iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: V Month of publication: May 2021

DOI: https://doi.org/10.22214/ijraset.2021.34602

Traffic Sign Recognition using CNN and Keras

Monika Sahu[1], Katyayani Sahu[2], Anjali Sisodia[3], Deepshikha Ogre[4]
[1, 2, 3, 4]Department of Computer Science & Eng, New GEC Raipur, Sejbahar (C.G),CSVTU Bhilai, India

Abstract: Detection and recognition of traffic signs is very important and could potentially be used for driver assistance to reduce accidents and eventually in driverless automobiles.Also traffic signs are essential part of day to day lives. They contain critical information that ensures the safety of all the people . As there are number of traffic signs throughout the world , it is almost impossible for human beings to remember them and identity their meaning which create huge traffic accidents and human loss throughout the world so it is important to establish this project that will remember the traffic signs of all the country throughout the world. Traffic signs classification is the process of identifying which class a traffic sign belongs to. In this project with the help of deep learning , different traffic signs are identified and classified into different categories which helps in reducing various traffic accidents and also reduces human time to remember different traffic signs . In this paper , traffic sign recognition using Convolutional Neural Network(CNN) is implemented ,the CNN will be trained by using GTSRB dataset of 43 different classes containing 50,000 images of traffic signs The results will show 94% accuracy.
Keywords: Convolutional Neural Network, Traffic Sign Recognition, Tensorflow, Keras.

I. INTRODUCTION

Traffic sign detection and recognition has gained importance with advances in image process thanks to the advantages that such a system could give. The recent developments and interest in self-driving cars has conjointly exaggerated the interest during this field. A traffic sign detection and recognition system can give the flexibility for good cars and good driving. In self-driving cars, many passengers fully depend on the car for traveling. But to achieve level 5 autonomous, it is necessary for vehicles to understand and follow all traffic rules. In the world of Artificial Intelligence and advancement in technologies, many researchers and big companies like Tesla, Uber etc are working on autonomous vehicles and self-driving cars. So, for achieving accuracy in this technology, the vehicles should be able to interpret traffic signs and make decisions accordingly.

Without traffic signs, all the drivers would be clueless about what might be ahead to them and roads can become a mess. The annual global roach crash statistics say that over 3,280 people die every day in a road accident. These numbers would be much higher in case if there were no traffic signs. On the other hand, researchers and big companies are working extensively on proposing solutions to self-driving cars like Tesla, Uber, Google, Audi, BMW, Ford, Toyota, Mercedez, Volvo, Nissan, etc. These autonomous vehicles need to follow the traffic rules and for that, they have to understand the message conveyed through traffic signs.

There are several different types of traffic signs like speed limits, no entry, traffic signals, turn left or right, children crossing, no passing of heavy vehicles, etc. Traffic signs classification is the process of identifying which class a traffic sign belongs to. The goal of the Traffic Sign Recognition is to build a deep neural network model that can classify traffic signs present in the image into different categories. With this model, different traffic signs are read , understand and classified into different classes which are a very important task for all autonomous vehicles. The earlier Computer Vision techniques required lots of hard work in data processing and it took a lot of time to manually extract the features of the image. Now, deep learning techniques have come to the rescue and using this a traffic recognition system for autonomous vehicles is build.The dataset used for this project is the GTSRB (German traffic sign recognition benchmark). It contains a Train folder that has traffic sign images in 43 different classes, a Test folder that has over 12,000 images for testing purposes. A test.csv file that contains the path of the test images along with their respective classes. To implement this project , Keras will be used which is a popular deep learning framework for python and some additional library scikit-learn, numpy, PIL, pandas, Tkinter and jupyternotebook.

II. LITERATURE SURVEY

There are many researches in the literature dealing with Road TSR problem :

A. Narejo, S., Talpur, S., Memon, M., y Rahoo, A. (2020):" the authors proposed an automated system for traffic sign recognition using convolutional neural network.". They applied some techniques on the data for exploration and then visualized the data through EDA techniques. And finally, they build in this study a classification model based on the CNN. Afterwards, model is trained and validated and then based on the validated model, they attempted a test. And afterwards, they deployed their classifier.

