Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Mini-Project Report

on

“Traffic Sign Recognition System”

COMP 484 - Machine Learning

(For partial fulfillment of 4th Year/ 1st Semester in Computer Engineering)

Submitted by:

Junth Basnet (05)

Sandip Dulal (63)

Abin Sainju (65)

Submitted to:

Dr. Bal Krishna Bal

Associate Professor

Department of Computer Science and Engineering

Submission Date:

March 11, 2020

Abstract

Traffic sign detection and recognition are crucial in the development of intelligent vehicles. An improved traffic sign detection and recognition algorithm for intelligent vehicles is proposed to address problems such as how easily affected traditional traffic sign detection is by the environment, and poor real-time performance of deep learning-based methodologies for traffic sign recognition. In this work, we propose a novel deep network for traffic sign classification that achieves outstanding performance on GTSRB surpassing the best human performance of 98.84%.We apply Convolutional Networks (ConvNets) to the task of traffic sign classification. . ConvNets are biologically-inspired multi-stage architectures that automatically learn hierarchies of invariant features.We have achieved the state-of-the-art performance of 99.22% on GTSRB dataset. Compared with other algorithms, the proposed algorithm has remarkable accuracy and real-time performance, strong generalization ability and high training efficiency.

Keywords: Advanced Driver Assistant System, Image Classification, Traffic Sign, Deep Networks, Convolutional Neural Network(ConvNets)

Table of Contents

List of Figures

List of Tables

Acronyms/ Abbreviations

ADAS: Advanced Driver Assistance Systems

GPU: Graphics Processing Unit

GTSRB: German Traffic Sign Recognition Benchmark

IJCNN: International Joint Conference on Neural Networks

CLAHE: Contrast Limited Adaptive Histogram Equalization

CNN: Convolutional Neural Network

1. Introduction

1.1. Problem Definition

Traffic signs classification is one of the foremost important integral parts of autonomous vehicles and advanced driver assistance systems (ADAS) [8], [5], [9], [4], [10]. Most of the time drivers missed traffic signs due to different obstacles and lack of attentiveness. Automating the process of classification of the traffic signs would help reduce accidents. Traditional computer vision and machine learning-based methods were widely used for traffic signs classification [17], [3], but those methods were soon replaced by deep learning-based classifiers. Recently deep convolutional networks have surpassed traditional learning methods in traffic signs classification. With the rapid advances of deep learning algorithm structures and the feasibility of its high-performance implementation with graphical processing units (GPU), it is advantageous to relook the traffic signs classification problems from the efficient deep learning perspective. Classification of traffic signs is not such a simple task, images are affected to adverse variation due to illumination, orientation, the speed variation of vehicles, etc. Normally wide-angle camera is mounted on the top of a vehicle to capture traffic signs and other related visual features for ADAS. These images are distorted due to several external factors including vehicle speed, sunlight, rain, etc. Sample images from GTSRB dataset are shown in Fig. 1.

1.2. Motivations for Doing the Project

Traffic sign recognition has direct real-world applications such as driver assistance and safety, urban scene understanding, automated driving, or even sign monitoring for maintenance. With the rapid development of economy and technology in modern society, automobiles have become an indispensable means of transportation in the daily travel of people. Although the popularity of automobiles has introduced considerable convenience to people, it has also caused numerous traffic safety problems that cannot be ignored, such as traffic congestion and frequent road accidents. Traffic safety issues are largely caused by subjective reasons related to the driver, such as inattention, improper driving operation and non-compliance with traffic rules, and smart cars have become an effective means to eliminate these human factors [5], [8]. Self-driving technology can assist, or even independently complete the driving operation, which is of remarkable importance to liberate the human body and considerably reduce the incidence of accidents [6]. Traffic sign detection and recognition are crucial in the development of intelligent vehicles, which directly affects the implementation of driving behaviors. With the rapid advances of deep learning algorithm structures and the feasibility of its high-performance implementation with graphical processing units (GPU), it is advantageous to relook the traffic signs classification problems from the efficient deep learning perspective. Normally a wide-angle camera is mounted on the top of a vehicle to capture traffic signs and other related visual features for ADAS.

1.3. Objectives of the Project work

1. To build a convolutional neural network that classifies traffic signs with admirable accuracy, better real-time performance, stronger generalization ability and higher training efficiency

2. To learn and implement machine learning library Keras to experiment with different convolutional neural network architecture
3. To achieve the state-of-the-art performance on GTSRB dataset, that surpasses the best human performance of 98.84%.

2. Related Works

Traffic sign classification becomes a mature area with an increasing focus on autonomous driving research. Notable research work exists on detection and classification traffic signs for advanced driver assistance systems. Most of the works attempted to address the challenge involved in real-life problems due to scaling, rotation, blurring, etc. We will go through the overview of some relevant works since it is not possible to discuss all those research works.

Most of the works based on computer vision and machine learning algorithms that use data from several camera sensors mounted on the car roof at different angles. A number of existing approaches to road-sign recognition have used computationally-expensive sliding window approaches that solve the detection and classification problems simultaneously. But many recent systems in the literature separate these two steps. Detection is first handled with computationally-inexpensive, hand-crafted algorithms, such as color thresholding. Classification is subsequently performed on detected candidates with more expensive, but more accurate, algorithms. The classification has been approached with a number of popular classification methods such as Neural Networks [20], Support Vector Machines [18], etc.
In [19] global sign shapes are first detected with various heuristics and color thresholding, then the detected windows are classified using a different Multi-Layer neural net for each type of outer shape. These neural nets take 30x30 inputs and have at most 30, 15 and 10 hidden units for each of their 3 layers. While using a similar input size, the networks used in the present work have orders of magnitude more parameters.
In some of the work, researchers explore detection based on color features, such as converting the color space from RGB to HSV and then using color thresholding method for detection and classification by using a support vector machine. In color thresholding approach morphological operation like connected component analysis was done for accurate location. Bahlmann et al[16] have used color, shape, motion information and haar wavelet-based features for detection, classification of the traffic sign. By using SVM based color classification on a block of pixels Le et al [18] addressed the problems of weather variation.

German Traffic Sign Recognition Benchmark (GTSRB) is one of the reliable datasets for testing and validating traffic sign classification and detection algorithms.In the competition of GTSRB, the top-performing algorithm exceeds best human classification accuracy. By using a committee of neural networks Ciresan et al. [15] achieved the highest ever performance of 99.46%, which surpassed the best human performance of 98.84%. Their proposed committee composed of 25 networks each having 3 convolutional and 2 fully connected networks with traditional data augmentations and jittering. The main disadvantages of this committee are multiple networks, a huge number of parameters (around 90Millions) and dataset dependent handcrafted augmentations. Sermanet et al. proposed a multi-scale convolutional network [14] with 2 different features stages, which has achieved 98.31% accuracy in this dataset.

3. Datasets

This paper uses the German Traffic Sign Recognition Benchmark (GTSRB), which was presented at the 2011 International Joint Conference on Neural Networks (IJCNN) [1]. The internal traffic signs are collected from the real road traffic environment in Germany, and it has become a common traffic sign dataset used by experts and scholars in computer vision, self-driving, and other fields. The GTSRB comprises 51,839 images in 43 classes, which are divided into training and testing sets. A total of 39,209 and 12,630 images are provided in the training and testing sets, accounting for approximately 75% and 25% of the whole, respectively. Each image contains only one traffic sign, which is not necessarily located in the center of the image. The image size is unequal; the maximum and smallest images are 250 x 250 and 15 x 15 pixels, respectively.

