on how image from given dataset is trained by using two different algorithms in CNN model. Therefore, it brings some of the following different alphabet prediction. We had trained three different types of CNN, namely Manual architecture, AlexNet architecture and LeNet architecture and will be comparing the accuracy. The better classification method was then taken from there and is deployed in Django framework for better user interface.

VIII. ACKNOWLEDGMENT

We thank the Management, Principal, Head of Computer Science Department and the Professors of Jeppiaar Engineering College for providing the opportunity. A special thanks to our project Guide and Counsellors for extending a helping hand throughout the process. We would also like to thank our parents and friends for their help and Motivation throughout this project.

REFERENCES

[1] Z. Tan, J. Chen, Q. Kang, M. Zhou, A. Abusorrah and K. Sedraoui, "Dynamic Embedding Projection-Gated Convolutional Neural Networks for Text Classification," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 3, pp. 973-982, March 2022, doi: 10.1109/TNNLS.2020.3036192.

[2] C. N. Truong, N. Q. H. Ton, H. P. Do and S. P. Nguyen, "Digit detection from digital devices in multiple environment conditions," 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), 2020, pp. 1-3, doi: 10.1109/RIVF48685.2020.9140775.

[3] S. Arseev and L. Mestetsky, "Handwritten Text Recognition Using Reconstructed Pen Trace with Medial Representation," 2020 International Conference on Information Technology and Nanotechnology (ITNT), 2020, pp. 1-4, doi: 10.1109/ITNT49337.2020.9253330.

[4] R. Sethi and I. Kaushik, "Hand Written Digit Recognition using Machine Learning," 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), 2020, pp. 49-54, doi: 10.1109/CSNT48778.2020.9115746.

[5] Y. Patil and A. Bhilare, "Digits Recognition of Marathi Handwritten Script using LSTM Neural Network," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), 2019, pp. 1-4, doi: 10.1109/ICCUBEA47591.2019.9129291.

[6] W. Xu, X. Li, H. Yi and Z. Deng, "A Simple Conceptor Model for Hand-written-digit Recognition," 2019 3rd International Symposium on Autonomous Systems (ISAS), 2019, pp. 61-64, doi: 10.1109/ISASS.2019.8757783.

[7] S. K. Leem, F. Khan and S. H. Cho, "Detecting Mid-Air Gestures for Digit Writing With Radio Sensors and a CNN," in IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 4, pp. 1066-1081, April 2020, doi: 10.1109/TIM.2019.2909249.

[8] S. M, S. Surya and G. Priyanka, "Hand Written Indian Numeral Character Recognition using Deep Learning approaches," 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), 2018, pp. 1301-1304, doi: 10.1109/ICRIEECE44171.2018.9008959.

[9] A. Chakraborty, R. De, S. Malakar, F. Schwenker and R. Sarkar, "Handwritten Digit String Recognition using Deep Autoencoder based Segmentation and ResNet based Recognition Approach," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 7737-7742, doi: 10.1109/ICPR48806.2021.9412198.

