**Topic:** Traffic Sign Recognition in Selfless Driving Systems

## 1.0 Introduction

To increase traffic efficiency and safety, there has been an increase in interest in creating autonomous driving assistance systems in recent years. The use of visual information to recognise and understand traffic signs, moving objects, pedestrians, and other things in the driving environment has been a major area of interest for many manufacturers and academics. Traditional methods have depended on well-organized models of traffic signs, but new research has shown how crucial it is to consider actual aspects of the road environment.

The ability to create more advanced algorithms that can manage changes in lighting, weather, and perspective distortions that may impact the look of road signs has been made possible, in particular, by developments in computer vision and machine learning. For instance, current research has looked into the use of deep learning models to identify traffic signs in real-time video feeds, even in difficult situations like dim lighting or signs that are partially obscured.

An essential part of contemporary intelligent transportation systems (ITS) and autonomous driving is traffic sign recognition. It seeks to accurately recognise and categorise traffic signs, giving drivers, pedestrians, and other road user's crucial information. The multitude of sign sizes, colours, and designs, along with the complicated and ever-changing driving conditions, makes traffic sign detection and recognition a difficult subject. However, great progress has been achieved in the creation of efficient and dependable traffic sign recognition systems as a result of current developments in computer vision and deep learning.

The recent years have seen a huge increase in the use of traffic sign detection and recognition (TSDR). This is because such a system can offer a wide range of application such as upkeep of signs, driving support, intelligent autonomous vehicles and the likes.

Advances in artificial intelligence (AI), particularly in the areas of computer vision and machine learning, are crucial for the development of traffic sign recognition systems. Even in challenging and dynamic driving settings, AI techniques, such as deep learning, have shown substantial effectiveness in accurately and consistently classifying traffic signs because they can automatically develop discriminative features from raw image data without the requirement for manually created feature engineering, deep learning-based approaches have grown in popularity for TSDR.

High classification accuracy is achieved by these algorithms because they are able to grasp the intricate connections between the input images and related sign categories.

Traffic sign recognition has a number of potential advantages for improving efficiency and road safety. First of all, it can aid in lowering the amount of collisions brought on by driver error or a failure to pay attention to traffic signs.

## 2.0 Related works

Blauth et al. (2012) demonstrated a large-scale traffic sign identification system based on regional attributes and colour segmentation. The regions of interest (ROI) in the input image are first identified using a colour segmentation method in the proposed system, and they are later defined by regional characteristics including SIFT, SURF, and ORB. Following that, a support vector machine (SVM) classifier is fed the quantized local features using a bag-of-words (BoW) method.

On a dataset of German traffic signs, the system was able to attain excellent accuracy. However, this strategy has a number of drawbacks.

HOG descriptors are sensitive to changes in lighting and may not function well in low-light or high-contrast situations, which is one restriction. SVM classifiers' need for precise hyper parameter selection and perhaps poor generalization to new datasets are additional drawbacks.

The human feature selection requirement and time-consuming feature extraction method, which might not be ideal for all traffic signs, are two drawbacks of the proposed approach.

Researchers have suggested a few upgrades to solve these constraints. As an illustration,

Wu et al. (2016) suggested a multi-scale HOG feature extraction technique that increased the HOG descriptors' resistance to variations in lighting.

Li et al. (2016) suggested a hybrid feature extraction technique that incorporated HOG descriptors with colour and texture features. On a dataset of Chinese traffic signs, the method outperformed other cutting-edge techniques and produced results with excellent accuracy. There are, however, drawbacks to this strategy.

One drawback is that depending on the particular dataset and kinds of traffic signs, the effectiveness of the hybrid feature extraction method may vary. The technique may also be sensitive to variations in illumination and occlusion, which may have an impact on the aspects of colour and texture.

Deep residual learning framework ResNet, developed by He et al. (2016), dramatically enhanced deep neural network performance for image identification applications. By introducing residual connections, which enable the direct flow of information from one layer to another, omitting several intermediate layers, ResNet tackles the problem of disappearing gradients.

With this method, it is possible to train incredibly deep neural networks with up to 152 layers and yet obtain cutting-edge accuracy on a variety of image classification benchmarks.

ResNet has demonstrated higher performance in a number of computer vision applications, such as object detection and image recognition, but it also has certain drawbacks. ResNet has a number of drawbacks, including a high computational cost and memory requirement that might make deployment on devices with limited resources challenging.