B. The authors proposed a Convolutional Neural Network and Support Vector Machines (CNN-SVM) method for traffic signs recognition and classification. The coloring used in this method is YCbCr color space which is input to the convolutional neural network to divide the color channels and extracting some special characteristics. SVM is then used for classification.

C. Alexander Shustanov and Pavel Yakimov:,implemented a Method for Traffic Sign Recognition with CNN using GPU. Training of the neural network is implemented using the TensorFlow library and massively parallel architecture for multithreaded programming CUDA. The entire procedure for traffic sign detection and recognition is executed in real time on a mobile GPU. The experimental results confirmed high efficiency of the developed computer vision system.

D. Mohamed Yusof Radzak implemented *an algorithm for detection of traffic sign using convert region of interest (ROI) polygon to region mask method. The algorithm detects the traffic sign from the images captured from different environment and different position angle. The proposed method extracted the detected sign in black and white pixels and further classified into groups.They introduces the main difficulties in road sign recognition with further discussion on the potential trend of road sign recognition.*

E. Danyah A. Alghmghama , Ghazanfar Latif a, b*, Jaafar Alghazo a , Loay Alzubaidi ,proposed Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN .The proposed system works in real time detecting and recognizing traffic sign images. About 24 different traffic signs collected from random road sides in Saudi Arabia. The images were taken from different angles and including other parameters and conditions. A total of 2718 images were collected to form the database which we named Saudi Arabian Traffic and Road Signs (SA-TRS-2018). The CNN architecture was used with varying parameters in order to achieve the best recognition rates.

F. Attempted to make an early traffic sign recognition system. A system capable of automatic recognition of traffic sign could be used as assistance for drivers, alerting them about the presence of some specific sign (e.g. a one-way street) or some risky situation (e.g. driving at a higher speed than the maximum speed allowed). It also can be used to provide the autonomous unmanned some specific-designed signs. Generally, the procedure of a traffic sign recognition system can be roughly divided of two stages namely detection and classification.

1) *Detection:* The goal of traffic sign detection is to locate the regions of interest (ROI) in which a traffic sign is more likely to be found and verify the hypotheses on the sign's presence.
2) *Classification:* Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change their presentation of an image into something that is more meaningful and easier to analyse [5]. A binary image classification method is a digital image that has only two possible values for each pixel.

G. Kale and Mahajan (2015) the road sign recognition system is to be divided into two parts, the first part is detection stage which is used to detect the signs from a whole image, and the second part is classification stage that classifies the detected sign in the first part into one of the reference signs which are presents in the dataset. They used PCA (Principle Component Analysis) and ANN (Artificial Neural Network) techniques for detection and recognition with the dataset of different road signs from Maharashtra RTO (Regional Transport Office). Tool used by them was MATLAB.

H. Mrinal Haloi [2] defines a method for traffic sign recognition using a classification method called probabilistic latent semantic analysis. It is tested using the German sign traffic database. Shape classification and SIFT are initially used to represent the image and detect the traffic sign which is to be classified. PLSA uses a histogram of words concept, wherein each image is converted to a document based on some previously designed vocabulary. Further, the Fuzzy KNN classifier is used to retrieve an appropriate topic for the test image corresponding to the set of training images. 12000 images corresponding to 43 different signs are used. The PLSA method gave an overall recognition accuracy of 98.14%.

I. Yok-Yen Nguwi et al. target at developing an efficient and intelligent road sign recognition system based on the Singapore traffic sign database. A total of 2500 road and non-road signs have been used for training and testing. The non-road signs have been used to enhance the rejection capability of the system. Binary images are used as part of the system to keep the detection time minimal. Gaussian and laplacian derivatives are used to detect the edge and adaboost is used to remove the unnecessary parts of the image. Critical points in the image are located using the macro and micro SVM based independent component detectors. Gabor wavelets coupled with the Gini feature is used to shrink down the feature set. Support vector machines are used for classification. Using 200 samples for testing, an accuracy of 95% has been reported.

