UCCD3074 Deep Learning for Data Science
Group Assignment
Traffic Sign Classification

Research-based	☐	Application-based	√

Name	Eii Chee Hieng	Ong Chan Meng	Hazel Lim Benin
Programme	CS	CS	CS
ID	22ACB00282	21ACB06897	22ACB00107
Contribution	1/3	1/3	1/3

Chapter 1: Introduction

1.1 Problem Statement

Since the advent of self-driving cars, the ability of accurately identify and classify traffic signs has become increasingly important. Traffic sign classification (TSC) is a technique of recognizing the traffic signs according to their characteristics and it can be achieved using deep learning algorithms. However, there are several problems associated with TSC.

The first problem in TSC is the wide variety of traffic sign designs. The traffic signs have different characteristics and the design varies in each country. This diversity led to challenges of creating a model that can accurately identify them.

Besides that, environment conditions can significantly impact the performance of TSC system. Bad weather condition make the model difficult to capture the traffic signs, leading to misclassification. This might lead to serious impact such as accident if the system fails to identify the traffic signs, causing the autonomous self-driving car to make the wrong decision.

1.2 Motivation of the Project

The motivation for this project stems from the critical role that TSC plays in ensuring safety and functionality in autonomous driving technologies. As the world are adopting self-driving vehicles, advanced systems to interpret traffic signs accurately in real time is essential. Failing to recognize traffic signs can have serious consequences, such as car accidents.

While significant progress has been made in autonomous vehicle development, current TSC systems still struggle to meet the reliability and accuracy required for real-world implementation. Inconsistent lighting conditions, environmental factors, and variations in traffic signs across different countries create obstacles for these systems, leading to misclassifications or delays in processing traffic signs.

This project aims to address these challenges by developing a deep learning-based TSC system that improves on existing limitations. Accurate classification of traffic signs will not only enhance the safety of autonomous driving but also contribute to smoother traffic flow, fewer accidents, and an overall improvement in road safety. The success of this project could significantly reduce human errors on the road, reinforcing the reliability of autonomous systems and instilling confidence in the public's adoption of self-driving technologies.

1.3 Background of the Project

The field of TSC has evolved significantly with the advent of machine learning and deep learning technologies. In traditional TSC systems, hand-crafted features like colour histograms, edge detection, and shape descriptors were used to classify signs. However, these approaches had limitations in terms of generalizability, particularly when dealing with variations in lighting, weather, and different sign designs. These systems often struggled in complex real-world scenarios, leading to misclassifications.

With the Convolutional Neural Networks (CNNs), the field has seen a paradigm shift. CNNs have proven to be highly effective in image recognition tasks due to their ability to automatically extract features from images. This has significantly improved the performance of TSC systems, allowing to classify traffic signs more accurately across different conditions.

Despite these advancements, challenges remain. The accuracy of TSC models can be affected by poor-quality images and diversity of traffic signs across countries. This lack of generalizability limits the global scalability of TSC systems.

Moreover, the need for real-time processing adds another layer of complexity. Self-driving cars operate in dynamic environments, requiring the TSC system to process traffic sign images in real-time to make decisions almost instantaneously. Achieving high accuracy while maintaining low latency is crucial for the success of these systems.

1.4 Challenges of the Project

During the development of the system, several difficulties are faced. Addressing these issues was crucial to ensure the model could perform well in real-world scenarios.

First, the system failed to identify the traffic sign from an input image due to the low quality of the input image. Thus, various image processing techniques are applied to the input image to enhance the quality of the image.

Besides that, the imbalanced dataset is another challenge for certain traffic sign classes. A balanced dataset is important to produce a good model. However, some categories contained very less samples and could lead the model to be biased towards more frequent classes and struggling to appropriately classify less-represented signs. As shown in the figure 1.1, Traffic Sign Recogntion Database(TSRD) having some classes with two samples only. Thus, technique like data augmentation needed to handle the imbalance.

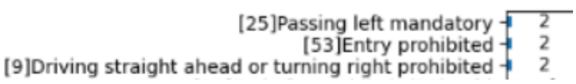

Figure 1.1: Classes with 2 samples

1.5 Project Scope

This project is to develop a TSC system using a deep learning algorithm, Convolutional Neural Network (CNN) model by using Python.

