Faculty of Electronic and Computer Technology and Engineering

DEVELOPMENT OF ROAD SIGN DETECTION USING DEEP LEARNING

NUR FATIN NAZIHAH BINTI MOHD FADZIL

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2024

DEVELOPMENT OF ROAD SIGN DETECTION USING DEEP LEARNING

NUR FATIN NAZIHAH BINTI MOHD FADZIL

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering Technology (Computer Systems) with Honours

Faculty of Electronic and Computer Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN COMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Development of Road Sign Detection System using Deep Learning

Sesi Pengajian : 2023/2024

Saya Nur Fatin Nazihah binti Mohd Fadzil mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
✓	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap:
No 43A Lorong Perdana 1/3
Bandar Perdana 08000,
Sungai Petani, Kedah

Ts. AHMAD FAIRUZ BIN MUHAMMAD AMIN
Pensyarah Kanan
Jabatan Teknologi Kejuruteraan Elektronik dan Komputer
Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer
Universiti Teknikal Malaysia Melaka

Tarikh: 12 JANUARY 2024

Tarikh: 13 JANUARY 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development of Road Sign Detection using Deep Learning" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	Fatin Nazihah.
Student Name	:	NUR FATIN NAZIHAH BINTI MOHD FADZIL
Date	:	12 JANUARY 2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature	:	
Supervisor Name	:	AHMAD FAIRUZ BIN MUHAMMAD AMIN
Date	:	13 JANUARY 2024
Signature	:	
Co-Supervisor Name (if any)	:	
Date	:	

DEDICATION

I would like to say a thousand thanks to both my parents Mohd Fadzil bin Othman and Suhaila binti Ithnin @ Abdul Jalil who gave me so many words of encouragement as I completed my final year project. My Mother helps me a lot while I try to implement my project. She gave me advice on how I finish my final project on time without any pressure and stress. They also prepared a comfortable place for me to find ideas and inspiration to complete my assignments. Apart from that, let's not forget my colleague Muhammad Afiq bin Rushdan who gave a lot of opinions and ideas for me to improve my work to be better. Apart from that, my supervisor Ts. Ahmad Fairuz bin Muhammad Amin thank you for all the help and feedback throughout my inquiries about the project regardless of the time.

ABSTRACT

Road sign detection plays a crucial role in intelligent transportation systems, aiding drivers in making informed decisions and enhancing overall road safety. In this project, we propose a robust road sign detection and recognition framework using deep learning techniques. The objective is to develop an intelligent system that can accurately detect and classify road signs from real-time video streams or images captured by onboard cameras. The proposed framework leverages the power of deep convolutional neural networks (CNNs) to learn discriminative features from road sign images. The model is trained on a comprehensive dataset of annotated road signs to improve its detection accuracy through pre-processing techniques, data augmentation, and fine-tuning. In addition to detection, the framework incorporates a recognition module that utilizes deep learning algorithms to classify the detected road signs into their respective categories. This enables the system to provide additional contextual information to drivers, such as speed limits, warnings, and other regulatory signs. The proposed road sign detection and recognition framework holds significant potential for integration into intelligent driver assistance systems, autonomous vehicles, and smart city applications. Enhancing the perception capabilities of vehicles can contribute to safer roads and more efficient transportation systems.

ABSTRAK

Pengesanan tanda jalan memainkan peranan penting dalam sistem pengangkutan pintar, membantu pemandu dalam membuat keputusan termaklum dan meningkatkan keselamatan jalan raya secara keseluruhan. Dalam projek ini, kami mencadangkan rangka kerja pengesanan dan pengecaman tanda jalan yang teguh menggunakan teknik pembelajaran mendalam. Objektifnya adalah untuk membangunkan sistem pintar yang boleh mengesan dan mengklasifikasikan tanda jalan dengan tepat daripada strim video masa nyata atau imej yang ditangkap oleh kamera onboard. Rangka kerja yang dicadangkan memanfaatkan kuasa rangkaian neural convolutional dalam (CNN) untuk mempelajari ciri diskriminasi daripada imej papan tanda jalan. Melalui gabungan teknik pra-pemprosesan, penambahan data dan penalaan halus, model ini dilatih pada set data komprehensif tanda jalan beranotasi untuk meningkatkan ketepatan pengesanannya. Selain pengesanan, rangka kerja itu menggabungkan modul pengecaman yang menggunakan algoritma pembelajaran mendalam untuk mengklasifikasikan papan tanda jalan yang dikesan ke dalam kategori masing-masing. Ini membolehkan sistem menyediakan maklumat kontekstual tambahan kepada pemandu, seperti had laju, amaran dan tanda kawal selia yang lain. Rangka kerja pengesanan dan pengecaman tanda jalan yang dicadangkan mempunyai potensi besar untuk disepadukan ke dalam sistem bantuan pemandu pintar, kenderaan autonomi dan aplikasi bandar pintar. Meningkatkan keupayaan persepsi kenderaan, ia boleh menyumbang kepada jalan raya yang lebih selamat dan sistem pengangkutan yang lebih cekap.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts Ahmad Fairuz Bin Muhammad Amin for his precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my parents for the financial support which enables me to accomplish the project. Not forgetting my fellow colleague, Muhammad Afiq bin Rushdan for the willingness of sharing his thoughts and ideas regarding the project.