It may also be more challenging to evaluate the learned features and comprehend the underlying network mechanics when residual connections are used.

An artificial neural network (ANN) classifier and a hybrid feature descriptor are used in the traffic sign identification system Abedin et al. (2016) presented. Histograms of oriented gradients (HOG), local binary patterns (LBP), and colour histograms make up the hybrid feature descriptor.

Using the features that were taken out of the input image, the ANN classifier is trained. A 97.5% accuracy rating for the suggested method was obtained when it was tested against the GTSRB dataset. The manual feature selection, which could not be the best option for all traffic signs, and the use of a single ANN classifier, which might not generalise effectively to new data, are two shortcomings of the proposed approach.

Researchers have suggested a few upgrades to solve these constraints. For instance, Zhang et al. (2018) proposed a system for recognising traffic signs that incorporated deep convolutional neural networks (CNNs) and HOG descriptors. On a big dataset of traffic signs, the system produced good accuracy and outperformed earlier methods in terms of resistance to changes in illumination and occlusion.

The ability of traffic sign recognition systems to generalise is improved by the application of domain adaption techniques. A domain adaptation strategy, for instance, was proposed by Zhang et al. (2019) to adjust a pre-trained CNN model to new domains of traffic signs with various distributions of forms, colours, and backgrounds.

Overall, while Li et al. (2016)'s hybrid feature extraction technique has showed promising for traffic sign identification, there are still issues with this strategy that need to be resolved in order to increase its robustness and generalisation skills. The development of more sophisticated feature extraction techniques and the investigation of domain adaption approaches should be the key goals of future study.

Researchers have also looked into using different classifiers, like neural networks. As an illustration, overall, the HOG descriptor and SVM classifier strategy has showed promise for recognizing traffic signs, but there are still issues that need to be resolved for it to become more robust and generic. Future work should concentrate on creating more sophisticated feature extraction techniques and investigating the application of different classifiers.

### 2.1. Objectives

1. The objective of this report is channeled towards Deep Learning Methods for Traffic Sign Recognition in Selfless Driving Systems: A Comparative Study on various deep learning models.
2. This study's goal is to create and assess deep learning algorithms for traffic sign recognition in difficult situations including dim lighting and partially covered signs.
3. Compare the models' output in terms of precision and computational efficiency.
4. The study took approximately 2 month for completion with the circumstances including dataset.

## 3.0 Methodology

### 3.1. Convolutional Neural Network

The convolutional layer, which extracts features from the input data to enable the CNN to learn patterns in the input data, is the most crucial layer in a CNN. Convolutional layer filters are programmable parameters that the CNN modifies during training to enhance network performance. A feature map that emphasises the areas of the input data that are similar to each filter is created when each filter is applied to the input data using a convolution operation. A stack of feature maps, each of which corresponds to a separate filter in the layer, is the output of the convolutional layer.

The feature maps created by the convolutional layer are made less dimensional by the pooling layer, which comes after it. By doing this, the network is made more effective and the number of parameters that must be learned is decreased. The feature maps' small rectangular sections are used by the pooling layer to combine them into a single value. Max pooling, which uses the maximum value in each region as the summary value, is the most popular type of pooling.

The output of the pooling layer is used by the fully connected layer to provide a prediction. Similar to hidden layers in other neural networks, this layer is made up of a group of neurons that are all coupled to those in the layer above. The output of the layer is determined by adding the weights assigned to each neuron in the fully connected layer, then applying an activation function to the weighted sum of the inputs.

In conclusion, CNNs are a potent variety of neural network that can benefit from the organisation of input images. In comparison to other neural networks, they are defined in a more logical manner and have more layers, including the convolutional layer, the pooling layer, and the fully connected layer. The pooling layer lowers the dimensionality of the feature maps, the convolutional layer retrieves features from the input data, and the fully connected layer produces a prediction. Together, these layers enable the CNN to recognise patterns in the incoming data and produce precise predictions.

## 3.2. Transfer Learning

In the realm of machine learning (ML), transfer learning is a well-known research subject that seeks to apply the knowledge discovered while completing one job to a related task. It entails applying a previously trained model to a fresh set of issues and making use of the understanding gained from one endeavour to enhance generalisation about another.

