SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES(AUTONOMUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TRAFFIC SIGN DETECTION FOR AUTOMATIC SELF DRIVING CAR SYSTEM USING DEEP LEARNING

UNDER THE GUIDANCE OF:
M.PUSHPANJALI(M.TECH)

PRESENTED BY:

M.PAVANI(19751A0568)

T SAKTHI SWARUPA(19751A05B9)

V HARSHITHA(19751A05C6)

Y ROHITHA(19751A05C7)

Contents:

Abstract:

- The use of deep learning models for traffic sign detection has become increasingly popular in recent years, particularly in the field of autonomous driving.
- In this approach, a dataset of traffic sign images is collected and preprocessed, and a deep learning model, such as a convolutional neural network (CNN), is trained on the data.
- The model is then evaluated on a test set and deployed in a real-world scenario. By using deep learning models, traffic signs can be detected and recognized with high accuracy and speed, making them an ideal solution for real-time applications.
- Overall, traffic sign detection using deep learning models is a challenging but promising area of research that has the potential to improve road safety and facilitate the development of autonomous driving technologies.

Introduction:

- Traffic sign detection is a critical task in the field of computer vision, particularly in the context of autonomous driving. The ability to accurately detect and recognize traffic signs in real-time can greatly enhance the safety and efficiency of transportation systems.
- Traditional approaches to traffic sign detection relied on handcrafted features and machine learning algorithms, but these methods often struggled to achieve high accuracy and robustness.
- In recent years, the use of deep learning models has revolutionized the field of computer vision, including traffic sign detection.
- Deep learning models, such as convolutional neural networks (CNNs), are capable of automatically learning features from raw data, making them well-suited for tasks such as traffic sign detection.
- By training a CNN on a large dataset of traffic sign images, the model can learn to recognize traffic signs with high accuracy and speed, even in complex and challenging environments.
- We will discuss the steps involved in building a traffic sign detection system using deep learning models. We will cover data collection and preprocessing, model selection and training, evaluation, and deployment.

Literature Review

- **TITLE:** Traffic Flow Prediction With Big Data: A Deep Learning Approach
- **YEAR:** 2020
- **AUTHOR:** Lv Yisheng, Yanjie Duan.
- **ABSTRACT:**
- Accurate and timely traffic flow information is important for the successful deployment of intelligent transportation systems. Over the last few years, traffic data have been exploding, and we have truly entered the era of big data for transportation.
- Existing traffic flow prediction methods mainly use shallow traffic prediction models and are still unsatisfying for many real-world applications.
- This situation inspires us to rethink the traffic flow prediction problem based on deep architecture models with big traffic data. In this paper, a novel deep-learning-based traffic flow prediction method is proposed, which considers the spatial and temporal correlations inherently.
- A stacked autoencoder model is used to learn generic traffic flow features, and it is trained in a greedy layerwise fashion. To the best of our knowledge, this is the first time that a deep architecture model is applied using autoencoders as building blocks to represent traffic flow features for prediction. Moreover, experiments demonstrate that the proposed method for traffic flow prediction has superior performance

Cont……

- **TITLE:** Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN
- **YEAR:** 2019
- **AUTHOR:** Danyah A. Alghmghama , Ghazanfar Latif, Jaafar Alghazo, Loay Alzubaidi
- **ABSTRACT:**
- Accidents and eventually in driverless automobiles. In this paper, Deep Convolutional Neural Network (CNN) is used to develop an Autonomous Traffic and Road Sign (ATRS) detection and recognition system. The proposed system works in real time detecting and recognizing traffic sign images.

- The contribution of this paper is also a newly developed database of 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. 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. Experimental results show that the proposed CNN architecture achieved an accuracy of 100%, thus higher than those achieved in similar previous studies

Existing Methodology

- The existing system of traffic sign detection includes traditional computer vision techniques, such as color segmentation, edge detection, and template matching.
- By using Logistic regression algorithm, the code made on.
- It is one the most administered and managed machine learning algorithm.
- Based on inputs i.e., images, the algorithm predicts the real valued output.
- Traffic sign prediction is based on these inputs. There are many advantages of traffic management, the very important being reduction in accident and in time consumption.

Disadvantages

- Accuracy is low
- Time consumption is high
- Due to low accuracy, not reliable

Proposed Methodology:

- Matrix called "Activation Map" or "Feature Map". The output layer is made up of several convolutional layers that extract features from the image. CNN can be optimized with the help of hyper parameter optimization. It finds hyper parameters of a given deep learning algorithm that deliver the best performance as measured on a validation set. Hyper parameters must be set before the learning process can begin. The learning rate and the number of units in a dense layer are provided by it. Convolutional Neural Network Architecture :

1.Convolution Layer:

- This layer is major building block in convolution process. It performs convolution operation to identify various features from given image. It basically scans entire pixel grid and perform dot product. Filter or kernel is nothing but a feature from multiple features which we want to identify from input image.

2. Pooling Layer

- This layer is used for down sampling of the features. It reduces dimensionality of large image but still retains important features. It helps to reduce amount of computation and weights. One can choose Max pooling or Average pooling depending on requirement. Max pooling takes maximum value from feature map while average takes average of all pixels

3.Activation Function

- This layer introduce non linear properties to network. It helps in making decision about which information should be processed further and which not. Weighted sum of input becomes input signal to activation function to give one output signal.This step is crucial because without activation function output signal would be simple linear function which has limited complex learning capabilities.

4.Flattening Layer

- The output of the pooling layer is in the form of a 3D feature map, and we need to transfer data to the fully connected layer in the form of a 1D feature map. As a result, this layer transforms a 3*3 matrix to a one-dimensional list.

5. **Fully connected Layer**

- Actual classification happens in this layer. It takes end result of convolution or polling layer by flattened layer and reaches a classification decision. Here every input is connected to every output by weights .It combines the features into more attributes that better predicts the classes

Advantages

- Accuracy is improved.
- Data extraction process is highly load.
- Pattern recognition is defined deeply through pooling layer.
- Time consumption is low.

System Requirement:

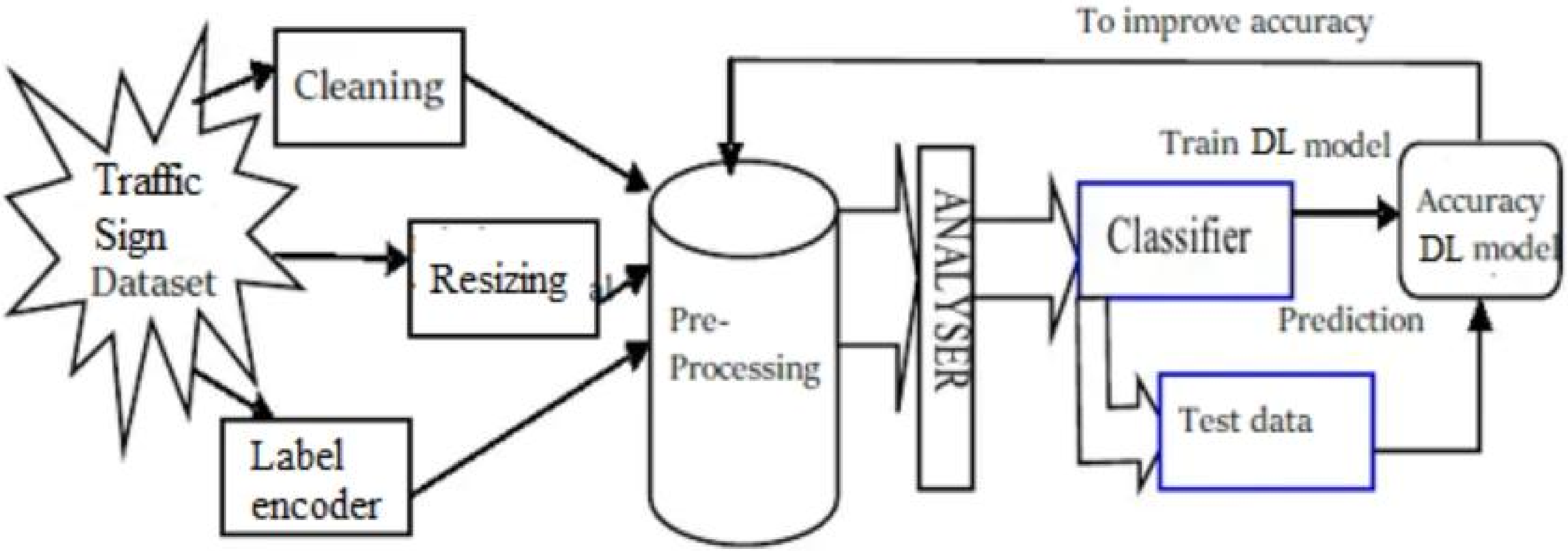
Software Requirements

▪	IDE	:	Anaconda ,Jupyter
▪	Programming Language	:	Python

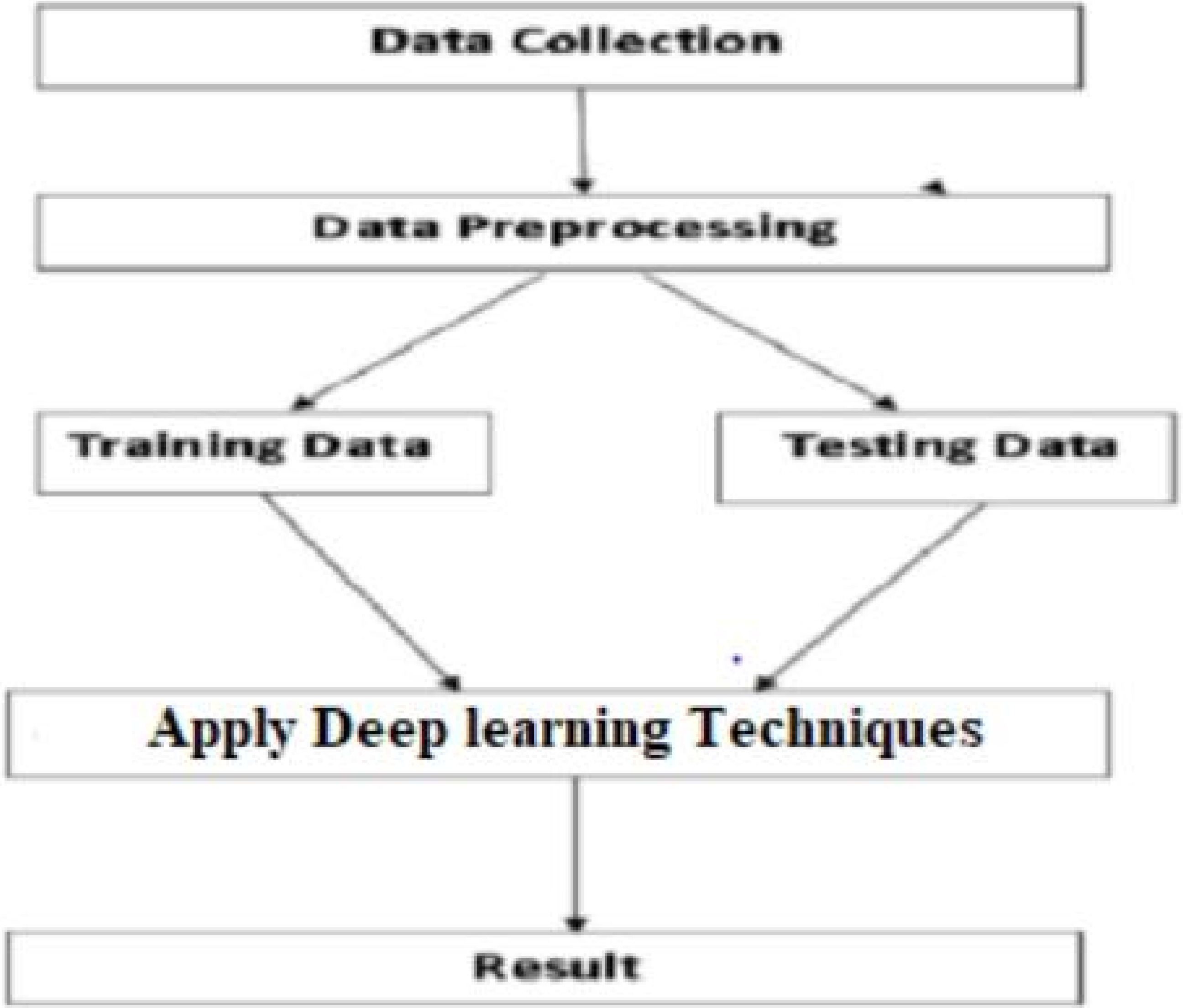
Hardware Requirements

▪	PROCESSOR	:	Intel i5(min)
▪	RAM	:	4 GB DD RAM
▪	HARD DISK	:	250 GB

Architecture Diagram:



Data Flow Diagram:



Modules:

Data Collection:

- A rich dataset is needed in object recognition based on neural network in order to train the system and evaluate its result.

  For the purpose of traffic signs classification, we used the German Traffic Sign Benchmark (GTSB).
- The reason for this is largely due to the lack of traffic sign data sets. Training and verifying a deep convolution neural network traffic sign recognition model requires a large amount of traffic sign data as a basis.
- However, the open traffic sign datasets in India is relatively scarce compared with developing countries. More well-known traffic sign datasets now include GTSRB in Germany, GTSDB in Germany and KUL in Belgium. In this paper, GTSRB, GTSDB traffic sign datasets is used to traffic sign detection and recognition.
- These datasets include many types of complex traffic signs such as sign tilt, uneven lighting, traffic sign with distraction, occlusion and similar background colors, as well as actual scene maps.

Data Augmentation:

Since the image classes are heavily imbalanced, we augment the training data to get balanced distribution among the classes. We mirror and rotate the images to create new augmented data set.

Splitting of data:

After cleaning the data, data is normalized in training and testing the model. When data is spitted then we train algorithm on the training data set and keep test data set aside. This training process will produce the training model based on logic and algorithms and values of the feature in training data. Basically aim of feature extraction is to bring all the values under same scale. A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

Training set:

A training set is a portion of a data set used to fit (train) a model for prediction or classification of values that are known in the training set, but unknown in other (future) data. The training set is used in conjunction with validation and/or test sets that are used to evaluate different models.

Test set:

