Traffic Sign Recognition

Robin CR - RET17EC126
Sanjay MS - RET17EC136
Vidhu Krishnan - RET17EC178
V S Achuthan - RET17EC186

S8 ECE Gamma
Rajagiri School of Engineering and Technology
Guided By: Mr. Ajai V Babu

June 16, 2021

TABLE OF CONTENTS

Problem Statement

Problem Statement

- Traffic signs are sometime ignored by the drivers, which may lead to accidents - Traffic sign recognition

Objective

Objective

To detect the traffic sign boards and to provide a voice message according to the sign.
Our goal is realized through the following objectives :

- Dataset preparation.
- Design an efficient algorithm for detecting traffic signs on the road.
- Train a model to detect traffic signs, classify them into one or more classes.
- Evaluate the detection accuracy and performance of the trained model.

Block Diagram

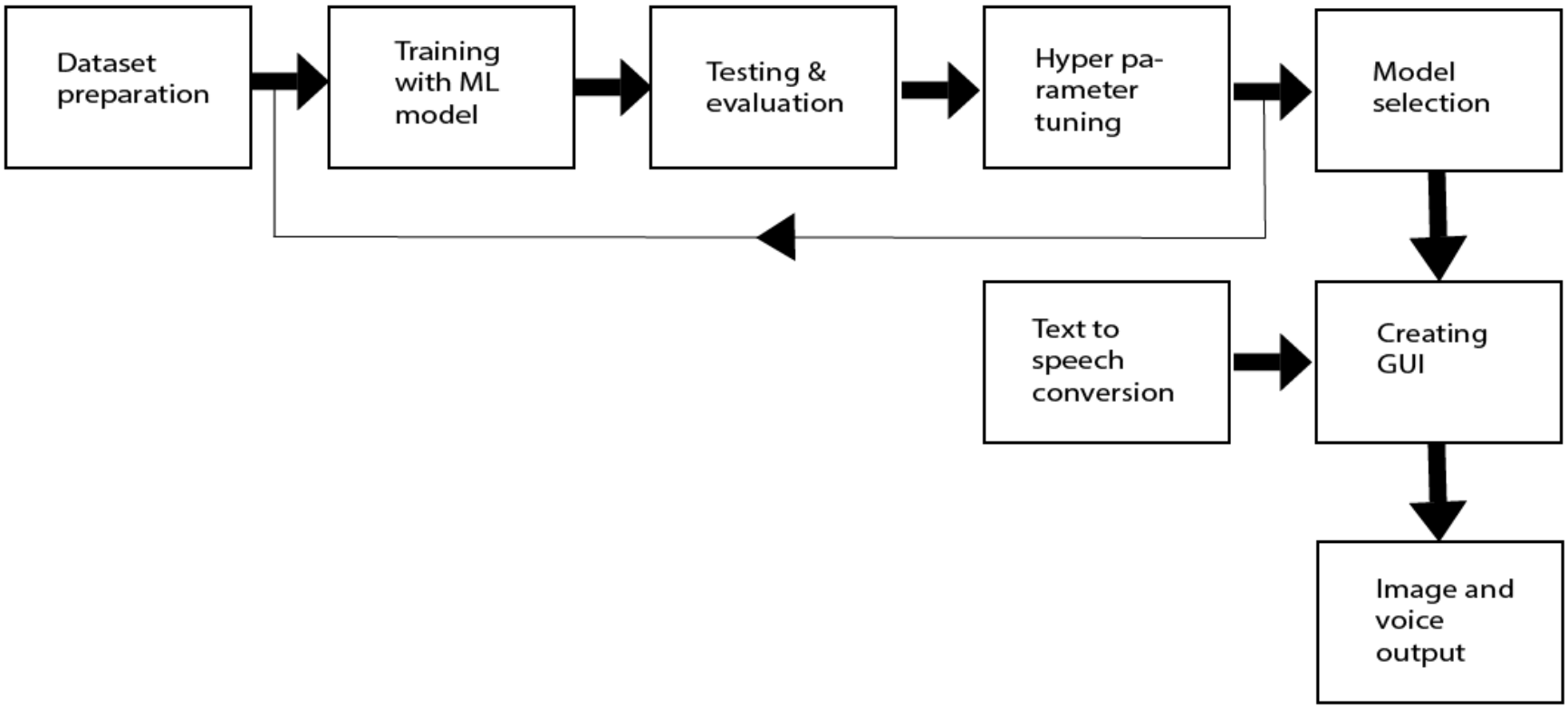

Figure 1: Block Diagram

Milestones Completed

Dataset Preparation

- In our project we have used a German traffic sign dataset from the site kaggle.com . The dataset include 43 signs. The signs which are common among German and Indian signs are filtered.
- After thorough research 12 signs are selected, that if the driver didn't notice any of these sign boards the chances of getting to accident is very high.
- Hindusthan Times published a report on road accidents in India on 2019. The traffic signs were selected based on the collision accidents mentioned in the report.

Milestones Completed

Dataset Summary

- Number of training examples = 34799
- Number of validation examples = 4410