J. The authors developed a new dataset consisting of 100,000 images and also proposed a traffic sign detection and classification method based on a robust end-to-end CNN. The method achieved 84% accuracy.

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 V May 2021- Available at www.ijraset.com

K. The authors proposed an efficient algorithm for traffic sign detection on low cost embedded systems. Their method consists of color thresholding, shape detection and sign validation. They utilized an efficient color thresholding technique based on the red-blue angle color transformation (RBAT) and the red color normalized. Ellipse fitting technique is also employed for detecting the circular signs. HOG is employed for validation. Their method achieved 97% accuracy.

L. The authors analyzed the spatial transformers and stochastic optimization methods for deep neural network for traffic sign recognition. They finalized this with a proposed system that achieved 94% accuracy.

III. METHODOLOGY

A. CNN Image Classification

CNN is one of the category to do images recognition, images classification and that is what we are using for our model. A Convolutional Neural Network(CNN) is made up of convolutional and pooling layers. At each layer, the features from the image are extracted that helps in classifying the image.

Usually an activation function (ex-relu) will be applied to the convoluted values to increase the non-linearlity.

B. There are two main parts of a CNN Architecture

1) A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction.
2) A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

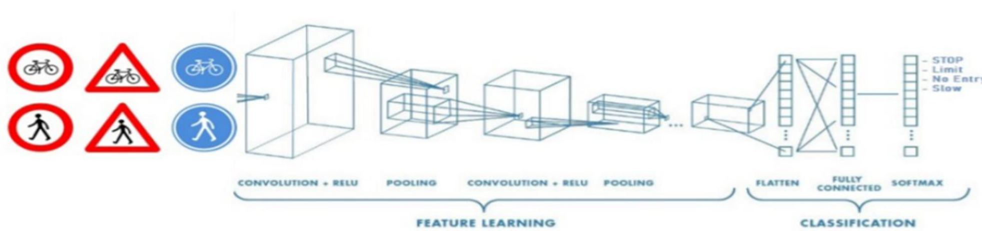


Fig:1 CNN Traffic Sign Classification

a) Convolution Layers: There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

- *Convolutional Layer:* This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM). The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.
- *Pooling Layer:* In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations. In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer
- *Fully Connected Layer:* The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

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 V May 2021- Available at www.ijraset.com

IV. PROPOSED APPROACH

In the paper, approach for building traffic sign classification is explained in following steps:

