

CNN Based Approach for Traffic Sign Recognition System

Karan Singh¹, Nikita Malik^{2*}¹ Bhai Parmanand Institute of Business Studies (BPIBS), GGSIP University, Shakarpur, New Delhi-110092² Dept. of Computer Applications, Maharaja Surajmal Institute (MSI), C-4, Janakpuri, New Delhi-110058

* Corresponding Author email:

nikitamalik@msijanakpuri.com

Article History

Received: 26 May 2021

Revised: 10 August 2021

Accepted: 15 September 2021

Published: 26 September 2021

Student(s)

- Karan Singh

Academic Year: 2020-2021

Course Level: Master

Course Name: MCA

Course year: 1st Year/2nd Semester

Mentor(s)

- Nikita Malik

ABSTRACT

Machine Learning (ML) involves making a machine able to learn and take decisions on real-life problems by working with an efficient set of algorithms. The generated ML models find application in different areas of research and management. One such field, automotive technology, employs ML enabled commercialized advanced driver assistance systems (ADAS) which include traffic sign recognition as a part. With the increasing demand for the intelligence of vehicles, and the advent of self-driving cars, it is extremely necessary to detect and recognize traffic signs automatically through computer technology. For this, neural networks can be applied for analyzing images of traffic signs for cognitive decision making by autonomous vehicles. Neural networks are the computing systems which act as a means of performing ML. In this work, a convolutional neural network (CNN) based ML model is built for recognition of traffic signs accurately for decision making, when installed in driverless vehicles.

Keywords: Machine Learning, Convolutional Neural Network, Traffic Signs.

1 Introduction

Traffic signs play a significant role in managing the road traffic, avoiding any accidents, and ensuring safety by conveying different norms or rules that need to be followed on the road. There are several types of traffic signs like speed limits, no entry, traffic signals, turn to left or right, children crossing, no passing of heavy vehicles, etc. To make them easily readable and recognizable, traffic signs are often designed to be of a particular shape and color with symbols inside such that there is a significant difference between the sign and the background. Traffic sign classification is the process of identifying which class a traffic sign belongs to [1].

Neural networks can be used to make a traffic sign recognition system capable of recognizing and understanding the different traffic signs and help the vehicles in which they are installed to decide and abide by these rules [2]. In other words, the problem statement of this work is: To train and build a neural network supported model which can recognize different traffic signs and take the adequate decision when imported in a vehicle. Such a neural network model can help in controlling accidents which are caused as a result of over-speeding or dizziness of driver.

1.1 Machine Learning

Machine learning (ML) is a branch of Artificial Intelligence (AI) which deals with structured data and implements human-like learning capabilities in machines by modifying its algorithmic implementation as demanded by the problem. In the system being aimed at in this work, supervised learning is implemented where the output provided with dataset's input helps in generating desired algorithm for the sake of doing classification task accurately on traffic sign images. Such types of systems act like being guided by some force in their early phase due to sufficient information provided to them for the needed result types; which is absent in unsupervised tasks [3]. Deep learning, in contrast to machine learning, deals with many layers to get the desired result and thus, said to be composed of different layers of inter-connected neural networks, which helps in further efficient data interpretation without requiring any sort of human intervention. Its accuracy and precision quality depends on the input data quality.

1.2 Convolutional Neural Networks

A neural network can be referred to as a machine learning algorithm which is composed of input, hidden and output layers and does processing of data which is needed for simulating behavior of electro-chemical signals generated by the neurons in various conditions. Neural networks find their applications in doing regression related tasks on attributes of continuous nature and mining data from classification-based problems. Deep learning relies on ANN (Artificial Neural Networks) which are layered network models which are designed to perform computational work like the one done by neurons which results in different types of biological behavior in animals and humans by analyzing patterns. The main concept which enables machines to think like humans is achievable via Neural Networks. Convolutional neural network (CNN) is the backbone neural network which enables face and image recognition which comes under machine learning's aim of providing computer vision [4]. CNN model is best suited for image classification and is used to classify images into different categories to which they belong. When CNN models are employed, much of the data pre-processing tasks are not required [5]. Its main operations are Convolution-where extraction of image features is done, Pooling-for dimensionality-reduction while not losing important information related to features, ReLU activation function-for non-linearity CovNet model and Classification of images into different classes to which they belong [1].

1.3 Literature Survey

For making auto-working and smart vehicles lot of efforts are already made in form of convolution network trained models with different proposed methodologies for detection and recognition. A recent approach [6] having wide applications for traffic sign detecting and recognizing intelligent transportation systems that informs driver about precaution measures and sign related information, uses color, shape and ML algorithm-based methods, and provided comparative information on the same. Color space, segmentation method, features, and shape detection method are the terms considered in the review of the detection module. The paper presents a comparison between these methods and used datasets from different countries.

