AN AUTOMATED SYSTEM FOR TRAFFIC SIGN RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

Sanam Narejo
Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro, (Pakistan).
E-mail: sanam.narejo@faculty.muet.edu.pk ORCID: https://orcid.org/0000-0002-3537-8949

Shahnawaz Talpur
Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro, (Pakistan).
E-mail: shahnawaz.talpur@faculty.muet.edu.pk ORCID: https://orcid.org/0000-0002-2660-6145

Madeha Memon
Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro, (Pakistan).
E-mail: madeha.memon@gmail.com ORCID: https://orcid.org/0000-0003-2147-0944

Amna Rahoo
Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro, (Pakistan).
E-mail: f16cs05@students.muet.edu.pk ORCID: https://orcid.org/0000-0002-9376-6528

Recepción: 13/11/2020 **Aceptación:** 13/11/2020 **Publicación:** 13/11/2020

Citación sugerida **Suggested citation**
Narejo, S., Talpur, S., Memon, M., y Rahoo, A. (2020). An automated system for traffic sign recognition using convolutional neural network. *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Noviembre 2020,* 119-135. https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

ABSTRACT

TSR (Traffic Sign Recognition) represents an important feature of advanced driver assistance system, contributing to the safety of the drivers, autonomous vehicles as well and to increase driving comfort. In today's world road conditions drastically improved as compared with past decades. Obviously, vehicle's speed increased. So, on driver's point of view there might be chances of neglecting mandatory road signs while driving. This paper explores the system to helps the driver about recognition of road signs to avoid road accidents. TSR is challenging task, while its accuracy depends on two aspects: feature extractor and classifier. Current popular algorithms mainly deploy CNN (Convolutional Neural Network) to execute both feature extraction and classification. In this paper, we implement the traffic sign recognition by using CNN, the CNN will be trained by using the dataset of 43 different classes of traffic signs along with TensorFlow library. The results will show the 95% accuracy.

KEYWORDS

Convolutional Neural Network, Traffic Sign Recognition, Autonomous Vehicles, Exploratory Data Analysis

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

1. INTRODUCTION

Road and traffic signs must be properly installed in the necessary locations and an inventory of them is ideally needed to help ensure adequate updating and maintenance. Meetings with the highway authorities in both Scotland and Sweden revealed the absence of but a need for an inventory of traffic signs. An automatic means of detecting and recognizing traffic signs can make a significant contribution to this goal by providing a fast method of detecting, classifying and logging signs. This method helps to develop the inventory accurately and consistently. Once this is done, the detection of disfigured or obscured signs becomes easier for human operator. Road and traffic sign recognition is the field of study that can be used to aid the development of an inventory system (for which real-time recognition is not required) or aid the development of an in-car advisory system (when real-time recognition is necessary). Both road sign inventory and road sign recognition are concerned with traffic signs, face similar challenges and use automatic detection and recognition. A road and traffic sign recognition system could in principle be developed as part of an ITS (Intelligent Transport Systems) that continuously monitors the driver, the vehicle, and the road in order, for example, to inform the driver in time about upcoming decision points regarding navigation and potentially risky traffic situations (Fleyeh, 2008).

The aim of intelligent transport systems is to increase transportation efficiency, road safety and to reduce the environmental impact with the use of advanced communication technologies (Sermanet, & LeCun, 2011; De la Escalera, Armingol & Mata, 2003). Automatic TSR, as an important task of Advanced Driver Assistance Systems and ITS has been of great interest in recent years. The road signs are placed on either roadside or above the roads. These signs provide mandatory information regarding to guiding, warning and regulating the behaviors to drivers in order to make driving safer and easier.

There are several different TSR like speed limits, no entry, traffic signals, turn left or right, children crossing, no passing of heavy vehicles, etc. Traffic sign classification/recognition is the process of identifying the, which class traffic sign belongs to. TSR has a direct impact on the safety of drivers, and damages can be easily produced due to their ignorance. Automatic systems are developed to assist the drivers, based on detection and recognition of signs which corrects the most unsafe driving behaviors.

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

The main purpose of advanced driver assistance systems is to collect significant information for drivers in order to reduce their effort in safe driving. Because drivers must pay attention to various conditions, including vehicle speed and orientation, passing cars and to many more. So, if driver assistance systems collect such information, it will greatly reduce the burden of drivers. Thus, road signs are designed in such a way that attracts a driver's attention with colors and simple geometric shapes.

The research work available on recognition of traffic signs for local roads is quite sparse and still at the preliminary stage. Mostly it is focused on recognition of traffic signs for local roads and they are also still at the preliminary stage and focused on recognition of traffic signs through static images. In this work, algorithms were developed in "Google Colab" environment to recognize traffic signs while vehicles in motion. Project is mainly focused on automatic recognition of warning signs placed in local roads captured by image clips. Traffic signs were recognized based on its geometrical characteristics and color information (Gunawardana 2010). In this project, we develop a deep NN (Neural Network) model that that can classify the traffic signs present in the image into different categories. With this model, we can read and understand traffic signs which are very important task for autonomous vehicles.

2. LITERATURE SURVEY

There are many researches in the literature dealing with Road TSR problem. According to Kale and Mahajan (2015), the road sign recognition system is to be divided into two parts, the first part is detection stage which is used to detect the signs from a whole image, and the second part is classification stage that classifies the detected sign in the first part into one of the reference signs which are presents in the dataset. They used PCA (Principle Component Analysis) and ANN (Artificial Neural Network) techniques for detection and recognition with the dataset of different road signs from Maharashtra RTO (Regional Transport Office). Tool used by them was MATLAB. For TSR, mostly the preferable model for image classification is CNN. The recognition system is implanted into autonomous vehicle 'Eurecar'. The system was followed by HSV (Herpes Simplex Viruses) and Hough transform algorithms for extracting ROI (Return on Investment) of traffic signs. Gaussian blurring was also applied as canny edge detectors. Then extracted area is given to CNN

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

model. They used dataset of 6 types of (Korean-Version) traffic signs to train the CNN model. The obtained results demonstrated low accuracy; it shows overlapping results when several proposal regions point to a same traffic sign. Development of clustering algorithm is considered as a future work for robust recognition system (Jung et. al. 2016). Most of the research used color segmentation technique with C-CNN with GERMAN traffic signs dataset for detection. The C-CNN method consists of selecting a set of ROIs by applying a color thresholding on the input image, thus reducing the search space. Then, a trained CNN is used to classify the ROI (whether it contains a traffic sign or not), followed by another CNN with the same architecture, that is used to recognize the detected traffic signs. Therefore, 2 datasets are selected, one for detection and another one, to recognize the traffic sign. Therefore, CNN was trained to recognize two classes: traffic sign/no traffic sign. It was concluded that C-CNN is slow and sensitive to weather conditions (Boujemaa *et al.*, 2017).

Researchers presented a three-stage real-time TSR system, consisting of a segmentation, a detection and a classification phase. They combine the color enhancement with an adaptive threshold to extract red regions in the image. The detection is performed using an efficient linear SVM (Support Vector Machine) with HOG (Histogram of Oriented Gradients) features. The tree classifiers, K-d tree and Random Forest, identify the content of the traffic signs found. A spatial weighting approach is proposed to improve the performance of the K-d tree. The Random Forest and Fisher's Criterion are used to reduce the feature space and accelerate the classification. They presented that only a subset of about one third of the features is enough to attain a high classification accuracy on the GTSRB (German Traffic Sign Recognition Benchmark (Zaklouta & Stanciulescu 2014). The paper proposed a method for Traffic Sign Detection and Recognition using image processing for the detection of a sign and an ensemble of CNN for the recognition of the sign. TensorFlow is used for the implementation of the CNN. They have achieved higher than 99% recognition accuracies for circular signs on the Belgium and German data sets (Vennelakanti *et al.*, 2019). Research based on TSR methods proposed mechanism for real time TSR using CNN. The training database was established by field sample collection, with which the neural network model was trained. SGD (Stochastic Gradient Descent) optimizer is utilized during training to improve the learning efficiency. The test results show that the proposed

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

method achieves good performance in speed, accuracy and robustness for real time TSR (Xu *et al.*, 2018).

R-CNN was the first to use this strategy, but it is very slow for two reasons (Girshick *et al.*, 2014). Firstly, generating category-independent object proposals is costly; it takes about 3s to generate 1000 proposals for the Pascal VOC 2007 images (Simonyan & Zisserman 2014). Secondly, it applies a whole deep CNN to every candidate proposal calculated, which is obviously very inefficient and time consuming. To improve efficiency, the SPP-Net (Spatial Pyramid-Pooling Network). Girshick *et al.* (2015) calculated a convolutional feature map for the entire image and extracts feature vectors from the shared feature map for each proposal. This speed up the R-CNN approach about 100 times. They have proposed the Fast R-CNN model, which is a faster version of the R-CNN approach. He *et al.* (2015) proposed RPNs (Region Proposal Networks), which generate object proposals using convolutional feature maps. This allows the generator of the object proposal to share the convolutional features of the whole image with the detection network. With this technique detection system can achieve a frame rate of 5 fps on a powerful GPU (Graphics Processing Unit). Szegedy *et al.* (2016) improved the network architecture, to achieve a frame rate of 50 fps in testing, with competitive detection performance.

In current traffic management systems, there is high probability that driver may miss some of the traffic signs on the road because of overcrowding due to neighboring vehicles. So, we have introduced the TSR system with an aim of detecting and recognizing all the emerging traffic signs.

3. METHODOLOGY

Initially, we opted for the data collection. We applied some techniques on the data for exploration and then visualized the data through EDA techniques. And finally, we build in this study a classification model based on the CNN. Afterwards, model is trained and validated and then based on the validated model, we attempted a test. And afterwards, we deployed our classifier. As we mentioned earlier that for classification, we have used CNN model, it is based on convolution of filters and images or raw inputs.

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

In this section, we will describe the CNN based approach. A flowchart of the main phases is shown in Figure 1.

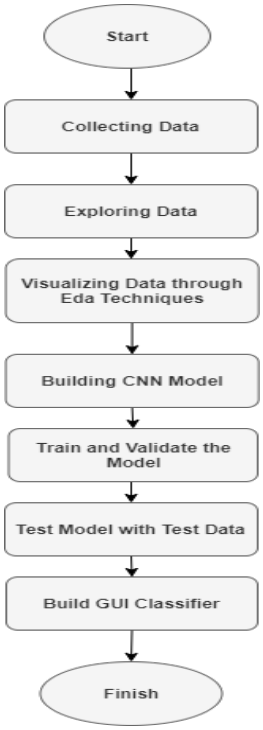

Figure 1. Flowchart of methodology.

In NNs, CNN is one of the main categories to do images recognition, images classifications. CNN image classifications take an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion).

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The calculation of the convolutional layer can be simplified as shown in Equation (1):

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

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

Where, k w_{ij} is the weight value of the convolution kernel; i_j x the input pixel value corresponding to the weight of the convolution kernel; the offset corresponding to the first convolution kernel; k b is the bias; f () x is the activation function. CNN commonly used activation functions as an RLU (Rectified Linear Unit).

The Figure 2 is a complete flow of CNN to process an input image and classifies the objects based on values.

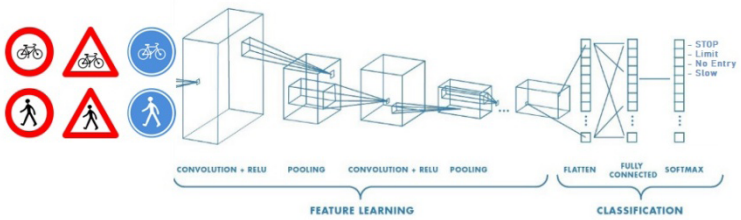

Figure 2. Neural network with many convolutional layers.

Convolution Layer is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

In Non-Linearity (ReLU): ReLU stands for RLU for a non-linear operation. Its purpose is to introduce non-linearity in our ConvNet. Since, the real-world data would want our ConvNet to learn would be non-negative linear values.

The Pooling Layer would reduce the number of parameters when the images are too large. Spatial pooling also called sub-sampling or down-sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

Fully Connected Layer is attached flattened the matrix into vector and feed it into a fully connected layer like a NN. The feature map matrix will be converted as vector (x_1, x_2, x_3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as SoftMax or sigmoid to classify the outputs as cat, dog, car, truck etc.

In the subsequent sections, we present the detailed steps of our methodology.

4. DATA PREPROCESSING

For this project, we have selected the publicly available dataset from Kaggle. The dataset contains more than 50,000 images of different traffic signs. It is further classified into 43 different classes. Few of them are shown in Figure 3. Prior to training a NN, data is divided in training and test sets.

Figure 3. Dataset of traffic signs.

The 'train' folder contains 43 folders each representing a different class. The range of the folder is from 0-42. With the help of OS module, we must iterate overall the classes and append images and their respective labels in the data and label list.

An image is made up of pixels and each pixel has 3 values to specify its color i.e. RGB (Red, Green Blue). In order for machines to understand the image, we converted the image into numbers. For this purpose, we use the PIL (Python Imaging Library) that can perform many image manipulations tasks. The subsequent step was of resizing the images into some uniform criteria. Therefore, we resize all the images to a fixed size, such as,30x30.Let's traverse through all the classes, open the image using PIL and also resize the image to 30x30 dimensions. Then we will append the data and label in the X and Y list respectively. After storing all the images and their labels into lists (data and labels), the list was further transformed into NumPy arrays for feeding the model. Nevertheless, finally the shape of

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

data is (39209, 30, 30, 3) which means that there are 39,209 images of size 30x30 pixels and the last 3 means the data contains colored images (RGB Value).

4.1. APPLYING EDA TECHNIQUES

In order to apply the EDA techniques, we have divided the process in three steps. Initially, we have focused on in-depth understanding for the dataset. Subsequently, some data cleaning is applied. EDA is the practice of using visual and quantitative methods to understand and summarize a dataset without making any assumptions about its contents. It is a crucial step to take before diving into machine learning or statistical modeling because it provides the context needed to develop an appropriate model for the problem at hand and to correctly interpret its results. Afterwards, we have attempted to find some correlations available in the dataset.

Step-1: Understanding the Data: Tables 1-2 present the details of data set by summarizing the first and last 5 rows respectively, its attributes and associated values. The dataset contains 12630 rows and 8 columns. This indicates that the data is enough to be used for any Deep learning architecture.

Table 1. Showing head function.

Data No.	Width	Height	ROI.X1	ROI.Y1	ROI.X2	ROI.Y2	Class Id	Path
12625	42	41	5	6	37	36	12	Test/12625.png
12626	50	51	6	5	45	46	33	Test/12626.png
12627	29	29	6	6	24	24	6	Test/12627.png
12628	48	49	5	6	43	44	7	Test/12628.png
12629	32	31	6	5	27	26	10	Test/12629.png

Table 2. Showing tail function.

Data No.	Width	Height	ROI.X1	ROI.Y1	ROI.X2	ROI.Y2	Class Id	Path
0	53	54	6	5	48	49	16	Test/00000.png
1	42	45	5	5	36	40	1	Test/00001.png
2	48	52	6	6	43	47	38	Test/00002.png
3	27	29	5	5	22	24	33	Test/00003.png
4	60	57	5	5	55	52	11	Test/00004.png

Step-2: Cleaning the Data: After understanding the data it was further examined null or missing values. Data set is free from missing values as depicted in Table 3.

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

Table 3. Showing cleaning process.

Width	0
Height	0
RoiX1	0
RoiY1	0
RoiX2	0
RoiY2	0
ClassId	0
Path	0
dtype	Int64

Step-3: Analyzing the Relationship: Correlation matrix is used to summarize the data as an input into more advanced analysis. We get visual representation of correlation matrix by heatmap. The heatmap is a way of representing the data in a 2D (Two-Dimensional) form. The data values are represented as colors in the graph. The goal of the heatmap is to provide a colored visual summary of information. It is a graphical representation of data where the individual values contained in a matrix are represented as colors. It is a bit like looking a data table from above. It is useful to display a general view of numerical data, not to extract specific data point. Figure 4 shows heatmap.

Figure 4. Heatmap.

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.

We used bar to plot graphs for any column's visual representation. Here we used width and height column of dataset with bar plot function. Figure 5 shows bar plot for input.

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

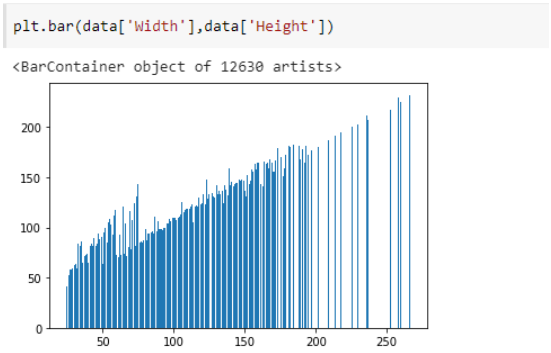

Figure 5. Bar plot.

5. EXPERIMENTAL SETUP FOR CNN

To classify the images into their respective categories, we developed and trained. The CNN have proved the state of the art in image classification tasks and this is what we will be using for our model. A CNN is made up of convolutional and pooling layers. At each layer, the features from the image are extracted that helps in classifying the image.

Apart from this, a dropout layer is also added, which is used to handle the over fitting of the model. The dropout layer drops some of the neurons while training. Finally, the model is compiled with cross entropy measures, because the dataset has multi classes to be classified. The Table 4 shows layers in details of CNN model along with parameters.

Table 4. Architecture of CNN model.

Layer Number	Layer Type
L1	Conv2D (32x5x5), ReLU
L2	Conv2D (32x5x5), ReLU
L3	MaxPool2Dlayer(pool_size=(2,2))
L4	Dropout layer (rate=0.25)
L5	Conv2D (64x5x5), ReLU
L6	Conv2D (64x5x5), ReLU
L7	Flatten Layer (1 Dimension)
L8	DenseFullyconnected layer (256 nodes,ReLU)
L9	Dropout layer (rate=0.5)
L10	Dense Layer (43 nodes, softmax)

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

5.1. TRAIN AND VALIDATE THE MODEL

After building the architecture of model. The model is defined, and the data is ready. To start the training of our model we initialize the training set, validation set, batch size and no of epochs.

So, for classifier, we opted for multiple architectures of CNN model. We experimented with various number of batch sizes and activation functions. The Table 5 shows the details of all our implemented models.

Table 5. CNN trained model summary.

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

However, as we find out that CNN4 has outperformed to the rest of other models. Therefore, we used this model to further demonstrations associated with it.

Therefore, this model is trained with batch size 64. And after 110 epochs, the accuracy was stable. Our model got 95% accuracy on the training dataset. We plot the graph for accuracy and loss. Figures 6-7 showing accuracy and loss respectively.

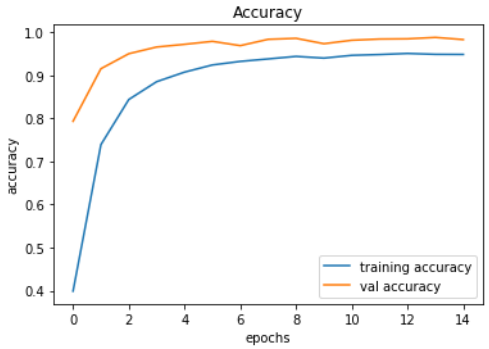

Figure 6. Accuracy of trained model CNN4.

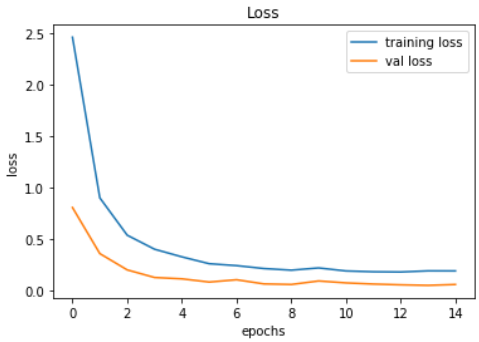

Figure 7. Loss of trained model CNN 4.

Finally, we build a graphical user interface for our traffic signs classifier. The GUI (Graphical User Interface) is built for uploading the image and to predict the traffic sign we must provide the same dimension we have used when building the model. A GUI will save a lot of time in testing and seeing the results of our model prediction.

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

From the interface of the GUI application, we will ask the user for an image and extract the file path of the image. Then we use the trained model that will take the image data as input and provide us the class our image belongs to.

6. RESULTS

This section presents the series of experiments carried out to validate the CNN approach for recognition of road signs. To get the road images, we use camera of 16 megapixels, to get the good distribution of images at different angles and positions. The images in each class are randomly chosen.

By observing Figures 8-9, we got an understanding of how autonomous vehicles can take advantage of CV (Computer Vision) and Deep Learning techniques to automatically recognize and classify from multiple classes.

Figure 8. Showing results of speed Limit(30km/hr).

Figure 9. Showing results of turn right ahead.

7. CONCLUSION

In this research work, we demonstrated and developed an efficient alert traffic sign detection and recognition system. Both color information and the geometric property of the road signs are used to classify the detected traffic signs. The experiment shows that the system can achieve a high detection rate of 95%.System is giving accurate results under different illumination conditions, weather conditions, day light conditions and different speed levels of the vehicle. We have successfully classified the traffic signs classifier with 95% accuracy and visualized how our accuracy and loss changes with time, which is pretty good from a

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

simple CNN model. The techniques implemented in this research can be used as a basis for developing general purpose, advanced intelligent traffic surveillance systems.

ACKNOWLEDGEMENT

We are thankful to the Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro, Pakistan, for providing facilities to conduct this research work.

REFERENCES

Boujemaa, K.S., Berrada, I., Bouhoute, A., & Boubouh, K. (2017). Traffic Sign Recognition Using Convolutional Neural Networks. In *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1-6. https://doi.org/10.1109/WINCOM.2017.8238205

De la Escalera, A., Armingol, J. M., & Mata, M. (2003). Traffic Sign Recognition and Analysis for Intelligent Vehicles. *Image and Vision Computing, 21*(3), 247-258. https://e-archivo.uc3m.es/bitstream/handle/10016/7089/traffic_escalera_IVC_2003_ps.pdf

Fleyeh, H. (2008). *Traffic and Road Sign Recognition* (Doctoral Dissertation). Napier University. https://www.diva-portal.org/smash/get/diva2:523372/FULLTEXT01.pdf

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (**2015**). Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(1), 142-158. https://doi.org/10.1109/TPAMI.2015.2437384

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

Gunawardana, P.M. (**2010**). *Automatic Detection and Recognition of Traffic Signs* (Doctoral Dissertation). University of Colombo, Sri Lanka.

He, K., Zhang, X., Ren, S., & Sun, J. (**2015**). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(*9), 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

Jung, S., Lee, U., Jung, J., & Shim, D.H. (**2016**). Real-Time Traffic Sign Recognition System with Deep Convolutional Neural Network. In *IEEE 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI),* pp. 31-34 https://www.semanticscholar.org/paper/Real-time-Traffic-Sign-Recognition-system-with-deep-Jung-Lee/17a608865ba3c6b2c27fcb18973c27139560954a

Kale, A.J., & Mahajan, R.C. (**2015**)**.** A Road Sign Detection and the Recognition for Driver Assistance Systems. *IEEE International Conference on Energy Systems and Applications,* pp. 69-74.

Sermanet, P., & LeCun, Y. (**2011**)**.** Traffic Sign Recognition with Multi-Scale Convolutional Networks. In *The 2011 International Joint Conference on Neural Networks,* pp. 2809-2813. https://doi.org/10.1109/IJCNN.2011.6033589

Simonyan, K., & Zisserman, A. (**2014**). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint.* https://arxiv.org/abs/1409.1556

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (**2016**)**.** Inception-V4, Inception-Resnet and the Impact of Residual Connections on Learning. *arXiv Preprint*. https://arxiv.org/abs/1602.07261

Vennelakanti, A., Shreya, S., Rajendran, R., Sarkar, D., Muddegowda, D., & Hanagal, P. (**2019**). Traffic Sign Detection and Recognition Using a CNN Ensemble. In *2019 IEEE International Conference on Consumer Electronics (ICCE),* pp. 1-4. https://doi.org/10.1109/ICCE.2019.8662019

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135

Xu, S., Niu, D., Tao, B., & Li, G. (**2018**). Convolutional Neural Network Based Traffic Sign Recognition System. *IEEE 5th International Conference on Systems and Informatics (ICSAI),* pp. 957-961.

Zaklouta, F., & Stanciulescu, B. (**2014**). Real-Time Traffic Sign Recognition in Three Stages. *Robotics and Autonomous Systems, 62*(1), 16-24. https://doi.org/10.1016/j.robot.2012.07.019

https://doi.org/10.17993/3ctecno.2020.specialissue6.119-135