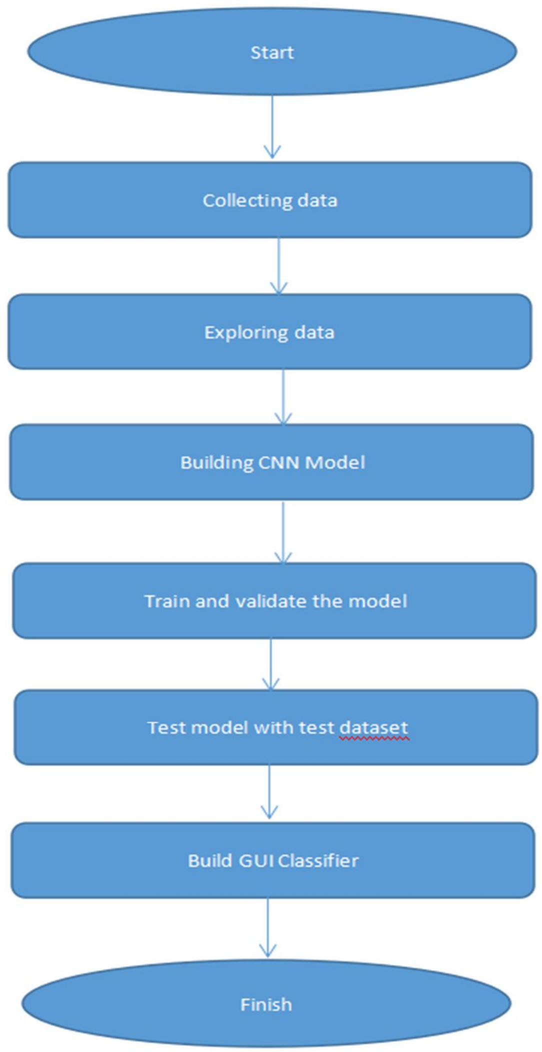


Fig. 2. Flowchart of Methodology

A. Explore & Preprocess Data

In this step, 'train' folder contains 43 folders each representing a different class. The range of the folder is from 0 to 42. With the help of the OS module, iterate over all the classes and append images and their respective labels in the data and labels list.

The PIL library is used to open image content into an array.Stored all the images and their labels into lists (data and labels). convert the list into numpy arrays for feeding to the model.The shape of data is (39209, 30, 30, 3) which means that there are 39,209 images of size 30×30 pixels and the last 3 means the data contains colored images (RGB value).With the sklearn package, use the train_test_split() method to split training and testing data. From the keras.utils package, use to_categorical method to convert the labels present in y_train and y_test into one-hot encoding.

B. Build a CNN model

To classify the images into their respective categories, build a CNN model (**Convolutional Neural Network**). CNN is best for image classification purposes.2 convolution layers followed by 2 max-pooling layers are added.Then there is a Flatten layer and finally, there are 2 dense layers and 3 dropout layers is used to handle the over-fitting of the model. The dropout layer drops some of the neurons while training but not when we are predicting.

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 V May 2021- Available at www.ijraset.com

The architecture of our model is:

1) 2 Conv2D layer (filter=32, kernel_size=(5,5), activation=”relu”)
2) MaxPool2D layer (pool_size=(2,2))
3) Dropout layer (rate=0.25)
4) 2 Conv2D layer (filter=64, kernel_size=(3,3), activation=”relu”)
5) MaxPool2D layer (pool_size=(2,2))
6) Dropout layer (rate=0.25)
7) Flatten layer to squeeze the layers into 1 dimension
8) Dense Fully connected layer (256 nodes, activation=”relu”)
9) Dropout layer (rate=0.5)
10) Dense layer (43 nodes, activation=”softmax”)

Model is compiled with Adam optimizer which performs well and loss is “categorical_crossentropy” because we have multiple classes to categorise.

C. Train and Validate the Model

Now the model is defined and the data is ready. To start the training of our model model.fit() function is used which takes the training set, validation set, batch size and no of epochs. After training the model for 15 epochs , save the model in a traffic_recognition.h5 file.

```
Epoch 1/20
981/981 [==============================] - 104s 106ms/step - loss: 2.3886 - accuracy: 0.3940 - val_loss: 1.1408 - val_accuracy:
0.6991
Epoch 2/20
981/981 [==============================] - 104s 106ms/step - loss: 1.1947 - accuracy: 0.6447 - val_loss: 0.5877 - val_accuracy:
0.8406
Epoch 3/20
981/981 [==============================] - 103s 105ms/step - loss: 0.8536 - accuracy: 0.7407 - val_loss: 0.3868 - val_accuracy:
0.8936
Epoch 4/20
981/981 [==============================] - 101s 103ms/step - loss: 0.7331 - accuracy: 0.7767 - val_loss: 0.3278 - val_accuracy:
0.9028
Epoch 5/20
981/981 [==============================] - 101s 103ms/step - loss: 0.6102 - accuracy: 0.8145 - val_loss: 0.2362 - val_accuracy:
0.9341
Epoch 6/20
981/981 [==============================] - 120s 122ms/step - loss: 0.4675 - accuracy: 0.8584 - val_loss: 0.1419 - val_accuracy:
0.9601
Epoch 7/20
981/981 [==============================] - 109s 111ms/step - loss: 0.3842 - accuracy: 0.8848 - val_loss: 0.1517 - val_accuracy:
0.9526
Epoch 8/20
981/981 [==============================] - 118s 121ms/step - loss: 0.3288 - accuracy: 0.9004 - val_loss: 0.0939 - val_accuracy:
0.9716
Epoch 9/20
981/981 [==============================] - 98s 100ms/step - loss: 0.3078 - accuracy: 0.9095 - val_loss: 0.0993 - val_accuracy:
0.9727
Epoch 10/20
981/981 [==============================] - 114s 117ms/step - loss: 0.2727 - accuracy: 0.9198 - val_loss: 0.0839 - val_accuracy:
0.9774
```