Finally, my highest appreciation goes to my parents, friends, and family members for their love and prayers during the period of my study. An honourable mention also goes to my mother for all the motivation and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Road sign detection and recognition is one important function of autonomous vehicles and has recently attracted much interest [1]. An autonomous car or a self-driving car (SDC) is one of the Intelligent Transportation Systems (ITS). The primary goal of the autonomous car system is to collect important data for drivers to decrease their efforts in driving safely. Drivers must be aware of a variety of factors such as the vehicle's speed and orientation, vehicle distance, passing automobiles, and potentially hazardous or unusual incidents out ahead. If the autonomous car system can collect this information in advance, it will reduce the difficulty of driving for drivers, which will also make it safer and easier [2]. Road sign detection is crucial for autonomous cars to effectively comprehend and respond to traffic signs. The two primary factors contributing to car crashes in Spain involve exceeding the speed limit and distractions in general. Therefore, the two main goals of this effort, which focuses on traffic sign recognization (TSR) for driving assistance, are to help drivers stay under the speed limit and prevent distractions while driving [3].

Computer vision algorithms are frequently used by road sign detection systems to analyze the images or video feeds that are being acquired by the cameras mounted on autonomous vehicles. This algorithm can detect and identify a variety of road signs, including yield signs, yield limits, stop signs, and other traffic signs [4]. Research interest in traffic sign recognition (TSR) has increased in recent years. The automatic detection and interpretation of traffic signs using computer vision techniques is known as traffic sign recognition (TSR). Traffic sign recognition (TSR) detects traffic signs' location from digital images or video frames, given a specific classification. The traffic sign recognition (TSR) methods make use of visual information such as the shape and colour of traffic signs.

1.2 Road Sign Detection Impact on Society and Global Issues

Deep learning systems that accurately detect road signs can help improve traffic safety. Giving them timely alerts and reminders about traffic laws, speed limits, hazardous curves, pedestrian crossings, and other important information, can assist drivers. As a result, there may be fewer crashes, injuries, and fatalities on the roadways. Detecting road signs also can help with effective traffic management. Traffic authorities can collect information on traffic patterns, congested areas, and compliance with traffic laws by correctly detecting road signs. Utilizing this data helps improve traffic flow, plan infrastructure upgrades, and reduce bottlenecks. Road sign detection using deep learning is essential for the development and use of autonomous vehicles. Computer vision systems are used by self-driving automobiles to accurately detect and respond to road signs. Autonomous vehicles can travel highways more safely and make wise decisions based on the identified indications by utilizing deep learning algorithms. Lastly, enhancing traffic sign identification can help reduce the negative effects of transportation on the environment. Vehicles can spend less time idle, decreasing emissions, by enhancing traffic flow and reducing congestion. The promotion of eco-friendly driving activities can also be aided by the reliable detection of traffic signs relating to such practices as speed restrictions for fuel-efficient driving.

1.3 Problem Statement

Traffic sign recognition (TSR) is one important system in autonomous vehicles because it detects traffic signs. This autonomous vehicle is important nowadays because the driver sometimes overlooks the signs because of distractions or lack of attention. At the same time, this traffic sign recognition (TSR) system is still in an immature phase with very limited ability. There are still some challenges as an object to recognize in natural ecosystems. Traffic sign recognization (TSR) usually includes shape-based and colour-based methods to recognize the sign [5]. Road sign detection is a critical task in intelligent transportation systems, enabling automated analysis of traffic regulations and assisting drivers in making informed decisions. However, accurately detecting road signs in real-world scenarios poses significant challenges due to variations in lighting conditions, occlusions, different sign sizes, and complex backgrounds. Traditional computer vision techniques struggle to handle these complexities effectively, leading to suboptimal performance and potential risks for drivers. The problem addressed in this project is to develop an accurate and robust road sign detection system using deep learning techniques [6]. The goal is to create an intelligent system capable of identifying road signs from real-time video streams or images captured by onboard cameras with high precision while considering various environmental factors and maintaining real-time processing capabilities. Existing road sign detection methods cannot often generalize well to diverse and challenging scenarios, leading to false detections or missed signs. Moreover, these methods may not efficiently handle changes in lighting conditions, occlusions, and variations in sign appearances caused by factors such as weather or degradation.

Therefore, a solution is needed that can effectively detect road signs under these challenging conditions, ensuring reliable performance in real-world applications. The proposed system aims to address the limitations of traditional techniques by leveraging the power of deep learning, specifically convolutional neural networks (CNNs). By training a CNN model on a comprehensive dataset of annotated road sign images and optimizing it to handle variations in appearance and environmental conditions, the system will be capable of accurate and robust road sign detection [7], [8]. The successful development of this road sign detection system will contribute to enhancing the performance of intelligent transportation systems, improving road safety, and providing valuable assistance to drivers. By enabling real-time and reliable detection of road signs, the system will assist drivers in complying with traffic regulations, reducing the risk of accidents, and promoting efficient and safe transportation. Overall, the project aims to address the problem of accurate and robust road sign detection using deep learning techniques, providing a solution that overcomes the limitations of existing methods and enhances the safety and efficiency of intelligent transportation systems.