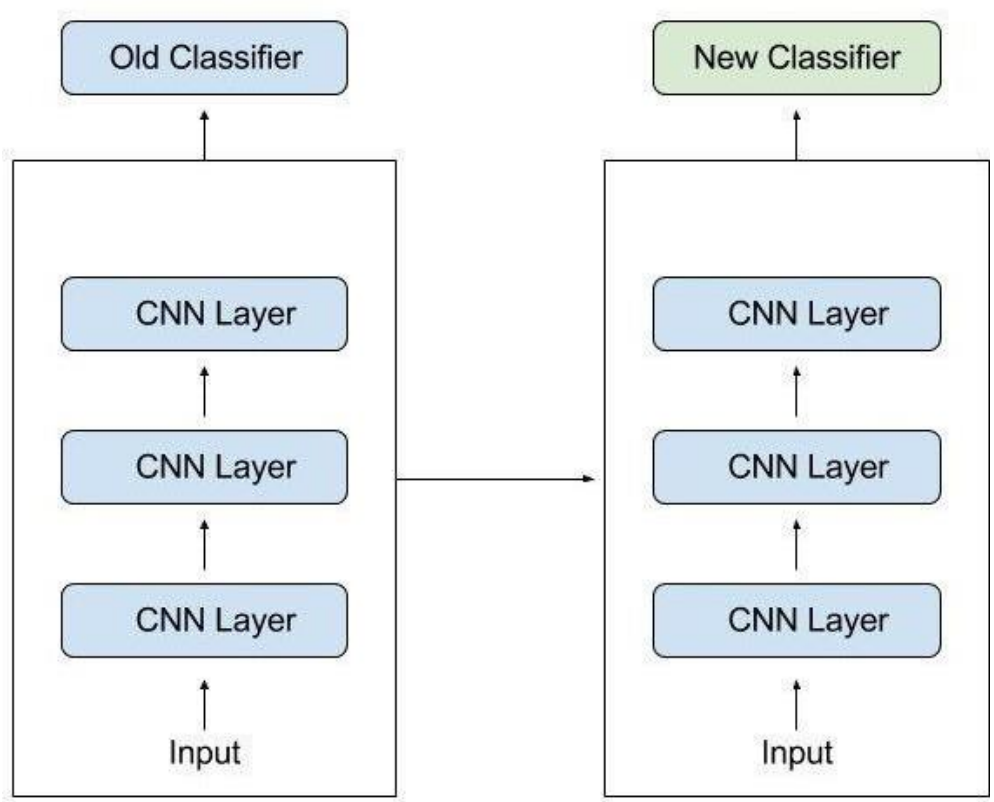



Fig. 1. Transfer Learning Architecture.

## 3.3. Model Architecture

Traffic sign recognition methods go through a number of stages throughout development, including data preprocessing, feature extraction, and classification. In the preprocessing stage, the input photos are improved for quality and clarity, noise and distortion are reduced, and changes in lighting and contrast are corrected. Finding pertinent properties, such as shape, colour, and texture that can aid discriminate between various sorts of signs is the task of the feature extraction step. During the classification stage, machine learning techniques are used to group the features into distinct types of traffic sign.

The codes were created and run using the open-source web application Jupyter Notebook. A user-friendly interface for creating and running Python programmes.

To make the deep learning models easier to implement, several libraries were imported. These libraries include Sklearn, Matplotlib, NumPy, and Keras. The deep learning models were created and trained using Keras, and the numerical calculations were performed using NumPy.

Sklearn was used to divide the dataset into training and testing sets, while Matplotlib was used to visualise the models' performance.

Utilising categorical accuracy, which gauges the proportion of correctly identified phot, the models were assessed. A confusion matrix, which shows the number of properly and erroneously identified photos for each class, was also used to visualise the performance of the models.

The process, in its entirety, includes importing pertinent libraries, creating and training deep learning models using Keras, refining the models, assessing the models' performance using category accuracy, and visualising it using a confusion matrix.

In order to train the system and assess its performance in object recognition using convolutional neural networks, a substantial dataset is required. For the classification of traffic signs.

# 4.0. Experiment

## 4.1. Data preprocessing

The dataset utilised for training and testing the deep learning models in the Traffic Sign Classification and Recognition project was adopted from Kaggle. In this dataset, there were more than 6,000 images of traffic signs from 10 distinct classes.

The images include a wide range of traffic sign kinds, including warning signs, regulation signs, and information signs.

The data were preprocessed to make sure they were suitable for input to the deep learning models before being used in the dataset for training and testing. To lower the computation required to train the models, they were specifically shrunk to 32x32 pixels.


