TRAFFIC SIGN DETECTION USING DEEP LEARNING ALGORITHMS

MSc Research Project
Data Analytics

SANIKA KHANDALKAR
Student ID: x19266355

School of Computing
National College of Ireland

Supervisor: Dr. Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student ID:	X19266355
Programme:	MSc Data Analytics
Year	2021-2022
Module:	MSc Research Project
Supervisor:	Dr. Christian Horn
Submission Due Date	15/08/2022
Project Title	TRAFFIC SIGN DETECTION USING DEEP LEARNING ALGORITHMS
Word Count	7985
Page Count	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …………SANIKA ATUL KHANDALKAR………

Date: ………15/08/2022……………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

TRAFFIC SIGN DETECTION USING DEEP LEARNING ALGORITHMS

Sanika Khandalkar
X19233655

Abstract

Traffic sign detection, along with road architecture, is a vital component of the advanced driver assistance system. Previously, typical traffic sign recognition system algorithms were used, however, due to a more advanced and upgraded globe. The only traditional model available for detecting traffic signs was insufficient. At the same time, because of its limitations, diving deeper into the recognition or detection aspect was a little tough. Deep learning methods, on the other hand, were introduced for the implementation and detection of traffic signs. The dataset was obtained from Kaggle. The dataset includes over 50,000 traffic sign photos and 43 identified classes. A transfer learning-based model for traffic sign detection and implementation was presented in this work, which steadily minimizes the quantity of training data required. when compared to other machine learning or deep learning models, while also reducing computational expenses by using transfer learning models such as Resnet 50, VGG16, VGG19, and so on. In addition, the Convolution neural network has been proposed, in which layer-wise feature extraction was conducted using multiple convolutions and pooling processes, which were then compared and analyzed. Finally, the transfer learning-based model is retrained numerous times using statistical analysis and fine-tuning parameters at different learning rates. when CNN and Adam Optimiser were utilized, the results reveal that the CNN model can get up to 98.95 percent recognition rate in traffic sign identification. The classification and evaluation report comprises accuracy, F-1 score, and recall, as well as several additional aspects tailored for review and analysis. This research might aid in the identification of different forms of traffic infrastructure, such as roadway marking and roadside provisions. Python tools such as matplotlib, Keras, seaborn, pandas, and many others were used to run and deploy the code. For programming, Google collab and Jupyter notebook were also used.

Keywords: Deep learning, CNN, VGG16, VGG19, Resnet50, Traffic sign, multiclassification

1 Introduction

The authority provides traffic signs such as left, right, restricted zone, U-turn, and so on. Advanced Driver Assistance Systems (ADAS) in high-end car manufacturers such as Mercedes-Benz, Jaguar, BMW, and others assist drivers on the road, for example, TSR (Traffic Sign Recognition) by generating a notification to remind drivers to adhere to a certain speed limit based on different geographical zones. As a result, developments in performance and intelligence will benefit the business and are likely to increase dramatically in the future, making traffic sign identification using computer technology critical.

Our study, like this one, is based on the GTSRB dataset, which is simple but extremely powerful. Despite its size, the dataset contains no legal or ethical issues that should be noted. Traffic sign detection and recognition have been a critical component that should benefit from computational assistance. Deep learning algorithms were used in this study to detect and classify traffic signs. The dataset includes over 50,000 images divided into 43 categories. The dataset is substantial and typical. Deep learning algorithms are more powerful detection models. Deep learning algorithms, as well as the Convolutional neural network approach and transfer learning, have been widely used in image processing research studies. As a result, the intensity of the model using such approaches tries to achieve good model performance. In this study, four deep learning algorithms were applied to this dataset. Because the dataset is in standard form, no additional data cleaning was required. Data augmentation was not performed; instead, the data was resized using 32*32 and one hot encoding was applied. Later, the data was split into training and testing with a 9:1 ratio, which means that 90% of the data was taken by the training data and 10% by the testing data. Later, CNN with Adam optimiser was run on the dataset, followed by transfer learning-based models Resnet50, VGG16, and VGG19.

As a result of the research on traffic sign detection, 4 algorithms were implemented, each with a different performance associated with the epoch, time, speed, and so on.

In this study, the convolutional neural network outperformed all other transfer learning models. The optimiser used in this approach was Adam optimiser and loss of categorical entropy. The activation of Relu was used. Previously, the SDG optimiser was run while performing the implementation part, but the results were unsatisfactory. Using an Adam optimiser, the same CNN model was considered after many permutations and combinations. As a result, Adam optimiser produced an excellent result. As a result, Adam is truly said to be the SDG's replacement

Figure1- Multiclassification

1.2 Research Question and Objectives

RQ:" How well does the deep learning model recognize multiclassified traffic signs?"

> *"How successfully can Deep Learning methods (CNN, VGG16, VGG19, and ResNet50) use transfer learning models to categorize whether all multiclassified data are detected or not?"*

Objective: The traffic sign identification research study employs a deep learning algorithm. In this work, the dataset contains 50,000 or more images that have been labeled with 43 different classes. The key goal of this model implementation and study is that there are many research papers listed below that have done extremely well by using the transfer learning model such as Resnet 50, VGG 16, and so on. As a consequence, in this study, four deep learning algorithms will be implemented, and the model's accuracy score, computational time, speed, and other characteristics will be noted using a comparative analysis. Finally, in traffic sign categorization, which deep learning model is superior to other learning models will be discussed.

2 Related Work

This section that is directly related to previous papers used on similar topics and using the same type of algorithms is generally referred to as related work. As a result, this paper serves as a resource to which we can refer while carrying out and designing the current research project. The Related Work section is divided into three sections.

Traffic sign recognition using different CNN Architecture

(Zhao, Zheng, Xu, and Wu, 2019) suggested a deep learning model for object detection that is advanced. In this situation, they detected an item, such as face detection, pedestrian detection, and quiet object detection. Rather than just categorizing the photos, they wanted to know where the items in each image located. This is known as object localization. For pedestrian detection, they applied CNN-based algorithms for object detection and multi-level semantics to find the item in the picture. Similarly, in face detection, they used a CNN model that did not produce good results, so they improved it by designing a meaningful network cascade, devising novel optimization losses, changing and upgrading generic detection pipelines, and comprehending and studying multi-task shared neural networks. CNN features that will help you make better judgments. According to them, they might have created a better model with a better outcome by applying CPRSD, DSSC, and NLDP, which are put on the multi-scale representation, and super pixel segmentation.