After 20 epochs ,94% accuracy is achieved .

D. Plot the Accuracy Graph

With the help of matplotlib functions,graph of training and validation accuracy is plotted.

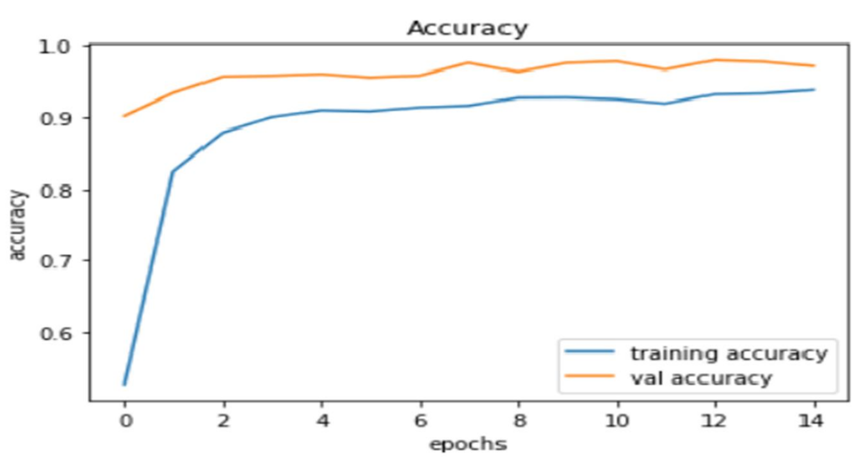


Fig 3. Accuracy of trained model CNN

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 V May 2021- Available at www.ijraset.com

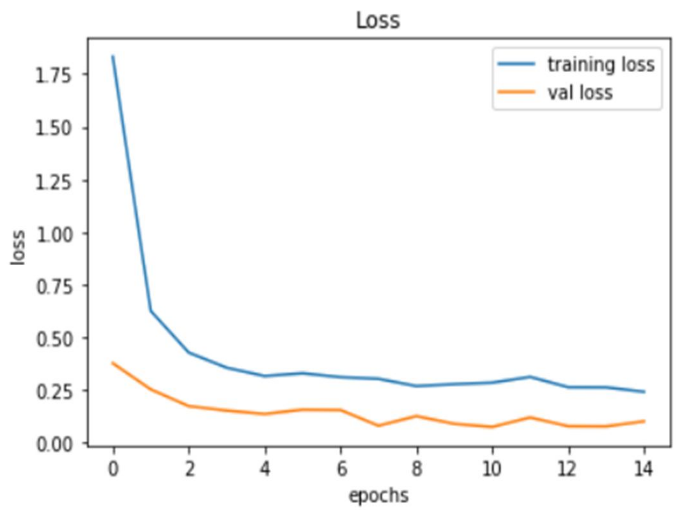


Fig 4. Loss of trained model CNN

E. Test Our Model with test Dataset

Then model is tested having test folder that contains around 12,000 images. The test.csv file contains the path of the image along with the label of the class. Iimage path and labels using pandas are extracted. Then to predict the model, resize images to 30×30 pixels and make a numpy array containing all image data. From the sklearn.metrics, import accuracy_score and observed how model predicted the actual labels. 98% accuracy in testing model is achieved.