Some of the important datasets and methods worth noting are Faster Region CNN (FRCNN) [7] and improved LeNet-5 network model that incorporated deep learning models for task accomplishment. A few other SSD and SVM based models were also proposed. CNN models already trained on ImageNet dataset were employed for recognition tasks including detection followed by classification of sign. Faster RCNN was deployed with ResNet model and SSD along with inception V2 for performance analysis on GTSRB dataset with performance standard measure being used as frames per second (fps) and mAp. Not a much significant difference was achieved but performance of YOLO v2 [8] with Coco CNN was higher than previous two for classification in real time. Another approach was to normalize image at preprocessing stage given as input to VGG-16 of SSD algorithm, with model built with a single softmax layer at the final end, five convolution layers stacked as first layer with middle layer being three layer that were fully

connected. Training batch size was 50 and 20 with 20,000 iterations to produce 96% accuracy and a learning rate of 0.001. An approach proposed by Dan Ciresan et al. [9] equalized sizes of various dataset images and used contrast normalizing techniques at pre-processing stage to normalize high and low contrast images results of which were fed to 9 layer model with three maxpooling and convolution layer between seven hidden layer at input and two fully connected layers by Dan Ciresan et al. [9], that was trained on eight different datasets. The grayscale images produced from model with images from different datasets with original dataset images were used further inclusive of contrast normalized images. Images were translated, scaled and rotated at a specified uniformly distributed range prior to every epoch making recognition rate to be 98.73% with CNN and rate of 99.15% rate for MLP combined with CNN. CNN combined with MLP gave 99.15% and CNN alone gave 98.73% recognition rate. However, the 'no vehicle' sign was not classified accurately by both. Approach incorporated was [10] proposed for task mentioned afore on basis of CNN with considering adverse environmental situations also. Noise due to rain, dirt, shade, and lighting distorted clarity of real time image. Noise in dataset and other challenges detected by virtue of VGG-16 architecture alike model based on CNN. Different methods can be used for reducing distortions from images, for contrast and noise using CLAHE and to remove error due to rain by using ResNet supported architecture can be used to remove rain from the images. A deep CNN with U-Net architecture is used to localize traffic signs from the preprocessed image. Finally, for the classification task, two convolution blocks, each with two convolution layers, a ReLU activation layer followed by a Maxpool layer, with a dropout layer added after each block was used. An approach [11] to segment traffic sign from background scene that involves gabor filter for linear analysis of texture using orientation and frequencies in image at certain region which is very useful for texture discrimination for spatial domain. Steps used were reading input, applying gabor filter with edge detection then YCBCR conversion process, feature segmentation feature selection extraction, correlation and performance matrix for performance analysis. It used two datasets namely GTSDB and GTSRB for both recognition and detection. The dataset included scenes from maps and complex data signs to verify algorithm's ability and efficiency [12]. Clustering algorithm with neural network to increase detection rate with maximizing pooling positions as per proposed algorithm of Mao [13] and Qian [14] can be supposed to give much high real-time efficiency.

2 Methodology And Implementation

2.1 Overview

The steps followed in this work, right from the dataset preparation to obtaining results are presented in Fig.1. The paper includes a tested approach along with a suggested approach for traffic sign detection and recognition.

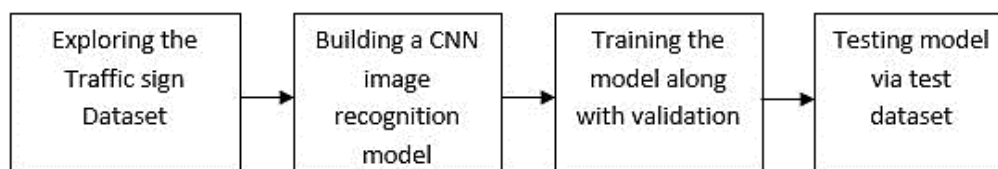


Figure 1: Steps followed for obtaining results

The approach followed for building this computer vision model involves - Importing the data-set and loading it, Performing EDA (Exploratory Data Analysis) on dataset and visualizing data-set variables and their characteristics, Pre-Processing data so that model is finely tuned, Designing the model using CNN, and Training and testing the designed model with training and testing samples.

2.2 Dataset and its Preprocessing

The CIFAR-10 dataset used was collected by German data scientists and is found to be containing more than 45,000 images which are collection of different traffic signs in various size and clarity ranges. A

reasonable amount of varying nature is noticed in the dataset as some traffic signs are provided with many images as compared to others having fewer images in the records. Images are classified in different classes in our folder from where data is extracted into our Python module with Tensorflow background environment. ETL (Extract, Transform, Load) tools are used here, by virtue of which, the data is converted into suitable format for transforming sample raw data into understandable format. The train dataset used consists of 43 folders within the range of 0-42 over which iteration is done using OS module. The function of conversion of image content into array is provided by the Python Imaging Library (PIL), in which all labels associated which images are appended in data lists. One-Hot Encoding and conversion to categorical labels of train and test dataset is done by the keras.utils packages. The shape of dataset is (39209,30,30,3) signifying the number of images, their height x width values and RGB (red, green and blue) notations respectively. Splitting is done via `train_test_split()` method provided by Sklearn package. If image data is normalized by making pixel values ranging in mid of -1 and +1, numerical instabilities are reduced. For better model performance, some transformations can also be performed on the generated augmented and finely refined data, by changing the brightness, rotation of image etc., by using OpenCV library. Also, rejecting color or converting RGB supported images to gray-scale can help in saving efforts on training.