First, we get our dataset from Chinese Traffic Sign Database website [1]. The dataset consists of 58 categories and has divided into train and test sets which contain 4170 and 1994 images respectively. Before the training, several image processing techniques are performed to improve the contrast of image and increase the samples of dataset. Then, the model is then trained and validated to ensure accuracy and resilience. Next, a fine-tuning process is performed to achieve better performance. Finally, the performance of the trained model is tested using test datasets and compare the results with the training's results.

1.6 Objectives

The primary objective of this project is to develop an effective TSC system that can accurately categorize traffic signs from images. The system is going to enhance the road safety and support advance technology. The project aims to achieve several objectives:

- To apply suitable algorithms to classify traffic signs
- To develop a reliable system that has high accuracy of TSC
- To implement a system that can well perform under challenging conditions

1.7 Literature Reviews

You Only Look Once version 5 (YOLOv5)

YOLOv5 is an object detection algorithm that can process real-time images [2]. YOLOv5 uses three steps to process the images. First, it extracts the features from images, followed by aggregating these features, and then produce output.

According to [3], YOLOv5 takes an image and resizes it to 448x448 pixels before dividing it into a SxS grid. Each grid cell predicts the objects and their positions in the image. It creates a class probability map and bounding boxes around detected objects, allowing it to recognize many objects when passing through a single network.

The YOLOv5 model was trained using the German Traffic Sign Recognition Benchmark (GTSRB) dataset. It was trained with an input size of 160x160 pixels for 100 epochs and achieve excellent results in TSC. Table 1.1 shows the performance of the model.

Precision	93.7%
Mean Average Precision (mAP)	94.5%
Recall	93.8%
Minimum Miss Rate	6.2%

Table 1.1: Traffic sign classification results

In short, YOLOv5 can effectively balance speed and accuracy, enabling simultaneous recognition of many indicators without sacrificing detection precision.

Teacher and Student Model

Besides that, according to the study of [4], a lightweight CNN architecture which comprises two neural networks: teacher and student while teacher is a larger and more complex network than student. Therefore, the student model is intended to be lightweight and efficient, which is ideal for resource-constrained situations. During the training process, the teacher's knowledge is transferred to the student. This method allows the student model to perform effectively while requiring fewer parameters and computational power. The process is known as knowledge distillation. Figure 1.2 shows the flow of the teacher-student model.

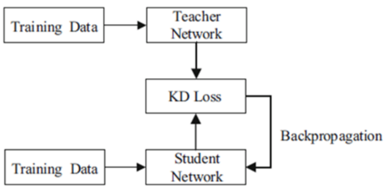

Figure 1.2: Flow of teacher-student model [5]

The teacher-student model is tested with two datasets, GTSRB and Belgian Traffic Sign Classification dataset (BTSC). The student model with 50% channel pruning achieve accuracy of 99.38% in GTSRB dataset and 98.89% in BTSC dataset.

Convolutional Neural Network

Convolutional Neural Network is one of the most renowned deep learning neural networks. CNN now provide a more scalable solution for tasks such as object recognition and image categorization[6]. One of the examples of image categorization is TSC.

Furthermore, the CNN architecture can be modified for more specific tasks by adjusting the number and types of layers, incorporating specialized layers, or using transfer learning. For example, paper [6] modified CNN by combining with residual models, which mitigate the Vanishing Gradient Problem.

In [7], a high recognition rate of 99% is achieved by a six layered CNN model in classifying Malaysia traffic signs. The German Traffic Sign Detection Benchmark (GTSDB) dataset demonstrated respectable results for traffic sign detection and recognition when the CNN model and SVM-based colour transformation combined, as suggested by [8].