1.4 Project Objective

The objective of this project is help to overcome the problem statement. The objectives are based on the problem statement stated as follows:

a) To develop a deep-learning system that is capable of reliably detecting and recognising various traffic signs, such as warning, yield, and speed limit signs.

b) To recognise the pattern of road signs using Python.

c) To analyse the performance and accuracy of several proposed images with the test of the dataset.

1.5 Scope of Project

The scope of this project are as follows:

a) This project will focus on developing deep-learning algorithms for road sign detection. Thus, pre-processing steps in pre-processing the raw image input will be implemented in this project. However, that dataset for deep-learning training and testing will be realized by using the dataset from predetermined libraries from similar applications.

b) The outcome of this bachelor's degree final year project is to detect road sign images using deep learning and to present the recognition accuracy across several deep learning algorithms.

c) In this project there are 43 classes of dataset that use such as regulatory sign, warning sign and guide sign.

1.6 Thesis Organization

The introduction of the project includes the problem, objectives, job scope, and limitations covered in Chapter 1. Chapter 1 is more detailed and explains the background of the project with its related real-life problems. The theoretical part of this thesis takes place in Chapter 2. Starting with explaining more specifically about road sign detection systems. This part shows related works and studies for this project and ways to improve the system efficiency using deep learning methods. After that, Chapter 3 describes the methodology for developing a road sign detection system using deep learning such as using Python language. Next chapter 4 is the result and analysis of the project and lastly, chapter 5 conclusion of all the progress made in this project.

d)

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Nowadays, traffic sign detection is the most important system in autonomous vehicles. This literature review provides research on traffic sign detection using many methods such as traditional traffic sign detection solution method, artificial neural network (ANN), and convolutional neural network (CNN). After that, explain the related work of development of the road sign detection system.

2.2 Traffic Sign Detection

Traffic sign detection is researched to identify areas with or without traffic signs in complicated scene photos. The goal of traffic sign detection is to enable intelligent transportation systems to analyse and understand the traffic environment, providing valuable information to drivers or autonomous vehicles. Traffic sign detection is also used to extract the unique features represented by traffic sign patterns [5]. Traffic sign detection techniques are essentially divided into two groups or categories which are traditional methods and other one is linked with deep learning methods.

2.3 Traditional Traffic Sign Detection Solution Method

Computer vision techniques are used in traditional traffic sign detection to identify and locate traffic signs in pictures and movies. These techniques frequently involve several phases, such as feature extraction, pre-processing, and classification. The detection of traffic signs using traditional techniques has performed well in a range of circumstances. They may be constrained, though, by their reliance on hand-crafted features and their inability to adapt to changes in perspective, lighting, or occlusion [9]. It's crucial to remember that, despite the widespread usage of these traditional computer vision techniques in the past, more recent methods that make use of deep learning and convolutional neural networks (CNNs) have proven significant improvements in the accuracy and resilience of traffic sign detection.

2.3.1 Using Feature Extraction Methods

Feature extraction methods are an important step in many machine learning and computer vision applications, including object detection, recognition, and classification. The purpose of feature extraction is to identify relevant information from the input data that may be utilized to differentiate between various classes or categories. People often considered colour and shape features to complete traffic-sign identification and classification tasks. Using colour-based segmentation is one common method for traffic sign detection [10]. To do this, the input image must be segmented according to the traffic sign colour, which typically is known in advance For instance, red signs can be identified through the image's red channel's thresholding, while blue signs can be identified using the image's blue channel's thresholding. In regards to colour features, instead of using RGB (Red, Green, Blue), the images were converted to other colour spaces like HSV (Hue, Saturation, Values) [11].

2.3.2 Using Machine Learning

For classifying traffic signs, a variety of machine learning methods were used, including ensemble classifiers, support vector machines (SVM), random forests, decision trees, and neural networks. Data is used by machine learning algorithms for training, validating, and testing. The validation and testing data are used to evaluate the model's performance generalization, while the training data is used to fine-tune the model's parameters and discover underlying patterns. Machine learning techniques typically pick out specific visual characteristics and use them to categorize different types of traffic signals. The distinct characteristics include Histogram of Gradient (HOG), Scale Invariance Feature (SIFT), Haar-life features, and others [11]. A collection of non-pruned random decision trees that were all constructed using random training data forms the ensemble method known as Random Forest. The output of classification is produced by the majority of votes cast among all decision trees [12]. Machine language algorithms offered a multitude of advantages for classifying traffic signs but they were unable to handle elements like different image sizes and aspect ratios, which had to be finished manually. The features generation procedure was therefore usually difficult and susceptible to errors [13].

2.4 Artificial Neural Network (ANN)

The term artificial neural network (ANN) describes the connections present between the images in each system's different layers. The first layer of this system consists of input images that are used to transmit data through synapses to the second layer, and then further synapses move data to the third layer, which is the output image. In general, more complicated systems have more layers with a greater number of output and input layers. Weights are variables that synapse holds and use to alter data during calculations. A neutral network is an artificial model of the road sign [14].