2.3 Language, Platform and Libraries

Python language is very useful for building machine learning models, artificial intelligence supported applications and performing data analysis. This interpreted language evaluates the code fast, has easy-to-use syntax and supports thousands of libraries for analysis of real-world scenarios, and therefore this work is based on Python programming [6]. The widely used python libraries of Numpy and Pandas for data manipulation have also been used. Jupyter Notebook is the open source web-based application which supports Python language along with other languages. It allows for easy debugging and re-running of individual snippets and serves to be very useful in data training, exploratory data analysis, data visualization, data pre-processing and data augmentation along with several other benefits. TensorFlow is a machine learning platform having comprehensive libraries, flexible tools and is a core open source library for building ML supported applications [15]. It has been used in this work for training the NN model sequence to sequence. With its highly intuitive APIs (Application Programming Interfaces) for instance Keras, it makes debugging easier and execution can be done readily [16]. Keras is one of the most widely used framework for deep learning. It's a vast and central system within Tensorflow for machine learning workflow, training done on hyper parameters and deploying various problems with solutions. Some functions of Keras include classification of image and text, support during data loading and providing a range of utilities for converting raw data into a proper dataset, accurate feature normalization and, rescaling images and applying transformations on them [17].

2.4 Parameters employed

The designed ML model's architecture consists of the following:

- Input:** CIFAR-10 dataset's typical image is holder of its dimensions 32 x 32 x 3, where the depth refers to the number of RGB channels in the image.
- CONV:** The second layer handles the computation of dot product between the input image region sharing connection and a neuron's weight.
- RELU:** The third layer gives the desired dot product which is the resultant of applying a function for activation.
- POOL:** The fourth layer is termed as pool layer and is used to perform down-sampling on images, for example, if the image dimensions are 32 x 32 x 12 it will be reduced to 16 x 16 x 12. It can also be said that it reduces spatial dimensions i.e. the height and width of an image.
- Fully connected layer:** It is used to calculate class score which leads to draw the final result on volume (a x b x c), where c represents categories, for example, if c = 10 then this represents categories of CIFAR-10.

The CNN learns the filters' values during the model training process. Some parameters are necessitated to specify here like the number of filters, network architecture being used, size of filter and many other, prior to the training process [2]. Recognition of unseen patterns from image becomes more efficient when number of filters are increased in number. Also, the extraction level and quality of pattern evaluation gets improved to a significant level. Some of the important model parameters are:

Filters- The convolution operation on an image is used to identify dependencies of the image. A feature map is produced as a result when any filter slides over an input image [18]. Different feature maps are generated in turn when convolution operation is performed with another filter. The filters and image are stored for operation as numeric matrices.

Epochs- It is an entity used to signify how many times the entire dataset is passed forward and backward through the neural network. Although the dataset is too big in size to feed to the system even once, but it is required to make multiple submissions to get multiple tests generated optimal results.

Batch size- Different from number of batches, it is a number which represents the number of training examples in a batch. As the whole dataset can't be passed at once to the neural network model, batch numbers are nothing but the division of a dataset so that it is easy to process with accuracy.

Learning rate- There is an iterative optimization algorithm in machine learning called as gradient descent. This iterative algorithm works multiple times to get optimal results. A parameter here in is known as learning rate. This also helps in making an under-fit graph to fit optimally.

Loss regularization- It refers to the changes or transformations that are made to reduce the generalization error, and not the training error. These modifications are done on the learning algorithm. It is an important factor to prevent over-fitting and maintain precision along with accuracy. Some of the methods used to perform data regularization include using a dropout layer, introducing weight penalty, augmentation on the dataset and incorporating early stopping for effectively tuning hyper-parameters like epochs, number of batches etc.