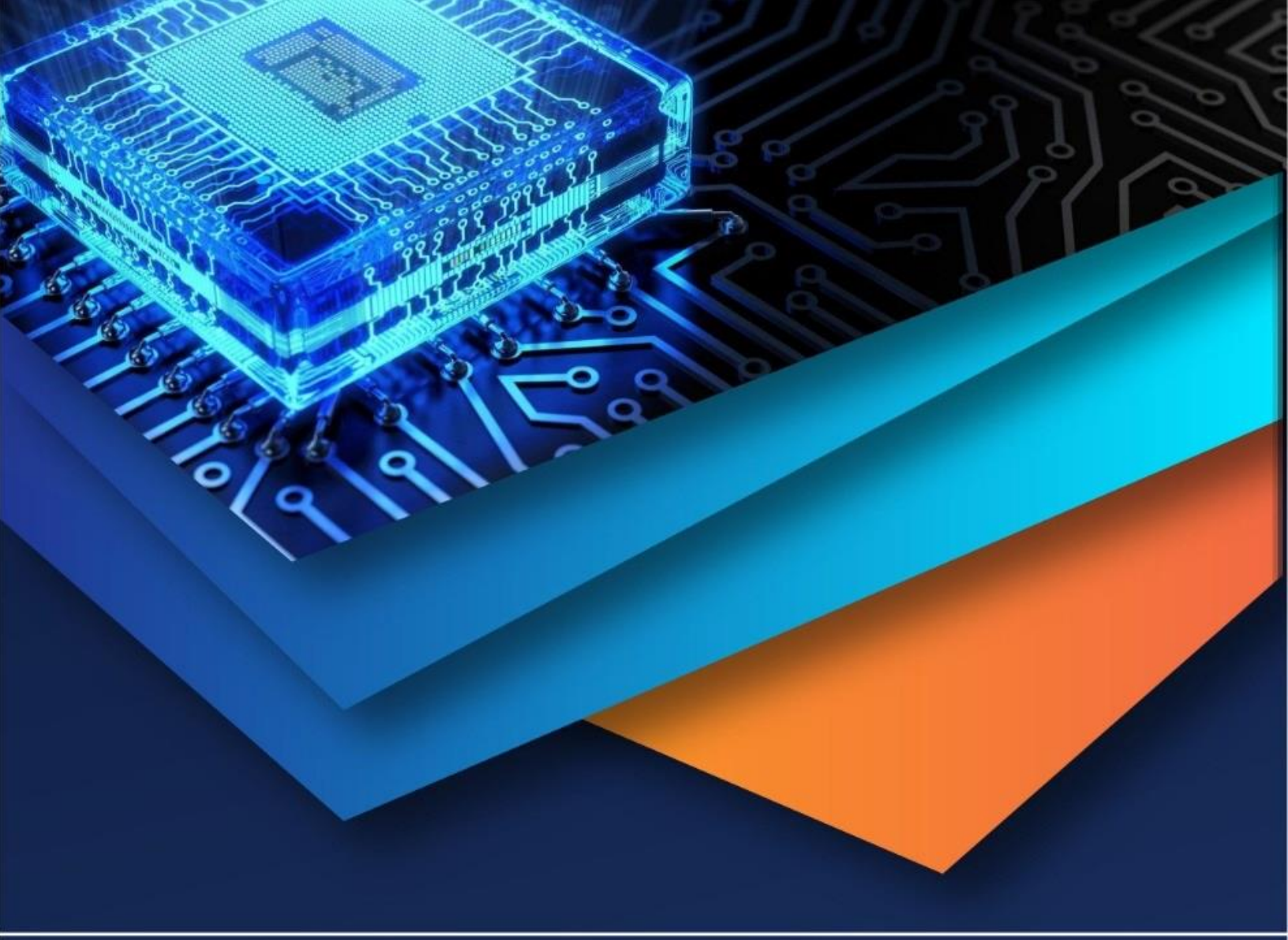
[10] R. Fernandes and A. P. Rodrigues, "Kannada Handwritten Script Recognition using Machine Learning Techniques," 2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 2019, pp. 1-6, doi: 10.1109/DISCOVER47552.2019.9008097.

[11] J. Li, G. Sun, L. Yi, Q. Cao, F. Liang and Y. Sun, "Handwritten Digit Recognition System Based on Convolutional Neural Network," 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), 2020, pp. 739-742, doi: 10.1109/AEECA49918.2020.9213619.

[12] P. Dhande and R. Kharat, "Recognition of cursive English handwritten characters," 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 199-203, doi: 10.1109/ICOEI.2017.8300915.

[13] S. E. Warkhede, S. K. Yadav, V. M. Thakare and P. E. Ajmire, "Handwritten Recognition of Rajasthani Characters by Classifier SVM," 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), 2021, pp. 1-5, doi: 10.1109/ICITIIT51526.2021.9399590.

[14] V. V. Mainkar, J. A. Katkar, A. B. Upade and P. R. Pednekar, "Handwritten Character Recognition to Obtain Editable Text," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 599-602, doi: 10.1109/ICESC48915.2020.9155786



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)