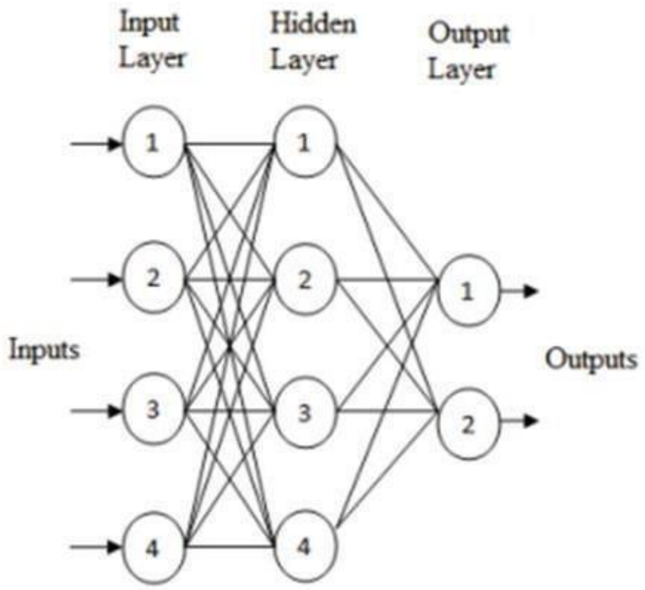


Figure 2.1 Basic Diagram of Artificial Neural Network. [14]

2.5 Convolutional Neural Network (CNN)

In image and video recognition applications, Convolutional Neural Networks (CNNs) is a type of neural network that is commonly used. Multilayer perceptrons, which are fully connected, are found in CNNs. In a multilayer perceptron, completely related implies that every neuron in one layer is coupled to each neuron in the next layer. Overfitting of the data can be efficiently avoided using this architecture. Their specific applications include natural language processing, image, and video recognization, medical image analysis and recognition, and others [15].

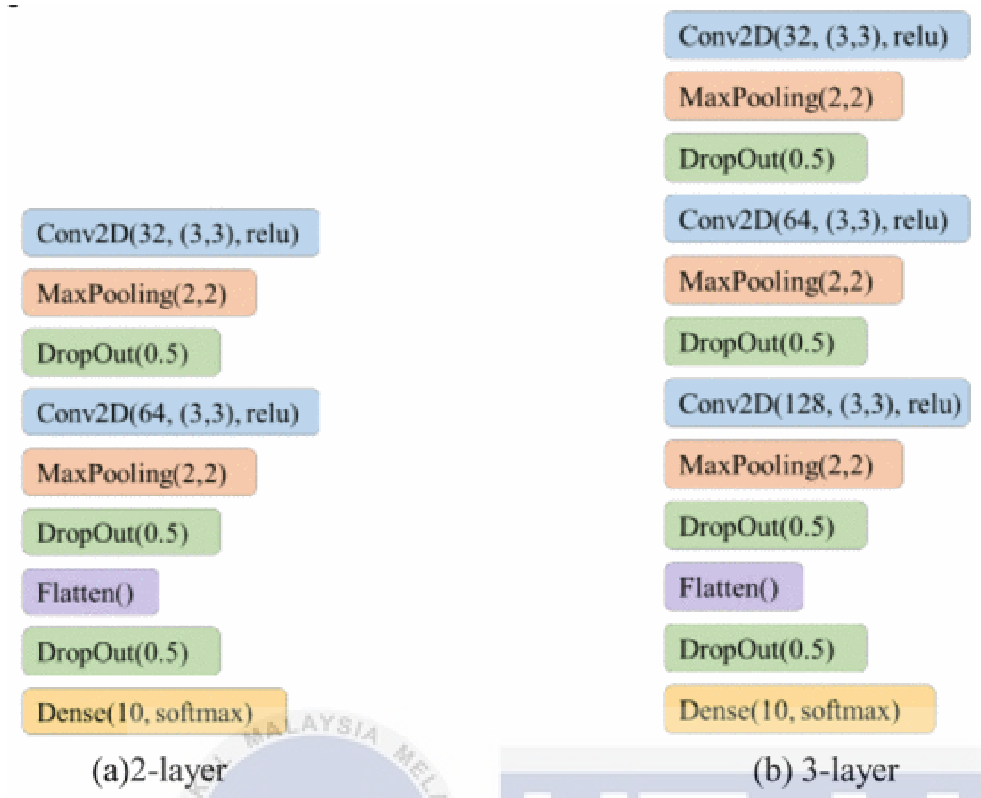


Figure 2.2 Structure of two CNN network. [16]

There are two different types of fundamental CNN networks. Both networks used Dropout and Maxpooling layers. The first CNN network contains two convolutional layers in Figure 2.2(a), while the second CNN network has three convolutional layers in Figure 2.2(b). This is the sole difference.

