

UNDERGRADUATE PROJECT PROGESS REPORT

|  |  |
| --- | --- |
| **Project Title:** | **A Convolutional Neural Network Approach for the Recognition of Traffic Signs** |
| **Surname:** | **Wu** |
| **First Name:** | **Yile** |
| **Student Number:** | **201918020102** |
| **Supervisor Name:** | **Happy Monday** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **2023-01-13** |

**Table of Contents**

[1 Introduction 3](#_Toc124467318)

[1.1 Background 3](#_Toc124467319)

[1.2 Aim 3](#_Toc124467320)

[1.3 Objectives 3](#_Toc124467321)

[1.4 Project Overview 4](#_Toc124467322)

[1.4.1 Scope 4](#_Toc124467323)

[1.4.2 Audience 4](#_Toc124467324)

[2 Background Review 4](#_Toc124467325)

[3 Project Technical Progress 5](#_Toc124467326)

[3.1 Methodology 5](#_Toc124467327)

[3.1.1 Approach 5](#_Toc124467328)

[3.1.2 Technology 6](#_Toc124467329)

[3.2 Testing and Evaluation 6](#_Toc124467330)

[3.3 Design and Implementation 6](#_Toc124467331)

[3.3.1 Model design 6](#_Toc124467332)

[3.3.2 Data Process 7](#_Toc124467333)

[3.3.3 Training verse Validation for Accuracy and Loss 7](#_Toc124467334)

[3.3.4 GUI 8](#_Toc124467335)

[4 Project Management 9](#_Toc124467336)

[4.1 Activities 9](#_Toc124467337)

[4.2 Schedule 10](#_Toc124467338)

[4.3 Project Version Management 11](#_Toc124467339)

[4.4 Project Data Management 11](#_Toc124467340)

[4.5 Project Deliverables 11](#_Toc124467341)

[5 Professional Issues and Risk: 11](#_Toc124467342)

[5.1 Risk Analysis 11](#_Toc124467343)

[5.2 Professional Issues 12](#_Toc124467344)

[5.2.1 Legal Issue 12](#_Toc124467345)

[5.2.2 Social Issue 12](#_Toc124467346)

[5.2.3 Ethnical Issue 12](#_Toc124467347)

[5.2.4 Environmental Issue 12](#_Toc124467348)

[6 References 13](#_Toc124467349)

# Introduction

## Background

Traffic sign recognition is becoming more and more promising since the recent interested development of unman driving cars. Moreover, human error made in traffic sign misidentification has always been a risk when driving. There have been cases of death associated with misidentification of traffic signs in China over the year. However, traffic sign recognition is of great help since it is able to remind drivers of the signs, and automobile as well. The motivation for the development of traffic sign recognition system is to save lives and mitigate traffic accident as well as to improve accuracy of traffic sign recognition system. The rest of the paper is organized as follows. Section 1.2 covers the aim of this project and section 1.3 discusses the objectives of this study. Section 1.4 explains the significance and targeted audience of this project. Important and relevant literature related to this project is discussed in section 2. Section 3 details the approach, technology and the management plan of this project while the project management activities, project schedule and deliverables are presented in section 4. Finally, risk analysis as informed by current progress, also identification and discussion of professional issues are detailed in section 5.

## Aim

The aim of this project is to develop a traffic sign recognition system based on a mobileNetv2 convolutional neural network model.

## Objectives

The objectives are as follows.

1: Finish a background review of the existing CNN technology.

2: Design the CNN architecture.

3: Train and test the CNN architecture with a public dataset named Chinese traffic sign using appropriate technology.

4: Evaluate the model with test data and accuracy.

5: Explain the work to related audience.

## Project Overview

### Scope

The project seeks to develop a mobileNetV2 convolutional neural network (CNN) capable of identifying and classifying traffic signs which would greatly benefit road traffic management. Developing a computer vision system for traffic sign recognition with high precision is paramount.

### Audience

The project would bring a large benefit to government agency, automobile manufacturers, driving school owners and trainees within the domain who can utilize this computer vision technology to recognize traffic with high accuracy.