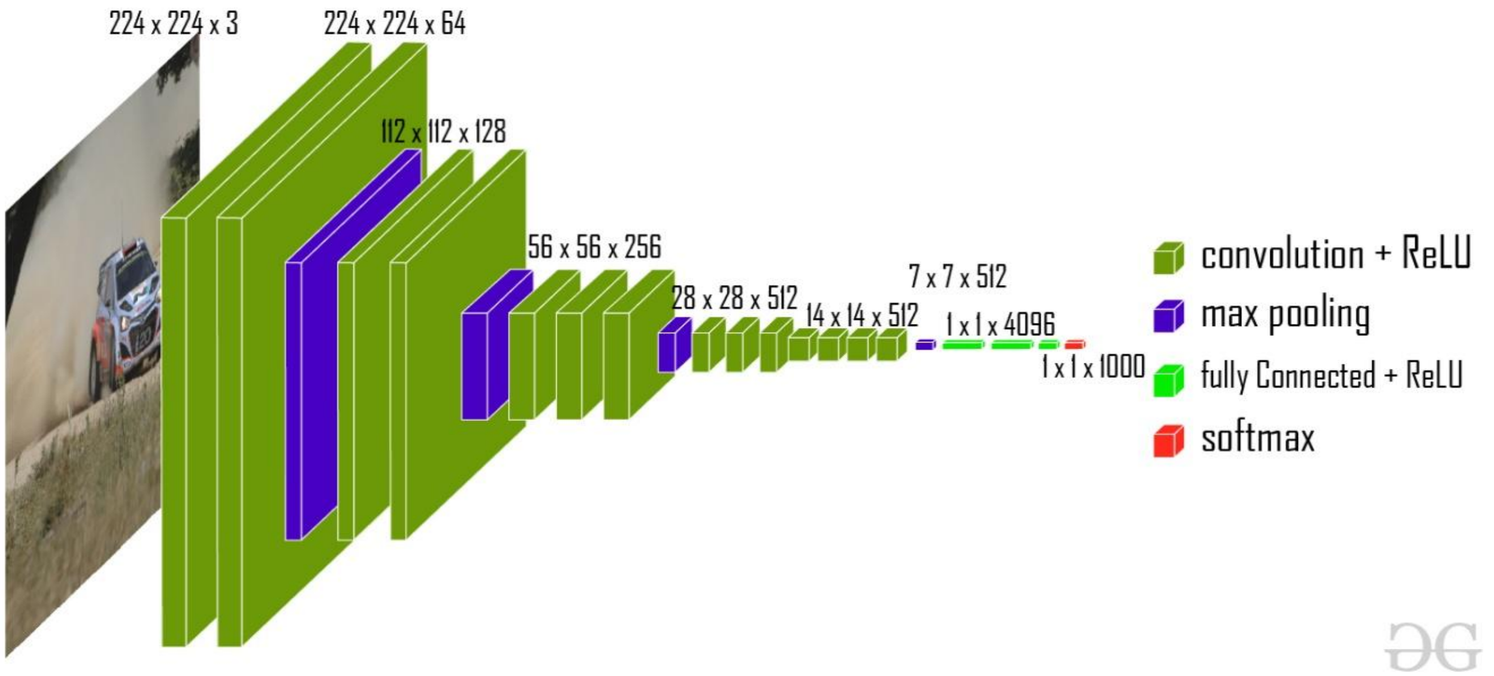
- A test set is needed for an evaluation of the trained model and its capability for generalization.
- The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It's crucial to use different subsets for training and testing to avoid model over fitting, which is the incapacity for generalization we mentioned above.

Validation Dataset:

- The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters.
- The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

Pre-trained CNN Model (VGG-16):

- VGG(Visual Geometry Group) -16 is a convolutional neural network that is 16 layers deep.
- You can load a pretrained version of the network trained on more than a million images from the ImageNet database .
- The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.



For building this model, we will use Tensor Flow to stack up the VGG-16 model and freeze the convolution blocks so we can utilize it as a feature extractor.

- This model achieves *92.7% top-5* test accuracy on ImageNet dataset which contains *14* million images belonging to 1000 classes.

- The ImageNet dataset contains images of fixed size of *224*224* and have RGB channels. So, we have a tensor of *(224, 224, 3)* as our input. This model processes the input image and outputs a vector of *1000* values.

Performance Matrices:

- Data was divided into two portions, training data and testing data, both these portions consisting 70% and 30% data respectively. All these two algorithms were applied on same dataset using Enthought Canaopy and results were obtained.

- Predicting accuracy is the main evaluation parameter that we used in this work. Accuracy can be defied using equation. Accuracy is the overall success rate of the algorithm**. Accuracy=(TP+TN) / (P+N)**

Confusion Matrix:

- It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc.

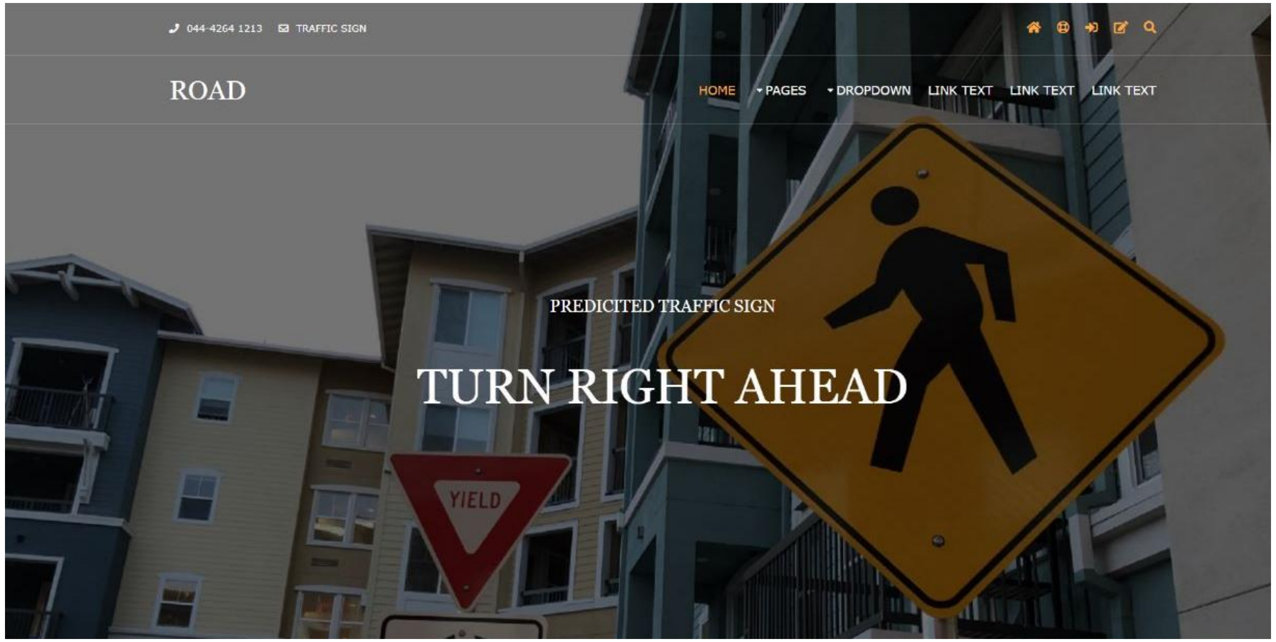
- It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem.

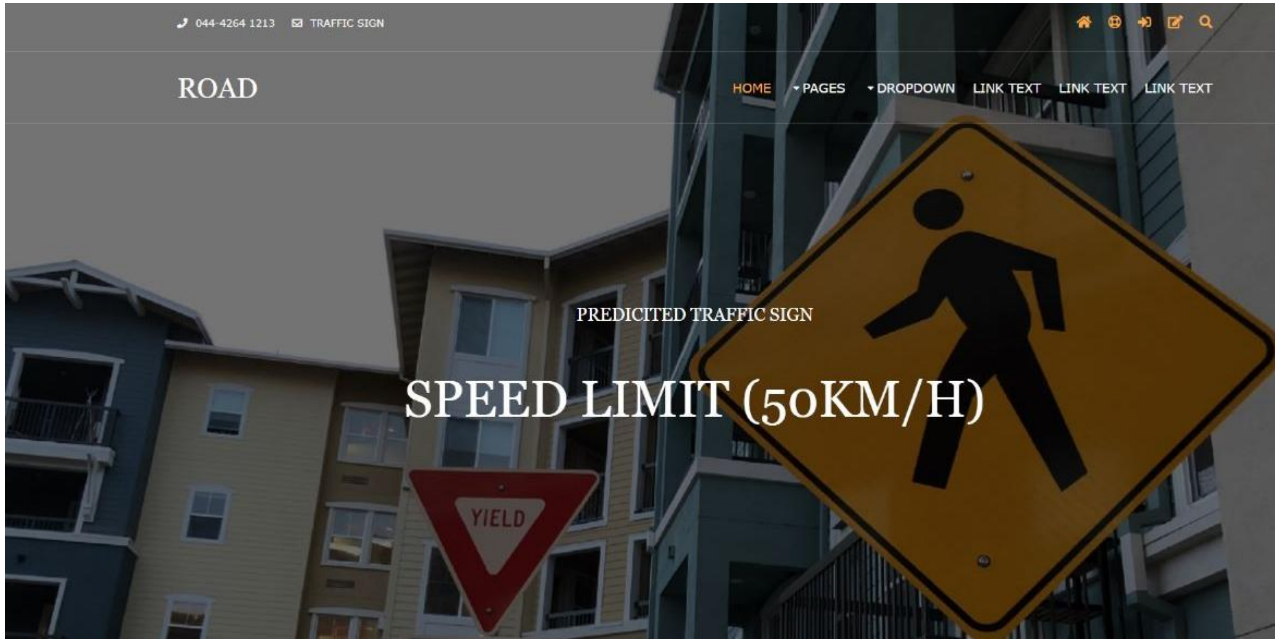
Output

044-4264 1213
TRAFFIC SIGN
ROAD
HOME
PAGES
DROPDOWN
LINK TEXT
LINK TEXT
LINK TEXT
TRAFFIC SIGN PREDICITION
Predict the sign on the road using Transfer Learning.
Choose File No file chosen
Image CNN Algotithm
Predict

044-4264 1213
TRAFFIC SIGN
ROAD
HOME
PAGES
DROPDOWN
LINK TEXT
LINK TEXT
LINK TEXT
TRAFFIC SIGN PREDICITION
Predict the sign on the road using Transfer Learning.
Choose File 00001.png
Image VGG16 Algotithm
Predict

044-4264 1213
TRAFFIC SIGN
ROAD
HOME
PAGES
DROPDOWN
LINK TEXT
LINK TEXT
LINK TEXT
TRAFFIC SIGN PREDICITION
Predict the sign on the road using Transfer Learning.
Choose File 00001.png
Image CNN Algotithm
Predict
CNN
VGG16


044-4264 1213
TRAFFIC SIGN
ROAD
HOME
PAGES
DROPDOWN
LINK TEXT
LINK TEXT
LINK TEXT
PREDICITED TRAFFIC SIGN
TURN RIGHT AHEAD
YIELD


044-4264 1213
TRAFFIC SIGN
ROAD
HOME
PAGES
DROPDOWN
LINK TEXT
LINK TEXT
LINK TEXT
PREDICITED TRAFFIC SIGN
SPEED LIMIT (50KM/H)
YIELD

References :

- Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. Proceedings of the International Conference on Neural Information Processing Systems (NIPS).

- Zhao, H., Shi, J., Qi, X., Wang, X. (2018). Real-Time Traffic Sign Recognition System for Intelligent Vehicles. IEEE Transactions on Intelligent Transportation Systems, 19(3), 864-875.

- Ciresan, D., Meier, U., Gambardella, L., Schmidhuber, J. (2012). Deep, big, simple neural nets for handwritten digit recognition. Neural computation, 22(12), 3207-3220.

- Gupta, A., Mittal, A., Sharma, S. (2019). Traffic sign detection and recognition using deep learning and image segmentation. International Conference on Communication, Computing and Electronics Systems (ICCCES).

- Zhang, X., Zhang, L., Sun, Y. (2019). Deep Learning Based Traffic Sign Detection and Recognition: A Survey. IEEE Access, 7, 137027-137043.
- Ma, J., Li, Y., Wang, H., Yang, L. (2021). Traffic Sign Recognition Based on Deep Learning: A Review. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3777-3796.

Thank you!