2.5.1 Convolutional Layer

A convolutional layer is the most important building block of a convolutional neural network (CNN). To build a series of feature maps that highlight different aspects of the input, a variety of filters are applied to the input data, which frequently creates an image. Each filter is a tiny scaled matrix that moves over the input information and performs a dot product at every location to produce a scalar value that indicates the existence of a certain feature. An initial process will end up in a 2D feature map [17]. Finally, a convolutional neural network (CNN) was used to classify traffic signs. The following three hyperparameters control the convolutional layer's output volume: Stride, depth, and zero-padding. The number of neurons in a layer connected to the same area of the input volume depends on the depth of the output volume. Stride controls the distribution of columns' depths around the input's physical dimensions, which are its width and height. The worth of stride (S) must be any integer bigger than zero [18]. The length of S is typically less than three in real life. The output volume will have reduced spatial dimensions as the stride length increases due to less receptive fields overlapping. Control with zero padding the volume of the output in terms of space. To figure out the number of neurons that can fit in a specific volume, use equation (2.1):

$$n = \frac{W - K + 2P}{S} + 1 \tag{2.1}$$

where W is the size of the input volume. The convolutional layer neurons' kernel size is indicated by the letter K. While S and P stand for the stride length and zero-padding amounts respectively. Usually, we set the zero-padding as P=(K-1)/2 while S =1, which equalizes the spatial size of the input and output volumes.[18]

2.5.2 MaxPooling Layer

Convolutional neural networks (CNNs) use a layer called pooling to reduce the spatial dimension of feature maps while retaining the most crucial data. The pooling layer provides a pooling function to each local region of the feature map and operates on each feature map independently [19]. Maximum pooling and average pooling are the two most often utilized pooling operations in CNNs. While average pooling uses the average value for each local region on the feature map, max pooling uses the maximum value [20]. The most popular pooling variant has a stride of 2 down samples and a filter size of 2x2. The volume's depth remains unchanged.

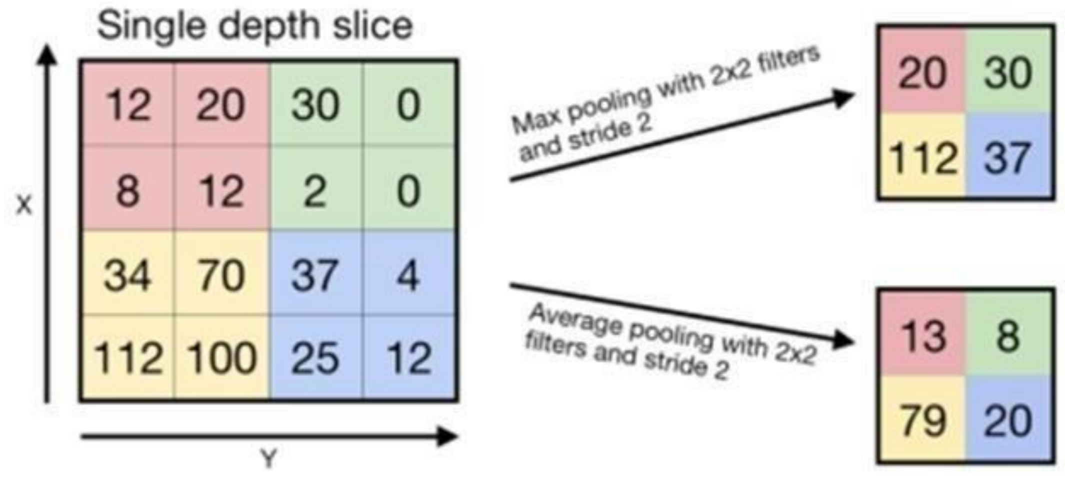


Figure 2.3Max and average pooling with a filter of size 2x2 and stride 2. [20]

There have also been other pooling functions proposed in the literature, such as L2 pooling and stochastic pooling. In CNNs, the pooling layer has two primary functions. First, it helps in reducing the spatial dimensionality of the feature maps, lowering the model's parameter count, and aiding in the avoidance of overfitting. Second, it aids in strengthening the model's resistance to changes in the input data such as translation, rotation, and scaling [21].

2.5.3 DropOut Layer

The dropout layer is a layer that removes nodes from the neutral network at random to prevent overfitting. Here, we choose a ratio of 0.5, which indicates that 50% of the nodes in each dropout layer will be dropped out [22]. In a nutshell, Dropout replicates the parallel training of numerous neural networks that have different topologies. The neural network is effectively kept from growing overly dependent on any one particular set of input properties or neurons. The dropout layer is disabled during inference or testing, and the entire network is used for making predictions. To maintain the overall signal strength, the weights of the neurons are scaled up by the dropout rate. The dropout rate is a hyperparameter that controls the percentage of input neurons that will be lost at random. Depending on the size of the training dataset and the complexity of the neural network, typical dropout rates range from 0.1 to 0.5. Overall, the Dropout layer is an effective technique for avoiding overfitting in neural networks and has been shown to improve CNN and other deep learning models' performance on a variety of tasks [17].