# Background Review

One of common problems for CNN model is to find a balance between accuracy and the network depth since large network with small data would lead to model overfitting. It is an obvious fact that one single layer is not enough for achieving satisfactory accuracy. Haque et al. (2021) describe a light-weight CNN model for traffic sign recognition. The architecture consists of four convolutional layers, two overlapping max-pooling layers and then followed by one fully-connected layer. The authors implemented overlapping max pooling and sparsely used stride convolution to made training faster in order to reduce overfitting issue. The benefit of a light-weight structure is to lower computational cost with a considerable depth of four convolutional layers. The proposal model achieved 98.97% accuracy as the authors claimed.

Vennelakanti et al. (2019) propose a CNN Ensemble to identify traffic signs. The authors use a feed-forward network with six convolutional layers, 3 max pooling layers and 2 fully connected layers. The suggested ensemble model of three convolutional neural networks aggregates the output which is a great technique of improving the accuracy. The authors implemented fully connected hidden layers with dropout between the layers to avoid overfitting problem. The authors claimed that their proposed model achieved 98.11% accuracy for triangular traffic signs and 99.18% accuracy for the circular traffic signs. Shustanov and Yakimov (2017) suggest three CNN layers of one full-connected layer model with soft max as a classifier. The authors claimed that their model achieved 99.94% accuracy.

Moreover, Sun et al. (2019) propose a light-weight CNN classifier which is consist of two convolutional layers, two pooling layers with two full connected layers and achieved test accuracy of 98.2%. The authors implemented RELU activation function in each layer of the CNN in order to learn complex features as well as adding dropout to prevent overfitting. Alghmgham et al. (2019) suggest a CNN network with two convolutional layers, two max pooling layers with one dropout layer and 3 dense layers. The proposed model uses a large number of layers with the technique of utilizing two pooling layers to prevent overfitting problem by decreasing the dimension of feature maps. However, the novelty of their work is the utilization of leaky RELU to overcome the problem of dead neurons which is common when normal RELU is used. Basically, leaky RELU does not output zero when the input values are less than zero, instead it outputs negative value. The authors claimed that their model achieved 100% accuracy.

In addition, Howard et al (2017) propose MobileNets for mobile and embedded version application that are based on depth-wise separable convolutions, aiming to build light weight neural networks. Depth-wise separable convolutions is a depth-wise convolution which is the channel-wise DK\*DK spatial convolution. Then followed by a pointwise convolution, which is 1\*1 convolution to change the dimension. Compared to standard convolution cost, less computation can be achieved but only slightly reduction in accuracy.

Lastly, Sandler et al (2018) propose a new mobile architectural, MobileNetV2 that based on MobileNets. MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers overall. Moreover, MobileNetV2 introduces a new CNN layer, the inverted residual and linear bottleneck layer. It enables high accuracy in mobile applications. The new layer builds on the depth-wise separable convolutions which is introduced above in MobileNets. More importantly, the MobileNetV2 network is built around this new layer. In MobileNetV2, there are two types of blocks. One is residual block with stride of 1 and the other one is block with stride of 2 for downsizing. There are 3 layers for both types of blocks. The first layer is 1×1 convolution with ReLU6. The second layer is the depth-wise convolution. The third layer is another 1×1 convolution.

# Project Technical Progress

## Methodology

### Approach

The core deep learning model that this project focuses on is MobileNetV2, which is a convolutional neural network model that aims to achieve a good performance on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. Non linearity comes from the intermediate expansion layer which uses lightweight depth-wise convolutions to filter features. Moreover, Keras API was adopted in the program, which enable to build layers and model. The optimizer utilized in the project is Adam, which is can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. Additionally, the dataset that apply to the project is Chinese Traffic Signs. This dataset contains 5998 traffic sign images of 58 categories. Each image is a zoomed view of single traffic sign. As for the data processing, the approach is Keras image data generator. It generates batches which contains the data of tensor images and help loop over the data in batches.

### Technology

The software adopted for coding the model is jupyter notebook and the program language is Python and Pycharm is used to code GUI of the model. The hardware adopted to implement the code is Intel Xeon Cascade Lake (2.5 GHz).

## Testing and Evaluation

Firstly is to test the data to make sure the dataset satisfies the requirement. The Annotation of the dataset provide image properties like file name, width, height and traffic sign coordinates within image and category. And all images originally collected by camera under nature scenes or from BAIDU Street View. Therefore, this dataset is satisfied for traffic sign recognition.