Fig.1. *Preprocessed image.*

## 4.2. Base Model

### 4.2.1. VGG16

The VGG16 architecture served as the study's base model. The architecture consists of 5 max-pooling layers, 3 fully connected layers, and 13 convolutional layers. With 10 output classes, a softmax layer has taken the role of the previous fully linked layer.

## 4.3. Pre-trained Model

### 4.3.1. ResNet152

A 152-layer deep neural network with numerous residual blocks connected by skip connections makes up the ResNet152 architecture. These skip connections give the network the ability to learn the identity mapping, which makes it easier for gradients to go through the network and lessens the issue of vanishing gradients. Two or three convolutional layers, batch normalisation, and a shortcut connection that skips one or more layers are all included in each residual block.

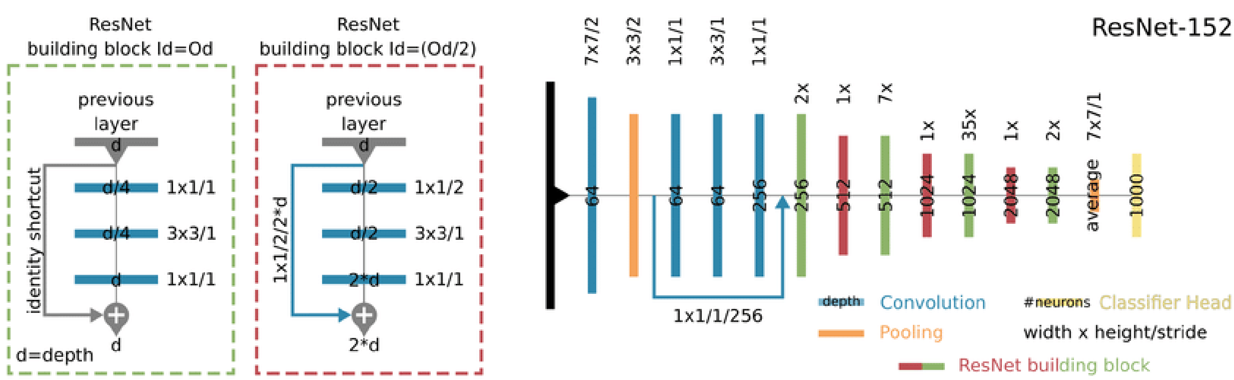



Fig. 5. *ResNet152 Architecture*

### 4.3.2. DenseNet121

By leveraging shorter connections between the layers, the DenseNet (Dense Convolutional Network) design aims to increase the depth of deep learning networks while also improving training efficiency. A convolutional neural network called DenseNet connects every layer to every other layer below it. For example, the first layer is connected to the second, third, fourth, and so on levels, and the second layer is connected to the third, fourth, fifth, and so on layers. In order to maximise information flow between network tiers, this is done. Each layer receives input from all the layers that came before it and transmits its own feature maps to all the layers that will follow it in order to maintain the feed-forward nature.

Contrary to Resnets, it concatenates the features rather than combining them by summation. The 'ith' layer, which contains 'i' inputs, is made up of feature maps from all the convolutional blocks before it. Every 'I-i' layer after it receives its own feature maps. Instead of only 'I' connections as in conventional deep learning architectures, this adds '(I (I+1))/2' connections to the network. As a result, it needs fewer parameters than conventional convolutional neural networks because no unimportant feature maps need to be learned.

## 4.3. Training

The categorical cross-entropy loss function and the Adam optimizer were used to train the model, which had a learning rate of 0.001. The model was trained using a 32-person batch across 10 epochs.

A deep convolutional neural network architecture called VGG16 was created by the Visual Geometry Group (VGG) at the University of Oxford. The 16 layers include of 3 completely linked layers, 13 convolutional layers, and more. With state-of-the-art performance at the time of its release, the network was trained on the extensive ImageNet dataset for picture categorization.

The VGG16 architecture served as the study's foundational model. The architecture consists of 5 max-pooling layers, 3 fully connected layers, and 13 convolutional layers. With 43 output classes, a softmax layer has taken the role of the previous fully linked layer.

The VGG16 deep learning algorithm has been used to classify traffic signs in the context of Traffic Sign Classification and Recognition. Convolutional and pooling layers are used by the network to extract useful characteristics from an input image of a traffic sign. For the purpose of producing a classification output, the extracted features are flattened and then passed through a number of fully connected layers.