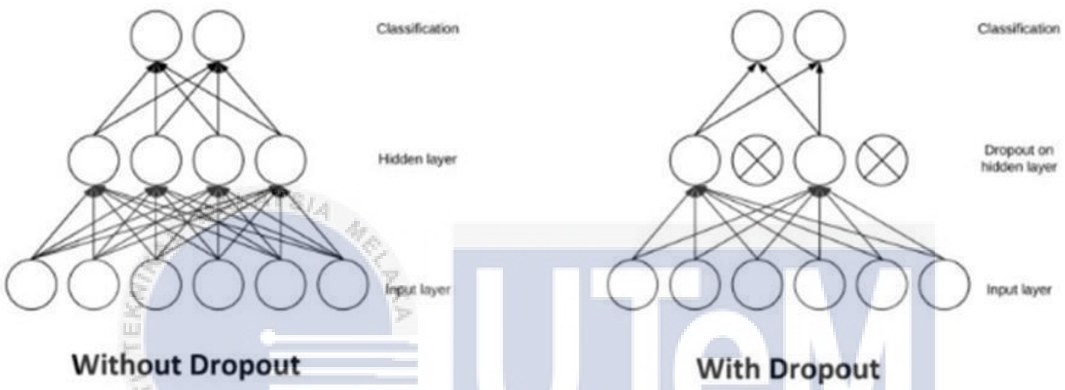


Figure 2.4 Show the different pictures of the circle Without DropOut and With DropOut [17]

2.5.4 Flatten Layer

Convolutional neural networks (CNNs) frequently include the Flatten layer, which is typically used to transform the output of the previous convolutional and pooling layers into a 1-dimensional vector. In the network, this allows the data to be processed and fed into subsequent fully connected levels or other kinds of levels. In essence "flattening" the spatial dimensions, the flatten layer takes the multi-dimensional output from the preceding layer and flattens it into a single long vector. For instance, the flatten layer will convert the output of the preceding layer into a 1-dimensional vector of size (batch_size, height*width*channels) if it contains dimensions (batch_size, height, width, and channels).

The flatten layer's function is to convert the input's spatial representation such as an image into a format that can be fed into fully linked layers, which need a 1-dimensional input. The flattened vector can then be processed by fully connected layers to carry out operations like classification and regression [23].

2.5.5 Dense Layer

A dense layer, sometimes referred to as a completely linked layer, is a soft of layer in convolutional neural networks (CNNs) that connects every neuron from the preceding layer to every neuron in the next layer. In a CNN design, dense layers are often applied at the very end to carry out classification, regression, or other more complex operations [24]. In a CNN architecture, convolutional and pooling layers are often placed before dense layers. In the input data, these previous layers extract regional information and record spatial hierarchies. After being flattened, the outputs from these layers are fed into one or more dense layers, which are used to capture high-level abstractions and global relationships. Convolutional neural networks (CNNs) may efficiently train hierarchical reprentations from the input data and produce predictions based on these learned features by combining convolutional layers for feature extraction and dense layers for high-level processing [18].

2.6 Model Convolutional Neural Network (CNN)

Blocks of convolutional layers and a pooling of 14 layers compose the conventional CNN design, which is followed by a fully connected layer and a Softmax layer, AlexNet, VGGNet, LeNet, and NiN are a few examples of such CNN models [16].

2.6.1 LeNet

Convolutional neutral network (CNN) architecture LeNet, commonly referred to as LeNet-5, was created in the 1990s by Yann LeCun and others. It is one of the original CNN models and was created primarily for recognizing handwritten digits. LeNet was crucial in popularizing CNNs and in laying the groundwork for contemporary deep learning. LeNet accepts grayscale photos as input, ofter in the form of 32x32 pixel images. LeNet uses two layers of convolution. Each convolutional layer performs convolutions on the input image and creates feature maps by applying a series of teachable filters. Local patterns and image features are captured by the filters [23]. LeNet employs the sigmoid activation function throughout the network. The sigmoid function squashes the output of each neuron between 0 and 1, enabling non-linear transformations and capturing non-linear relationships in the data. LeNet's architectural principles have had an ongoing impact on the development of CNNs, despite the fact that it was created primarily for digit recognition. LeNet's ideas and revelations have impacted later CNN models, and many modern designs have developed from its fundamental ideas.

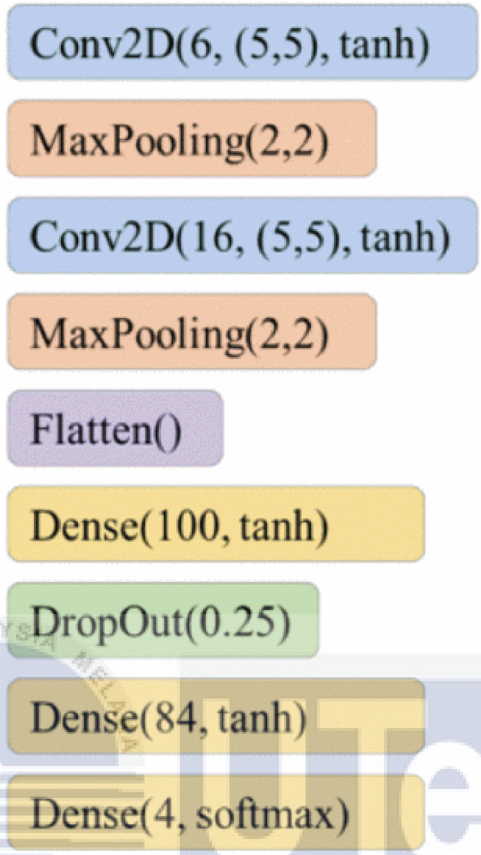


Figure 2.5 Structure of LeNet [16].