Dropout rate- Some nodes in neural networks are randomly chosen by the dropout layer for their movement along their incoming and outgoing connections. This layer operates on hidden layer as well as the input layer. It can also be said as an ensembling technique, and the training neural network dropout can be seen as an ensemble collection of 2^n thinned networks. It is useful for different random subsets of neurons which help in learning more robust features. The different parameters of the CNN model being used are:

- 0 input layer
- 1 Type- conv2d layer, filters=32, kernelsize=5,5, activation function="relu"
- 2 Type-conv2d layer, filters=32, kernelsize=5,5, activation function="relu"
- 3 Type-MaxPool2D, pool_size=2,2
- 4 Type- DropOut layer , dropout rate=0.25
- 5 Type-Conv2D layer , filters=64 , kernel_size=3,3 activation function="relu"
- 6 Type-Conv2D layer , filters=64 , kernel_size=3,3 activation function="relu"
- 7 Type-MaxPool2D , pool_size=2,2
- 8 Type-DropOut layer , dropout rate=0.25
- 9 Type-Flattening Layer
- 10 Type-Densing layer size=256, activation function="relu"
- 11 Type-DropOut layer , dropout rate=0.5
- 12 Type-Densing layer size=43, activation function="SoftMax"

Another suggested approach for recognition and detection could be as shown in Fig. 2:

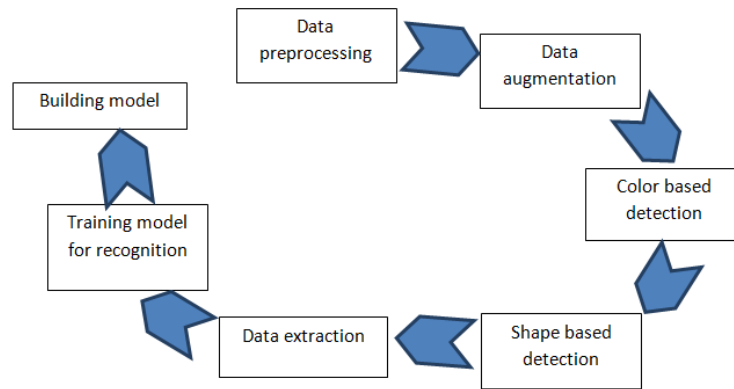


Figure 2: Another approach for recognition and detection task

2.5 Building the Model

Traffic signs are surrounded with a complete red circle across Indian states. Therefore, detection task's main accomplishment can be done by correctly identifying them from the input provided. Next step is pre-processing the data to remove any sort of error that affects accuracy, one of which is caused due to problem in sensor transmission and range of sensor being limited that results in contrast error in image. Detection can be done using brightness histograms with pixel's number count on y-axes and black to white color change values on x-axis. Values on y-axis can be either normalized or absolute. Concentration of values in plot is at specific regions only and improper utilization of brightness range denotes presence of low image contrast. Data is not balanced for different classes identified as images, as they aren't distributed uniformly over all classes. Some classes have more and some have lesser image than a particular useful number in every class such that the result of recognition and detection task is not biased around two or three classes. Resolving this issue requires incorporating resampling wherein classes with majority samples are under sampled and minority classes oversampled also decision to choose a better matrix to evaluate performance of algorithm implemented on model and adding synthetic samples also helps. After locating the erroneous points in data correcting them requires using data augmentation techniques, where batches from training images are taken and subjected to various image transformations like shearing, rotation, scaling, flipping, shadowing etc. further these transformed images are replaced with original batch and training CNN model using transformed dataset. This makes machine learning model more efficient in recognition of needed data without hindering the class labels in turn making classifier more generalized.

For these transformation related tasks related to implementing real-time computer vision OpenCV is used [19]. Rotations to augment data should not be up to degree of 10 as more rotation can damage data and performance and may lead to wrong recognition. Translating images slightly downward moves every image point with respect to a new coordinate system specified to look at all images in a proper uniform frame. There could be hindrance in identification due to less efficiency of capturing device captured raw images can include noisy points that can be AWGN or random.

Lack of proper light some part of image could be found having sharp edges and undesired color changes than other parts while detecting. To make things easy, reduce noise and have smoothen images blurring was done which is of two types namely GaussianBlur() and medianBlur() for removal of sharp images and noise respectively. Objects can be detected either on basis of color or shape, with color being a powerful aspect represented in RGB over Cartesian space as a measure of gathering information from image. RGB system for coloring lies on tri-chromatic theory. Image processing can be done using various available color spaces like YIQ, HSV, HSL etc. in which hue, saturation and value are important aspects. Color space representation can be changed to HSV where co-ordinate systems are cylindrical for giving information. Phi,z and rho are changed to hue, value and saturation. A particular color range can be defined using HSV colour space for defining color ranges which include red of particular importance. An image with

appropriate weight is generated by combining both upper and lower boundaries to a single one. Insights of red colour range are gained from the final image. Detection on basis of color generates data from object on basis of color-based detection. That's the reason for using detection on basis of shape as color based can give desired or undesired information.