V. RESULTS

A. Build ing a Graphical User Interface (GUI):

Build a graphical user interface for traffic signs classifier with Tkinter. Tkinter is a GUI toolkit in the standard python library. Then make a new file in the project folder Save it as gui.py and then run the code by typing python gui.py in the command line.

In this file, load the trained model 'traffic_classifier.h5' using Keras. And then build the GUI for uploading the image and a button is used to classify which calls the classify() function. The classify () function is converting the image into the dimension of shape (1, 30, 30, 3). This is because to predict the traffic sign , same dimension which was used when building the model is provided. Then predict the class, the model.predict_classes (image) returns us a number between (0-42) which represents the class it belongs to. Use the dictionary to get the information about the class.

Fig 5. Showing results of General caution.

Fig. 6. Showing results of Stop signal.

VI. CONCLUSION

In this research work,an efficient traffic sign detection and recognition system is developed. The dataset went through a preprocessing ,building CNN model and training and testing stages. It got partitioned into training, testing and validating datasets. The final Deep CNN architecture proposed in this work consists of two convolutional layers, two maxpooling layers ,three dropout layer and 2 dense layers. .We have successfully classified the traffic signs classifier with 94% accuracy in 20 epochs and visualized how our accuracy and loss changes with time, which is pretty good from a simple CNN model. The techniques implemented in this research can be used as a basis for developing general purpose, advanced intelligent traffic surveillance systems. Future work will include increasing the size of the dataset and publishing it so that it can be used by other researchers for benchmarking purposed.

VII. ACKNOWLEDGEMENT

We wish to express our sincere gratitute to Dr. R.S. Parihar, Principal and Dr. R. H. Talwekar ,H.O.D of Department of Computer Science & Engineering(CSE) of Govt. Engineering College(GEC) Raipur,for providing us an opportunity to do our project work on “Traffic Sign Recognition”.We also sincereourly thank our project guide Ms.Monika Sahu for her guidance and encouragement in carring out this reasearch work.

RREFERENCES

[1] Narejo, S., Talpur, S., Memon, M., y Rahoo, A. (2020):” An automated system for traffic sign recognition using convolutional neural network.” 3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Noviembre 2020.