Three completely connected layers, comprising one output layer, one flattening layer, two convolution layers, two pooling layers, and one input layer are all present. In Figure 2.4, this is shown. The input layer's purpose is to accept the preprocessed image, which includes the various traffic sign maps. The extraction of features takes place on the first convolution layer. The convolution kernel consists of six different types of 5 by 5 matrices. For feature dimension reduction or down sampling, the initial max-pooling layer is utilized. Pooling is a type of nonlinear down-sampling that, by lowering the network parameters, can reduce the amount of calculation and, to a certain extent, control over-fitting. Similar to the first convolution layer, the second convolution layer serves the same purpose. High-dimensional data can be reduced to one dimension via the flattening layer. The layer with complete connectivity is in charge of producing categorization outcomes [24].

2.6.2 AlexNet

Convolutional neural network (CNN) architecture AlexNet attracted a lot of interest and advanced the science of deep learning. It was created by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, and it won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), significantly increasing the accuracy of image classification. AlexNet accepts RGB images as input that are typically 227x227 pixels in size. There are five convolutional layers in the architecture. These layers use a rectified linear unit (ReLU) activation function and narrow receptive fields (such as 3x3 and 5x5 filters) with a stride of 1[25]. The filters learn hierarchical characteristics while capturing local patterns. Local response normalization is incorporated by AlexNet between the convolutional and pooling layers. This normalizing method improves generalization by enhancing the contrast between various features. The network uses ReLU activation functions to provide non-linearity and allow the model to capture intricate correlations in the data. Convolutional and pooling layers are followed by three fully connected layers in the AlexNet architecture. The final completely connected layer contains 1,000 neurons, which corresponds to the number of classes in the ImageNet dataset at the time, while the previous two fully connected layers each had 4,096 neurons.[26]

2.6.3 VGGNet

Its object recognition algorithm was created and trained by Oxford's well-known VGG (Visual Geometry Group), which performed far better than the ImageNet dataset. It is popular not only because it performs well but also because the Oxford researchers made the structure and weights of the trained network available online. A 19-layer deep convolutional neural network called VGG-19. The VGG neural network architecture placed first in the image localization challenge and second in the image classification problem at the 2014 ILSVRC. Localization is the process of locating a certain object within a bounding box of an image. The process of identifying the nature of the object in an image is referred to as "classification." This suggests the presence of a category label, such "dog" or "vehicle."

The input for the VGG neural network is an RGB image of 224x224 pixels. The middle 224x224 patch in each image was cropped away by the authors in order to maintain a constant input image size for the ImageNet competition[15]. The smallest size possible while still catching left/right and up/down is a 33 receptive field, which is what convolutional layers have. Before the input goes through a ReLU unit, there are extra convolution filters that modify it linearly. After convolution, the stride is fixed to 1 pixels to maintain spatial resolution. ReLU, a creation of AlexNet that dramatically decreases training time, is used in all of VGG's hidden layers. VGG does not employ Local Response Normalization (LRN) since it extends training time and memory requirements without enhancing accuracy.

Instead of using large receptive fields like AlexNet, VGG uses relatively small ones. Consequently, it employs 33 with a 1 stride. With three ReLU units instead of just one, the decision function is more discriminative. Less parameters are also used (27 times as many channels compared to 49 times as many channels in AlexNet). To make the decision function more non-linear, VGG employs 11 convolutional layers without altering the receptive fields. Due to the convolution filters' tiny size, the -VGG model can contain a large number of weight layers; of course, more layers equal higher performance. This characteristic, meanwhile, is not exceptional. Convolutional neural network architectures have been around for a long, including the VGG design. It was created as a consequence of study on how to densify particular networks. The network uses minuscule 3x3 filters. Aside from that, the network is notable for its simplicity, with two additional components being two pooling layers and a fully linked layer. The VGG net deep learning model is one of the most popular image recognition algorithms available today[17].

2.6.4 ResNet

For computer vision applications, the Residual Network (ResNet) deep learning model is used. It is a design for a convolutional neural network (CNN) that can support a large number of convolutional layers—possibly thousands. Performance was negatively impacted by the limited number of layers that earlier CNN architectures could support. However, as more layers were added, researchers ran into the "vanishing gradient" problem. The backpropagation approach used to train neural networks lowers the loss function and determines the weights that minimize it by using gradient descent. If there are too many layers, the gradient will eventually become so small that it "disappears". With each successive layer, performance will likewise become saturated or degrade [25]. The ResNet "skip connections" function presents a fresh approach to the vanishing gradient problem. Convolutional layers that are initially inactive (many identity mappings; ResNet) are stacked, skipped, and the activations from the previous layer are recycled. Skipping speeds up the initial training process by reducing the number of layers in the network. After the network has been retrained with all layers expanded, the remaining parts, also known as the residual parts, are free to explore more of the feature space of the input image. With batch normalization and nonlinearity in between, the majority of ResNet models skip two or three layers at once. HighwayNets, a kind of more complex ResNet architecture, have the ability to learn "skip weights," which determine how many layers to skip on the fly.