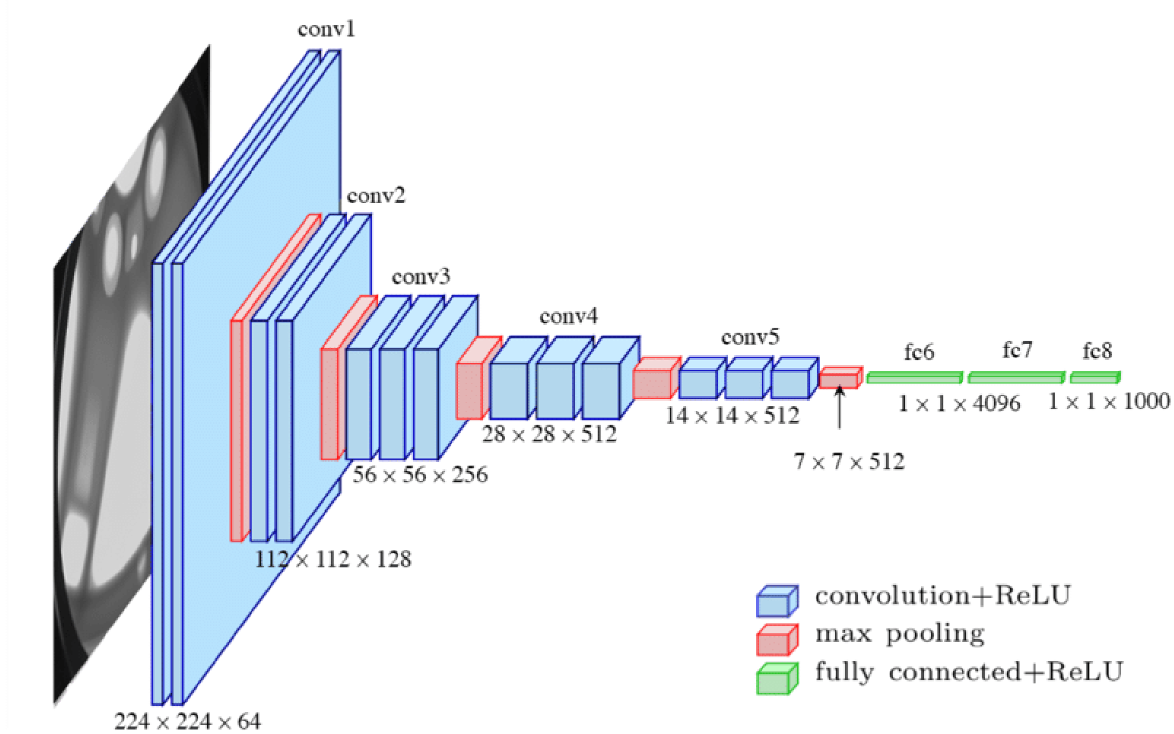



Fig. 1. *VGG16 Architecture*

The Traffic Sign Classification and Recognition dataset is utilised to train the VGG16 model for traffic sign recognition. Over 6000 images of 43 distinct kinds of traffic signs, each labelled, are included in the dataset. A training set, a validation set, and a testing set are created from the dataset. The training set is utilised to train the VGG16 model, the validation set is utilised to adjust hyperparameters and prevent overfitting, and the testing set is utilised to assess the model's effectiveness.

Categorical cross-entropy is used as the loss function and the Adam optimizer is used to optimise the VGG16 model during training. Using a 32-person batch size, the model is trained over several epochs. To increase the robustness of the model, training data is enhanced using techniques including rotation, zooming, and horizontal reversal.

Utilising criteria including accuracy, precision, and recall, the VGG16 model's performance is assessed on the testing set following training. High accuracy and precision on the testing set are indicators of a good model.

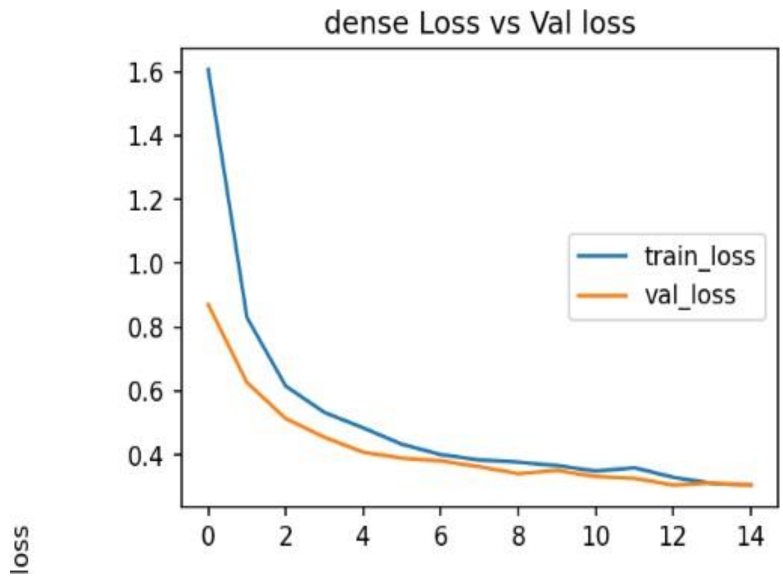
After 10 epochs, the VGG16 model had a categorical accuracy of 93.52%. The model's learning curve reveals that after 10 epochs, the model's convergence was attained. The model fared well in correctly identifying the majority of the traffic signs, according to the confusion matrix.

By stopping the convolutional layers of the pre-trained ResNet152 model and extracting features from fresh input images, the model can be used as a feature extractor in transfer learning.

## 5.0 Performance evaluation metrics

The performance of traffic sign classification experiments was evaluated by the following four evaluation metrics, which are described in Equations below. To more clearly reflect the difference in indicator values, each indicator value is multiplied by 100 as the final result.

After refining our CNN model for the classification phase, we plotted the learning curve and the confusion matrix (Figs. 1, 2, 3). The model's performance is compared based on the number of epochs using the learning curve. After 15 epochs, the model reaches its convergence. We may assess the categorization system's quality using the confusion matrix. Based on the model chosen, we plotted the learning curve and confusion matrix.



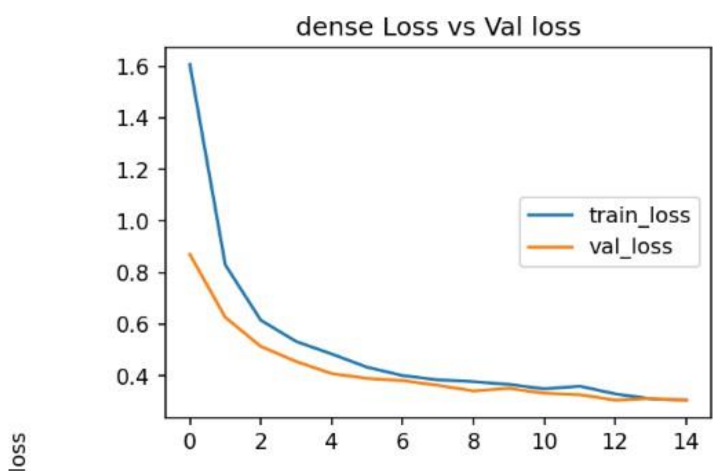




Fig. 3. Graph of train Accuracy vs Validation Accuracy

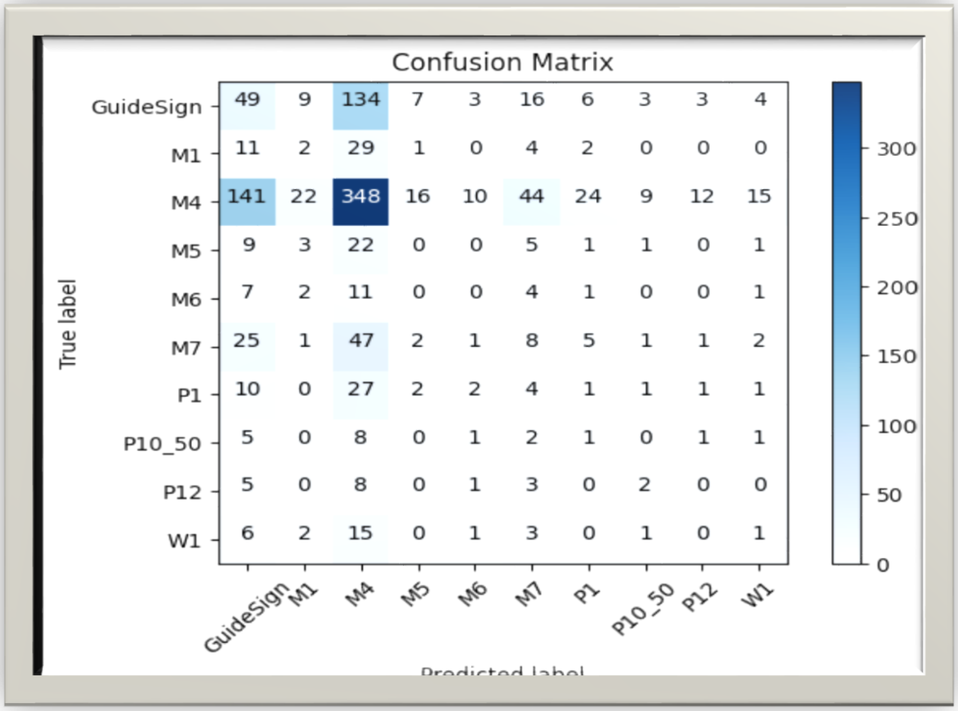



Fig. 4. Confusion Matrics graph

After refining our CNN model for the classification phase, we plotted the learning curve and the confusion matrix (Figs. 2, 3, 4). The model's performance is compared based on the number of epochs using the learning curve. After 15 epochs, the model reaches its convergence. We may assess the categorization system's quality using the confusion matrix. Based on the model chosen, we plotted the learning curve and confusion matrix.

Accuracy= (TP + TN)/ (TP + TN TP + FP + TN + FN)

Precision = TP / (TP + FP)

F1 − Score = (2 × Recall × Precision) / (Recall + Precision)

## 6.0 Result and Conclusion

The Basic CNN architecture is employed to validate the employment of several feature descriptors with a CNN-based architecture. Better accuracy is seen after experimenting with a branched CNN design. Over the fundamental structure.

According to the experimental findings, the three deep learning models—VGG16, DenseNet121 and ResNet152—were all capable of classifying traffic signs with a high degree of accuracy in autonomous driving systems.

A categorical accuracy of 89% was achieved by the VGG16 model, while 92% and 67.2% were attained by the DenseNet121 and ResNet152 models, respectively.

| Model Name | Validation Accuracy | Testing Accuracy | Training Time (Secs) | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| VGG16 | 0.89 | 0.88 | 252 | 0.86 | 0.91 | 0.88 |
| DenseNet121 | 0.92 | 0.89 | 290 | 0.9 | 0.9 | 0.9 |
| ResNet152 | 0.67 | 0.65 | 256 | 0.67 | 0.69 | 0.63 |

Given that all three models successfully utilised the dataset, the results highlight the promise of deep learning techniques for traffic sign identification in autonomous driving systems. DenseNet121, on the other hand, outperformed the other two models, suggesting that deeper and more complex networks might produce superior outcomes.

## References

Blauth, M., Kraft, E., Hirschenberger, F. & Böhm, M. Large-scale traffic sign recognition based on local features and color segmentation.

Deepika, S. Vashisth & S. Saurav. (2018) Histogram of oriented gradients based reduced feature for traffic sign recognition. - *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI).*

G. Huang, Z. Liu, L. Van Der Maaten & K. Q. Weinberger. (2017) Densely connected convolutional networks. - *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Islam, K. T., Wijewickrema, S., Raj, R. G. & O'Leary, S. (2019) Street sign recognition using histogram of oriented gradients and artificial neural networks. *Journal of Imaging*, 5 (4), 44. doi: 10.3390/jimaging5040044.

Kaiming He, Xiangyu Zhang, Shaoqing Ren & Jian Sun. (Jun 2016) Deep residual learning for image recognition. IEEE.

P. Sermanet & Y. LeCun. (2011) Traffic sign recognition with multi-scale convolutional networks. - *The 2011 International Joint Conference on Neural Networks.*

S. Chen, J. Zhou, X. Zheng & X. Ruan. (2018) Energy-efficient data collection scheme for environmental quality management in buildings.

Shao, F., Wang, X., Meng, F., Rui, T., Wang, D. & Tang, J. (2018) Real-time traffic sign detection and recognition method based on simplified gabor wavelets and CNNs. *Sensors*, 18 (10), 3192.

Wong, A., Shafiee, M. J. & St. Jules, M. (2018) MicronNet: A highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification. *IEEE Access,* 6 59803- 59810.

Yasmina, D., Karima, R. & Ouahiba, A. (Nov 2018) Traffic signs recognition with deep learning. IEEE.