(Zhang, Chang, and Bian, 2020) introduced the Mask RCNN method for detecting vehicle damage. They will use transfer learning and enhanced Mask RCNN to forecast the car damage detection problem. In this scenario, MaskRCNN was utilized to detect and segment vehicle damaged regions in traffic accidents. Their first purpose was to remedy the traffic collision caused by the damaged automobile. Car damage identification can occasionally lead to issues in terms of speed (lower) and accuracy. As a consequence, they employed MaskRCNN; however, when enhanced MaskRCNN was applied, accuracy increased by 21%, showing that Improved MaskRCNN was a superior choice. In this study, they will strengthen the remaining network, as well as adjust the internal structure, which assists in model reinforcement, and lastly upgrade the generalization ability, which results in the anchor box's parameters being balanced. Furthermore, in this research, they obtained high accuracy but only partially correct instance segmentation, and the damage was not segregated in certain places. As a result, the weather was usually a problem in identifying the car's damage image.

(Ahmed, Kamal and Hasan, 2022) proposed study on the detection of Robust Traffic Signs in difficult weather circumstances. Initially, the SOTA algorithms were applied to raw data, resulting in performance degradation under various difficult scenarios. In general, the major goal of this research was to identify the robust traffic sign in various climatic situations, as well as a strong performance model in class classification and detection. With the preceding improvement, the author presented a CCN MODEL. The major technique used in this research was a CNN-based challenge classifier, an encoder-decoder, a CNN architecture for image enhancement, and others for sign recognition and classification. Enhance-Net was utilized to concentrate on the performance and improvement of traffic sign areas. It was a video recognition and classification of traffic signs using CNN-based VGG16, Densenet121, and Resnet 5018. As a consequence, VGG16 has one of the greatest results when it comes to identifying images under certain climatic situations. In addition, the models were compared against SOTA, Faster R-CNN, and R-FCN, with the model excelling by a wide margin.

The Deep Learning model is used in the research on the identification of small traffic signals. In the suggested research project (Wan et al., 2020), they created a

YOLOv3 model for the effective detection and categorization of traffic signs. They proposed a fully optimized model and corresponding algorithm, which was helpful in getting an efficient result in the model's performance regarding the classification and detection of traffic signs in high resolution, followed by network pruning, four scale prediction, and loss function optimization. Initially, the model was constructed using a CNN model, which included Faster R-CNN and R-FCN, but the model was inefficient since the computational process was also resource-intensive. The YOLO model series was later deployed. The original YOLO was utilized at first, followed by Yolov3 -pruning, Yolov3-4SPB, and YoloV3-final. Finally, considering the network's ability to extract the traffic sign properly, network pruning, loss function, and so on, the Yolov3-Final model and its related efficient algorithm improved the model's accuracy and performance by roughly 94 percent and recall by 92 percent. Thus, when compared to the Yolov3 original model, the dataset utilized for detecting and classifying the small traffic sign was enhanced by the Yolov3-Final model. The author also stated future work in which object detection on the autonomous vehicle is performed by condensing the model and optimizing detection.

(Xu, Gong, and Liu, 2020) introduced an algorithm dubbed IMC-CNN that recognizes hazy and fuzzy pictures, including tiny objects, and is very useful in identifying car damage. Similarly, the IMC-CNN model includes an object localization network and an object classification network, whereas previously, they employed CLAHE and MSR to magnify the picture. Their major goal was to detect damage or severity in small and large vehicles when the image was not clear, therefore they used an object classification network with two convolution layers that decrease data loss while increasing the accuracy and classification of tiny and indistinct objects. Finally, when compared to each model, they concluded that when Image enhancement (MSR+CLAHE) was performed with the IMC-CNN model, accuracy was about 86.67%, but with simply Fast R-CNN accuracy was 61.33%. When only Fast R-CNN was used, a considerable decrease was found.

(Dwivedi et al., 2020) developed a Vehicle damage categorization that will assist insurance companies in processing claims more swiftly. Their dataset was not very huge; it was built manually by adding and collecting photographs from the internet. They employ the Pre-trained CNN model and YOLO object detector, which aids in recognizing particular areas of damage. Initially, they demonstrated an accuracy of 74.23%, but when evaluated, the precision increased to 77.78%, indicating that as precision grows, it aids in detecting tiny pictures. In addition, for a more clear and effective identification of the damaged car, researchers introduced the channel by incorporating classification and detection, which assisted them in achieving a positive outcome. To avoid difficulties like overfitting, data augmentation was used, allowing the limited dataset to grow in size. In addition, for a more clear and effective identification of the damaged car, they introduced the channel by adding classification and detection, which assisted them in achieving a good outcome. This undoubtedly aided in improving model correctness. Thus, using the aforesaid model, the identification of damaged automobiles proved effective, leading them to suggest a completely automated method that aids in the speedier filing of insurance claims. The sole factor discovered was that the dataset was small.

Transfer Learning in Image Classification

(Khaja Khan and Abdullah, 2022) proposed research on traffic sign recognition using a transfer learning approach. The Small CNN architecture was compared to the VGG, Alex Net, and some Resnet 50 Models. In the end, the three transfer learning approaches used for this pre-existing model were InceptionV3, Resnet 50, and Xception. All of these models were fine-tuned numerous times to improve the model's performance and efficiency. Because the number of parameters and the size of all three models was the same, training each model took approximately 9 to 10 hours. Finally, of the three main transfer learning models, InceptionV3 and Xception have the highest accuracy (around 96%) and the lowest loss (0.701). When the Resnet 50 model was used, it produced an unsatisfactory result with an accuracy of about 60.69 percent. As a result, if a comparative analysis is performed, InceptionV3 is the best choice for this research.

(Peng et al., 2017) proposed a model in which transfer learning models were used to detect and classify traffic signs. GAE, or Generalized Auto-Encoder, was also proposed for transferring knowledge between domains. The domain-invariant data was also obtained from the latent subspace. To construct the deep non-linear architecture, they added Local Coordinate coding based on the relational function at the same time for statistical and geometric properties. Essentially, after all of these changes, a few benchmark datasets of traffic sign detection were optimized, yielding an outstanding performance. As a result, when this approach was compared to other traditional models, it came out on top, The new research model outperformed the other traditional model. During the ongoing process, two subsets were considered: one for the source domain and the other for the target domain, which was a corrupted version; as a result, Latent suspace was used, which has the ability to reconstruct the domain invariant. In the future, the author hopes to collaborate with other traditional methods, such as a combination of global and local strategies that can be used for the neighbor across the domain.

(Li, Zhao, Chen and Zhang, 2018) A research effort on traffic sign identification using deep learning algorithms is proposed. In this study, the author employed an algorithmic model called Deep Sign, which consists primarily of three modules: PosNet for identifying the placement of traffic signs in static images, PathNet for categorizing the discovered patch in the picture, and a temporary filter for correcting the recognition result. Posnet is commonly used to maintain the traffic sign in one class and the background in the other. This study eventually implemented numerous deep learning techniques. As previously stated, a filter was applied to the model, and a model without the filter was also constructed. A research study on traffic sign detection has been proposed. As a consequence, filter processing increases the model's performance. whose precision and recall levels were enhanced by 2.3 percent and 1.2 percent, respectively. Thus, in this study, the basic method with three main modules called DeepSign was applied with numerous different models where they also did the filter process, resulting in greater performance when the filter processing is included in the model.

(Lin et al., 2018) The inception-v3 model was used to describe traffic sign recognition. The transfer learning model underpins the Inception-v3 model. According to the author, traditional algorithms do not do justice due to their limitations; to address this, they have introduced transfer learning methods for traffic sign detection and classification. Furthermore, the transfer learning model significantly reduces the time spent on training data while also assisting in the reduction of computational expenses. Belgium traffic sign dataset was used, which was later supplemented with data preparation methods. Finally, the convolution and pooling operation was carried out for comparison and analysis. Later, the existing transfer learning model was fine-tuned at various learning rates. As a result, after all of the processes and approaches were used and optimized, it was claimed that the Inception-v3 model, which is a transfer learning model, provided a better accuracy of about 99.18 percent with a learning rate of 0.05. At different learning rates, the epoch was around 5000 at the time. At the same time, transfer learning models are said to be extremely powerful models that can also be used in roadside protection facilities or lane marking.

Traffic Sign detection using Machine Learning Approach

(Stallkamp, Schlipsing, Salmen and Igel, 2011) proposed a traffic sign recognition research study using a multi-category classification dataset. Variations in the visual appearance of signs were observed in this research study as a result of distance, illumination, weather conditions, and so on. At the same time, the images were preprocessed using the precomputed feature. As a result, EDA becomes simple. It also lacks any background knowledge. The dataset contains the same 43 classes as before, but with some unbalanced frequencies. The two test datasets were required because each dataset required more than 12,500 images. As a result, a variety of baseline results, including Linear discriminant analysis on the K-nearest neighbor on HOG features and human performance, were reported. Thus, in the human performance to determine the human traffic sign on the images, around 350 randomly chosen images were presented to 36 test persons, who were isolated and some operations were performed on them. The LDA and k-NN approaches produced very similar results for the various HOG features. Finally, the machine learning model performed well on both the test dataset, with an accuracy of around 98.98%, which is comparable to human performance

(Liu, Chang and Liu, 2016) TSR is a research method based on traffic sign detection that was proposed. According to the author, there were some difficult problems to solve, such as different types, small sizes, complex driving scenes, and occlusion, among others. A few years ago, there was a model called TSD that was based on machine vision and pattern recognition. As a result, a review of TSD was added to this paper, and it was later divided into five categories: color-based methods, shape-based methods, color and shape-based methods, machine learning based methods, and Lidar based methods. There were some instances where there was a lack of comparison on the open-source dataset, so that section was re-implemented. In general, color-based methods are relatively simple, but shape-based methods have received little attention in recent years, with MSER and HCRE-based methods potentially achieving a high percentage of the region of interest. Finally, the conclusion was that when dealing with high-resolution images and small vague traffic signs, it is still very difficult for machine learning algorithms to maintain a balance between time consumed and good accuracy or model performance. Thus, according to the author, this method requires some assistance or help from other methods in order to achieve fast and excellent model performance.

(Maldonado-Bascon et al., 2007) research on the problem of road sign detection using a machine learning model such as a support vector machine. According to the author, one of the most important steps in terms of traffic signs and driver assistance roads is road detection and classification. The goal was to detect all of the signs in the Spanish traffic sign dataset that were circular, rectangular, triangular, and so on. The research is based on the Support vector machine's generalization properties. As a result, the system consists of three types very first type is Segmentation considering the color of the pixel and the second one was classifying and detecting the traffic sign with the help of shape classification with the use of linear support vector machine. and lastly the content recognition which actually based on the Gaussain-Kernel SVM's. Thus, SVM has given a positive output in the detection and the classification of the model. It is also clearly seen that the Support vector machine is same and uniform to translation, rotation, scale and many other senarios. Also, based on the abilities of SVMs pattern recognition, two main modules for shape classification based on linear SVMs and pattern recognition with Gaussian kernels have been developed. Recognizing the interior

Summary

Following a study of the preceding research work, traffic sign recognition and detection were done, and a few image processing papers were also mentioned. As a consequence, CNN was the most commonly employed model in detecting traffic signs, although its performance was not sufficient when compared to other models stated in a few articles. Machine learning algorithms were also implemented on the above-mentioned Spanish dataset (Maldonado-Bascon et al., 2007) where support vector machine was implemented. As a result, the machine learning model does not do complete justice to the as it requires a lot of computational time and the model's performance is also not convincing. Finally, after examining all of the preceding research papers, Deep Learning models were introduced in this research study employing the transfer learning model.

3 Research Methodology

The methodology is the main thing that is the backbone of this research project, followed by the implementation, which is discussed in the following sections. Essentially, the research Methodology would include all of the steps, including the construction of the research. and all of which contribute to the completion of the research. It begins with Data Selection, in which the details about the selected data are mentioned, and is followed by Data Preparation, in which data is initially collected and then prepared for data mining through data cleaning. The next step, Modelling, discusses the methods that will be appropriate for the research later on. The trained data is tested against the real-time dataset during evaluation. Finally, the Deployment or Implementation stage provides the outcomes and performance of each model used in this research project.

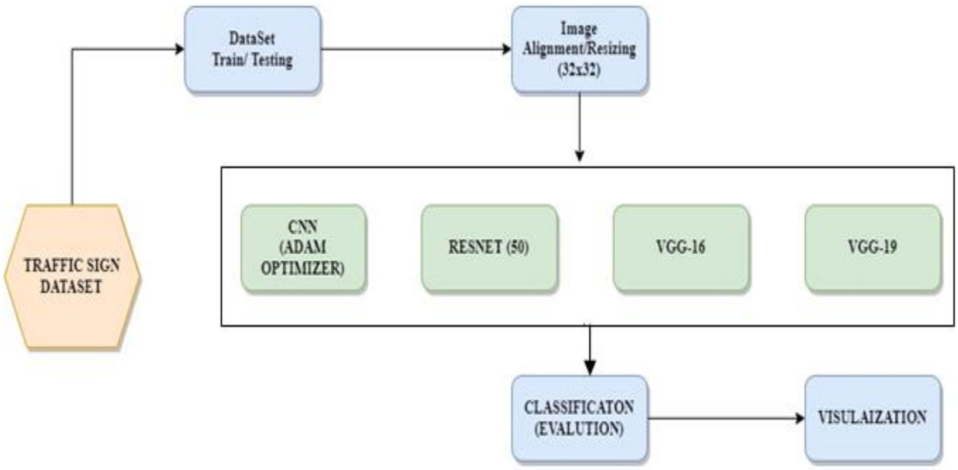

Figure 2: Research Design

3.1 Data Selection

There are numerous datasets available for image processing. In general, when it comes to image processing, there are a few concerns that must be addressed before selecting a dataset for a research project. These concerns include small datasets, which make it difficult to perform advanced and heavy models on, resulting in unsatisfactory performance. It is also critical to consider image quality. In such cases, data augmentation is performed, but avoiding such a problem is preferable for the research project. We chose the German Traffic Sign Recognition Benchmark dataset for this study. As traffic sign recognition has become increasingly important for advanced driver assistance systems, the traditional method for traffic sign detection no longer suffices due to its limitations. As a result, this research project was carried out using German traffic signs. Furthermore, this is an open-source dataset that is easily accessible on Kaggle. Given the ethical concerns, this dataset is ideal. Finally, this dataset contains images totaling more than 50,000 and classes 43. As a result, the dataset to work with is quite large. This dataset contains both single image and multiclass classification problems. The image pixel is small, which causes the image to appear hazy, but this is perfectly fine for research purposes. The image has no background and is entirely focused on the traffic sign. Overall, the data set requires very little preprocessing because the dataset is quite good and standard.

3.2 Data Pre-processing

When data is collected from Kaggle, it is first and foremost an open-source dataset that is freely available to the public. The German Traffic sign dataset was chosen for this research study in this scenario. This dataset is fairly typical. And have as little noise, errors, raw format, and other variables as possible. As a result, it is critical to work on clean data, and all extra noisy data, missing values, and so on must be processed. The images in the dataset were chosen and resized using data augmentation. This dataset contains 42 classes, the first of which is 0. According to the categories, each class has been classified.

3.3 Data Transformation

Prediction problems can be avoided by transforming data. Data transformation, which occurs after data pre-processing, is the most innovative stage in the study. In this case, 'train valid test split' is used. In general, it is critical to separate the data into training and testing sets. As a result, we considered training data to be 90% and testing data to be 10% of the total. There are 43 traffic sign recognition classes, which are divided into training and testing. 35288 images were used for training and 3931 images were used for testing, with a random size of 10 and a test size of 0.1. Later, one hot coding was used in y_train and y_test, resulting in uniform training data.

3.4 Model Selection

When the data is ready for the modelling approach, Implementation using deep learning models takes place. Model selection is typically performed after data preparation and transformation. As a result, four deep learning approaches were used in this study. Initially, the CNN model was performed using Adams Optimiser, followed by the transfer learning model such as Resnet 50, VGG16, and VGG19 Model, and as a result, multiple models were implemented to perform comparative analysis. As a result, these models are thought to be the best models for image processing data.

CNN WITH ADAM OPTIMISER

CNN, also known as the convolutional neural network, is a basic and widely used model that is commonly used for grid patterns. CNN is made up of three layers: convolution, pooling, and a fully connected layer. In CNN, the convolution and pooling layers primarily extract features, and the final layer, a fully connected layer, is used to map those extracted layers in the final output. (Mehta, Paunwala and Vaidya, 2019) perform the Adaptive moment estimation optimiser, which is widely used for network training. Typically, pixels in digital images are shown in two dimensions. Optimisers are used to improve the performance of the CNN model in order to make it more efficient and reliable., The Adam optimiser was optimized and implemented in this study, yielding excellent results. In CNN, the process begins with one layer passing its output to the next, and so on, as the extracted features become more complex. Adam optimiser is essentially a combination of ADAGRAD and RMSProp models that can handle complex problems.

RESNET 50

Resnet 50 is a deep learning model based on transfer learning models; it is also referred to as deep blocks of convolutional layers. Down sampling is performed in Resnet 50 by convolutional layers with a stride of 2, and batch normalization is also performed before and after the convolutional layer.
Resnet 50 structure is distinguished by the use of different colours. Resnet 50 is typically used to work with that specific variant that can work with 50 other neural network layers. The accuracy of Resnet 50 usually saturates in the beginning and then degrades while training deep learning models

VGG16

It is also known as the pre-trained VGG16 model, and it is an open-source model created by the CNN model. This model was eventually forwarded to the ILRVRC-2014. VGG16 is the result of combining 13 convolutional layers and 3 fully connected layers, for a total of 16 layers. As a result, it has a total of 16 CNN layers. Additionally, those 13 convolutional layers are divided into five different classes, each of which is followed by the max pooling layer. The RGB image on the VGG 16 architecture is 224*224 pixels. In general, it is composed of a stack of convolutional and max pooling layers, and when the feature map is received, it is flattened into a vector of size feature.

VGG19

VGG 19 is a Transfer Learning model. It consists of 19 CNN layers in total. To clarify, it has 16 convolutional and 3 fully connected layers with a filter size of 3*3 and a pad and side size of around 1 pixel. As the small grid is known as the kernel, it reduces the number of parameters and allows it to provide full coverage to the image. VGG19 primarily has three additional CNN layers for VGG16. The VGG19 model also performs a 2*2 max pooling operation with stride 2.

3.5 Evaluation

This research study is about traffic sign detection using deep learning methods. The dataset is a publicly available dataset that has 43 different classes and more than

50000 images. After performing the modelling, Once the model has been implemented .it is very important to check the performance of the model. The models here are evaluated by the confusion matrix in the classification problem, Basically, the confusion matrix is the matrix that has (TP) called True positive, (TN) called true negative, (FP) called false positive, and (FN) false negative. Generally, it provides the counts of actual and predicted labels.

Precision: Prediction is the measure of the true positive prediction out of the Positive prediction performed by the model.
Recall: Recall is used to check the positive prediction of the model out of the no of total positives in the dataset. The recall is also called Sensitivity
F1 Score: The F1 score is usually called s the harmonic mean of the precision and the recall, it is generally used for evaluating the occurrence where a high positive is not bearable.
Support: Support in the confusion matrix is generally used to measure the occurrence of true positives.

4 Implementation and Evaluation

The deep learning models used for traffic sign recognition have been implemented on the Jupyter Collab and notebook in this section. Later, the process and progress of each model in the study were evaluated and explained using the confusion matrix and other evaluation factors.

4.1 Preliminary Data Exploration

The first step in Preliminary Data Exploration is to understand and examine the type of file contained in the data. Because the primary goal is to analyse the data provided. It is critical to understand every aspect of the data, including its type, file type, number of rows, columns, missing values, noise error, and so on. The data contains a lot of disturbances, and it is critical to remove those disturbances. In this case, the dataset is massive, containing over 50,000 images and 43 classes. Because the dataset is so large, massive data augmentation is difficult. Another reason for having this dataset is that it is very standard and labeled. It is always profitable to invest in the labeled dataset. Finally, the data is divided into Train, Validation, and Test folders, with a proportion of 80% and 10%. The training data is divided into 80% and the test data is divided into 10% in this case. Further examination reveals that the train data contains over 35288 images with an image size of 32 * 32 and an RGB of 3. In the test data, the images are around 3921, with an image size of 32*32 and RGB equal to 3. For output, the Google Collaboratory Platform was used to work on this model. The GPU Tesla T4 also aided the Google Collaboratory platform. During data extraction, the dataset is uploaded to Google Drive and then mounted using drive mount features. After all of the primary changes and uploads are completed, the python libraries required for the existing research, such as TensorFlow, Keras, Sklearn, Matplotlib, and so on, are imported.

```python
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(data, labels, test_size= 0.1, random_state=10)

print("training_shape: ", x_train.shape,y_train.shape)
print("testing_shape: ", x_test.shape,y_test.shape)

y_train = tf.one_hot(y_train,43)
y_test = tf.one_hot(y_test,43)
```

```
training_shape:  (35288, 32, 32, 3) (35288,)
testing_shape:  (3921, 32, 32, 3) (3921,)
```

Figure3: Splitting data into train and test

4.2 Data Preparation

The dataset for traffic sign recognition contains over 50,000 images divided into 43 classes. The dataset in this scenario is quite large and standard. In this case, because the data is all about the images, there is no missing data, error, or other issues that are common in data with numbers. Most of the time, data augmentation is used. Data Augmentation was not performed in this dataset because the images are good in numbers; however, each image has been resized to a size of 32*32. There is also no need for resizing. On the dataset, one hot encoding was performed, which resulted in slightly better and excellent performance in the data's training model. As a result, 43 classes have already been labelled and saved. As a result, no labelling of the image was required. The dataset did not necessitate pixel scaling.

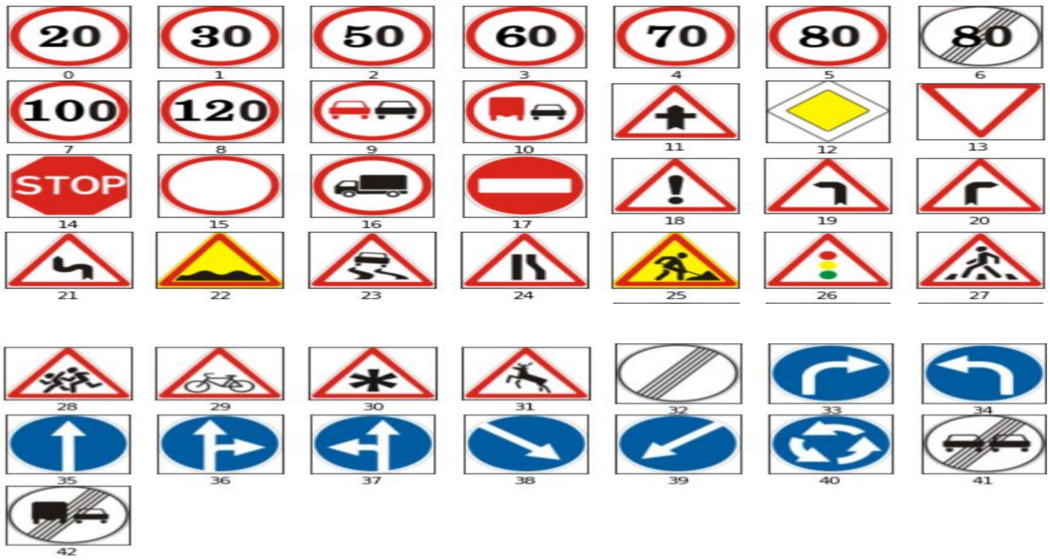

Figure5: All 43 labelled classes in the dataset

4.3 Model Implementation and Evaluation.

After data pre-processing, the main step is data modelling. when the image is properly sized and the model is ready for training the first model that can be used. Initially, CNN (convolutional neural network) was used, with the Adams optimisers. The second model was based on the transfer learning model RESNET 50, and it was followed by the VGG16 and VGG19, respectively. All of the models mentioned above are elaborated on below with appropriate modelling experience. As a result, it is a Deep Learning-based model. Furthermore, each model used has its own significance and popularity.

4.3.1 CNN with Adam's Optimiser.

CNN stands for Convolutional Neural Network, and it was the first model used in this study. The data was delivered to the CNN model, which has a total of 32 features. The Kernel is a small grid with a size of around (5,5). The Relu activation, also known as rectified linear function, is used on the Convolution neural network. The standard Relu activation value is max(x,0), which is the element maximum of 0 and the input. Basically, if the output is true or real, it will display the input output; otherwise, it will simply return zero. Finally, Models has a few CNN layers, the first two of which are Convo2D with filters of 32 and 64. As a result, the convo2D layer is useful for developing the convolution kernel, which is used to calculate or measure the total number of output filters present in the convolution. The third layer is Maxpool2d, which has a pool size of (2,2), and the next is Dropout, which is used to avoid overfitting and has a rate of 0.25. Other CNN layers, such as Flatten and Dense, were also implemented, giving it 3 Convo2D, 2 Maxpool, 3 Dropout, and 2 Dense network layers to identify hidden future trends. As a result, after the layers, the model is compiled using categorical cross-entropy by Adam optimisers.

Adam optimiser is a type of optimiser that combines the AdaGrad and RMSProp algorithms. which may be capable of managing the disturbance's sparse gradients as previously stated, the Adams optimisers and the loss function of binary cross entropy are used for model compilation because the goal of the project is to detect traffic signs using the Deep Learning algorithm. CNN datasets run for 10 epochs, with a batch size of 64. This provides relief for the consistency of the outcomes, including the training timeframe.

Following the compilation of the model, the accuracy and loss graph of training and validation has been drawn for each epoch, as shown below. As a result, the CNN model's accuracy after 10 epochs with the highest training and validation is 99.4%, and the training and validation loss is 0.0240. The test data accuracy is approximately 95.4 percent. Finally, the model performed well, as evidenced by its accuracy of around 95.4%, and by the fact that the training and validation values are very close to each other on the graph.

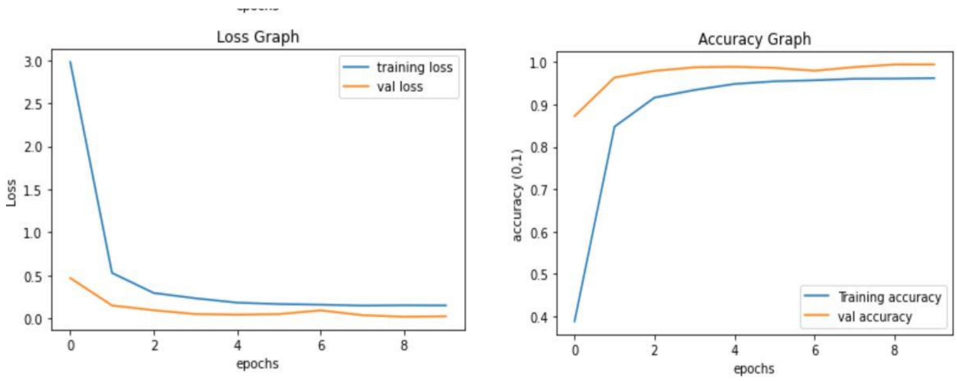

Figure 6: Accuracy and Loss graph

The Classification report has also been mentioned for all 42 classes, with the 0th class having a precision of 86% and a recall of 1.00. The 42nd class has a precision of around 86%, a recall of 90%, and an F1 of 88%.

accuracy			0.95	12630
macro avg	0.93	0.94	0.93	12630
weighted avg	0.96	0.95	0.95	12630

Figure 7: Classification report

The Heatmap is also mentioned, which clearly shows that the True Positives form a straight diagonal without any misclassification.

4.3.2 RESNET 50

The transfer learning foundation model is Resnet 50. Resnet 50 models are convolutional neural networks with 50 layers. Using the shortcut connection primarily skips the blocks of the convolutional layer. The Resnet 50 model also includes the implemented residual network structure. The reason for improved accuracy in the deep CNN model is that the use of residual network structure avoids or overlooks the issue of model deterioration.

In this study, the Resnet 50 model was used, with the weights and including top parameters set to zero. The models include GlobalAveragePooling2D()(X), which allows the fully connected layers to be replaced with the traditional CNN. This results in an excellent vector representation.

The dropout layer has been mentioned as a reason for overfitting, and the predictions have the Dense network the SoftMax activation.

In the beginning, the Input layer and the zero Padding2D have 0 parameters.

Thus, the third layer is Convo2D, which has a parameter of 9472 and an out shape of 16*16. The batch Normalization layer follows, with approximately 256 parameter values. Unlike Convo2D and ZeroPadding2D, the rest of the layers in the model summary section have the 0th parameter.

As a result, the total number of parameters achieved in this model implemented is approximately 23,675,819, and the trainable parameters are approximately 23,622,699. The Non-Trainable Parameter is 53,120.

Later, the model was compiled using an Adam Optimiser and Loos entropy, and the model's epoch run was 10, 20, and 30. The batch size was approximately 64. The validation training accuracy was around 69.2% after each epoch training, whereas the Validation for the loss was 1.14. The accuracy obtained in the test data was around69.2%.
The graph has been plotted using Matplotlib, and it shows the accuracy and loss in the training data by the count of each epoch. Finally, the graph is only 60% accurate because the accuracy and loss points are so close to each other.

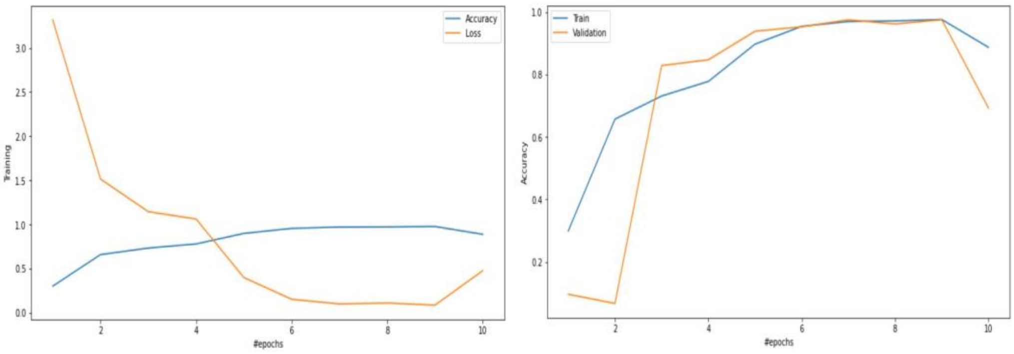

Figure8: Accuracy and Loss graph

In addition, the Classification report has been mentioned for all 42 classes, with the 0th class having precision of 0% and recall of 0.00.

accuracy			0.63	12630
macro avg	0.64	0.58	0.56	12630
weighted avg	0.71	0.63	0.63	12630

Figure 9: Classification report

The 42nd class has a precision of around 94%, a recall of 87%, and an F1 of 90%.
The Heatmap is also mentioned, which clearly shows that the True Positives do not form any of the straight diagonals, indicating that the model contains a large number of false negatives and misclassification data.

4.3.3 VGG16

The VGG16 model is likewise based on the transfer Learning model, which is based on the Convolutional neural network layers. The first 13 levels are the convolution network, while the last three layers are the fully connected layer. The input shape in the VGG16 model is about 32*32. The weight used in the VGG16 model is ImageNet because it improves computational effort by classifying pictures smoothly. In addition, the include top was set to False.

The first layer of VGG16 is the Input Layer, which has an output shape of 32*32 and parameters of 0; the second layer is the Convo2D, which also has an output shape of 32*32 with parameters of 1792. In the model summary, the Maxpool 2D layer contains 0 parameters, leaving just the Convo2D parameters as non-zero. As a consequence, the total number of parameters discovered in the VGG16 model was around 15,284,075 whereas the trainable parameters were 569,387 and non-trainable parameters were 14,714,688. The model was trained and ran using a batch size of 128 and a rating of 10 epochs. The initial epoch took around 9 seconds and 26ms/steps, with a loss validation of approximately 2.3033 seconds afterwards. Each epoch takes approximately 6 seconds to complete. Finally, the trained validation accuracy was about 89.2% and the validation loss entropy was around 49.2% after completion. Later accuracy on the test data was also approximately 53.7%.

The accuracy and loss graphs were also presented in Python using the matplotlib module. As a consequence, the training and validation accuracy peaks when the 1 and 2 epochs were running, achieving an accuracy of roughly 82% later on.

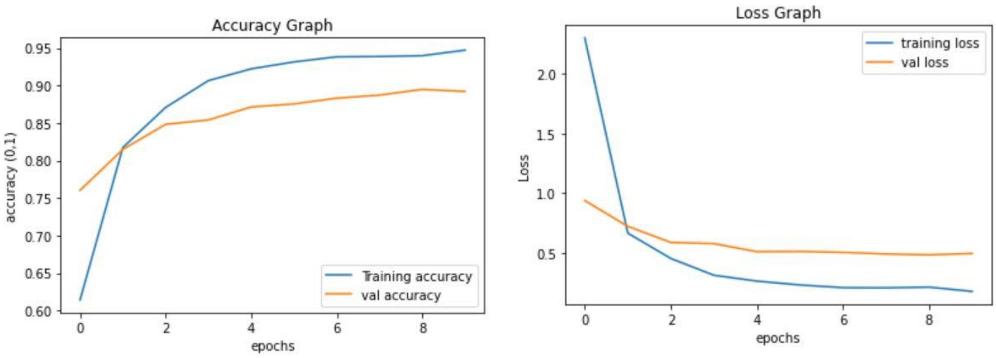

Figure10: Accuracy and Loss graph

The classification report was mentioned for all 42 classes, with accuracy of 10% and recall of 0.05 for the 0th class. The precision of the 42nd class is roughly 34%, recall is 33%, and F1 is 90%.

accuracy			0.54	12630
macro avg	0.47	0.45	0.44	12630
weighted avg	0.56	0.54	0.54	12630

Figure11: Classification Report

The Heatmap is also mentioned, demonstrating that the True Positives do not form any of the straight diagonals, indicating that the model contains a substantial number of false negatives and misclassification data. The specificity and sentivity were likewise categorized using all 43 classes. Initially, the sensitivity of the 0th class is approximately 99% and the specificity is around 05%. Similarly, in the previous class, the sensitivity was about 99% and the specificity was 33%.

4.3.4 VGG19

The VGG19 model is likewise a CNN model, with about 19 layers of convolution layers and a fully connected layer separated into 16 and 3 layers. In the VGG19 model's model implementation. Initially, the input shape was the same as the above-mentioned models, 32*32, and the weights were ImageNet, which aids in enhancing the model's performance once the data was downloaded.

The first layer was the input layer, which had an output shape of 32*32 and a parameter of zero. In the VGG19 model, the Convo2D layer has parameter values of 1792 and 36928 from the second and third layers, as a consequence, if you carefully read the summary, Maxpooling 2 dimensional layers comprised of 0th parameters, and the form of the output was likewise 8*8 and 16*16, with the first layer having an output size of 16*16 and the deeper layers having an output size of 8*8. Thus, the total parameter was about 20,593,771 and the trainable parameter was 569,387, whilst the non-trainable parameter was 20,024,384. The VGG19 models run for a total of ten epochs. The first epoch took 9 seconds and 29ms/step pace, the val accuracy reached was about 69.6%, and the val loss was around 1.2%, resulting in the last epoch running for 7 seconds and achieving validation accuracy of around 87.4%. When the test data was done, the accuracy reached was 50% due to the loss validation being approximately 53.6%.

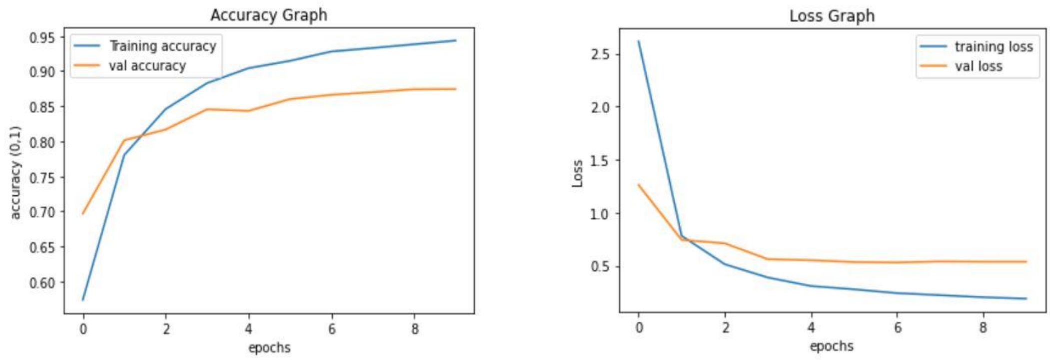

Figure 12: Accuracy and Loss graph

According to the classification reports, the precision, recall, and F1 on the 0th class were 0.00, however the 42nd class has an accuracy of 28%, recall of 26%, and F1 score of 27%.

accuracy			0.50	12630
macro avg	0.44	0.42	0.42	12630
weighted avg	0.52	0.50	0.50	12630

Figure13: Classification Report

After viewing the accuracy and loss graph, as well as the Heatmap, it is clear that this model VGG19 did not perform well for this dataset. Only half of justice was served. The remainder of the misclassification values were present.

4.4 Model Comparison and Discussion

When all four deep learning approaches were applied to the traffic sign dataset in the research study. The primary goal of this study was to investigate and comprehend all of the expectations such as model performance, accuracy, classification graph, and so on from each of the four deep learning models such as CNN, Resnet 50, VGG16, VGG19, and so on. Finally, all of these models perform admirably on training data. Convolution neural network with Adam optimizer provided outstanding training and validation accuracy of 95.6% and 98.2%, respectively.

In this study, the convolution neural network, popularly known as CNN, performed well on both the training and testing data sets. Whereas the testing data has a success rate of roughly 95%. A detailed examination of the classification graph reveals that the real positive images or data form a straight diagonal, implying that no misclassification occurred. (Khaja Khan and Abdullah, 2022) In this research work, CNN does not deliver good performance and accuracy when identifying traffic sign when compared to that performed by Inceptionv3 algorithms. When comparing these four existing deep learning models, Resnet 50 has a validation accuracy of around 70% and a test accuracy of 63%. Similarly, after implementing the VGG16 AND VGG19 models, the VGG16 model performed well in contrast to the VGG19 model. At the same time, in deep learning models, training time for each model is critical. For that situation, the batch size in CNN and Restnet 50 is 64, and the epoch is virtually the same in all models, which is 10 epochs. Whereas the VGG 16 and VGG19 models have a batch size of 128. In comparison to other deep learning models, CNN with Adam optimiser performed exceptionally well. Similarly, on CNN, each epoch runs for 236 seconds before being reduced to 234 seconds. The resnet model took only 37 seconds to run one epoch, and the time taken for the ninth and tenth epochs was only 26 seconds. The VGG 16 epoch took 9s to 6s to complete, but the VGG 19 epoch took less time than any other model (9sec to 7 seconds). In the study, the CNN model took the most time to train the model, although the duration decreased with each model.

As a result, in the German dataset, the CNN model works best at detecting traffic signs.

Rest models, such as Resnet 50, also performed well in training data, but not so well in testing data. The VGG16 and VGG19 did not perform well in testing data.

Model	Train Accuracy(%)	Test Accuracy(%)
CNN with Adam optmiser	95.60%	95%
Restnet 50	88.60%	63%
VGG16	94.70%	54%
VGG19	94.30%	50%

Figure 14: Model Comparison

5 Conclusion and Future Work

It is critical in this research effort to address the research study and research query that were proposed prior to the implementation of this research. This study's research question was, "How efficiently does the deep learning model detect traffic signs having Multiclassification?" As previously said, four deep learning algorithms were applied. As a consequence, the first research question reads, "When the model is implemented in the dataset, how is the model performance and accuracy obtained, concurrently looking at the computing times and the speed of each model" It can be challenging for the model to produce satisfactory results at times. As a result, all of the models perform well on training data, but Convolutional neural networks perform admirably on testing data as well.

The second study topic concerns whether or not these deep learning algorithms can detect traffic signs utilizing transfer learning approaches. Now, in response to this query, we have demonstrated that the methodologies and implementation of all the models resulted in CNN providing approximately 95% accuracy in detecting traffic signs on multiclassification data.

Few things were discovered throughout the data implementation time, such as the image in the dataset having a very low resolution, causing the image to appear very small. The dataset was quite large, with 43 different classes, making it difficult for the model.

In the future, the classes can be reduced and classified by including high-resolution photos, which aids in image clarity. Also, the system design in this study was not perfect, so if a system with a better configuration is employed in the future, it may perform better in terms of traffic sign detection.

6 Acknowledgment

I'd like to thank Dr. Christian Horn, my supervisor, for his advice during the implementation phase of the project. I would like to thank him for his consistent support and supervision, which helped the project go well.

References

Ahmed, S., Kamal, U. and Hasan, M., 2022. DFR-TSD: A Deep Learning Based Framework for Robust Traffic Arxiv.org. 2022. [online] Available at: cascaded convolutional neural network and mixed image enhancement. IET Image Processing, 14(17), pp.4621-4632. color model and shape context matching. *Journal of Electronic Imaging*, 25(5), p.051202.

Dwivedi, M., Malik, H., Omkar, S., Monis, E., Khanna, B., Samal, S., Tiwari, A. and Khaja Khan, M. and Abdullah, M., 2022. *A Transfer Learning Approach to Traffic* Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), pp.3212-3232

Li, D., Zhao, D., Chen, Y. and Zhang, Q., 2018. DeepSign: Deep Learning based Traffic Sign Recognition. 2018 International Joint Conference on Neural Networks (IJCNN),.

Lin, C., Li, L., Luo, W., Wang, K. and Guo, J., 2018. Transfer Learning Based Traffic Sign Recognition Using Inception-v3 Model. Periodica Polytechnica Transportation Engineering, 47(3), pp.242-250

Liu, C., Chang, F. and Liu, C., 2016. Multiview road sign detection via self-adaptive color model and shape context matching. Journal of Electronic Imaging, 25(5), p.051202

Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H. and Lopez-Ferreras, F., 2007. Road-Sign Detection and Recognition Based on Support Vector Machines. IEEE Transactions on Intel

Mehta, S., Paunwala, C. and Vaidya, B., 2019. CNN based Traffic Sign Classification using Adam Optimizer. 2019 International Conference on Intelligent Computing and Control Systems (ICCS),

Peng, X., Li, Y., Wei, X., Luo, J. and Murphey, Y., 2017. Traffic sign recognition with transfer learning. 2017 IEEE Symposium Series on Computational Intelligence (SSCI),

Stallkamp, J., Schlipsing, M., Salmen, J. and Igel, C., 2011. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. The 2011 International Joint Conference on Neural Networks,.

Wan, J., Ding, W., Zhu, H., Xia, M., Huang, Z., Tian, L., Zhu, Y. and Wang, H., 2020. An Efficient Small Traffic Sign Detection Method Based on YOLOv3. Journal of Signal Processing Systems, 93(8), pp.899-911.

Xu, K., Gong, H. and Liu, F., 2020. Vehicle detection based on improved multitask cascaded convolutional neural network and mixed image enhancement. IET Image Processing, 14(17), pp.4621-4632.

Zhao, Z., Zheng, P., Xu, S. and Wu, X., 2019. Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), pp.3212-3232

Zhang, Q., Chang, X. and Bian, S., 2020. Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN. IEEE Access, 8, pp.6997-7004.