- Number of testing examples = 12630
- Image data shape = (32, 32)
- Number of classes = 43

Milestones Completed

Data Augmentation

- During dataset preparation we encountered a problem called class skew. It arises when there is wide variation in training images among different classes.
- In our project there is around 500 images for "Hump" sign but only 40 images for "No parking" signal. So the solution is data augmentation.
- It will create new images by changing the properties such as orientation, size etc.., so that there is enough data for each class.

Milestones Completed

Building a Model

- A deep learning model is developed with the help of neural network.
- Convolutional Neural Network (CNN) algorithm is used in our project.
- The model for traffic sign detection is implemented with the help of CNN.
- CNNs can learn what characteristics in the filters are the most important. That saves a lot of time and trial and error work since there is no need to look for that parameters.

Milestones Completed

Voice Assistant Setup

- If a particular sign is detected a voice message is given according to the sign.
- Suppose "School Ahead" traffic sign is detected, then a voice message "School is Ahead, so please control your speed" is given by the system.
- This is implemented with the help of Google Text to Speech library in python.

Milestones Completed

Creating GUI

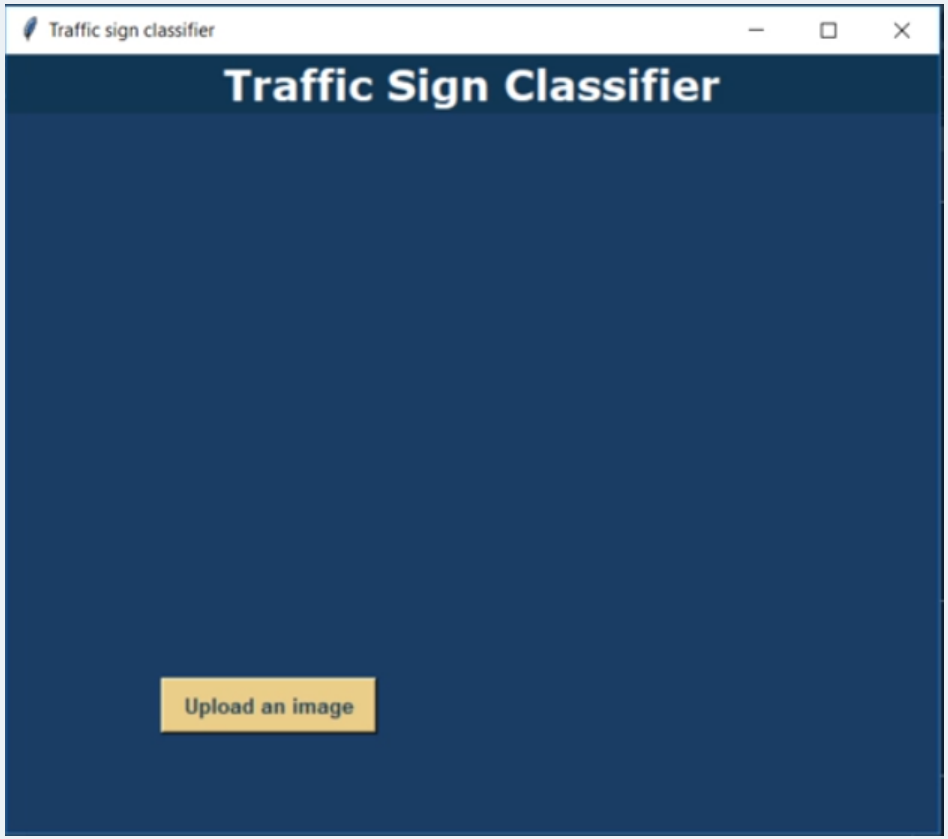

Figure 2: Graphical User Interface

Training in CNN

CNN consists of three layers Convolution layer, a Max Pooling layer, and a Softmax layer.

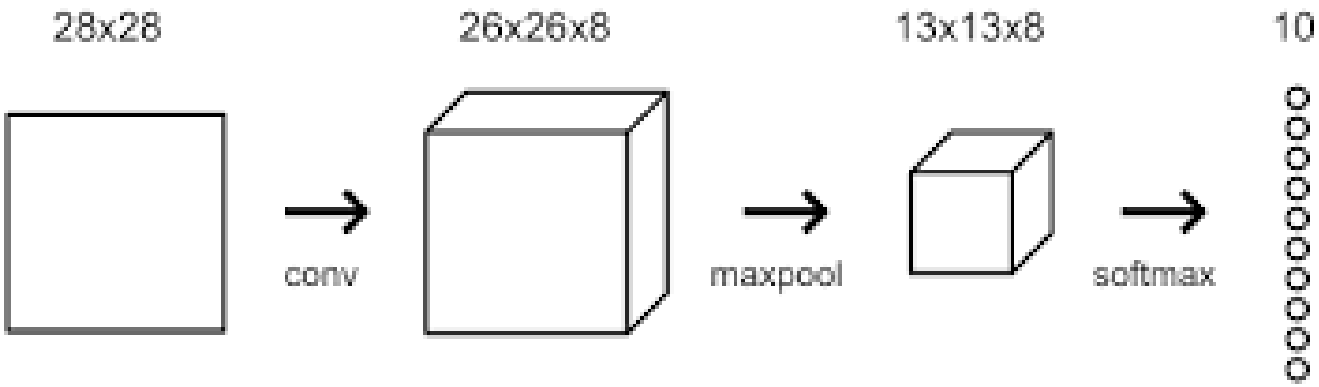

Figure 3: CNN layers

Training in CNN

Training a neural network typically consists of two phases:

- A forward phase, where the input is passed completely through the network.During the forward phase, each layer will cache any data (like inputs, intermediate values, etc) it'll need for the backward phase. This means that any backward phase must be preceded by a corresponding forward phase.
- A backward phase, where gradients are backpropagated and weights are updated.During the backward phase, each layer will receive a gradient and also return a gradient. It will receive the gradient of loss with respect to its outputs and return the gradient of loss with respect to its inputs.

Model Parameters

- No. of Filters = 60
- No. of nodes in hidden layers = 500
- Activation function in conv. layer = Relu
- Dropout = 0.5
- Activation function in output layer = Softmax
- Epochs = 10

Output of training

```
Epoch 1/10
981/981 [==============================] - 74s 75ms/step - loss: 3.7390 - accuracy: 0.2706 -
val_loss: 0.8561 - val_accuracy: 0.8009
Epoch 2/10
981/981 [==============================] - 92s 93ms/step - loss: 1.0241 - accuracy: 0.6987 -
val_loss: 0.3275 - val_accuracy: 0.9083
Epoch 3/10
981/981 [==============================] - 95s 97ms/step - loss: 0.5496 - accuracy: 0.8361 -
val_loss: 0.1706 - val_accuracy: 0.9462
Epoch 4/10
981/981 [==============================] - 96s 98ms/step - loss: 0.4135 - accuracy: 0.8801 -
val_loss: 0.1109 - val_accuracy: 0.9663
Epoch 5/10
981/981 [==============================] - 96s 98ms/step - loss: 0.2817 - accuracy: 0.9212 -
val_loss: 0.0887 - val_accuracy: 0.9755
Epoch 6/10
981/981 [==============================] - 98s 100ms/step - loss: 0.2694 - accuracy: 0.9227 -
val_loss: 0.0800 - val_accuracy: 0.9773
Epoch 7/10
981/981 [==============================] - 96s 98ms/step - loss: 0.2519 - accuracy: 0.9291 -
val_loss: 0.0708 - val_accuracy: 0.9809
Epoch 8/10
981/981 [==============================] - 98s 100ms/step - loss: 0.2126 - accuracy: 0.9398 -
val_loss: 0.0887 - val_accuracy: 0.9768
Epoch 9/10
981/981 [==============================] - 96s 98ms/step - loss: 0.2591 - accuracy: 0.9308 -
val_loss: 0.0646 - val_accuracy: 0.9821
Epoch 10/10
981/981 [==============================] - 97s 99ms/step - loss: 0.2193 - accuracy: 0.9406 -
val_loss: 0.1633 - val_accuracy: 0.9573
```

Figure 4: Accuracy of training

Constraints

- Road and Traffic conditions in various regions.
- Time constraints can restrict a certain property of the system. These could be events either stimuli from the environment (e.g. camera sends in an image) or responses from the system (e.g. control outputs an action).
- Response time is less than 300ms.

Testing the model

In order to calculate the efficiency of the model, we need to test it under various conditions. Real time images from the roads were taken. The images were taken under the following conditions.

- Morning
- Evening
- Tilted
- Real time using web camera

Test images - morning

Figure 5: Picture - morning

Test images - evening

Figure 6: Picture - evening

Test images - tilted

Figure 7: Picture - tilted

Real time/ video

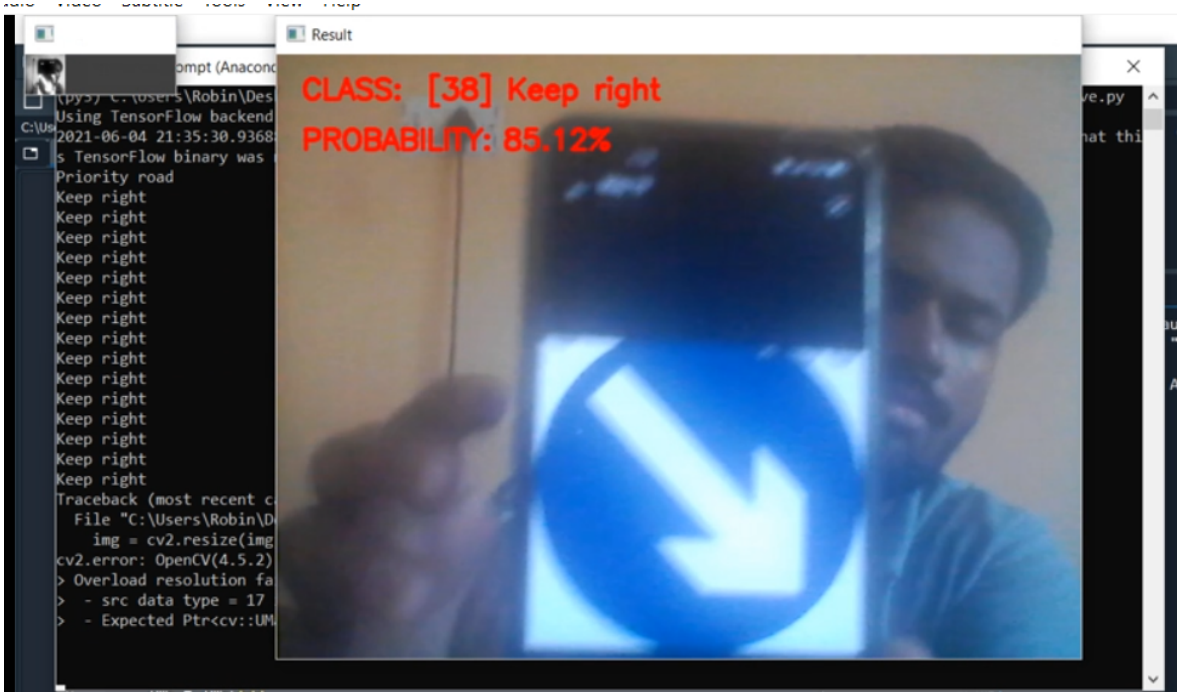

Figure 8: Real time

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

Accuracy

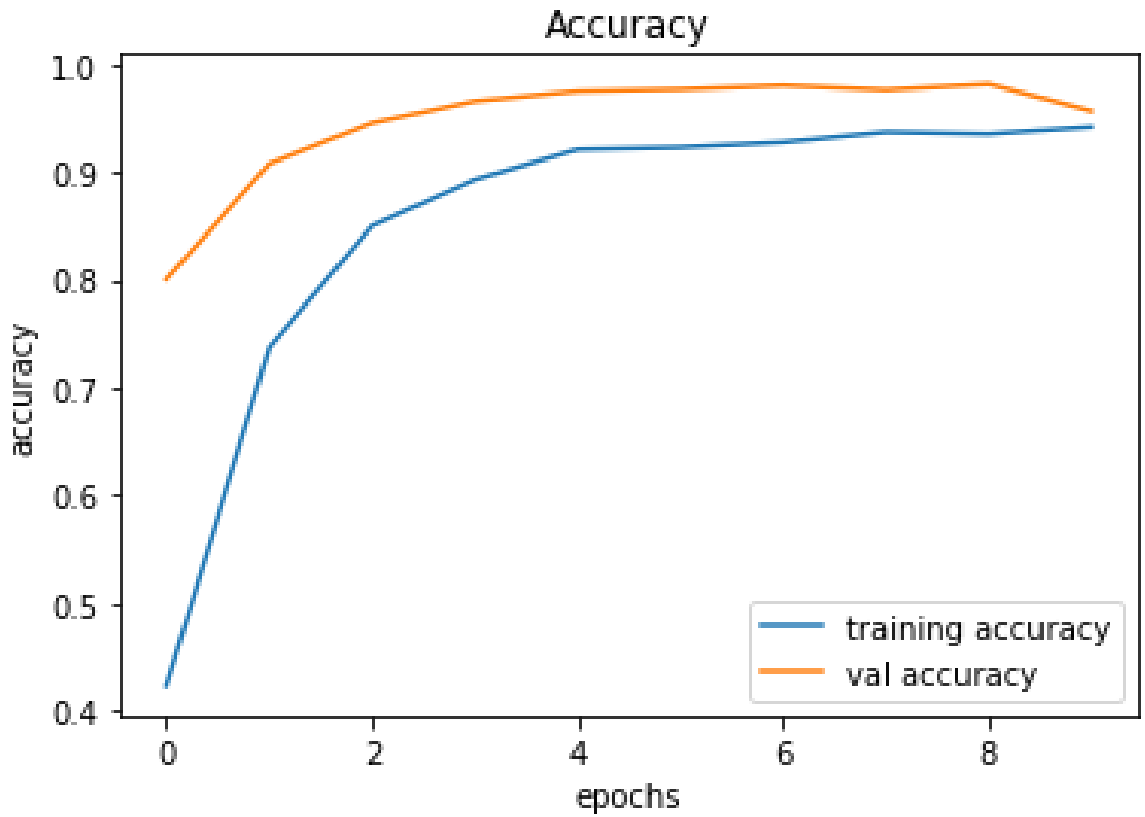

Figure 9: Accuracy plot

Loss

Cross entropy loss

Cross-entropy loss is used when adjusting model weights during training. The aim is to minimize the loss, i.e, the smaller the loss the better the model. A perfect model has a cross-entropy loss of 0.

$$loss = -\sum_{i=0}^{n} t_i \log(p_i) \tag{2}$$

n= total no. of classes
t = Truth label for i th class
p = Softmax probability for i th class
Categorical cross-entropy is used when true labels are one-hot encoded.