Secondly is to evaluate and test the model. The accuracy metric that uses calculates how often predictions equal labels. This metric creates two local variables, total and count that are used to compute the frequency with which prediction matches actual. This frequency is ultimately returned as binary accuracy. And the model achieves accuracy of 87%. Moreover, to calculate train loss and test loss, cross-entropy loss is used for loss function. And the model achieves the loss of 38%.

## Design and Implementation

### Model design

The CNN model adopted Keras pretrained MobileNetV2 model but attach own classifiers. This can be achieved by attaching classifier which is just one dense layer. And then also include one flatten layer before adding a dense layer. This is because the former layer output is 4D, so it is necessary to convert the 4D output from the convolution layer to 2D, since the dense layer accepts 2D input. The next step is to add one batch normalization layer to reduce the learning process and the number of epochs. The overall model is presented in the following figure 1. As for the activations, RELU and Softmax are adopted to introduce non-linearity. The callbacks used are early stopping and ReduceLROnPlateau, which enable the control of training process. Early stopping could be needed to log performance or to stop training if performance metric reaches a certain threshold. And using ReduceLROnPlateau is to reduce the learning rate when the metric stops improving for long time. In addition, cross-entropy loss is used for loss function. Lastly, the model utilized Adam as the optimizer.

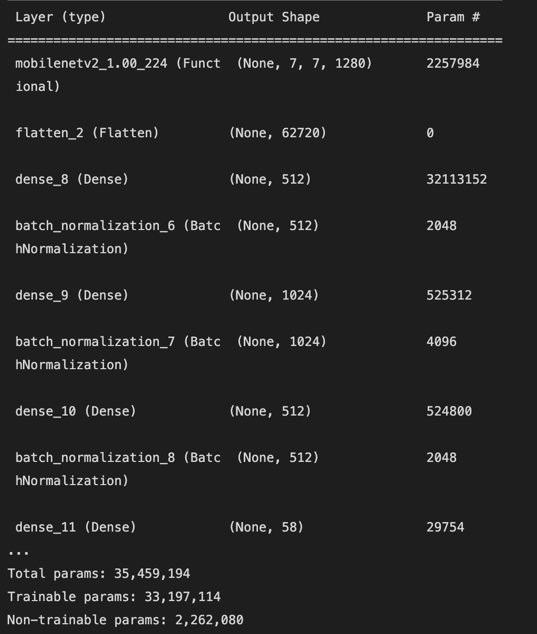


Figure 1. The model summary

### Data Process

In this step, Image data generator from keras is used to generate batches of image. Firstly, process the training dataset. Preprocess input method from mobilenet\_v2 is applied on each input. Use validation split since part of the training data need to be used as validation data. The model will set apart this part of the training data and will not train on it, then will evaluate the loss and any model metrics on this data at the end of each epoch. Then 15 for degree range in random rotations. And horizontal position and up-down position panning between 0 and 0.2. Then use flow from directory to take the path to image directory and then generate batches of augmented data. Then process the validation dataset. Use validation spilt and preprocess input method as above.

### Training verse Validation for Accuracy and Loss

The training and validation accuracy is presented in figure 2. Training and validation is presented in figure 3.