2.0 System Design

System Block Diagram and Description

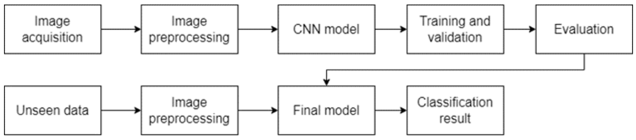

Figure 2.1 : Block Diagram of System

Figure 2.1 illustrates the block diagram of the TSC system, outlining the overall flow of the process. A more detailed workflow is presented in Figure 3.2 below. The acquired images, sourced from the TSRD, are divided into training, validation, and testing sets. Each of these sets undergoes a series of preprocessing steps before being input into the CNN model. The preprocessed training and validation sets are used to train and validate the CNN model, respectively. Once validated, the model is evaluated using the test set. If the evaluation results meet expectations, the model is saved for future use. The saved model is then integrated into a Graphical User Interface (GUI), which allows users to upload and classify images. These user-provided images, considered as unseen data, also undergo a series of preprocessing steps before being processed by the final model through the GUI.

System Flow Diagram and Description

First, research for traffic sign dataset is conducted and is obtained. Train, validation, and test sets are then created from the dataset. Lowering order bias requires train and validation set image scaling and shifting. Overfitting would occur if the network learned patterns based on data order. This isn't done for the test set because it won't be used for training or model evaluation.

Train and validation sets get one-hot encoding to transform labels into numerical data. The Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm normalizes and enhances all datasets. This strategy reduces photographic exposure effects like bright or dark traffic sign photos by increasing contrast.

After that, the normalized train set is augmented by randomly rotating images, translating them vertically or horizontally, shearing them, zooming in, and padding the border with the closest values using replicate padding mode. This approach fills empty pixels with values from the adjacent non-empty pixel to generate a new sample, expanding the train set size and distributing data more evenly across classes, especially those with few data samples. For imbalanced datasets with few samples in some classes, data augmentation is essential for pre-processing. Data augmentation reduces overfitting and improves model generalizability. Normalized validation and test sets validate and evaluate the training model. The evaluated model is stored locally. The model is then loaded into a GUI. Users can enter any traffic sign image and see the output in the GUI. Figure 3.3 illustrates GUI TSC results.

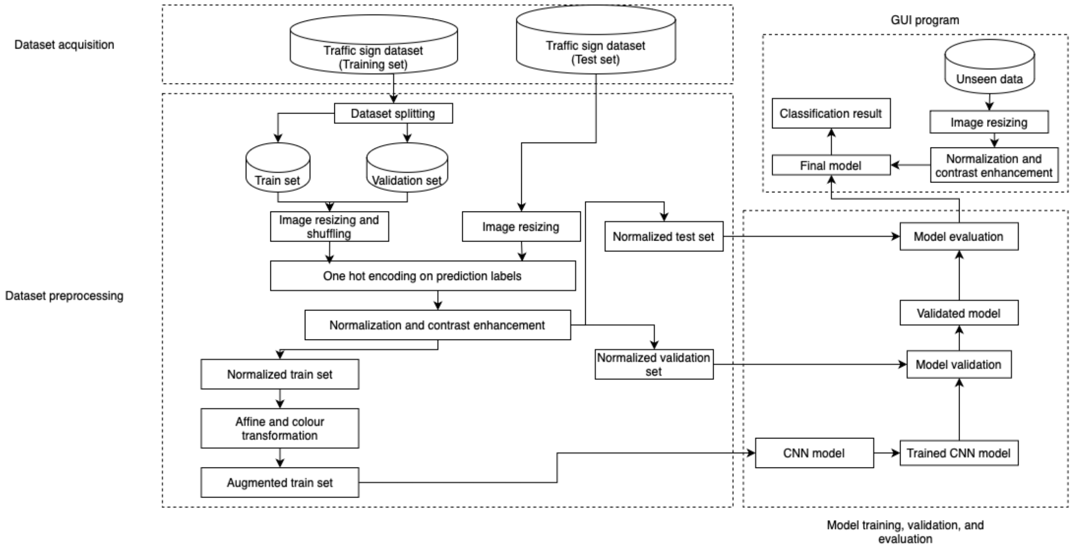

Figure 2.2: Flow of the system

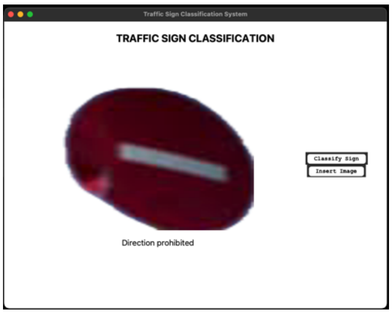

Figure 2.3: GUI

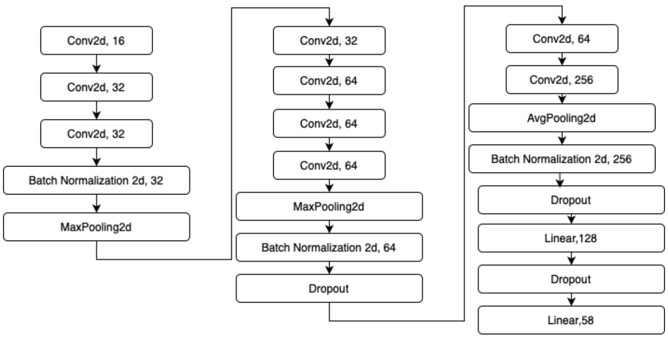

Figure 2.4: Layers of CNN model.

Figure 2.4 shows our CNN model consists of 20 layers. 6 types of layers were used to construct the model. Each type of layer performs different functionality.

1. Conv2D layer detects features from inputs.
2. MaxPooling2D layer reduces the spatial dimensions by taking maximum value.
3. Batch Normalization normalizes the output.
4. Dropout layer randomly drops units from network to reduce overfitting.
5. AveragePooling2D layer reduces the spatial dimensions by averaging values.
6. Linear layer connects every neuron from the previous layer to every neuron in the current layer.

3.0 Experiments and Evaluation

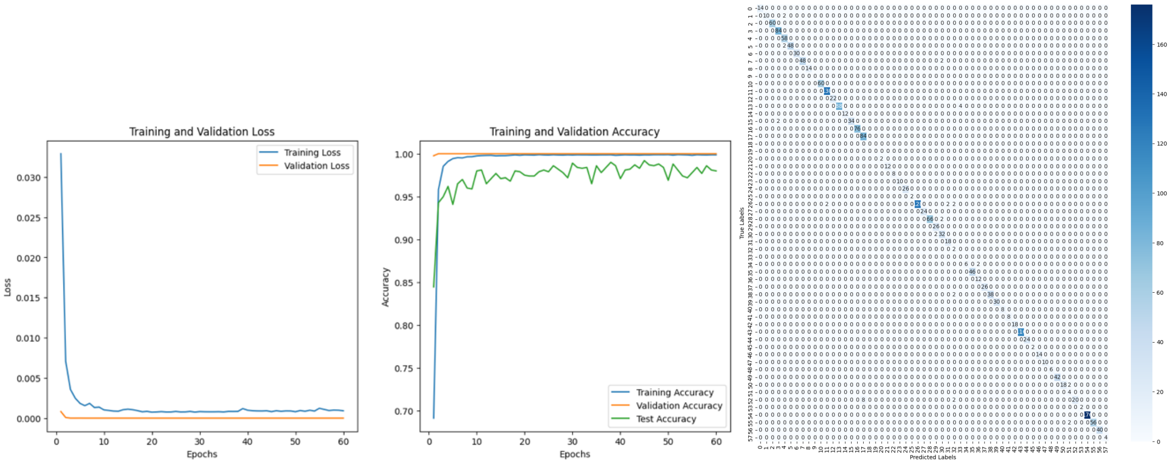

Figure 3.1: Learning curve of loss and accuracy & confusion matrix.

To achieve high accuracy of our model, the experiment has been performed using experience-based tuning to find the optimal hyperparameter setting as the grid search is very inefficient as the model is heavy requiring a lot of training time. Several image sizes, batch sizes, data augmentation transformations, learning rates, and gamma values were tested. The hyperparameters that yielded the highest accuracy are image size of 40x40, a batch size of 32, a learning rate of 0.001, and a gamma of 1 - 5e-5.