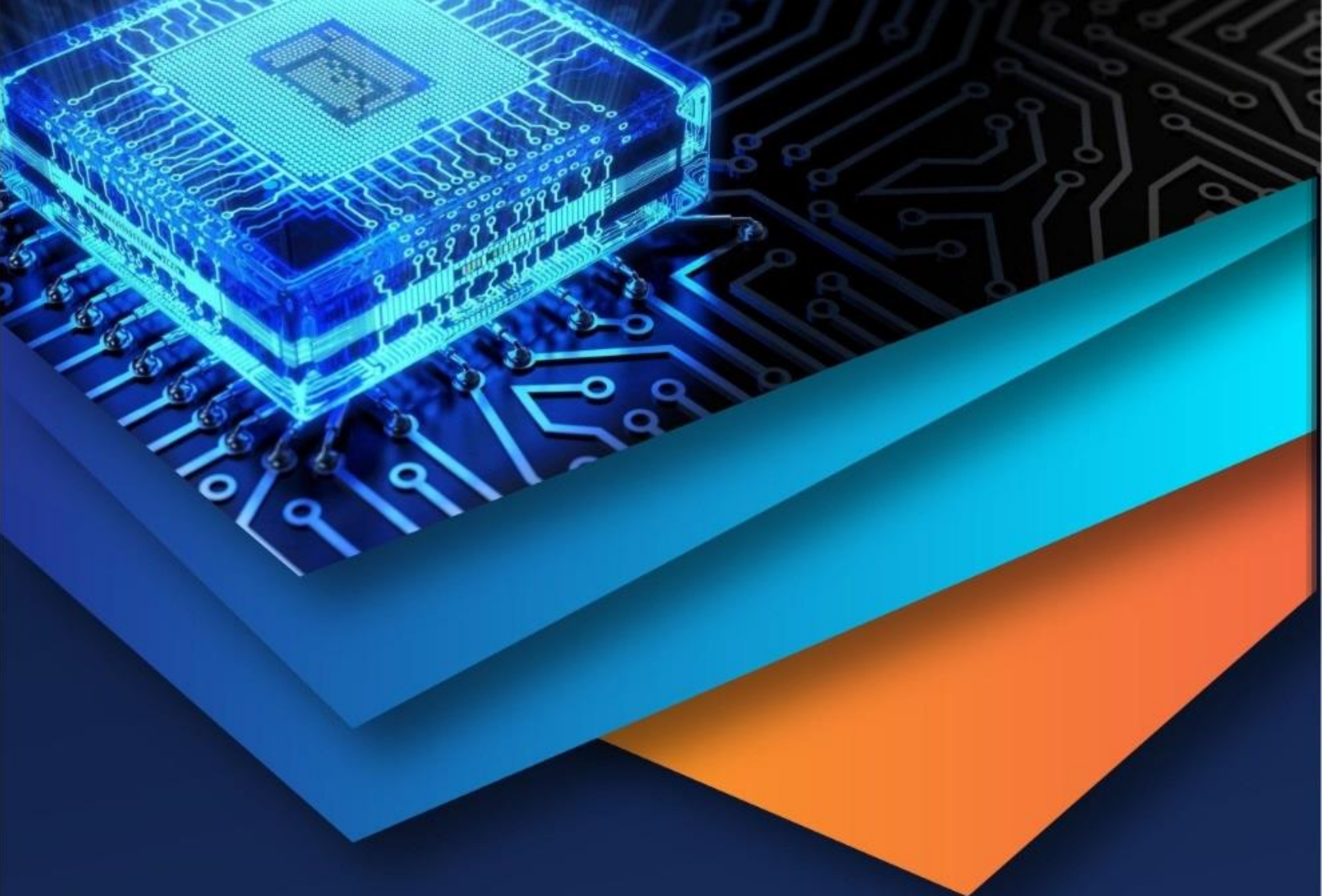
[2] Arasanipalai Mohan Aditya1, Sagar Moharir2 :”Study of Traffic Sign Detection and Recognition Algorithms”1, 2National University of Singapore, Institute of Systems Science, 25 Heng Mui Keng Terrace, 65162093, Singapore.

[3] Alexander Shustanov and Pavel Yakimov:”A Method for Traffic Sign Recognition with CNN using GPU”Samara National Research University, Moskovskoye shosse 34, Samara, Russia .

[4] Mohamed Yusof Radzak Electrical, Electronics and Automation Section, Universiti Kuala Lumpur Malaysian Spanish Institute, Kedah, Malaysia.”Study on Traffic Sign Recognition International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) Volume 2, Issue 6, June 2015, PP 33-39”.

[5] Danyah A. Alghmghama , Ghazanfar Latif a, b*, Jaafar Alghazo a , Loay Alzubaidi a .”Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN ”College of Computer Engineering and Sciences, Prince Mohammad bin Fahd University, Khobar, Saudi Arabia. b Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Malaysia .

[6] Arjun Dieep*1, Ivin Johnson k*2, Ashley Benny*3:”TRAFFIC SIGN DETECTION BASED ON CONVOLUTIONAL NEURAL NETWORK” International Research Journal of Modernization in Engineering Technology and Science Rosemol Xavier, Asst. Professor, Department of Computer Science, De Paul Institute of Science & Technology, Angamaly, Kerala, India..



10.22214/IJRASET

INDEX COPERNICUS
45.98

ISRA JIF
IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY