iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37610

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: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Offline Handwritten Character Recognition using Convolutional Neural Network

N. Devi[1], P. Leela Rani[2], AR. Guru Gokul[3], VM. Sivagami[4]
[1, 2, 3, 4]*Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur-602117, Tamil Nadu, India*

Abstract: ***This paper focuses on the task of recognizing handwritten Hindi characters using a Convolutional Neural Network (CNN) based. The recognized characters can then be stored digitally in the computer or used for other purposes. The dataset used is obtained from the UC Irvine Machine Learning Repository which contains 92,000 images divided into training (80%) and test set (20%). It contains different forms of handwritten Devanagari characters written by different individuals which can be used to train and test handwritten text recognizers. It contains four CNN layers followed by three fully connected layers for recognition. Grayscale handwritten character images are used as input. Filters are applied on the images to extract different features at each layer. This is done by the Convolution operation. The two other main operations involved are Pooling and Flattening. The output of the CNN layers is fed to the fully connected layers. Finally, the chance or probability score of each character is determined and the character with the highest probability score is shown as the output. A recognition accuracy of 98.94% is obtained. Similar models exist for the purpose, but the proposed model achieved a better performance and accuracy than some of the earlier models.***

Keywords: Devanagari characters, Convolutional Neural Networks, Image Processing

I. INTRODUCTION

In India, Hindi is the widely used language across many states and union territories. Hindi is different from other languages because of the fact that each character is not written separately. This leads to poor recognition by the various recognition algorithms. Character recognition is done by Optical Character Recognition (OCR). OCR converts the image of a character into digital format (ASCII). The input to OCR is the image of characters and a text file is produced as output.

Devanagari script is formed as a result of logical combination of its native symbols [1]. The writing technique in it is in an horizontal manner towards right from the left. The alphabetic script does not differentiate between the uppercase and lowercase letters [2]. There are 12 vowels and 36 consonants as shown in figure 1 and figure 2. Along with these letters, the script also constitutes a collection of mathras that is placed along with the alphabet either at the top, bottom or on either sides to act as vowel modifier. It also comprises of pure consonants those can be united with consonants to form conjuncts.

The vowels are used for dual purpose in Hindi and English languages as follows:

1) For generating their own sounds
2) For adjusting the sound of a consonant

A suitable modifier can be added to the consonant to achieve the second purpose. Fig. Represents the position of modifiers attached to the letter in Hindi. On examining the same, the modifiers can be classified as core, upper and lower based on the position. Some of the letters is a combination of any two categories of them [3]. In such cases, the core modifier is placed on either of the sides and the other one is placed above or below the core modifier.

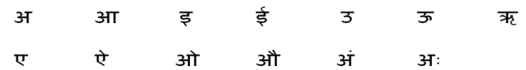


Figure 1. Hindi vowels

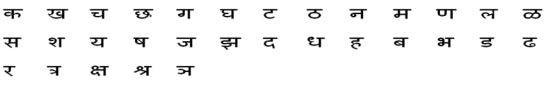


Figure 2. Hindi consonants

Figure 3. Hindi mathras

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Roughly all the consonants of the Devanagari script have a half character that attaches to the following character in the script to produce conjuncts. These are also called as combined characters. A few samples are shown in figure 3. The flat line above all the characters is Shirorekha or head line. Every word in Devanagari script can be divided into triplet zones as follows: a core zone, an upper zone and lower zone. The upper and core zones are separated by a flat line.

II. ARCHITECTURE OF CONVOLUTIONAL NEURAL NETWORK

The architecture of CNN mimics neurons in the human brain, in particular the visual cortex. Each neuron receives signals regarding an event and responds to that particular event. The response to multiple events by the neurons overlaps and covers the entire area of the cortex.

CNN accepts an input image, identifies various aspects in the image, assigns importance to those aspects and differentiates one from the other. To precisely find out the spatial and temporal relationships in an image, relevant filters are applied in CNN [4]. In CNN, preprocessing is very much reduced as the filters needed for the approach is learned by CNN itself instead of the parameters being decided by a human being. CNN is able to reduce the number of parameters used for fitting in recognition of an image and it strives to produce a better fit due to the ability of reusing the weights assigned.

A. Input Image

Let us consider the RGB image as shown in figure 4. The image has three colored planes, namely, Red, Green, and Blue. The various color models available are Grayscale, RGB, HSV, CMYK, etc. Here we consider RGB model. It would be computationally demanding to process images of larger dimensions. CNN reduces this intensive operation into an more less intensive operation by converting the image into a structure, that makes the computation less demanding and much care is taken in not losing critical features that would otherwise be essential to reconstruct the original image. So the model that is used for this kind of image processing should learn the essential features much faster and also should be applicable to enormous data sets.

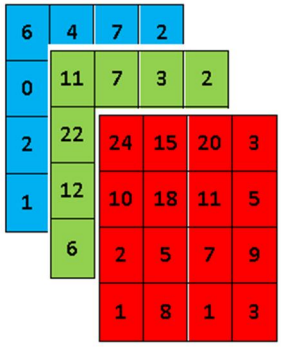


Figure 4. 4x4x3 RGB Image

B. Convolution Layer — The Kernel

In figure 5, the input image is 5x5x1, I. The input image is denoted using green color. Kernel/Filter K is part of a Convolutional layer and this K is responsible for the convolution on the input image. In fig 2, the kernel is denited using the color yellow. K is chosen to be 3x3x1 matrix.

Kernel/Filter,

K = 101
010
101

Since the Stride Length = 1 , the kernel performs a shift of 9. A matrix multiplication operation is carried out between K and the portion P of the image.

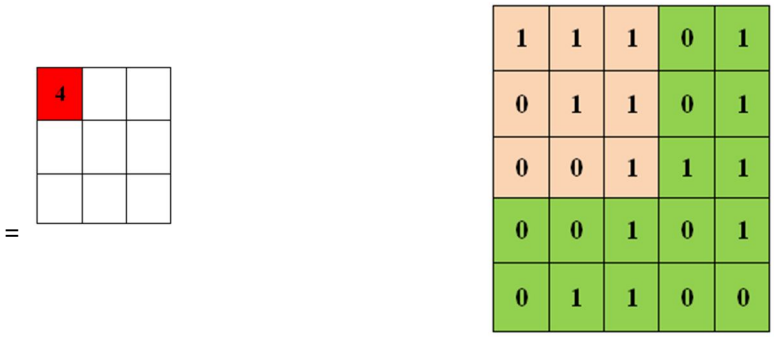


Figure 5. Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 feature

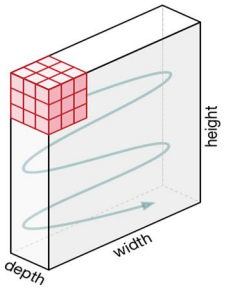


Figure 6. Shift in Kernel

The filter moves in the pattern as shown in figure 6. The filter starts from the shown position and traverses to the right, then it proceeds to the left most beginning and then traverses to the right. This traversal is done with a certain shift value till the entire image is covered. Figure 7 shows the traversal when the stride value is 2.

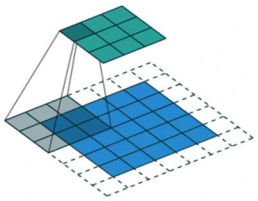


Figure 7. Convolution Operation with Stride Length = 2

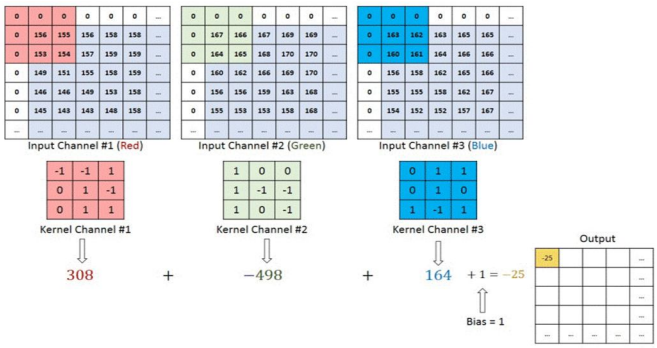


Figure 8 Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel

Figure 8 shows the convolution operation on images with multiple channels (e.g. RGB). The Kernel is same as that of the image depth. Then Matrix Multiplication is carried out between K_n and I_n stack ($[K_1, I_1]$; $[K_2, I_2]$; $[K_3, I_3]$). The results are finally added together to obtain a single convoluted value in the matrix.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

The high-level features such as color, gradient orientation, edges, are very crucial for regenerating an image. These features are extracted from the input image using CNN. There may be multiple Convolutional layers in this architecture. The Low-Level features are extracted from the image using the first convolution layer. Adding more number of convolution layers enables discovering more features. Valid Padding is performed in which the convoluted value has less dimension than the actual image. Similarly same padding is used to increase the number of dimensions in the convoluted feature or retain the same number of dimensions.

C. Pooling Layer

The Pooling layer decreases the computational intensiveness to process the data [5]. Pooling layer achieves this by reducing the number of dimensions.. The process of extracting the important features from an image is also done by the pooling layer as shown in figure 9 This layer is responsible for training the model.

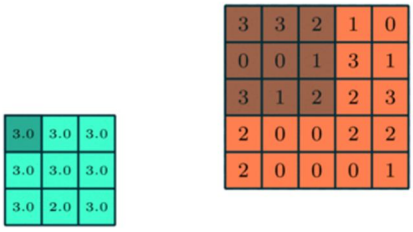


Figure 9. 3x3 pooling over 5x5 convolved feature

D. Categories of Pooling

Average pooling and Max Pooling are the two categories of Pooling as denoted in figure 10. Average Pooling calculates the average of all the values from the section of the image traversed by the Kernel. It just serves to reduce the number of dimensions in the image. Max Pooling locates and gives the maximum value from the section of the image traversed by the Kernel. It also serves to suppress the noise present in the image along with reduction in number of dimensions. So we can conclude that it is desirable to use Max Pooling.

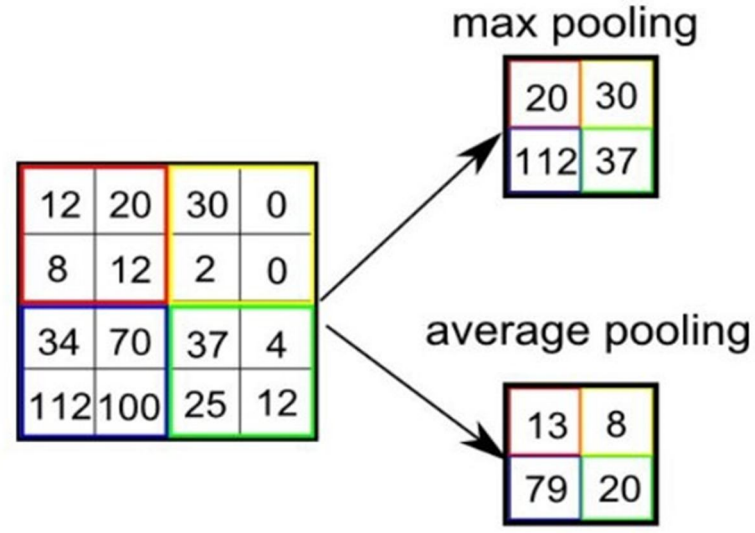


Figure 10. Categories of Pooling

More number of features can be identified by adding more number of Convolution Layer and the Pooling Layer. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. These steps enable the model to be trained to identify the features. The last step in this process would be to just roll out the final values and input this to the NN.

E. Classification — Fully Connected Layer (FC Layer)

A Fully-Connected layer [6] is added to the model to train it to accept non linear mixture of the essential features. Finally, the image should be levelled out into a column vector. This vector serves as the input to Neural Network and the concept of back propagation is applied to this model to train it. After a series of epochs, essential features and certain low-level features can be identified by the model. Using Softmax classification, these features can be classified. The same is denoted in figure 11.

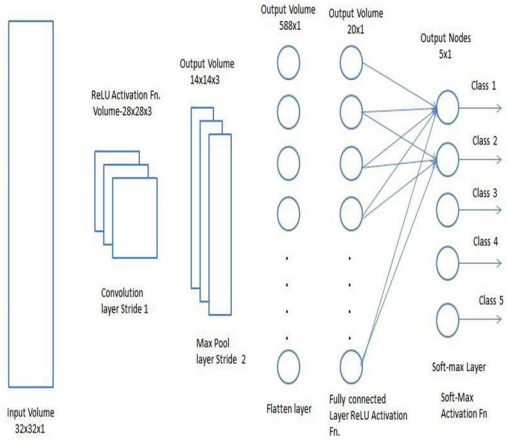


Figure 11. Addition of FC to the model

III. PROPOSED SYSTEM FOR HINDI CHARACTER RECOGNITION

The main architecture of this model involves the use of Convolutional Neural Network (CNN) layers for image processing and feature extraction. Fully connected layers are added for recognition purpose. Figure 12 depicts the general architecture of the proposed CNN model. Figure 13 shows the operations carried out on the images present in the image database.

The following phases are involved in the system:

A. Image Preprocessing

The input data i.e. the input image is given. It is first resized to 32x32 so that it is easier for the model to predict. Data augmentation is used to increase the size of the training set by performing slight alteration in the training images. Only gray scale images are used to train the model.

UCI

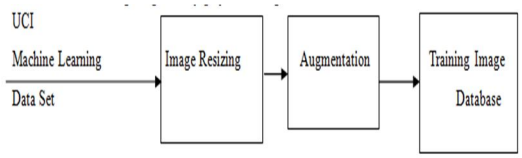


Figure 12. Architecture of Proposed System for HCR

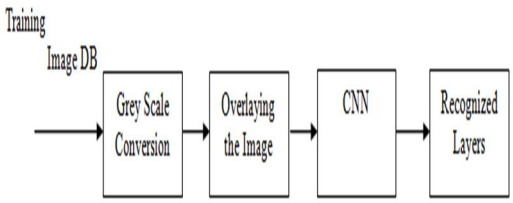


Figure 13. Operations on Image

B. CNN layers

After the image is preprocessed, it is fed to the CNN layers. As described earlier, CNN layers [7] are trained in such a way that they are capable of extracting the essential features in the image. Four layers of CNN are used. In each layer, three major operations are performed. The first is the application of the filter kernel. For all the layers, the filter kernel is of size 3x3. The first two layers involve 32 kernels each and the third and fourth layers involve 64 kernels each. The second operation is the application of ReLU activation function. This helps achieve non-linearity. At the end, pooling is applied which reduces the size of the feature maps.

C. Fully Connected Layers

After the image is processed by the CNN layers and the essential features are extracted, the flattened output from the final CNN layer is given to three fully connected layers. The first layer consists of 128 units, with all having Rectified Linear Unit (ReLU) as the activation function [8],[9].

After that, batch normalization is applied. The second layer consists of 64 units with ReLU followed by batch normalization. The final layer is the output layer which consists of 46 units that represent each class of output. The SoftMax activation function is used in the final layer. The chance or probability scores of all the characters are produced and the one with the maximum score is provided as the output. Loss function used in the model is Categorical Cross Entropy. The optimizer involved is the Adam optimizer.

IV. RESULTS AND DISCUSSIONS

The dataset for the model is obtained from the UC Irvine Machine Learning Repository [10]. In this dataset there are 46 classes of characters. This includes Devanagari alphabets and digits. For each class, around 2000 images are present. Training set is 85% of the data and test set is 15%. The data type is Gray scale image. The images are in .png format of resolution 32x32. The actual characters are centered within 28x28, and a padding of 2 pixels is added on all four sides. The summary of the model generated for recognizing Hindi characters is presented in the figure 14. Model training is followed by testing. In our dataset, 20% of the images are kept as the test set. The proposed model provides an accuracy of 97.64% on the training set and 98.94% on the test set.

V. CONCLUSION

This paper focuses on Hindi Character Recognition using Convolution Neural Networks Approach. This provides a validation accuracy of 98.94%. But this accuracy can be further improved by fine tuning the various parameters

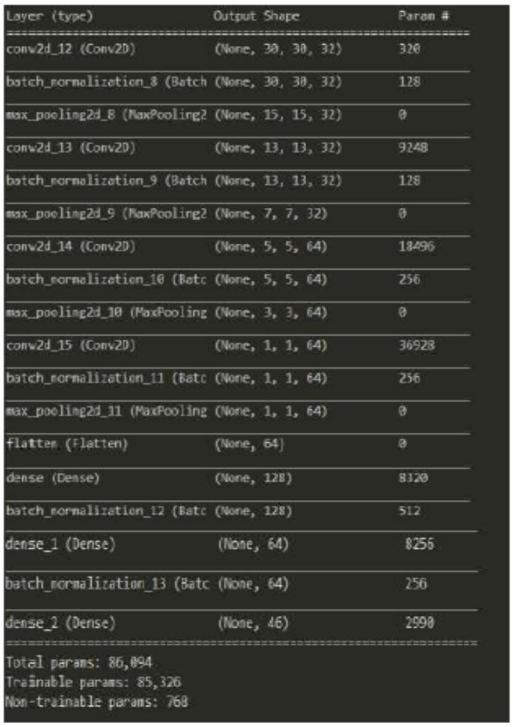


Figure 14. Generated Models in HCR

REFERENCES

[1] Gyanendra K.Verma, P. K., Shitala Prasad, 2011. Handwritten Hindi Character Recognition Using Curvelet Transform. In Information Systems for Indian Languages, pages 224-7. Springer. - https://doi.org/10.1007/978-3-642-19403-0_37.

[2] Deepti Khanduja, S. P., Neeta Nain, 2015. Hybrid Feature Extraction Algorithm for Devanagari Script. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 15:2:1-2:10. - https://doi.org/10.1145/2710018

[3] S. Acharya, A.K. Pant and P.K. Gyawali "Deep Learning Based Large Scale Handwritten Devanagari Character Recognition" ,In Proceedings of the 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 121- 6, 2015.

[4] D. Chaudhary and K. Sharma, "Hindi Handwritten Character Recognition using Deep Convolution Neural Network," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2019, pp. 961-5.

[5] Bottou, L., 2012. Stochastic Gradient Descent Tricks. In Neural Networks: Tricks of the Trade, volume 7700, pages 421-36. ISBN 9783642352898.https://doi.org/10.1007/978-3-642- 35289-8.

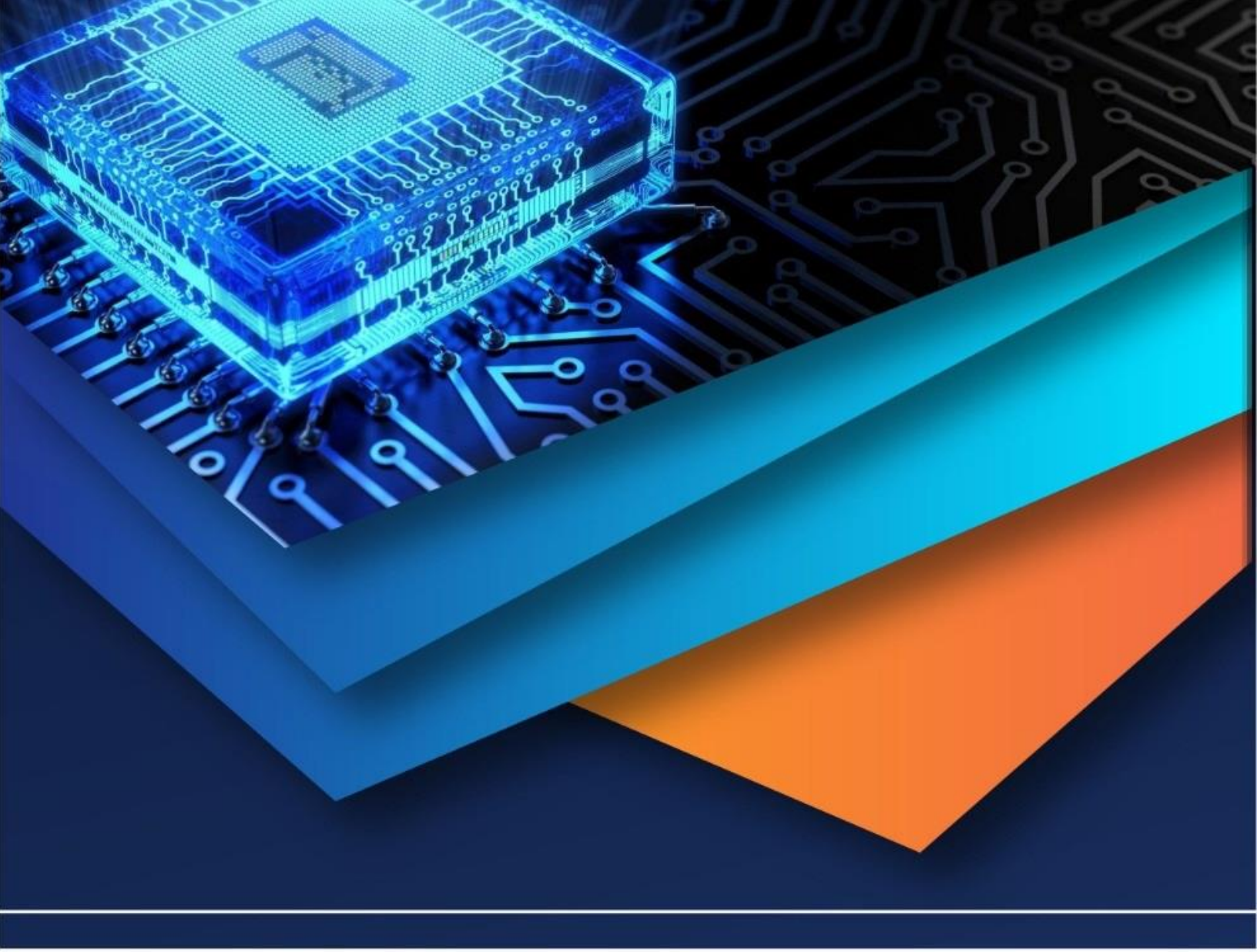
[6] Glorot, X. and Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9, pages 249-56. PMLR.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions", Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.

[8] K. Verma and M. Singh, "Hindi handwritten character recognition using convolutional neural network", International Journal of Computer Sciences and Engineering, vol. 6, no. 6, 2018.

[9] Bai, J., Chen, Z., Feng, B., Xu, B. 2014, Image character recognition using deep convolutional neural network learned from different languages. In: IEEE International Conference on Image Processing, pp. 2560–4.

[10] Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016 Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging 35, 1285–98.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY