



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** III **Month of publication:** March 2022

DOI: <https://doi.org/10.22214/ijraset.2022.40672>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Road Sign Detection System using Neural Networks and Tensor Flow

Rabia Hashim¹, Ravinder Pal Singh², Monika Mehra³

¹M. Tech, ECE, RIMT University, Punjab

²Technical head, Department of Research innovation and incubation, RIMT University, Punjab

³Head of Department, Department of ECE, RIMT University, Punjab

Abstract: Automated tasks have simplified almost everything we perform in today's environment. Due to a desire to focus only on driving, drivers regularly ignore signs placed on the side of the road, which can be harmful to themselves and others. To address this issue, the motorist should be informed in a method that does not require them to divert their concentration. Traffic Sign Detection and Recognition (TSDR) is critical in this case since it alerts the motorist of approaching signals. Not only are roads safer because of this, but motorists also feel more at ease when driving unfamiliar or difficult routes. Another typical issue is inability to read the sign. Driver assistance systems (ADAS) will make it easier for motorists to read traffic signs with the help of this software. We provide a traffic sign detection and recognition system that employs image processing for sign detection and an ensemble of Convolutional Neural Networks (CNNs) for sign recognition. Because of its high recognition rate, CNNs may be used in a wide range of computer vision applications. TensorFlow is used in CNNTSR (Traffic Sign Recognition), a key component of current driving assistance systems that improves driver safety and comfort. TensorFlow is used to implement CNNTSR (Traffic Sign Recognition). This article examines a technology that assists drivers in recognizing traffic signs and avoiding road accidents. Two things determine the accuracy of TSR: the feature extractor and the classifier. Although there are a variety of approaches, most recent algorithms use CNN (Convolutional Neural Network) to do both feature extraction and classification tasks. Using TensorFlow and CNN, we create traffic sign recognition. The CNN will be trained using a dataset of 43 distinct types of traffic signs. The accuracy of the findings will be 95 percent.

Keywords: Driver, Tensor flow, Data Sheet, Alert, CNNTSR, ADAS.

I. INTRODUCTION

To ensure continuous upgrading and maintenance, road and traffic signs must be placed in the proper areas. According to transportation specialists in Scotland and Sweden, a complete inventory of traffic signs is not only missing, but also crucial. By offering a quick mechanism to detect, categorize, and log signs, an autonomous traffic sign detection and identification system can assist achieve this aim. This strategy makes it easier to create an accurate and consistent inventory. Human operators will have an easier time finding distorted or obstructed signs when this stage is completed. For example, research into the recognition of road and traffic sign language can aid in the construction of an inventory system (which does not require real-time recognition) or an in-car warning system (which does require real-time recognition). Both road sign inventory and road sign identification are concerned with traffic signs, confront comparable challenges, and rely on automatic detection and recognition systems to perform their responsibilities. ITS and Advanced Driver Assistance Systems (ADAS) have risen in prominence in recent years because of autonomous TSR. As a result, they are either on the side of the road or over it. These signs, which contain necessary information about advising, admonishing, and warning drivers, can aid in the management and improvement of a driver's conduct. TSR includes speed limits, no entrance restrictions, traffic lights that require a left or right turn, children crossing the street, and no large vehicles passing, to name a few.

This is how you figure out what kind of traffic sign you're looking at. TSR has a direct impact on driver safety as a result of their ignorance. Automatic solutions based on signal detection and recognition are being developed to assist drivers in rectifying the most harmful driving behaviors.

Advanced driver assistance systems are designed to collect vital data from drivers and provide it to them in a user-friendly format, making safe driving easier. A variety of factors must be considered by drivers, including vehicle speed and direction, as well as passing vehicles. The stress on drivers will be significantly reduced if driver assistance technologies record this data. Colors and simple geometric forms are used to draw the driver's attention to traffic signals. The research on identifying local road traffic signs is limited and in its early phases. The main concentration has been on traffic signs for local roads, and they've been in the early stages of development, with a focus on traffic sign identification utilizing static images.

A. Advanced Driver Assistance Systems

ADS (Advanced Driver Assistance Systems) are technologies that attempt to give drivers with crucial information about road and traffic conditions, automate certain complicated or repetitive duties, and enhance overall road safety for both drivers and pedestrians. Human error is at fault for 94 percent of all automotive accidents, according to the National Highway Traffic Safety Administration [1.] Recognition mistakes, judgement errors, and performance defects are the most common forms of driving errors that result in accidents. Based on our findings, we've concluded that developing and implementing solutions to reduce or eliminate accidents should be a key priority. Advanced driver assistance systems (ADAS) are becoming more popular in automobiles.

Various alternative ADAS systems have been presented during the last two decades. For example, GPS navigation has been around since the 1990s and is a widely used technology. In recent years, further adaptive cruise control, adaptive light beam control, automated braking, automatic parking, collision avoidance systems, blind spot detection, driver tiredness detection, hill descent control, night vision, and lane departure warning systems have all been created. The purpose of these devices is to keep people and automobiles safe. These algorithms, on the other hand, mostly ignore the actions of the driver. Using a Traffic Sign Detection and Recognition (TSDR) approach, we want to show that driver gaze habit is a significant aspect of safety.

II. LITERATURE REVIEW

Template matching is another prominent approach for image processing and pattern detection. Template Matching (TM) is a machine vision technique that finds parts of an image (or a sequence of pictures) that match a certain visual pattern [2]. TSDR systems also employ this strategy. For example, image matching approaches for TSDR were developed by researchers in [3, 4], [5, 6], and [7, 8]. Gavrilu's shape-based method [6], which is based on distance transformations and template matching, is also worth highlighting. The first step in this procedure is to find the source pictures' edges. The next stage is to make DT images.

According to Kumar et al [7], capsule networks can recognize traffic indications. The structure of a multi-parameter deep learning network. From start to finish, Yuan et al. [8] developed a method for recognizing traffic signals. Multiple characteristics from pictures of various sizes may be extracted, and a vertical space sequence attention module can be used to gather background information around the detected image, resulting in strong detection performance in challenging road traffic situations. Several methodologies have increased traffic sign recognition accuracy, but algorithms still have advantages and disadvantages that are constrained by a variety of conditions. Interruptions such as severe weather, illumination changes, and signs fading, according to researchers, contribute to poor environmental adaptation and lower traffic sign detection accuracy [9,10,11].

To categorize recorded traffic signs and provide proper real-time input to smart vehicles, traffic sign identification employs an effective classification algorithm based on current dataset resources. When you're given a detection image to look at. A forward-learning and reinforcement system might be used to improve the capacity to categorize and recognize traffic signs by simulating the human brain's sensory cognitive process [12,13]. In this part, the limitations of the classic LeNet-5 model are examined, and the model is significantly improved to take advantage of CNN's excellent advancements in graphics recognition. The LeNet-5 Network Model's Deficiency was examined. Prof. Yann Lecun developed the LeNet-5 network model in 1998. Six convolutional and two pooling layers, as well as a single layer, make up the LeNet-5 network model. [14,15]

III. OBJECTIVES

To recognize traffic signals, neural networks and image processing techniques are employed in this article. Its primary goals are as follows:

- 1) Because it recognizes signals without diverting or breaking their focus, this proposed solution does not distract or annoy drivers.
- 2) Maintain a safe and efficient traffic flow by using signaling devices.
- 3) To raise awareness of road conditions among drivers.
- 4) Interpret traffic movement to assist in the safe functioning of crossing intersections.

IV. SYSTEM ANALYSIS

A. Background

Every traffic sign detection and identification system require a dataset. A significant number of samples of that item must be supplied to train and assess a detector for detecting an object based on multiple properties and classifiers. Several research groups have been working on traffic sign databases for detection, identification, and tracking during the last few years. The scientific community has free access to these datasets. A list of some of these datasets is shown in Figure 1.

Two big and well-known datasets are the German Traffic Sign (GTS) dataset and the Belgian Traffic Sign (BTS) dataset. Traffic sign detection and identification can be aided by the German Traffic Sign Recognition Benchmark (GTSRB) and the Belgian traffic sign categorization bench- mark.

The German Traffic Sign Recognition Benchmark (GTSRB) was first presented at the International Joint Conference on Neural Networks in 2011 and is utilized in this study (IJCNN). Traffic signs from Germany's real road traffic environment were captured and made into a standard traffic sign dataset by computer vision specialists, self-driving researchers, and other researchers. For the GTSRB, a total of 51,839 pictures were taken and categorized. The training and testing set each contain 39,209 and 12,630 photos, accounting for around 75 percent and 25% of the total, respectively. Each image has only one traffic sign, which is not always visible. As seen in Figure 2, the GTSRB has 43 distinct types of traffic signs. Therefore, each type of sign has its own library, which contains a CSV file and a multi-track image (each track includes 30 images). GTSRB requires an artificial dataset after picture preprocessing. Because of the GTSRB's changing number of traffic sign types, sample data may become skewed. Different types of traffic signs have different characteristics when it comes to categorization and identification, which has an impact on the overall traffic sign network model's generalization. A new fake dataset is created by selecting each attribute feature of a certain sample type at random from the value space.

Dataset	Description	Paper
GTSD	Country of origin: Germany Number of images: 5000 Classes: 43	[27]
GTSRB	Country of origin: Germany Number of images: 51839	[27]
ATIS	Country of origin: Austria Number of images: 176	[28]
BTSDB	Country of origin: Belgium Number of images: 726	[29]
ETSD	Country of origin: France Number of images: 1000	[30]
ITSD	Country of origin: France Number of images: 1000	[31]
USA	Country of origin: United States Number of images: 1000 Classes: 43	[32]

Figure 1 Data sets of Traffic

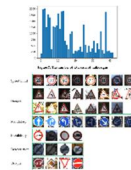


Figure 2 Types of traffic signs

V. SYSTEM ARCHITECTURE

In the field of image classification, the current generation of convolutional neural networks (CNNs) has shown impressive results. We're building a CNN-based model with enough layers to allow for adequate picture discrimination since our research focuses on a novel approach to traffic monitoring identification models based on leaf image categorization. The focus of this study will be on CNN and its architecture. Convolutional Neural Networks are composed of neurons with learnable weights and biases, and their structure and function are quite similar to those of traditional neural networks. For each input, a dot product is computed, and non-linearity is optionally applied. The whole network is scored using a single differentiable scoring function, from raw picture pixels on one end to class scores on the other. On the final (completely linked) layer.

A. Convolutional Neural Networks (CNNs / ConvNets)

In computer vision, a Convolutional Neural Network (CNN) is a deep learning system that detects and categorises characteristics in pictures. This multi-layer neural network can classify, segment, and recognise objects in images. For each input, a dot product is computed, and non-linearity is optionally introduced. On the last (fully connected) layer, a loss function (e.g., SVM/Softmax) remains, and all the tips/tricks we learnt for training traditional Neural Networks still apply.

B. Architecture Overview

There are several hidden layers that affect input in neural networks (a single vector). Neurons in a single layer of each buried layer, which is made up of a group of neurons, work fully independently and do not share any connections. It's the last layer, and it's used to represent class scores in classification settings. Layers used to build ConvNets: ConvNets are made up by layers, each of which transforms activations from one volume to another using a differentiable function. The Convolutional Layer, Pooling Layer, and Fully Connected Layer are the three primary types of layers used in ConvNet topology (exactly as seen in regular Neural Networks). We'll build a comprehensive ConvNet architecture by stacking these layers.

As an example, consider architecture: Overview. The following is an example of a simple CIFAR-10 classification ConvNet architecture: [INPUT - CONV — RELU — POOL — FC] [INPUT - CONV — RELU — POOL — FC] [INPUT - CON Adding a little more detail

INPUT [32x32x3] in this example a picture with 32x32 pixels and three colour channels R,G,B.

- The output of neurons associated to specific places in the input will be calculated by the CONV layer, which will compute a dot product between their weights and a little region in the input volume to which they are related. The volume may be [32x32x12] if we were to use 12 filters.
- RELU will apply an elementwise activation function to the zero layer, such as $\max(0,x)$. The volume is unchanged ([32x32x12]).
- POOL reduces the spatial extent (width and height), resulting in, for example, a volume of [16x16x12].
- FC (fully connected) has a volume of [1x1x10], with each number reflecting a class score, such as CIFAR-10. This layer's neurons will be linked to all of the numbers from the preceding volume.

Layer-by-layer, The RELU/POOL layers, on the other hand, will perform a predetermined purpose. To ensure that the ConvNet's class scores match the labelling in the training sample for each picture, the input vector will be utilised to train the parameters in the CONV/FC layers.

Convolutional Layer: A Convolutional Network's primary component, the Conv layer, is in charge of the majority of the computational work.

1) Summary

To summarize, the Conv Layer:

- Accepts a volume of size $W1 \times H1 \times D1$
- Requires four hyperparameters:
 - Number of filters KK ,
 - their spatial extent FF ,
 - the stride SS ,
 - the amount of zero padding PP .
- Produces a volume of size $W2 \times H2 \times D2$ where:
 - $W2 = (W1 - F + 2P) / S + 1$
 - $H2 = (H1 - F + 2P) / S + 1$ (i.e. width and height are computed equally by symmetry)
 - $D2 = KD2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D1$ weights per filter, for a total of $(F \cdot F \cdot D1) \cdot K$ weights and KK biases.

The output volume's dd -th depth slice (of size $W2H2W2H2$) is the result of appropriate convolution of the dd -th filter with a cadence of SS across the input volume, then offset by dd -th bias.

$F=3, S=1, P=1$ is a popular hyperparameter arrangement. Hyperparameters, on the other hand, are regulated by established standards and rules of thumb. See the ConvNet designs section below for further information on how to use ConvNet..

1x1 convolution: If the input is [32x32x3], 1x1 convolutions would be comparable to 3-dimensional dot products (since the input depth is 3 channels).

2) Pooling Layer

In a ConvNet architecture, at regular intervals between consecutive Conv layers, a Pooling layer is frequently incorporated. You may use it to progressively reduce the network's spatial size to manage overfitting, parameters, and computations. The Pooling Layer spatially resizes each slice of the input, functioning on each depth slice individually and using the MAX operation. Pooling layers with 2x2 filters applied with a stride of 2 are the most prevalent. Every depth slice in the input stream is down sampled by two in both width and height, eliminating 75% of the action potentials. Every MAX operation in this situation would need a maximum of four digits (little 2x2 region in some depth slice). In this situation, the depth dimension remains unchanged. The process of pooling is seen in Figure 3.

3) General Pooling

The pooling units may execute a variety of jobs in addition to max pooling, such as average pooling and L2-norm pooling. As a result, the average pooling operation has lost favour in favour of the maximum pooling operation, which has been shown to perform better in real-world scenarios. Figure 4 depicts striding

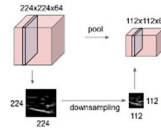


Figure 3 Illustration of Pooling

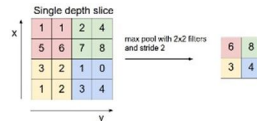


Figure 4 Illustration of Striding

4) Backpropagation

Remember that the backward pass of a max (x, y) operation may simply be understood as delivering the gradient to the input with the highest value during forward propagation. Gradient routing may be more efficient during training algorithm since the maximum activation index (also known as switches) is typically recorded during the forward run.

5) Normalization Layer

It has been hypothesized that ConvNet topologies use a variety of normalization layers, sometimes with the intention of simulating inhibiting mechanisms seen in real brains. As a result, these layers have become less popular. For an explanation of numerous types of normalizations, see Alex Krizhevsky's cuda-convnet library API.

6) Fully Connected Layer

Neurons in a layer of a normal neural network are entirely connected to all activations in the layer before it. As a consequence, using matrix multiplication and bias offset, their activations may be determined. In the notes section, you may learn more about neural nets.

7) FC->CONV Conversion

The ability to convert an FC layer to a CONV layer is the more practical of the two conversions. Consider a ConvNet architecture that reduces a 224x224x3 image to a linear activation volume of size 7x7x512 using a series of CONV and POOL layers (in an AlexNet architecture that we'll see later, this is accomplished by using 5 pooling layers that maintain and minor the input spatially by a factor of two each time, resulting in a final spatial size of $224/2/2/2/2/2 = 7$). To compute class scores, an AlexNet employs two FC layers with a total capacity of 4096 neurons, followed by a final FC layer of 1000 neurons. Each of these three FC levels can be converted, as previously indicated.

- Instead of a first FC layer that examines a volume of [7x7x512], use a CONV layer with a filter size of $F=7F=7$, resulting in an output volume of [1x1x4096].
- Instead of the second FC layer, use a CONV layer with a filter size of $F=1F=1$ and an output volume of [1x1x4096]. achieved by replacing the last FC layer with $F = 1F = 1$ in the same fashion, resulting in [1000 by 1x 1000].

The weight matrix WW in each FC layer would need to be changed (e.g., reshaped) for each of these conversions before it could be utilized in the CONV layer filters. With this conversion, it turns out that the original ConvNet can be "slid" over a large number of spatial regions in a larger image in a single forward pass.

8) ConvNet Architectures

There are only three types of layers in Convolutional Networks: CONV, POOL (we assume maximum pool unless otherwise specified), and FC (short for fully connected). A layer with initial feature semi will be used to express the Activation function's function. Many ConvNets are stacked on top of one another to create ConvNets..

VI. METHODOLOGY

Rather, we started with a data-gathering technique. We used a variety of approaches to investigate the data before presenting it using EDA tools. Finally, we developed a CNN-based classification model in this study. Following that, the model is trained and checked before being submitted to the test. Following that, our classifier was activated. The CNN model, as previously said, is based on a convolution of filters and pictures or raw inputs. Here, we'll discuss the CNN-based learning technique. Figure 5 is a flowchart depicting the critical phases. . Image recognition and classification is one of CNN's most important functions. CNN's image classification system analyses and categorizes the images it receives (Eg., Dog, Cat, Tiger, Lion). Each input picture will be passed through a succession of convolution operations with filters (Kernals), Pooling, and fully connected layers (FC), as well as the Softmax function to categorize each item with probabilistic values between 0 and 1, to train and analyse deep learning CNN models. Figure 6 demonstrates how to use equation to minimize the convolutional layer's computation time (1):

The convolution kernel's weight value, for example, is $k_{w_{ij}}$, and the input pixel branch of economics that deals with that weight is ijx . The first convolution kernel's offset is referred to as the offset. RLU activation functions are commonly used by CNN (Rectified Linear Unit). Figure 7 depicts how CNN analyses a picture and categorizes objects based on their values.



Figure 5 Flow chart of the keyphases of CNN

$$a^k = f \left(\sum_{i,j=1}^3 w_{ij}^k \times x_{ij} + b^k \right) \quad (1)$$

Figure 6 Equation of the convolution layer

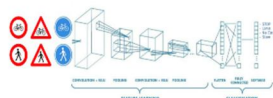


Figure 7 Processing a picture in CN:N

A. Neural Network with Many Convolutional Layers

The initial layer in the process of extracting features from an input picture is the Convolution Layer. By employing tiny input data squares, convolution retains the link between pixels. Two inputs are required to complete this mathematical procedure: an image matrix and a filter or kernel. Non-linearity (ReLU) is an abbreviation that stands for RLU and refers to a non-linear process. In our ConvNet, it's used to add non-linearity. Our ConvNet would need to learn non-negative linear values since the data in the actual world is non-negative linear. The pooling layer minimizes the number of parameters when the pictures are extremely huge. Spatial pooling, sometimes referred to as subsampling or down sampling, is a mapping approach that minimizes spatial complexity while keeping crucial data. There are a few different kinds of spatial pooling.:

- 1) Max Pooling
- 2) Average Pooling
- 3) Sum Pooling

The biggest element from the corrected feature map is used in max pooling. Taking the greatest factor might lead to the elements being averaged out. Summation is the process of adding up all of the components in a feature map.

B. Data Preprocessing

Kaggle's publicly accessible dataset was chosen for this investigation. To depict traffic signs, more than 50,000 photos are used. For it, 43 subcategories have been created. Figure 8 depicts a couple of them. Data separated into training and test sets is used to train a neural network.



Figure 8 Data sets pf traffic sign

The 'train' folder has 43 folders, each of which represents a distinct class. The folders range in size from 0 to 42. It is necessary to loop through all of the classes and add images and labels to the data-and-label list using the OS module. Each pixel has three values (RGB) that determine the color (Red, Green Blue). As a result, we turned the image to numbers for robots to grasp it better. The Python Imaging Library (PIL) is a powerful image manipulation tool. The images were then resized using a set of predetermined settings. A NumPy array was created by converting all of the images' data and labels into a list (data and labels). It doesn't matter how you slice it: (39209, 30, 30) indicates there are 39,209 30x30 pixel photographs, and the last three say there are 39,209 30x30 pixel photos.

C. Applying EDA Techniques

This is a three-step procedure for implementing EDA methods. We began by focusing on the dataset's enormous knowledge base. The following stages are used to clean the data.

- 1) *Step 1: Examine the Information:* Summarize the first and final five rows of the data collection, as well as the characteristics and values associated with them. Twelve thousand six hundred and thirty-three rows and eight columns make up the data set. As a result, any Deep Learning architecture might be supported by the amount of data available.
- 2) *Step 2: Data Cleansing:* Null or missing values were investigated further once the data had been analyzed.
- 3) *Step 3: Analyze the Relationship:* The data is summarized using a correlation matrix, which is then utilized as an input for a more advanced analysis. A heatmap can be used to display the correlation matrix. A two-dimensional (two-dimensional) data representation is a heatmap. Figure 9 displays the whole traffic flow. Experiment Recognize and sign



Figure 9 Entire Traffic Sign and recognition Experiment

VII. RESULTS AND OBSERVATIONS

As a result, we created and trained a classification system to classify the photos. We'll build our model with CNN because it's shown to be the best at picture classification on the market. Layers of convolutional and pooling convolutions are combined in a CNN. The characteristics of each layer are retrieved and utilized to categorize the picture. A dropout layer is also included to address model overfitting. During training, the dropout layer loses some neurons. Cross-entropy measures are used in the model since the dataset comprises numerous classes to categorize. Each layer of the CNN model has its own set of parameters, as shown in Figure 10, as listed in Table 4...

Layer Number	Layer Type
L1	Conv2D (32x3x3), ReLU
L2	Conv2D (32x3x3), ReLU
L3	MaxPool2D(layer/pool_size=(2,2))
L4	Dropout layer (rate=0.25)
L5	Conv2D (64x3x3), ReLU
L6	Conv2D (64x3x3), ReLU
L7	Flatten Layer (1 Dimension)
L8	DenseFullyconnected layer (256 nodes,ReLU)
L9	Dropout layer (rate=0.5)
L10	Dense Layer (43 nodes, softmax)

Figure 10 CNN Model Parameters

A. Train and Validate the Model

It's time to put the model's design to the test now that you've completed making it. The data has been prepared and a model has been developed. Setting up our model's training data, validation data, batch size, and number of epochs is the first step in the training process. As a result, in our classifier, we opted to utilize a range of CNN architectures. We tried a few different batch sizes and activation approaches. All the models that have been implemented are depicted in detail in Figure 11.

CNN Model No.	Batch Size	Epochs	Test Accuracy (%)
1	32	100	82.00
2	32	110	91.00
3	64	100	94.00
4	64	110	95.00

Figure 11 Information about the implemented model

The CNN4 model, on the other hand, has outperformed all of the others. We utilized it to make the concept further clearer. Therefore, a 64-batch model is used to train this model. Furthermore, the precision had remained steady after 110 epochs. On the training data, our model had a 95 percent accuracy rate. The graphs of accuracy and loss are drawn. This is depicted in Figures 12 and 13, with accuracy and loss, respectively.

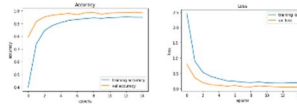


Figure 12 Right -shows loss Left- shows accuracy

We construct a graphical user interface for our traffic sign classifier towards the conclusion of the operation. Because the GUI (Graphical User Interface) is designed for uploading the photo into the system, we must use the same measurements that we used while developing the model in order to anticipate the road sign. Testing and viewing the outcomes of our model prediction will be much faster with a GUI.

We'll ask the user for a photo and then obtain the image's file location using the GUI application's interface. We'll use a trained model to receive the picture data as input and identify which class our image belongs in when we're ready.

B. Statistics and Analysis of Experimental Results

The capacity of the improved CNN network model to recognize a broad range of traffic signs is tested using photographs from six different types of traffic signs that are loosely stated in the dataset. The results of the classification and identification tests for six different traffic lights are shown in Figure 13.

Table 3. The classification and recognition test results of six categories of traffic signs.

Sequence Number	Traffic Signs Type	Test Images Number	TP	FN	Accuracy Recognition Rate (%)	Average Processing Time (ms/Frame)
1	Speed Limit	1000	997	3	99.7%	5.4
2	Danger	1000	999	1	99.9%	5.8
3	Mandatory	1000	997	3	99.7%	5.2
4	Prohibitory	1000	998	2	99.8%	4.9
5	Derestriction	1000	994	6	99.4%	6.4
6	Unique	1000	1000	0	100.0%	4.7
Total	-	6000	5985	15	99.75%	5.4

Figure 13 Identification tests for six signs

The letters TP (True Positive) and FN (False Negative) in Table 3 represent this. Because of the benefits of fixed outlines and distinguishing qualities, unique traffic signs perform better in classification and recognition tests. The average processing time per frame is 4.7 milliseconds, and the identification accuracy rate is 100.00 percent. Because of their homogenous shapes and comparable qualities, derestriction traffic signs perform the lowest in the tests. A 99.40 percent correct identification rate is also achieved with an average processing time per frame of 6.4 milliseconds. Six different types of traffic signs showed a 99.75 percent correct identification rate, with an average processing time per frame of 5.4 milliseconds for the whole collection. Therefore, the suggested traffic sign identification technique provides good real-time performance and flexibility, while the LeNet-5 network model with enhanced picture recognition provides excellent real-time performance. It's possible that sorting and analyzing test photographs will disclose misleading or missing images. Most of these photographs are the result of extremely poor resolution, motion blur, and significant tilt. To address this problem, more complicated network models must be created in the future, as well as larger datasets, to enable CNN in recognizing new traffic signals with interference features. As a result, the algorithm's stability and inclusivity are improving all the time.

VIII. CONCLUSION

In this study, we created and tested a system for detecting and recognizing warning traffic signals. To categorize the observed traffic signs, both color information and geometric aspects of the road signs are utilized. Experiments have shown that the system can identify objects with a 95% accuracy rate. As a result, the system produces accurate results in a variety of lighting circumstances, weather conditions, daylight conditions, and vehicle speeds. We were able to correctly categorise 95% of the traffic signs, and we were able to track changes in accuracy and loss over time, which is rather good for such a simple CNN model. As a result of this research, improved intelligent traffic monitoring systems that may be utilized for several purposes and applications may be developed...

REFERENCES

- [1] Shi, J.H.; Lin, H.Y. A vision system for traffic sign detection and recognition. In Proceedings of the 26th IEEE International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 18–21 June 2017; pp. 1596–1601.
- [2] Phu, K.T.; Lwin Oo, L.L. Traffic sign recognition system using feature points. In Proceedings of the 12th International Conference on Research Challenges in Information Science (RCIS), Nantes, France, 29–31 May 2018; pp. 1–6.
- [3] Wali, S.B.; Abdullah, M.A.; Hannan, M.A.; Hussain, A.; Samad, S.A.; Ker, P.J.; Mansor, M.B. Vision-based traffic sign detection and recognition systems: Current trends and challenges. *Sensors* 2019, 19, 2093. [CrossRef]
- [4] Wang, G.Y.; Ren, G.H.; Jiang, L.H.; Quan, T.F. Hole-based traffic sign detection method for traffic signs with red rim. *Vis. Comput.* 2014, 30, 539–551. [CrossRef]
- [5] Hechri, A.; Hmida, R.; Mtibaa, A. Robust Road lanes and traffic signs recognition for driver assistance system. *Int. J. Comput. Sci. Eng.* 2015, 10, 202–209. [CrossRef]
- [6] Lillo-Castellano, J.M.; Mora-Jiménez, I.; Figuera-Pozuelo, C.; Rojo-Álvarez, J.L. Traffic sign segmentation and classification using statistical learning methods. *Neurocomputing* 2015, 153, 286–299. [CrossRef]
- [7] Kumar, A.D.; Karthika, K.; Parameswaran, L. Novel deep learning model for traffic sign detection using capsule networks. arXiv 2018, arXiv:1805.04424.
- [8] Yuan, Y.; Xiong, Z.T.; Wang, Q. VSSA-NET: vertical spatial sequence attention network for traffic sign detection. *IEEE Trans Image Process* 2019, 28, 3423–3434. [CrossRef]
- [9] Li, Y.; Ma, L.F.; Huang, Y.C.; Li, J. Segment-based traffic sign detection from mobile laser scanning data. In Proceedings of the 38th Annual IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Valencia, Spain, 22–27 July 2018; pp. 4607–4610.
- [10] Pandey, P.; Kulkarni, R. Traffic sign detection using Template Matching technique. In Proceedings of the 4th International Conference on Computing, Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; pp. 1–6.
- [11] Banharsakun, A. Multiple traffic sign detection based on the artificial bee colony method. *Evol. Syst.* 2018, 9, 255–264. [CrossRef]
- [12] Liu, Q.; Zhang, N.Y.; Yang, W.Z.; Wang, S.; Cui, Z.C.; Chen, X.Y.; Chen, L.P. A review of image recognition with deep convolutional neural network. In Proceedings of the 13th International Conference on Intelligent Computing (ICIC), Liverpool, UK, 7–10 August 2017; pp. 69–80.
- [13] Rawat, W.; Wang, Z.H. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comp.* 2017, 29, 2352–2449. [CrossRef] [PubMed]
- [14] El-Sawy, A.; El-Bakry, H.; Loey, M. CNN for handwritten arabic digits recognition based on LeNet-5. In Proceedings of the 2nd International Conference on Advanced Intelligent Systems and Informatics (AISII), Cairo, Egypt, 24–26 October 2016; pp. 565–575.
- [15] Wei, G.F.; Li, G.; Zhao, J.; He, A.X. Development of a LeNet-5 gas identification CNN structure for electronic noses. *Sensors* 2019, 19, 217. [CrossRef] [PubMed]
- [16] Xiao, Z.T.; Yang, Z.J.; Geng, L.; Zhang, F. Traffic sign detection based on histograms of oriented gradients and boolean convolutional neural networks. In Proceedings of the 2017 International Conference on Machine Vision and Information Technology (CMVIT), Singapore, 17–19 February 2017; pp. 111–115.
- [17] Guan, H.Y.; Yan, W.Q.; Yu, Y.T.; Zhong, L.; Li, D.L. Robust traffic-sign detection and classification using mobile LiDAR data with digital Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2018, 11, 1715–1724.
- [18] Sun, Z.L.; Wang, H.; Lau, W.S.; Seet, G.; Wang, D.W. Application of BW-ELM model on traffic sign recognition. *Neurocomputing* 2014, 128, 153–159.
- [19] Qian, R.Q.; Zhang, B.L.; Yue, Y.; Wang, Z.; Coenen, F. Robust Chinese traffic sign detection and recognition with deep convolutional neural network. In Proceedings of the 2015 11th International Conference on Natural Computation (ICNC), Zhangjiajie, China, 15–17 August 2015; pp. 791–796.
- [20] He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
- [21] Yuan, Y.; Xiong, Z.T.; Wang, Q. An incremental framework for video-based traffic sign detection, tracking, and recognition. *IEEE Trans. Intell. Transp. Syst.* 2017, 18, 1918–1929. [CrossRef]
- [22] Zhu, Y.Y.; Zhang, C.Q.; Zhou, D.Y.; Wang, X.G.; Bai, X.; Liu, W.Y. Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing* 2016, 214, 758–766. [CrossRef]



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)