The dataset provided by the GTSRB competition [1] presents a number of difficult challenges due to real-world variabilities such as viewpoint variations, lighting conditions (saturation, low-contrast), motion-blur, occlusions, sun glare, physical damage, colors fading, graffiti, stickers and an input resolution as low as 15 x 15. Although signs are available as video sequences in the training set, temporal information is not in the test set. The present project aims to build a robust recognizer without temporal evidence accumulation. Sample images from the GTSRB dataset are shown in Fig. 1 and the distribution of images per sample is not uniform as shown in Fig. 2.
Dataset for this project is available at:
https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

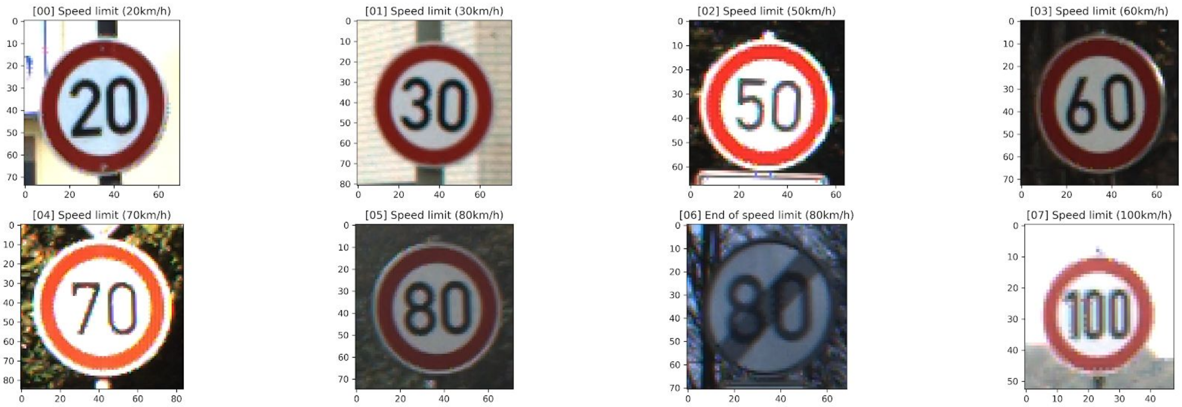


Fig.1. Sample images from GTSRB dataset

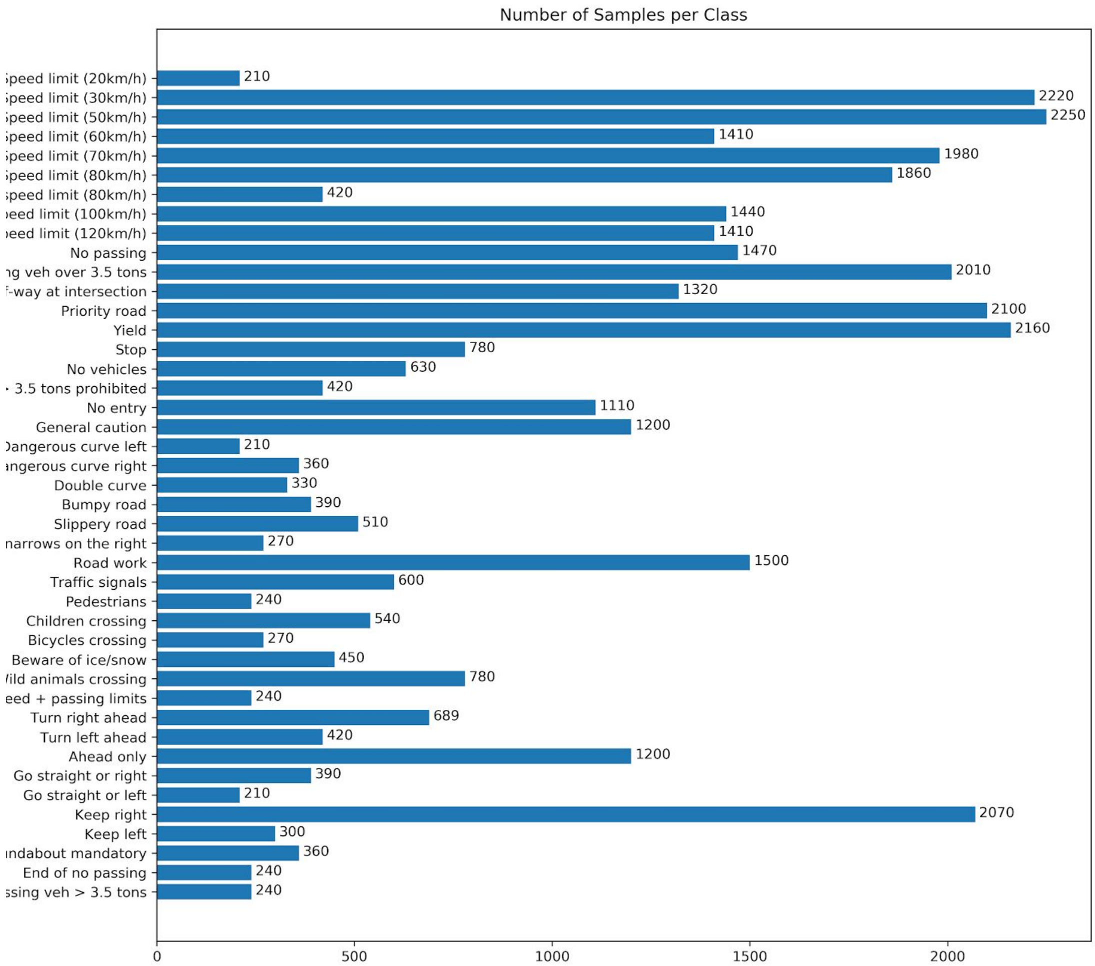


Fig. 2. Label distribution(number of images per class)

4. Methods and Algorithms Used

4.1. Image Preprocessing

All images are down-sampled or up sampled to 32x32 (dataset samples sizes vary from 15x15 to 250x250).The ROI in the traffic sign training image is not 100% in the center of the image, and some edge background information is included around the traffic sign. With the change of illumination conditions, these useless interference areas will increase the influence on traffic sign recognition, thereby undoubtedly raising the computational complexity of the training network and the misrecognition rate of traffic signs. Therefore, image preprocessing is necessary. Image preprocessing mainly includes the following three stages:

Contrast Limited Adaptive Histogram Equalization (CLAHE)

We used Scikit histogram equalization function, which not only normalizes the images but also enhances local contrast. CLAHE is an algorithm for local contrast enhancement, that uses histograms computed over different tile regions of the image. This approach enhances an image with low contrast, using a method called histogram equalization, which “spreads out the most frequent intensity values” in an image.Sample of histogram equalized images is shown in Fig. 3. The equalized image has a roughly linear cumulative distribution function as shown in Fig. 4.

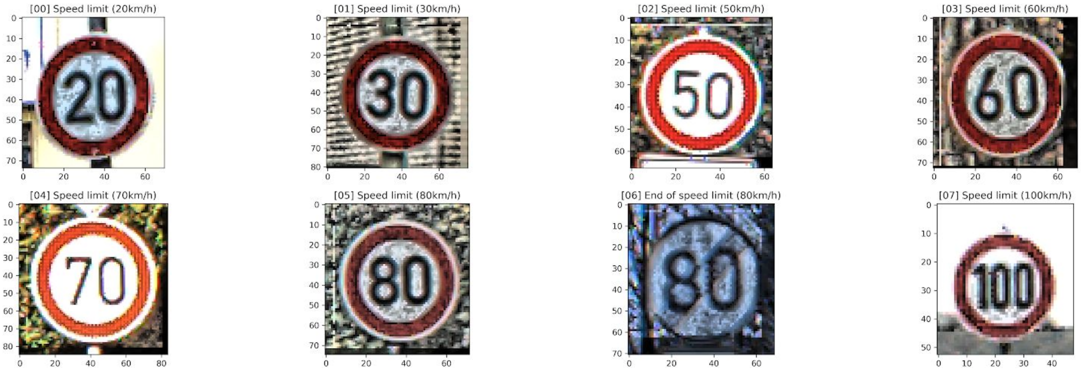


Fig.3. Sample of histogram equalized images

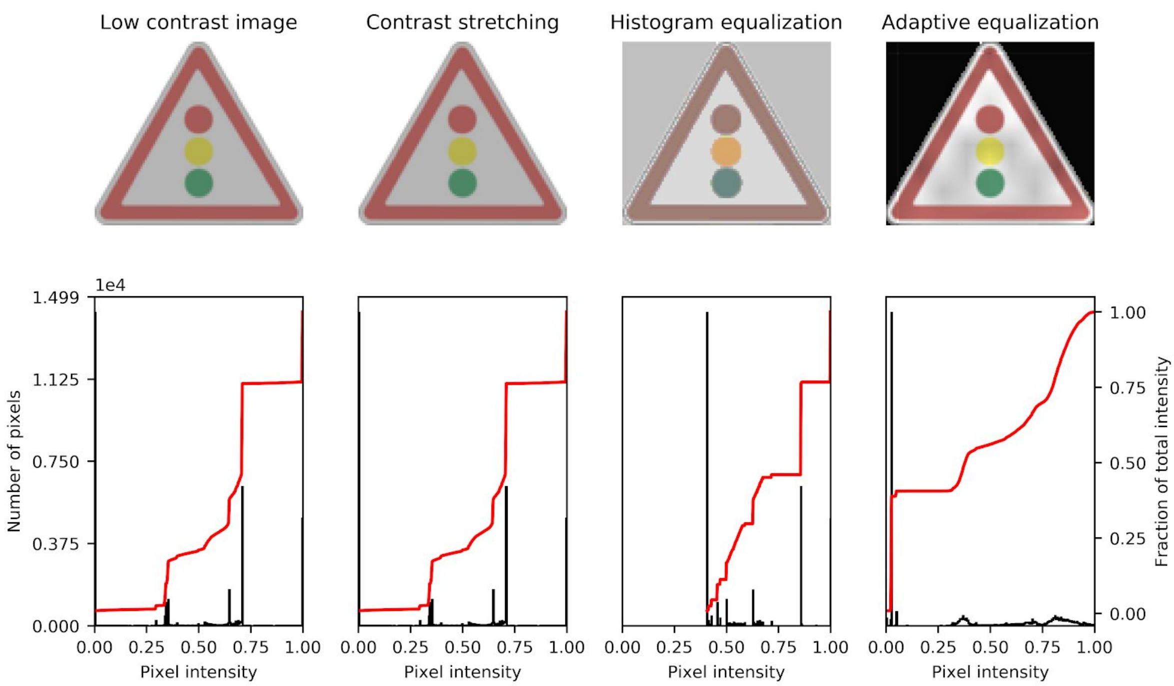


Fig. 4. Equalized image with roughly linear CDF

Image Augmentation

Traffic signs classification are affected due to contrast variation, rotational and translational changes. It is possible to nullify the effect of spatial transformations in an image undergo due to varying speed of vehicles camera by using multiple transformations to the input image.we build augmented dataset version 1 by adding 3000 images per class in original training set yielding 129,000 images in total and augmented dataset version 2 by adding 4500 images per class in original training set yielding 193,500 images in total.. ConvNets architectures have built-in invariance to small translations, scaling and rotations. When a dataset does not naturally contain those deformations, adding them synthetically will yield more robust learning to potential deformations in the test set.Other realistic perturbations would probably also increase robustness such as other affine transformations, brightness, contrast and blur.ImageDataGenerator class generates batches of tensor image data with real-time data augmentation.Sample of augmented images is shown in Fig. 5.

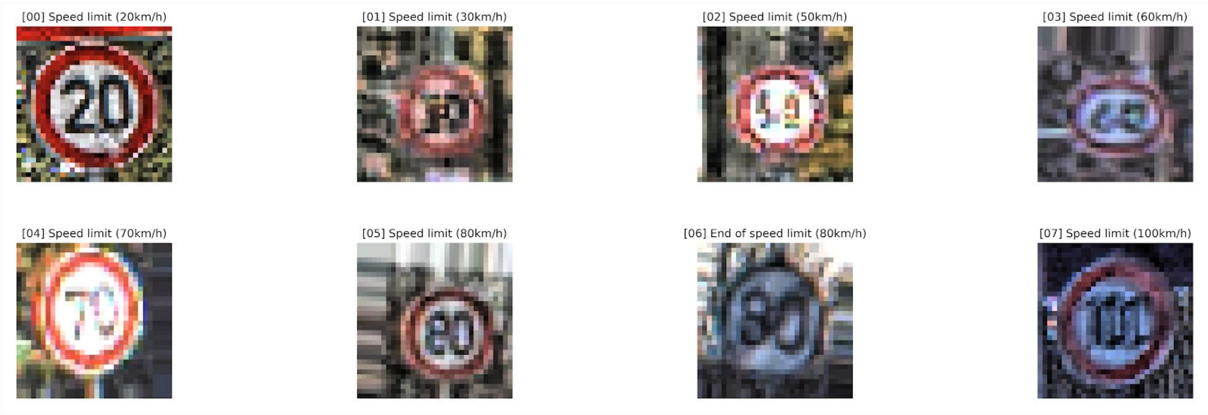


Fig. 5. Sample of augmented images (version 1)

Grayscaling

Converting an image with RGB channels into an image with a single grayscale channel. The value of each grayscale pixel is calculated as the weighted sum of the corresponding red, green and blue pixels as:

$$Y = 0.2125\ R + 0.7154\ G + 0.0721\ B$$

The grayscaled training images sample is shown in Fig. 6.

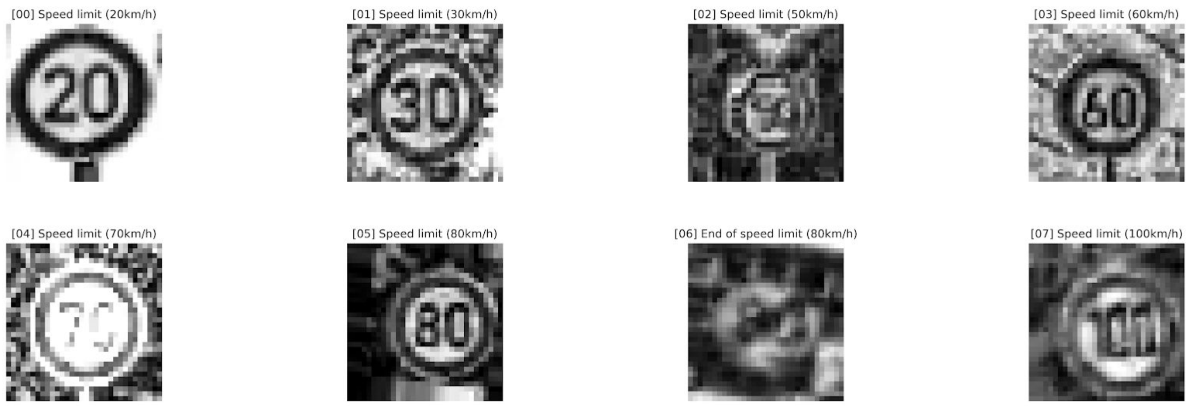


Fig. 6. Sample of grayscaled images

4.2. Model Architecture

Our model follows the guideline of classical LeNet-5 Convolutional Neural Network [2] with modification.Our First model is LeNet-5. After explorations, we build our second model LeNet-5 + Contrast Enhancement.The we used augmented dataset to the second model instead of original training images, LeNet-5 + Contrast Enhancement + Augmentation(3000) to reduce overfitting.The fourth model is Deep LeNet-5 + Contrast Enhancement + Augmentation(3000) and the fifth model is Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization.The details of the five models are discussed in this section.

4.2.1. LeNet-5

Professor Yann Lecun proposed the LeNet-5 network model in 1998, which was mainly used for digital recognition. The LeNet-5 network model consists of seven layers, including two convolutional layers, two pooling layers, two fully-connected layers and one output layer. The input image size is 32 × 32, and the output is a 10-dimensional classification vector, which can identify numbers from 0 to 9 [2].The classic LeNet-5 network model has good classification and recognition effects for a single target. However, in the traffic signs recognition training, it is difficult to ensure a high enough accurate recognition rate, the training network cannot converge, and the recognition efficiency of the network decreases dramatically. The learning rate and the iterations number of the training network are not adjusted accordingly,and the relevant parts are rationally optimized, thereby resulting in the emergence of the over-fitting phenomenon during training.

TABLE I
DETAILED DESCRIPTION OF LeNet-5 ARCHITECTURE

Name	Type	# Parameters	Output Shape
conv2d_1_input	InputLayer	0	None, 32, 32, 3
conv2d_1	Conv2D	456	None, 28, 28, 6
max_pooling2d_1	MaxPooling2D	0	None, 14, 14, 6
conv2d_2	Conv2D	2416	None, 10, 10, 16
max_pooling2d_2	MaxPooling2D	0	None, 5, 5, 16
flatten_1	Flatten	0	None, 400
dense_1	Dense	48120	None, 120
dense_2	Dense	10164	None, 84
dense_3	Dense	3655	None, 43

Activate Wi
Go to Settings t

4.2.2. LeNet-5 + Contrast Enhancement

We experimented with the same classical LeNet-5 Convolutional Neural Network as proposed in Gradient-Based Learning Applied to Document Recognition paper with the histogram equalized images rather than original training images.

TABLE II
DETAILED DESCRIPTION OF DEEP-LeNet-5 ARCHITECTURE

Name	Type	# Parameters	Output Shape
conv2d_1_input	InputLayer	0	None, 32, 32, 3
conv2d_1	Conv2D	2432	None, 28, 28, 32
conv2d_2	Conv2D	25632	None, 24, 24, 32
max_pooling2d_1	MaxPooling2D	0	None, 12, 12, 32
conv2d_3	Conv2D	51264	None, 8, 8, 64
conv2d_4	Conv2D	102464	None, 4, 4, 64
max_pooling2d_2	MaxPooling2D	0	None, 2, 2, 64
flatten_1	Flatten	0	None, 256
dropout_1	Dropout	0	None, 256
dense_1	Dense	263168	None, 1024
dropout_2	Dropout	0	None, 1024
dense_2	Dense	524800	None, 512
dropout_3	Dropout	0	None, 512
dense_3	Dense	22059	None, 43

4.2.3. LeNet-5 + Contrast Enhancement + Augmentation(3000)

LeNet-5 + Contrast Enhancement + Augmentation(3000) uses classical LeNet-5 Convolutional Neural Network with histogram equalized images and augmented dataset version 1 to reduce overfitting and increase generalization accuracy.

4.2.4. Deep LeNet-5 + Contrast Enhancement + Augmentation(3000)

Deep LeNet-5 + Contrast Enhancement + Augmentation(3000) has 2 more convolutional layers than classical LeNet-5 network architecture and dropout layers are added to reduce overfitting.This model is used with histogram equalized and augmented dataset version 1 images.

4.2.5. Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization

Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization also has 2 more convolutional layers than classical LeNet-5 network architecture.Dropout and L2 Regularization is also used in this model to reduce overfitting of model and increase generalization accuracy.Regularizers allow to apply penalties on layer parameters or layer activity during optimization. These penalties are incorporated in the loss function that the network optimizes.This model is applied to histogram equalized and augmented dataset version 2 images.

The **ReLU** function is selected as the activation function. Compared with the traditional Sigmoid and Tanh functions, the ReLU function is simple in calculation but effectively solves the gradient disappearance and explosion problem of the two functions. By making a part of the neuron output to 0, the network can be sparse, which helps reduce computational complexity and accelerate network convergence. Therefore, this function performs well in deep network training [11].

Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [7]. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters.The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. The hyper-parameters have intuitive interpretations and typically require little tuning.The Adam method can effectively solve the problems of learning rate disappearance, slow convergence and large fluctuation of loss function in the optimization process, thereby possessing a good convergence mechanism.

Deep learning solutions to classification problems usually employ the **softmax** function as their classification function (*last layer*) [11]. The softmax function specifies a discrete probability distribution for *K* classes, denoted by

$$\sum_{k=1}^{K} \mathbb{P}(k)$$

The **dropout** is added to the fully-connected layers.The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different “thinned” networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods [12].

5. Experiments

5.1. Experimental Environment

Software environment: Windows 10 64-bit operating system, PyCharm 2019.3 (Professional Edition), TensorFlow 2.1.0, Python 3.8.0 64-bit, Keras 2.3.1.

Hardware environment: Intel (R) Core (TM) i5-6500 CPU@3.20GHz processor, 8.00 GB memory, 2 TB mechanical hard disk.

5.2. Hyperparameter Tuning

The proposed network was trained and tested using the machine learning framework Keras.We extensively evaluate our proposed deep networks on GTSRB (German Traffic Sign Recognition Benchmark) [1] using our modified network architecture and also with original classical LeNet-5 Convolutional Neural Network.

5.2.1. LeNet-5

The number of trainable parameters for LeNet-5 model is 64,811, the batch size is 64,The learning rate is 1e-4 with Adam optimizer.Also, it has been observed that learning rate primarily influences training process. We tried several learning rates 1e-2, 1e-3, 1e-6 and 1e-4 works best.ReLU activation function is used in the intermediate layers and softmax activation function is employed as classification function (last layer).All images are down-sampled or up sampled to 32x32.Detailed description of network parameters is shown in TABLE III.

5.2.2. LeNet-5 + Contrast Enhancement

The network architecture and the hyperparameters for LeNet-5 + Contrast Enhancement model is same as that of first model. Only difference is that the model accepts histogram equalized images as input and generalization accuracy seems to increase for this model.Detailed description of network parameters is shown in TABLE III.

5.2.3. LeNet-5 + Contrast Enhancement + Augmentation(3000)

The network architecture and the hyperparameters for LeNet-5 + Contrast Enhancement + Augmentation(3000) is same as that of LeNet-5 model but the network accepts histogram equalized and augmented dataset version 1 images which helps reduce overfitting and increase generalization accuracy.Detailed description of network parameters is shown in TABLE III.

5.2.4. Deep LeNet-5 + Contrast Enhancement + Augmentation(3000)

For Deep LeNet-5 + Contrast Enhancement + Augmentation(3000) model, the number of trainable parameters is 991,819.Dropout (50%) was used for the fully connected layer and the number of neurons in the first fully connected layer increases from 120 to 1024 neurons and in the second fully connected layer, the number of neuron increases from 84 to 512 neurons.Other hyperparameters are same.Detailed description of network parameters is shown in TABLE III.