Figure 3.1 shows the learning curve of loss and accuracy of our TSC system showing our model is good fit as the test and validation curve are very close, the test curve showing very high accuracy. The confusion matrix showing most of samples are correctly predicted as most of numbers in not diagonal elements are small, this indicates the model have a high accuracy and does not overfit to training data.

	precision	recall	f1-score	support
micro avg	0.995803	0.951856	0.973333	1994.0
macro avg	0.928725	0.872620	0.895777	1994.0
weighted avg	0.998003	0.951856	0.972250	1994.0
samples avg	0.951856	0.951856	0.951856	1994.0

Figure 3.2: precision, recall, F1-score and support of TSRD test set.

Author	Algorithm	Accuracy (%)
Soni et al. (2019) [9]	LBP, HOG, PCA, SVM	84.44
Sapijaszko et al., (2019) [10]	DWT, DCT, MLP	94.9
X. R. Lim et al., (2023) [11]	Ensemble Learning	96.16
Our models	CNN	97.99

Table 3.1: Comparison results on TSRD dataset.

Figure 3.2 shows the micro average of precision is 0.9958, recall 0.9519, and F1-score of 0.9733, which is show the model having a strong overall performance. Table 3.1 compares the accuracy of various models on the TSRD dataset, highlighting that our model achieves the highest accuracy among these 4 models which is 97.99%.

4.0 Conclusion

In conclusion, this project develops a TSC system using CNN model that aims to solve various issues such as wide variation of traffic sign designs and the impact of environmental factors like poor lighting and weather conditions. It achieved a great performance at 98% of accuracy on test set by applying various preprocessing techniques including the data augmentation to solve the imbalance of dataset and some regularization techniques to reduce model overfitting on train set.

This project can be further improved by using several techniques including transfer learning method used in the teacher and student network, so the model can maintain high accuracy and reduce the computational power.

References

[1] https://nlpr.ia.ac.cn/, "Chinese Traffic Sign Database," National Nature Science Foundation of China(NSFC), [Online]. Available: https://nlpr.ia.ac.cn/PAL/TRAFFICDATA/RECOGNITION.HTML. [Accessed 28 August 2024].

[2] M. Karthi, V. Muthulakshmi, R. Priscilla, P. Praveen and K. Vanisri, "Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset," *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES),* pp. 1-6, 2021.

[3] O. Nacir, M. Amna, W. Imen and B. Hamdi, "Yolo V5 for Traffic Sign Recognition and Detection Using Transfer Learning," *2022 IEEE International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, pp. 1-4, 2022.

[4] W. A. Haque, S. Arefin, A. Shihavuddin and M. A. Hasan, "DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements," *Expert Systems with Applications,* vol. 168, no. 0957-4174, p. 114481, 2021.

[5] J. Zhang, W. Wang, C. Lu, J. Wang and A. Kumar, "Lightweight deep network for traffic sign classification," *Annals of Telecommunications,* vol. 75, 2019.

[6] J. C. Antony, G. M. K. Dheepan, V. K, V. Vikas, and V. Satyamitra, "Traffic sign recognition using CNN and Res-Net," *EAI Endorsed Transactions on Internet of Things*, vol. 10, Feb. 2024, doi: 10.4108/eetiot.5098.

[7] M. M. Lau, K. H. Lim, and A. A. Gopalai, "Malaysia traffic sign recognition with convolutional neural network," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, IEEE, Jul. 2015, pp. 1006–1010. doi: 10.1109/ICDSP.2015.7252029.

[8] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu, "Traffic sign detection based on convolutional neural networks," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Aug. 2013, pp. 1–7. doi: 10.1109/IJCNN.2013.6706811.

[9] D. Soni, R. K. Chaurasiya, and S. Agrawal, "Improving the classification accuracy of accurate traffic sign detection and recognition system using hog and LBP features and PCA-based Dimension Reduction," *SSRN Electronic Journal*, 2019. doi:10.2139/ssrn.3358756

[10] G. Sapijaszko, T. Alobaidi, and W. B. Mikhael, "Traffic sign recognition based on multilayer perceptron using DWT and DCT," *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2019. doi:10.1109/mwscas.2019.8884897

[11] X. R. Lim, C. P. Lee, K. M. Lim, and T. S. Ong, "Enhanced traffic sign recognition with ensemble learning," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, p. 33, Apr. 2023. doi:10.3390/jsan12020033