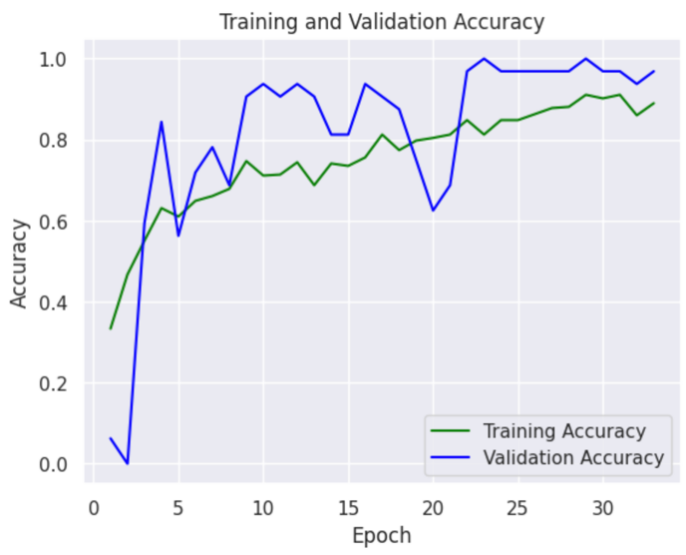


Figure 2. Training and Validation Accuracy

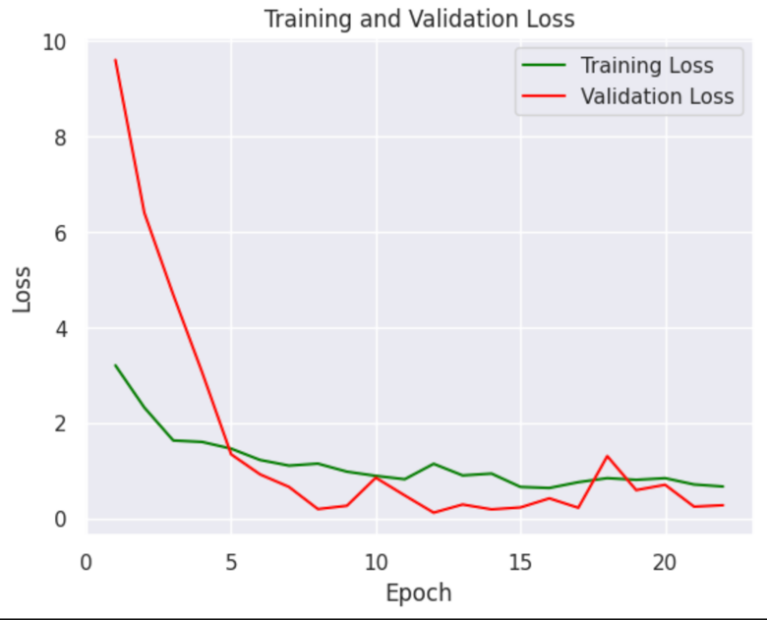


Figure 3. Training and Validation Loss

### GUI

The adopted Python framework to complete GUI is Tkinter, which is the standard Python interface to the GUI toolkit. First, use Tkinter to build the GUI window, which is the foundational element of a Tkinter GUI. Windows are the containers in which all other GUI elements live. Then the second step is create widgets, such as text boxes, labels, and buttons, which are all contained inside of windows. The last step is to use Tkinter command to realize the function of uploading the image and classifing it with the completed model mentioned above. First, define the functions as a callback, then assign the name of the function to the command option of the widget. The GUI is presented in figure 4.