Loss

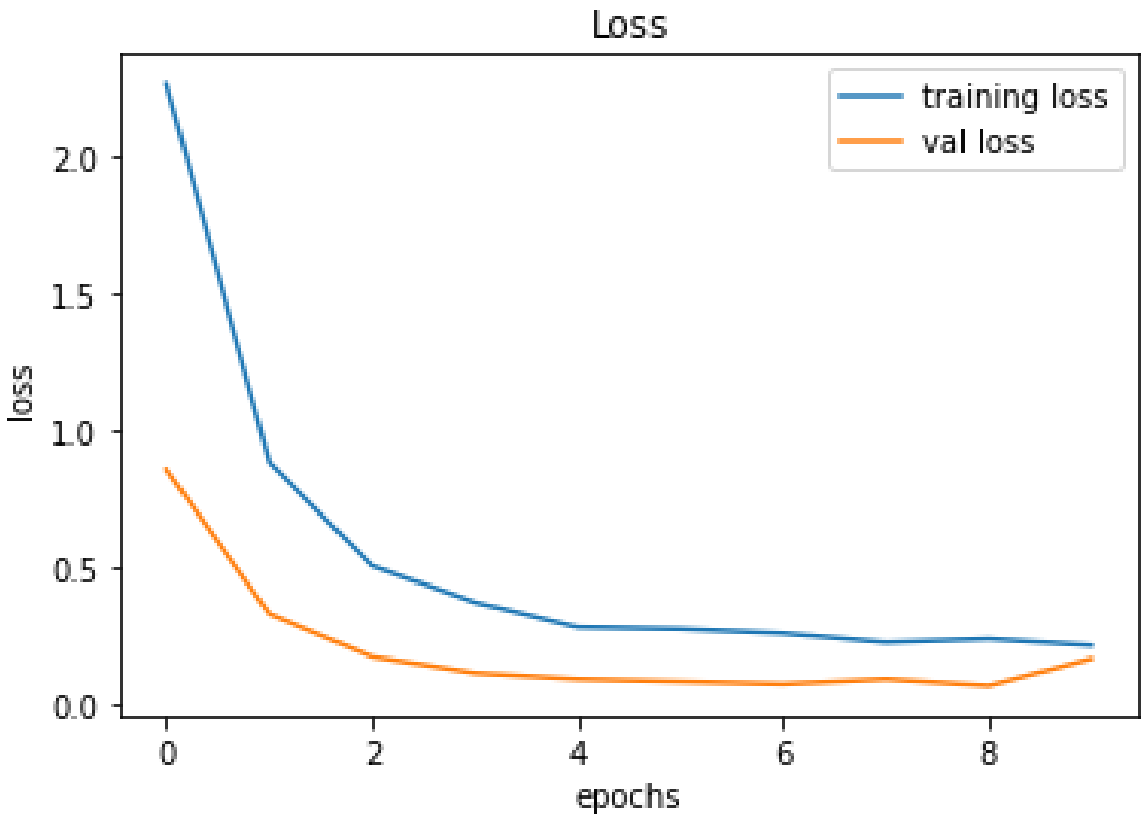

Figure 10: Loss plot

Results

- The German Traffic sign dataset containing over 43 traffic signs were preprocessed and trained using CNN algorithm.
- A User Interface is build for traffic sign detection.
- The text to speech conversion is done for Voice assisted message system.
- The model is tested with real time images and works perfectly.
- The CNN algorithm shows an accuracy of 0.9573.
- Categorical cross-entropy is calculated as a part of evaluating the performance of model.CNN shows a loss of 0.1633.

Future Scope

- In the world of Artificial Intelligence and advancement in technologies, research in autonomous driving will dominate in such a way that the vehicles should be able to interpret traffic signs and make decisions accordingly.
- A lot of methods will be integrated based on colour, shape, priority etc, which can lead to an efficient model in respective countries.
- Natural obstacles such as fog, smoke etc cause noise while image acquisition, models will be developed which will overcome these challenges.

Conclusion

- The project focuses on a new way of traffic sign detection using CNN which is very effective among the Indian traffic signs on the road without the use of extensive and expensive evaluations.
- The model developed remains to be a foundation that can be adjusted to different types of vehicles, such as bus or trucks, by considering the properties and constraints that defines each one.
- This also serves as a meaningful development in field of neural networks and its learning systems, as well as provide valuable insights for future progress.

Reference

[1] Ma Weixin, Xiong Changzhen, Wang Cong and Shan Yanmei." A traffic sign detection algorithm based on deep convolutional neural network",*Eng.Appl. Artif. Intelli.*, 48, 2016.

[2] Kang Kim, Hee Seok Lee. "Simultaneous traffic sign detection and boundary estimation using convolutional neural network" *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[3] T.Tan, J.Zhang and L. Ma. "Invariant texture segmentation via circular gabor Filters", *Proc. Object Recognit. Supported User Interact. Service Robot*, pages 901-904, 2002.

[4] Bei Tong ,Hengliang Luo, Yi Yang and Fuchao Wu." Traffic sign recognition using a multi-task convolutional neural network" *IEEE Transactions on Intelligent Transportation Systems*, 2017.

THANK YOU