TABLE III
DETAILED DESCRIPTION OF DEEP NETWORK PARAMETERS

Name (5 visualized)	batch_size	Runtime	L2_Regul...	dense_la...	dense_la...	total_params	dropout	learning_...
(1) LeNet-5	64	7m 38s	-	120	84	64811	-	-
(2) LeNet-5 + CLAHE	64	7m 5s	-	120	84	64811	-	-
(3) LeNet-5 + CLAHE + AUG(v1)	64	19m 51s	-	120	84	64811	-	-
(4)Deep LeNet-5 + AUG(v1)	64	1h 14m 7s	-	1024	512	991819	0.5	-
(5)Deep LeNet-5 + AUG(v2)	64	1h 44m 23s	0.001	1024	512	991819	0.5	0.0001

5.2.5. Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization

Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization is the final proposed model for this project.We have achieved the state-of-the-art performance of 99.22% on GTSRB dataset, which surpassed the best human performance of 98.84%.The number of trainable parameters is 991,819.Dropout (50%) was used for the fully connected layer.The learning rate is 1e-4 with Adam optimizer and L2 Regularization of 1e-3 is also used to reduce the overfitting.Detailed description of network parameters is shown in TABLE III.

6. Evaluation of Results (Evaluation Metrics)

Root-Mean-Squared-Error (RMSE) is used as performance evaluation criteria of different learning algorithms. Denote the desired vector of the true labels as $y \epsilon \mathbb{R}^n$, where *n* is the number of labels. $\hat{y} \epsilon \mathbb{R}^n$ is the vector of predicted labels, then RMSE is defined as:

$$RMSE(\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

We have achieved the state-of-the-art performance of 99.22% on GTSRB dataset, which surpassed the best human performance of 98.84%.Overall accuracy comparisons with different high performing approaches are shown in TABLE IV.

TABLE IV
COMPARISON OF ACCURACY OF 5 PROPOSED MODELS

Name (5 visualized)	val_accuracy	val_loss
(1) LeNet-5	0.978	0.1185
(2) LeNet-5 + CLAHE	0.9752	0.1246
(3) LeNet-5 + CLAHE + AUG(v1)	0.9596	0.2016
(4)Deep LeNet-5 + AUG(v1)	0.9771	0.08474
(5)Deep LeNet-5 + AUG(v2)	0.9922	0.09161

Comparison of 5 models according to validation accuracy is shown in Fig. 7.

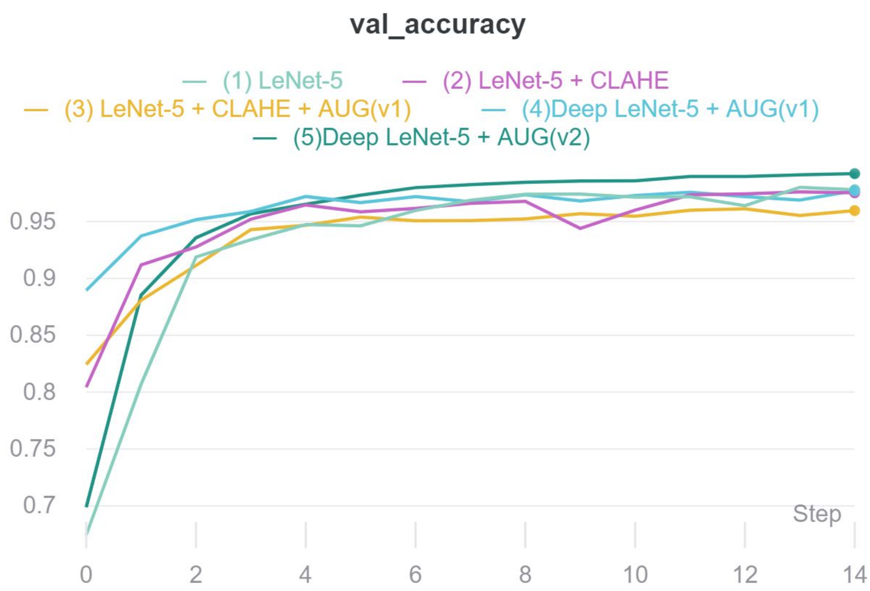


Fig. 7. Accuracy comparison of 5 proposed models

The precision, recall and F1-Score and support of Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization model is shown in TABLE V.

TABLE V
PRECISION, RECALL AND F1-SUPPORT OF 5th MODEL

	precision	recall	f1-score	support
Speed limit (20km/h)	0.92	1.00	0.96	60
Speed limit (30km/h)	0.95	0.99	0.97	720
Speed limit (50km/h)	0.98	0.98	0.98	750
Speed limit (60km/h)	0.96	0.92	0.94	450
Speed limit (70km/h)	1.00	0.97	0.98	660
Speed limit (80km/h)	0.94	0.94	0.94	630
End of speed limit (80km/h)	0.98	0.96	0.97	150
Speed limit (100km/h)	0.95	0.99	0.97	450
Speed limit (120km/h)	0.99	0.96	0.97	450
No passing	1.00	0.99	0.99	480
No passing veh over 3.5 tons	1.00	1.00	1.00	660
Right-of-way at intersection	0.99	0.93	0.96	420
Priority road	1.00	0.98	0.99	690
Yield	1.00	1.00	1.00	720
Stop	1.00	1.00	1.00	270
No vehicles	0.95	0.98	0.96	210
Veh > 3.5 tons prohibited	0.98	1.00	0.99	150
No entry	1.00	1.00	1.00	360
General caution	0.99	0.85	0.91	390
Dangerous curve left	0.98	0.95	0.97	60
Dangerous curve right	0.97	1.00	0.98	90
Double curve	0.84	0.99	0.91	90
Bumpy road	0.96	0.89	0.93	120
Slippery road	0.82	0.99	0.89	150
Road narrows on the right	0.92	0.99	0.95	90
Road work	0.97	0.98	0.98	480
Traffic signals	0.90	0.98	0.94	180
Pedestrians	0.69	0.87	0.77	60
Children crossing	0.98	0.99	0.99	150
Bicycles crossing	0.91	1.00	0.95	90
Beware of ice/snow	0.86	0.77	0.81	150
Wild animals crossing	0.98	1.00	0.99	270
End speed + passing limits	0.92	1.00	0.96	60
Turn right ahead	1.00	1.00	1.00	210
Turn left ahead	1.00	1.00	1.00	120
Ahead only	1.00	0.96	0.98	390
Go straight or right	0.94	1.00	0.97	120
Go straight or left	0.91	1.00	0.95	60
Keep right	1.00	1.00	1.00	690
Keep left	1.00	1.00	1.00	90
Roundabout mandatory	0.97	0.97	0.97	90
End of no passing	0.91	1.00	0.95	60
End no passing veh > 3.5 tons	0.99	0.91	0.95	90
accuracy			0.97	12630
macro avg	0.95	0.97	0.96	12630
weighted avg	0.97	0.97	0.97	12630

Confusion matrix of Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization is shown in Fig. 8.

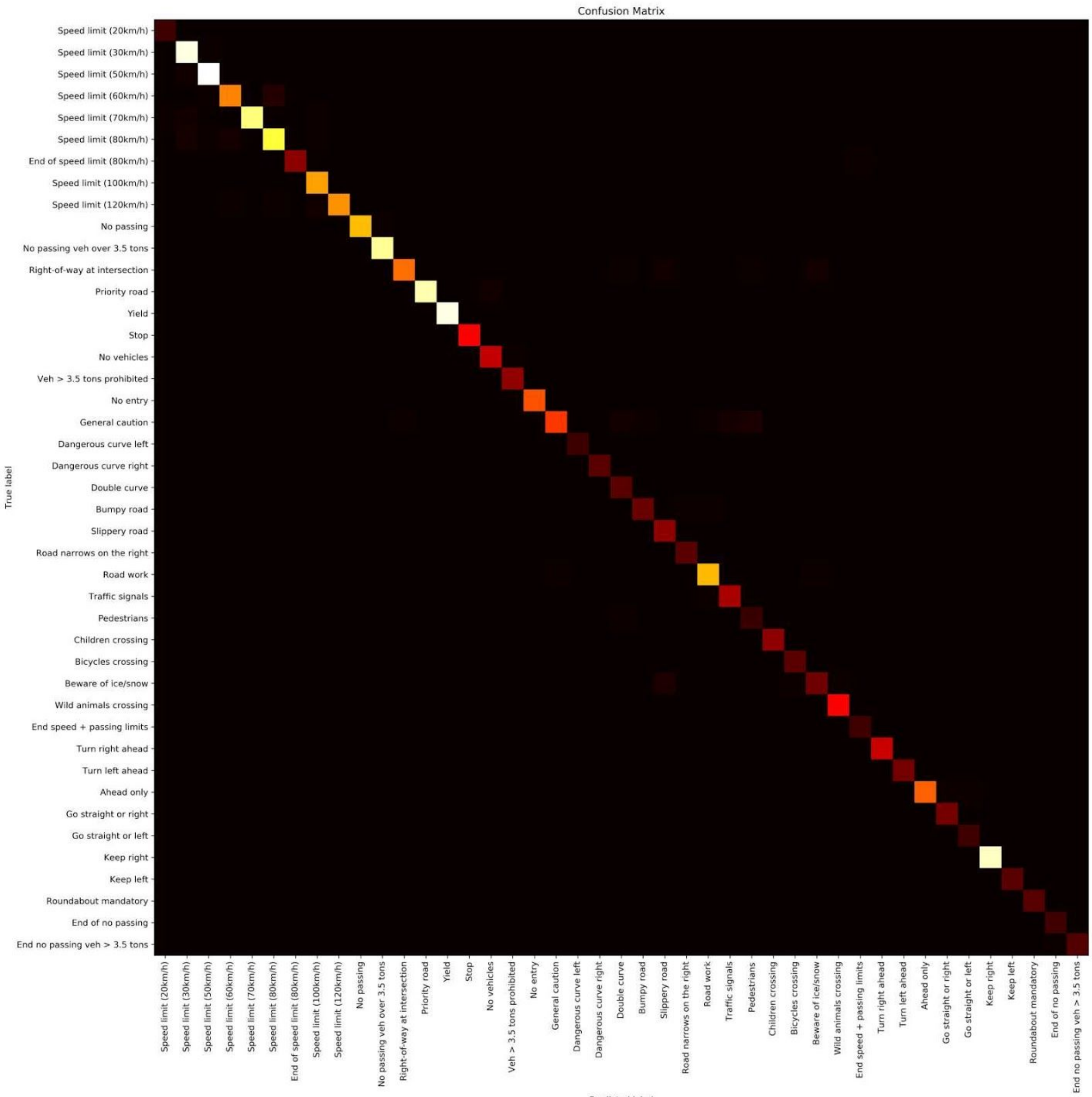
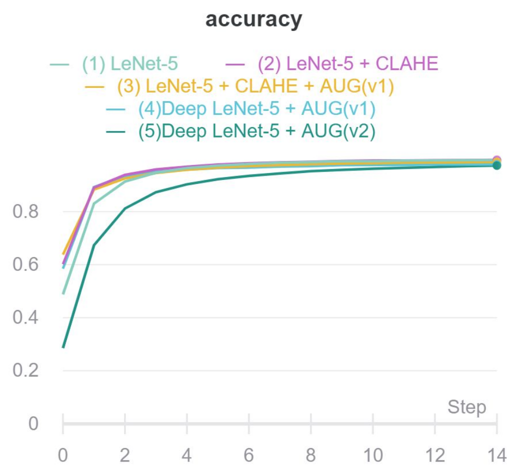
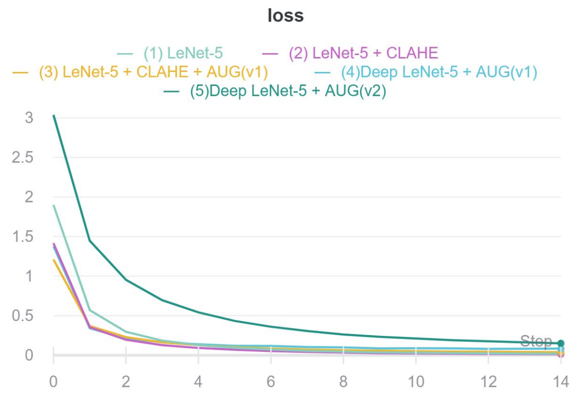


Fig. 8. Confusion matrix of 5th model

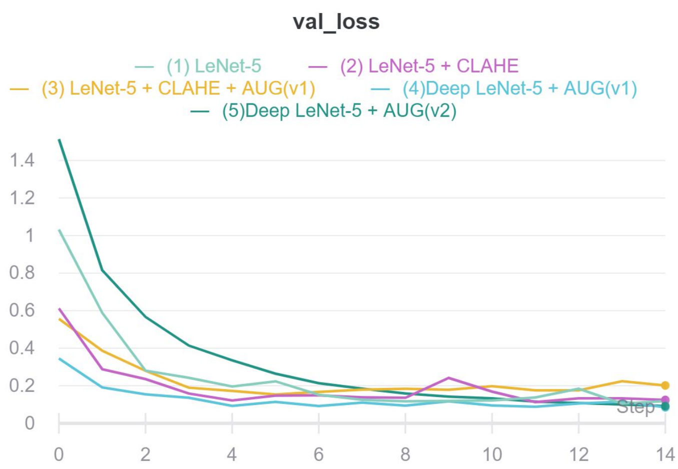
The accuracy, loss and validation loss is shown in Fig. 9 (a), (b) and (c) respectively.



(a)



(b)



(c)

Fig. 9. (a) Accuracy comparison of 5 proposed models (b) Loss comparison of 5 proposed models (c) Validation loss comparison of 5 proposed models

7. Discussion on Results

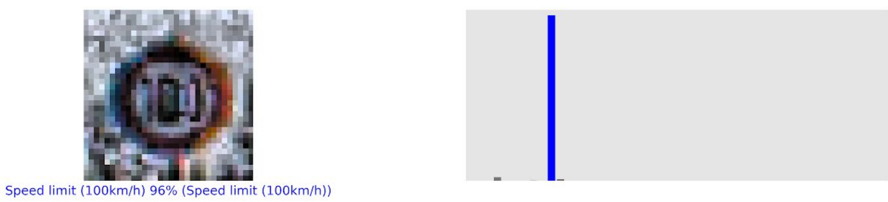
We have achieved the state-of-the-art performance of 99.22% on GTSRB dataset, which surpassed the best human performance of 98.84% as shown in table.The validation accuracy and training accuracy comparison of 5 different models are shown in TABLE VI.

TABLE VI
VALIDATION AND TRAINING ACCURACY COMPARISON OF 5 MODELS

Name (5 visualized)	val_accuracy	accuracy
(1) LeNet-5	0.978	0.9944
(2) LeNet-5 + CLAHE	0.9752	0.9955
(3) LeNet-5 + CLAHE + AUG(v1)	0.9596	0.9876
(4)Deep LeNet-5 + AUG(v1)	0.9771	0.9781
(5)Deep LeNet-5 + AUG(v2)	0.9922	0.9738

As seen in table, the first three models are clearly overfitting.To reduce overfitting dropout (50%)is used in Fully Connected Layer of Deep LeNet-5 + Contrast Enhancement + Augmentation(3000) model and 2 Convolutional Layer with 1024 and 512 neurons are added respectively.Finally, Augmented dataset version 2 containing 4500 images per sample is introduced to the final model Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization and L2 Regularization with 1e-3 is also used to increase the generalisation accuracy to 99.22%.

Some prediction done by Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization model is shown in Fig. 10.



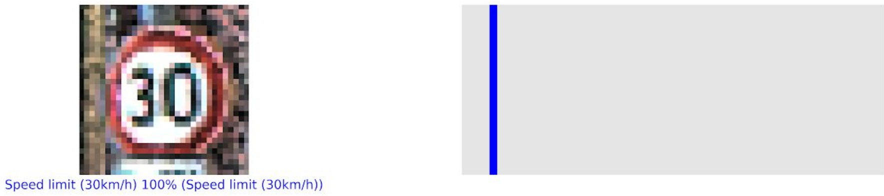


Fig. 10. Prediction done by 5th model

The results could have been better by using Deep Inception based Convolutional Neural network as proposed in Traffic Sign Classification Using Deep Inception Based Convolutional Networks[13].we could have deep network consisting of spatial transformer layers and a modified version of inception module specifically designed for capturing local and global features together. This features adoption allows the network to classify precisely intraclass samples even under deformations. Use of spatial transformer layer makes this network more robust to deformations such as translation, rotation, scaling of input images. Unlike existing approaches that are developed with hand-crafted features, multiple deep networks with huge parameters and data augmentations, This method addresses the concern of exploding parameters and augmentations. The network architecture that could be used is shown in Fig. 11.

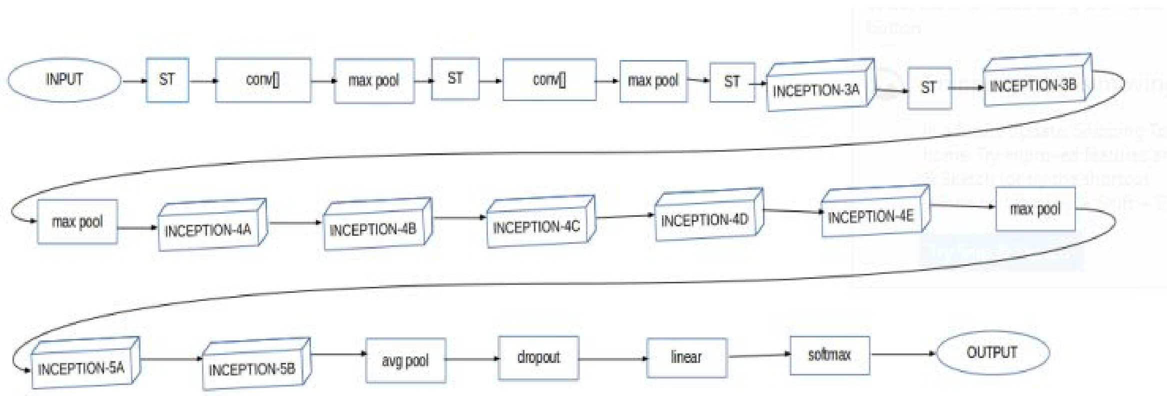


Fig. 11. Modified Inception Network

8. Contributions of each group member

Name	Work
Junth Basnet	1. Research on different network architecture 2. Model architecture design and visualization 3. Implementing the CNN model using Keras deep learning library
Sandip Dulal	1. Research 2. Data preprocessing and visualization
Abin Sainju	1. Research 2. Testing model 3. Documentation

9. Code

Source Code is available at Github:

https://github.com/Junth19/Traffic-Signs-Recognition-System

Training visualisation, saved models and performance comparison of 5 proposed models is available at:

https://app.wandb.ai/junth/traffic-sign-recognition-classifier

10. Conclusion and Future Extensions to the Project

We presented a Convolutional Network architecture with state-of-the-art results on the GTSRB traffic sign dataset implemented with the keras deep learning library.Using Deep LeNet-5 + Contrast Enhancement + Augmentation(4500) + Regularization architecture, 99.22% accuracy is obtained which surpassed the best human performance of 98.84%.The proposed algorithm has more admirable accuracy, better real-time performance, stronger generalization ability and higher training efficiency than other algorithms. The accurate recognition rate and average processing time are significantly improved.

Future work should investigate the impact of deeper layers.The impact of input resolution should be studied to improve both accuracy and processing speed.More diverse training set deformations can also be investigated such as brightness, contrast, shear and blur perturbations to address the numerous real-world deformations.In the future, the inclusiveness and anti-error recognition of the traffic sign recognition algorithm can be further optimized and improved to exploit the overall performance of the algorithm.

References

[1] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In Proceedings of the IEEE International Joint Conference on Neural Networks, pages 14531460. (2011).

[2] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. (1998)

[3] Haloi, M. A novel pLSA based Traffic Signs Classification System. arXiv preprint arXiv:1503.06643. (2015).

[4] Haloi, M., Jayagopi, D. B. Characterizing driving behavior using automatic visual analysis. In Proceedings of the 6th IBM Collaborative Academia Research Exchange Conference (I-CARE) on I-CARE 2014 (pp. 1-4). ACM. (2014).

[5] McCall, J. C., Trivedi, M. M. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. IEEE transactions on intelligent transportation systems, 7(1), 20-37. (2006).

[6]Yang, J.; Coughlin, J.F. In-vehicle technology for self-driving cars: Advantages and challenges for aging drivers. Int. J. Automot. Technol., 15, 333–340.(2004).

[7] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arxiv:1412.6980 (2014).

[8] Haloi, M., Jayagopi, D. B. A robust lane detection and departure warning system. In 2015 IEEE Intelligent Vehicles Symposium (IV) (pp. 126-131). (2015).

[9] Haloi, M., Jayagopi, D. B. Vehicle Local Position Estimation System. arXiv preprint arXiv:1503.06648. (2015).

[10] Braunagel, C., Kasneci, E., Stolzmann, W., Rosenstiel, W. Driver-activity recognition in the context of conditionally autonomous driving. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems (pp. 1652-1657). IEEE. (2015).

[11] A. F. Agarap. Deep learning using rectified linear units (relu). arXiv preprint arxiv:1803.08375, (2018).

[12] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929–1958, (January 2014).

[13] M. Haloi. Traffic sign classification using deep inception based convolutional networks. arXiv preprint arxiv:1511.02992 (Nov. 2015)

[14] Sermanet, P., LeCun, Y. Traffic sign recognition with multi-scale convolutional networks. In Neural Networks (IJCNN), The 2011 International Joint Conference on (pp. 2809-2813). IEEE (2011)

[15] Sermanet, P., LeCun, Y. Traffic sign recognition with multi-scale convolutional networks. In Neural Networks (IJCNN), The 2011 International Joint Conference on (pp. 2809-2813). IEEE (2011)

[16] Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M., Koehler, T. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In Intelligent Vehicles Symposium, 2005. Proceedings. IEEE (pp. 255-260). IEEE (2005)

[17] Le, T. T., Tran, S. T., Mita, S., Nguyen, T. D. Real time traffic sign detection using color and shape-based features. In Intelligent Information and Database Systems (pp. 268-278). Springer Berlin Heidelberg (2010)

[18] Lafuente-Arroyo, S, Gil-Jimenez, P, Maldonado-Bascon, R, Lopez Ferreras, F, and Maldonado-Bascon, S. Traffic sign shape classification evaluation i: Svm using distance to borders. In Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, pages 557 – 562, (2005).