Information regarding circular objects can be captured using `HoughCircle()` from OpenCV that is able to identify particular circular objects from gray scale images. Radius and co-ordinate of centre information is generated with count of circles detected. Red-circle related information are detected at this stage. Mask and Rect are used for extracting and cropping features followed by digit separation by contour methods further recognized by recognition process. Bilateral filtering is done next to reduce noise from dataset while keeping sharp edges of images. For reducing complexities that occur in case of a colored image, gray scaling is used. To fix the biased class issue, all different classes are made to have an arbitrary number of image records in them which is identified from dataset and could be generally double than highest image record. Observing the dataset gives the maximum number of records belonging to class label 2. Also, to increase image contrast local histogram equalization is employed.

To determine if information taken by neuron is useful or not in neural network, activation function is important. Output generated over a non-linear transformation from given input is fed as input to next layer. Importance of activation function lies to carry on back-propagation process. Rectified linear unit is one of important ReLU activation functions which is used generally. It increases network's efficiency and eases computing tasks by activating a limited number of neurons at a time, back-propagation of errors is also easy and multiple layers of neurons can be activated by ReLU. Another function is Softmax used for classification related problems and tells the confidence score of prediction after being applied on network's final layer. It requires exponentiation done to value taken by node and performing normalization on summing the values generated. Easy to interpret probability score are generated by vector yielded by Softmax function with respect to dataset's class labels. CNN derives its nature from processing in human brain thus have architectural efficiency in processing image. They have an advantage of capturing both dependencies special and temporal which cannot be done using feed-forward neural nets. CNN model architecture used comprises of 4 max-pooling and convolution layers in deep learning model for classification of unlabeled image data. Convolution layer takes an image of shape (30, 30, 1) with kernel size chosen to be (5, 5) for the layers because grayscale images are already pre-processed to these channels. Addition of max pooling layer after this step decreased the overfitting over dataset and training time needed. With single-dimension vector as input, two fully connected layers are added requiring flattening for input. The activation function used is Softmax reason being presence of multiple classes [20] for classification process. Number of epochs for this suggested approach could be around 700.

Iteration over different classes is done via OS module and for array implementation of image content, PIL library can be used. The dataset is loaded on the very first note and then is explored, visualized, pre-processed and augmented for designing an efficient model which could classify images with better precision. During data visualization, it is noticed that some items are represented more than others. It is also observed that images can be represented differently by changing their brightness levels, or they can also be different from one another despite being under same class labels with different clarity in the images and different part in an image with respect to the background window [21].

Building a CNN model is another major step for classification to be done for various input images. The model thus built is thereafter compiled with loss being "categorical_crossentropy" and "adam" being taken as an optimizer.

2.6 Testing and Validating the Model

The dataset is split into an 80-20 percentage for model training and testing purposes. Training and validating the built ML model is the third broad step wherein the model is tested on a validation set using the `model.fit()` function. The number of times the entire data set is passed forward and backward through the

model is 15 and the batch size is taken to be 32 and 64. The final error after training a model is influenced by random weights taken during the initial phase of model building on the training data set, and trying to minimize the error only by increasing the number of epochs can result in over-fitting, a problem that needs to be avoided during training of a model [22].

Testing the built model is the next major step. The downloaded data-set consists of a test.csv file and the images therein, with their respective class labels, and the path for reaching them, are extracted using the Pandas library of Python. For doing prediction on the test dataset, image dimensions of height and width are resized to 30 x 30 pixels and all of the generated image data is stored in a Numpy array of Python. Sklearn.metrics package helps in getting the accuracy of the built model with a metric known as accuracy_score which came out to be 0.942. The final broad step is to save the model by using model.save() function which comes under Keras library of Python.

3 Result and Discussion

The following important results are observed about the different data samples used in training, testing and validation for correctly identifying and classifying the traffic sign images:

- The training set shows an accuracy of nearly 95%
- The validation set produces result accurately with an accuracy percentage of 92.4%
- The testing set produces results with 90.3% accuracy

Precision was found to be 97.8% and recall to be 98.06% as per information derived from confusion matrix shown in Fig.4.

To evaluate the performance of the model, the chosen metric was accuracy, which is suitable for the traffic sign dataset. Analysis of performance of model when tested upon validation data is done through confusion matrix generally depicted in a tabular form and is useful for classification problem to know how many class labels are correctly classified and how many wrong [23]. It consists of two columns and two rows labeled as TP, TN, FP, FN for correct observation of class and prediction, both negative observation and prediction, negative observation but correct prediction, and incorrect prediction but correct observation. The model was found to saturate after 10 epochs whose number can be decreased to decrease the computation cost.

The accuracy can be increased with improved usage of hyperparameters like by tweaking the learning rate or increasing the number of epochs. Ensemble ANN model is another option to increase the observed accuracy. The graphs plotted for measure of loss and accuracy is shown in Fig. 3.

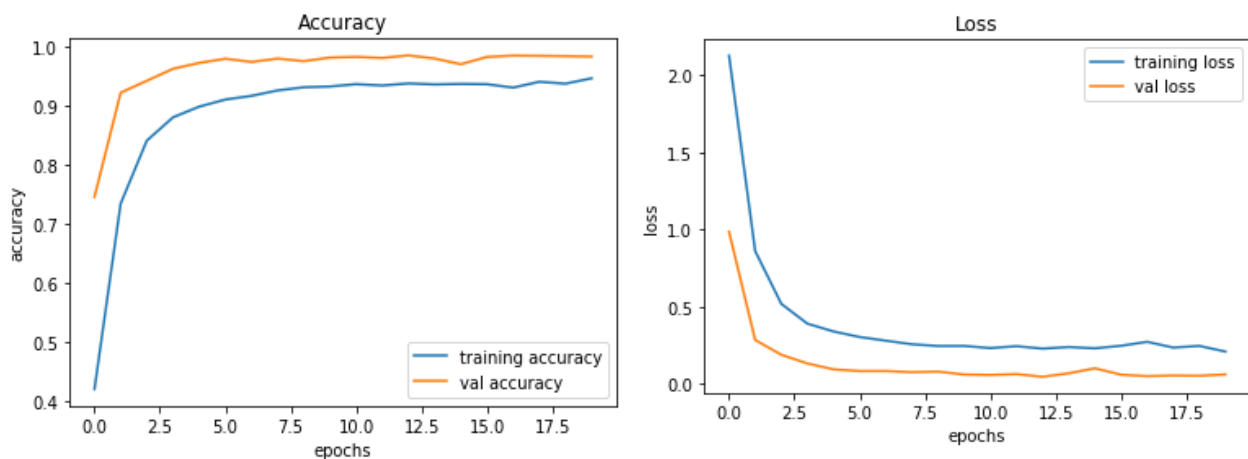


Figure 3: Graphs representing accuracy and loss

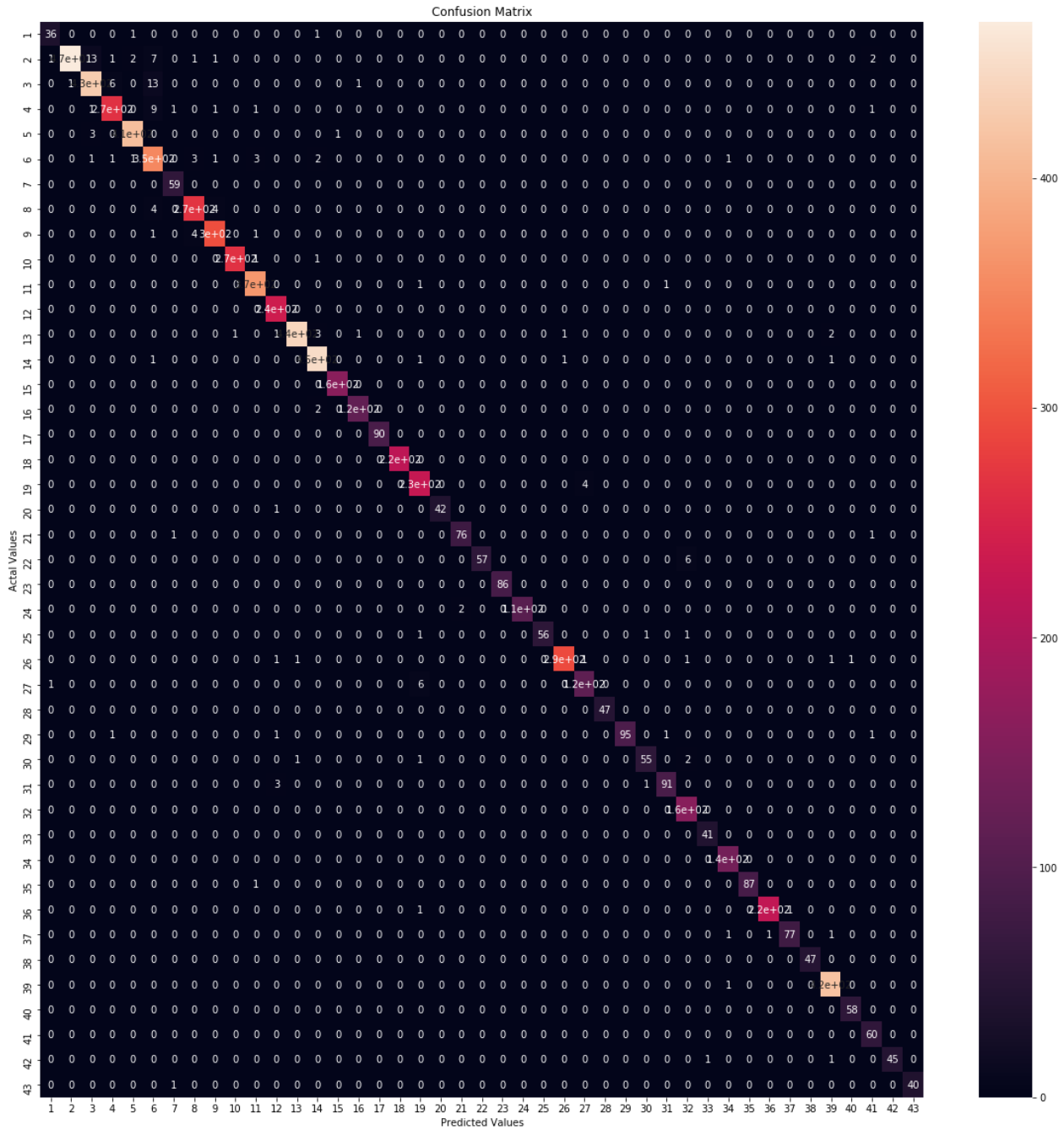


Figure 4: *Confusion matrix*

It is also observed that the model performance is significantly improved with the dataset batch size of 64 as compared to other batch sizes. In case of computer vision tasks such as the traffic sign recognition, the model should be made deeper for matching a human’s performance. Spatial Transformer Layer is a specialized layer which gives an accuracy of 99.61%. More parameter tuning and different architecture selection for modeling can also help improve the model.

As per the Changzhen approach [24] for Chinese traffic signs, point to be noted is that deeper models combined with other proposal network are very useful for real-time application. They proposed deep CNN networks implemented along with region proposal network of faster RCNN for Chinese dataset. The dataset consisted of images from various internet sources and real time captured images also, these were augmented then by application of brightness layers and motion blurring, and attained with three different

types of models from which the highest precision of classification was found for model named ZF. On test upon 33 videos, the detection rate of proposed model was very high i.e. 99%. Therefore, the model proposed in this paper can be enhanced by adopting approaches of Changzhen approach also for testing we can opt to use videos, further testing on such different types of data generating source can precisely tell insights about model working for different situations.

Signs are of different colors, shape and size for various places across globe. Variations can be of much affect to system in recognition and detection phase. As the suggested approach takes under consideration sign with higher similarity to signs present under GTSRB dataset. Thereby a system should be trained for detecting and extracting information on a manner generalized for every sort of sign as an important task for accomplishment. Recognizing them will require training model with a heavier and more generalized dataset. Mapping with global locations and signs so as to deal with challenges that can hinder real time image capture would also be needed as an important step. When combined with smart weather condition analyzing system the approach could be made more specific to deal with noise, rain and other problems for making a vehicle autonomous in true sense, which is main purpose of these sort of techniques [25]. Spatial Transformer Layer is a specialized layer which gives an accuracy of 99.61% and could be more useful than another generic version for capturing. It could yield good results when used along with properties derived from soft margin classifier and fuzzy shape recognizer. More parameter tuning and different architecture selection for modeling can also help improve the model.

4 Conclusions

Through this work, a model for traffic sign recognition system is successfully implemented using convolutional neural networks, needed for vehicles as a measure for ensuring road safety. The whole task is implemented on Jupyter Notebook using Python programming for machine learning and its strong libraries of deep learning. The parameters of the designed neural network model are finely tuned in order to get good accuracy results with precision of 97.8% and recall to be 98.06%. Accuracy on training set was 95% and 90.3% on testing set. Important point to be noted is that taking a large number of CNNs can increase learning rate of model and images for training are generally augmented but real time image capturing can't be augmented much fast so quick reliability is needed. Further challenges include integration with systems so as to capture images in dark and long distant and optimization with spatial methods. Errors in computer vision while capturing image during fog or smog should also be taken under consideration. A major challenge can be to integrate this recognition feature with mapping system. The study can be applied for making transport systems' vehicles and other related stuffs act intelligent as well as autonomous to the environment it is subjected to. Its implementation in real world requires integration with other fields of technology that subject to both hardware and software incorporated to give desired actions to be followed up with a less intervention from user. The capturing device should also be able for backup method in case of any sort of failure. CenterNet, like other CNN models, can be used to make model for adaptable. Extraction of image's feature can be improved along with understanding sign recognition and related features like driving behavior for a responsive system for high and low level of traffic.

5 Declarations

5.1 Study Limitations

The model's performance can be further improved by performing random cropping for training and validation data. More detection algorithms can be tried out-unsupervised learning can play a significant role in extracting certain features at different stages of the model building.

5.2 Competing Interests

The authors declare no competing interests.

How to Cite this Article:

K. Singh and N. Malik, "CNN Based Approach for Traffic Sign Recognition System", *Adv. J. Grad. Res.*, vol. 11, no. 1, pp. 23–33, Sep. 2021. <https://doi.org/10.21467/ajgr.11.1.23-33>

References

- [1] A. W. Harley, "An Interactive Node-Link Visualization of Convolutional Neural Networks," *Advances in Visual Computing Lecture Notes in Computer Science*, pp. 867–877, 2015.
- [2] S. Saini, S. Nikhil, K. R. Konda, H. S. Bharadwaj, and N. Ganeshan, "An efficient vision-based traffic light detection and state recognition for autonomous vehicles," *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.
- [3] S. Raschka, & V. Mirjalili, "Python Machine Learning: Machine Learning and Deep Learning with Python. Scikit-Learn, and TensorFlow". Second edition ed. , 2017
- [4] D. Michie, D. J. Spiegelhalter, & C. C. Taylor, "Machine learning, neural and statistical classification", (1994).
- [5] U. Karn, "An intuitive explanation of convolutional neural networks". *The data science blog.* , 2016.
- [6] A. Ellahyani, I. E. Jaafari, and S. Charfi, "Traffic Sign Detection for Intelligent Transportation Systems: A Survey," *E3S Web of Conferences*, vol. 229, p. 01006, 2021.
- [7] E. Peng, F. Chen, & X. Song, "Traffic sign detection with convolutional neural networks" . *International conference on cognitive systems and signal processing* , pp. 214-224 . Springer , Singapore , 2016, November.
- [8] P. Garg, D. R. Chowdhury, and V. N. More, "Traffic Sign Recognition and Classification Using YOLOv2, Faster RCNN and SSD," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019.
- [9] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.
- [10] U. Kamal , S. Das , A. Abrar , & M. K. Hasan , "Traffic-sign detection and classification under challenging conditions: a deep neural network based approach", *IEEE video and image processing cup* , 2017.
- [11] D. Karthikeyan , Enitha C, Bharathi S, Durkadevi K. "Traffic Sign Detection and Recognition using Image Processing". *International Journal Of Engineering Research & Technology (IJERT)*, 8(8), NCICCT–2020 , 2020.
- [12] Z. He, Z. Xiao, and Z. Yan, "Traffic Sign Recognition Based on Convolutional Neural Network Model," *Chinese Automation Congress (CAC)* , 2020.
- [13] X. Mao, S. Hijazi , R. Casas, P. Kaul, R. Kumar, & C. Rowen , Hierarchical CNN for traffic sign recognition. In *2016 IEEE Intelligent Vehicles Symposium (IV)* (pp. 130-135). IEEE, 2016, June.
- [14] R. Qian ,Y. Yue, F. Coenen, & B. Zhang . Traffic sign recognition with convolutional neural network based on max pooling positions. "12th International conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD)" (pp. 578-582). IEEE , 2016, August.
- [15] A. Géron , "Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems", O'Reilly Media , 2019.
- [16] M. Abadi , P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, & X. Zheng, "Tensorflow: A system for large-scale machine learning". *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283) , 2016.
- [17] N. Ketkar, "Introduction to Deep Learning," *Deep Learning with Python*, pp. 1–5, 2017.
- [18] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning," *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2018.
- [19] H. H. Aghdam, E. J. Heravi, and D. Puig, "A Practical and Highly Optimized Convolutional Neural Network for Classifying Traffic Signs in Real-Time," *International Journal of Computer Vision*, vol. 122, no. 2, pp. 246–269, 2016.
- [20] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," *The 2011 International Joint Conference on Neural Networks*, 2011.
- [21] D. J. MacKay, & D. J. Mac Kay, "Information theory, inference and learning algorithms". Cambridge university press, 2003.
- [22] M. Singh, M. K. Pandey , & L. Malik , "Traffic sign detection and recognition for autonomous vehicles". *Int J Adv Res Ideas Innov Technol (IJARIIT)*, vol. 4. No. 2. , 1666-1670, 2018.
- [23] A. Kumar, T. Singh, and D. K. Vishwakarma, "Intelligent Transport System: Classification of Traffic Signs Using Deep Neural Networks in Real Time," *Lecture Notes in Mechanical Engineering Advances in Manufacturing and Industrial Engineering*, pp. 207–219, 2021.
- [24] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, "A traffic sign detection algorithm based on deep convolutional neural network," *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, 2016.
- [25] C. Dewi, R.-C. Chen, and S.-K. Tai, "Evaluation of Robust Spatial Pyramid Pooling Based on Convolutional Neural Network for Traffic Sign Recognition System," *Electronics*, vol. 9, no. 6, p. 889, 2020.

Publish your books with AIJR publisher-

- ✓ Publish with ISBN and DOI.
- ✓ Publish Thesis/Dissertation as Monograph.
- ✓ Publish Book Monograph.
- ✓ Publish Edited Volume/ Book.
- ✓ Publish Conference Proceedings
- ✓ Retain full copyright of your books.

Submit your manuscript at books.aijr.org

Publish your research article in AIJR journals-

- ✓ Online Submission and Tracking
- ✓ Peer-Reviewed
- ✓ Rapid decision
- ✓ Immediate Publication after acceptance
- ✓ Articles freely available online
- ✓ Retain full copyright of your article.

Submit your article at journals.aijr.org