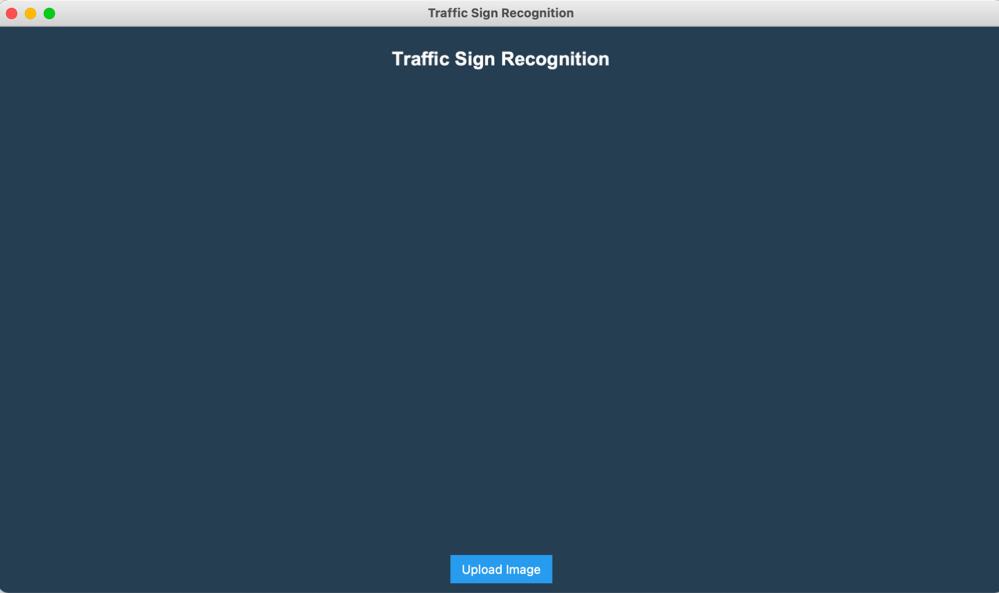


Figure 4. GUI of Traffic Sign Recognition

# Project Management

## Activities

|  |  |  |
| --- | --- | --- |
| Objective | Complete Task | Uncomplete Task |
| Finish a background review of the existing CNN technology. | * Conduct a systematic research of current CNN models. * Complete a feature competitive analysis. * Complete a literature review. * Conduct the requirement gathering. |  |
| Design the CNN architecture. | * Draw a scratch of the traffic sign recognition model including details the CNN architecture. * Study the dataset and summary features. * Complete GUI of the proposed product. |  |
| Train and test the CNN architecture with a public dataset named Chinese traffic sign using appropriate technology. | * Pre-process the images from dataset. * Realize the CNN architecture with code according to the scratch. * Train the CNN model with dataset. * Test the CNN model with dataset. |  |
| Evaluate the model with test data and accuracy. | * Obtain the accuracy of the proposed model. | * Reflect the model. |
| Explain the work to related audience. |  | * Complete the report. * Conduct the presentation. |

## Schedule

The schedule of the project is presented in the following Gantt chart.

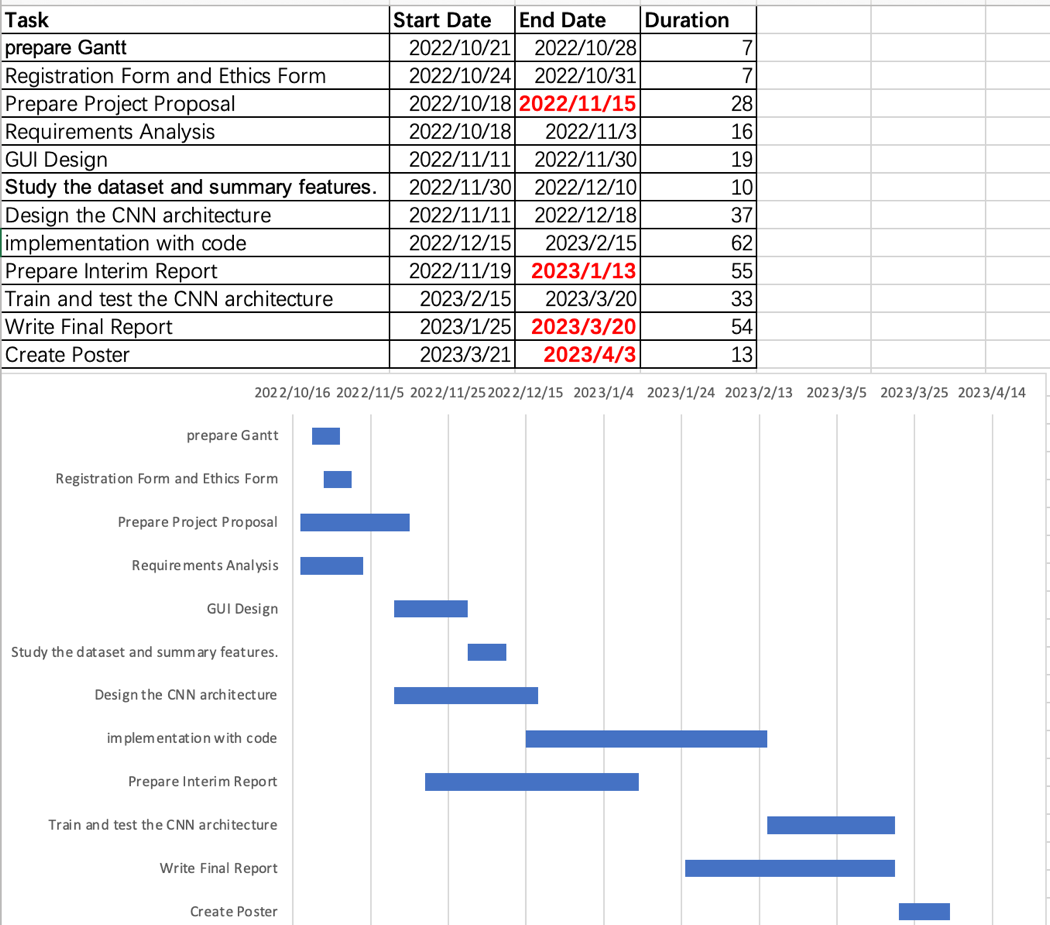


Figure 6. Gantt Chart

## Project Version Management

The source codes would be uploaded to this link <https://github.com/Haleywuyile/Traffic-Sign-Recognition> on GitHub.