[19] de la Escalera, A, Moreno, L, Salichs, M, and Armingol, J. Road traffic sign detection and classification. Industrial Electronics, IEEE Transactions on, 44(6):848 –859, (December 1997).

[20] Torresen, J, Bakke, J, and Sekanina, L. Efficient recognition of speed limit signs. In Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on, pages 652 – 656, (2004).

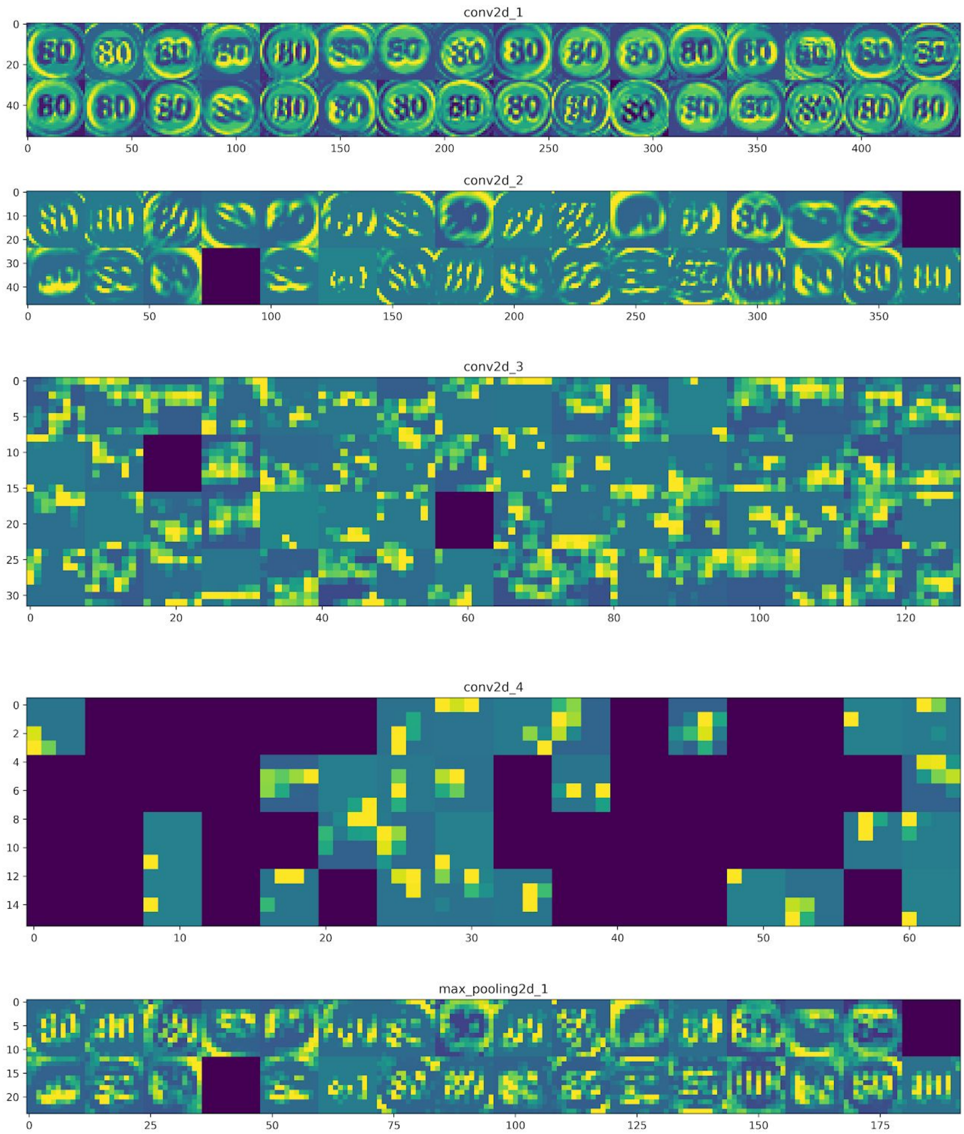
Appendices

Visualizing what convnets learn

The representations learned by convnets are highly amenable to visualization, in large part because they’re representations of visual concepts.

Visualizing intermediate convnet outputs (intermediate activations)

Visualizing intermediate activations consists of displaying the feature maps that are output by various convolution and pooling layers in a network, given a certain input.



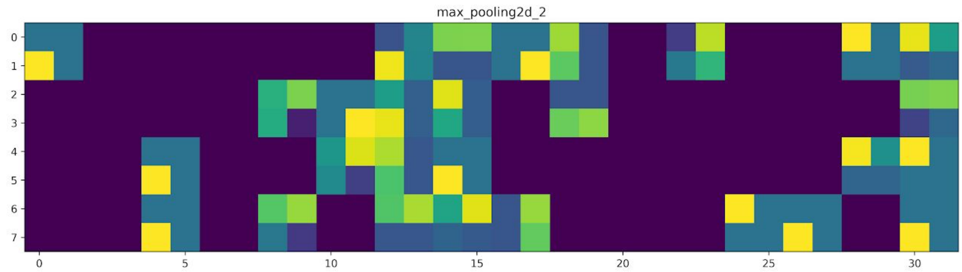
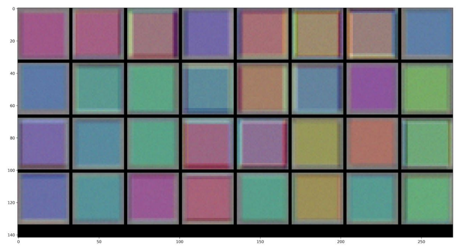
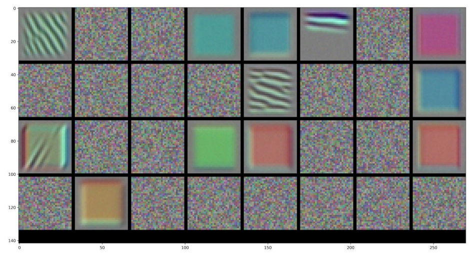


Fig. 12. Intermediate activations

Visualizing convnets filters

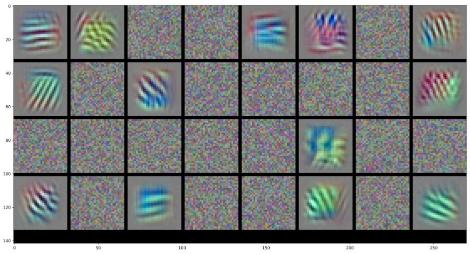
This can be done with gradient ascent in input space: applying gradient descent to the value of the input image of a convnet so as to maximize the response of a specific filter, starting from a blank input image. The resulting input image will be one that the chosen filter is maximally responsive to.



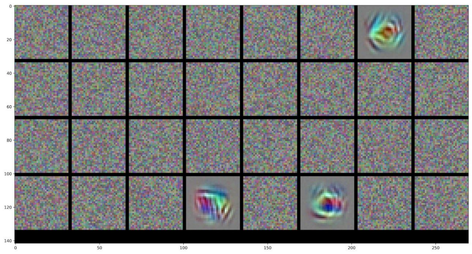


Conv2d_1

Conv2d_2



Conv2d_3



Conv2d_4

Fig. 13. Convnets filters