## Project Data Management

All the related documentation would be uploaded to this link <https://github.com/Haleywuyile/Traffic-Sign-Recognition> on GitHub.

## Project Deliverables

The following are the deliverables:

1. Weekly meeting logs includes;
2. Progress report
3. Next steps and supervisor comments
4. Requirements gathering
5. Deep learning workflow model
6. Testing documentation
7. Reports includes;
8. Proposal
9. Interim report
10. Final report
11. Ethics forms
12. Useful links of literature.
13. Source code

# Professional Issues and Risk

## Risk Analysis

Firstly, the risk that this project faces now by current progress is technical, which is one sudden drop of validation accuracy in around the epoch of 20. Secondly, the technical risk has been solved which is about the hardware. The model used to train on a MAC book computer. However, the training process is extremely slow since the GUP memory on this computer is small. So the change the hardware is necessary. The strategy towards the risk is renting GPU server, which helped to speed up the process indeed. In addition, since this hardware risk, there is change of project plan. The process of training model had to extend since learning how to use a GPU server takes time. Lastly, the future risk of the project might be improving the accuracy. From the above figure 2, the training and validation accuracy of the model is not stable at last. So there is need to refine the model.

## Professional Issues

### Legal Issue

According to the Data Protection Act (2018), the dataset should fairly, lawfully and transparently use for specified, explicit purposes and in a way that is adequate, relevant and limited to only what is necessary. This means that the Chinese Traffic Signs dataset should only be used to build traffic sign recognition system, and cannot be used for other purposes.

### Social Issue

The social issues is to improve the understanding of technology, appropriate application, and potential consequences. Although there is neural network to help the drivers to recognize the traffic signs, it is still an auxiliary tool. The accuracy of the system is not 100 percent. Therefore, it is not sensible to rely entirely on the system to make decisions.

### Ethnical Issue

According to IEEE Code of Ethics (2020), members should accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment. Therefore, it is the developer’s duty to increase the accuracy of the model constantly, since the result of prediction would effect decision of users and may put them in danger.

### Environmental Issue

As for the environmental issues, if put model into business, there would be facilities to hold the compute system, therefore would use large amount of electricity. The government would bound company with law and strategy to solve that is renewable energy.

# References

1. Alghmgham, D. et al. (2019) ‘Autonomous Traffic Sign (ATSR) Detection and Recognition using Deep CNN’, *Procedia Computer Science,* 163(12), pp.264-274. Available at: <https://www.sciencedirect.com/science/article/pii/S1877050919321477> (Accessed: 6 November 2022).
2. Andrew G, H. et al. (2017) ‘MobileNets: Efficient convolutional neural networks for mobile vision applications’, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Available at: <https://arxiv.org/abs/1704.04861v1> (Accessed: 2 January 2023).
3. GOV.UK. (2018) *Data Protection.* Available at: <https://www.gov.uk/data-protection> (Accessed: 6 November 2022).
4. IEEE. (2020) *IEEE Code of Ethics.* Available at: <https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf> (Accessed: 6 November 2022).
5. Shustanoy, A. and Yakimov, P. (2017) ‘CNN Design for Real-Time Traffic Sign Recognition’, *Procedia Engineering,* 201(9), pp.718-725. Available at: <https://www.sciencedirect.com/science/article/pii/S1877705817341231> (Accessed: 6 November 2022).
6. Sandler, M. et al. (2018) ‘MobileNetV2: Inverted Residuals and Linear Bottlenecks’, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.* Available at: <https://arxiv.org/abs/1801.04381> (Accessed: 3 January 2023).
7. Sun, Y. and Ge, P. and Liu D. (2020) ‘Traffic Sign Detection and Recognition Based on Convolutional Neural Network’, *2019 Chinese Automation Congress (CAC)*, 2019, pp. 2851-2854. Available at: <https://ieeexplore.ieee.org/document/8997240/references> (Accessed: 6 November 2022).
8. Vennelakanti, A. et al. (2019) ‘Traffic Sign Detection and Recognition using a CNN Ensemble’, *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, pp. 1-4. Available at: <https://ieeexplore.ieee.org/document/8662019/references> (Accessed: 6 November 2022).
9. Wasif Arman, H. et al. (2021) ‘DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements’, *Expert Systems with Applications*, 168 (5), pp.114-127. Available at: <https://www.sciencedirect.com/science/article/pii/S0957417420311283> (Accessed: 6 November 2022).