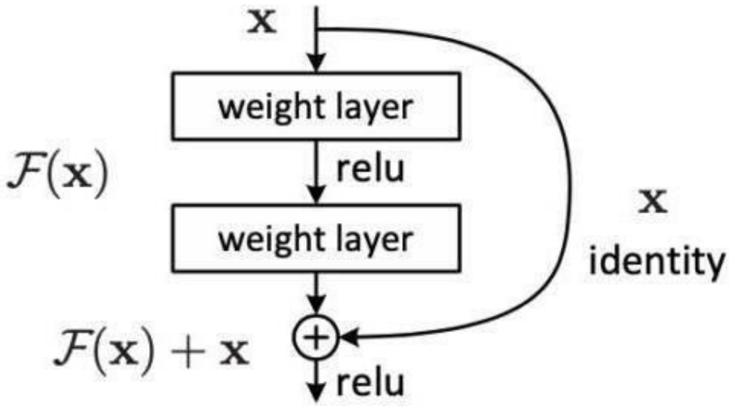


Figure 2.6 Residual learning a building block [25]

A typical residual block is depicted in the image above. Python code for this can be written as output = F(x) + x, where x is an input to the residual block and an output from the previous layer, and F(x) is a convolutional neural network (CNN) block. The gradient flow during backpropagation is slowed down using this strategy, allowing the network to scale to 50, 100, or even 150 layers. Skipping a connection does not increase the network's processing load. In many other neural network architectures, such as UNet and Recurrent Neural Networks (RNN), this technique of adding the input of the previous layer to the output of a following layer is increasingly widely used [25].

2.6.5 DenseNet

Another well-liked ResNet variant is DenseNet, which makes additional connections in an effort to address the problem of disappearing gradients. By physically connecting each layer to every other layer, the creators of DenseNet made sure that there was a maximum amount of information flowing across the network layers [24]. By enabling each layer to receive extra inputs from its preceding layers and transmit the feature map to following layers, this model keeps the feed-forward capabilities. Figure 2.6 is an example using the model:

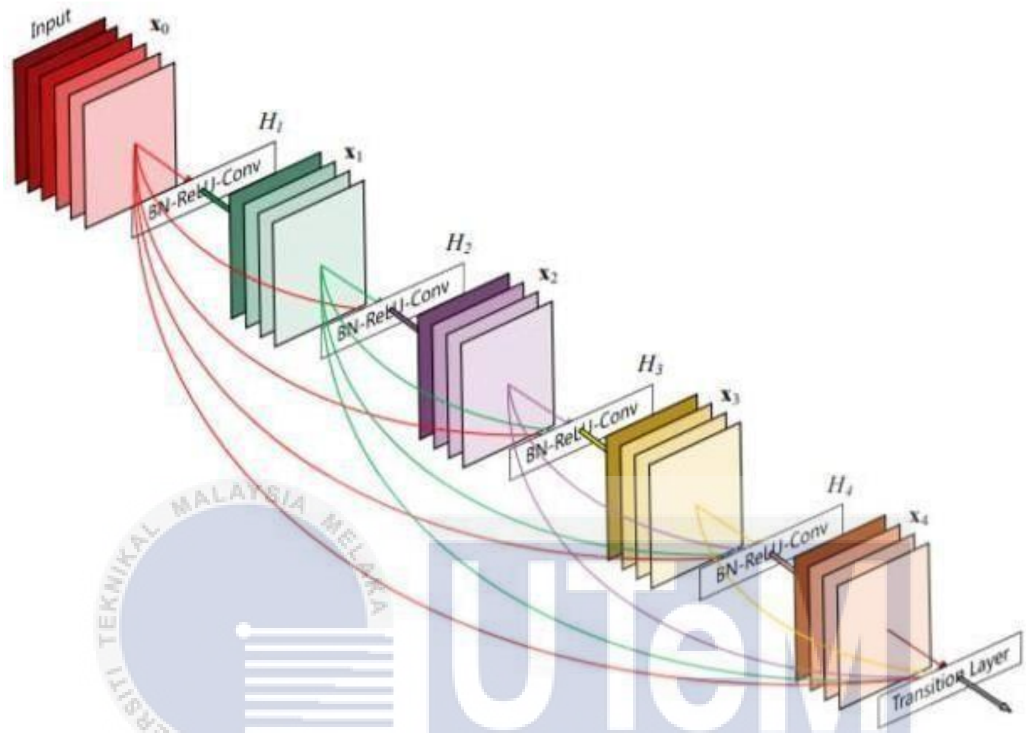


Figure 2.7 The model of DenseNet [24]

The inventors of DenseNet assert that the solution not only fixes the vanishing gradient issue but also permits feature reuse in the network. As a result, this method uses fewer parameters than traditional networks, where each layer acts as a separate state, reading from the layer before it and writing to the layer after it. The common convolution network model change the state and transmits crucial information[24]. Contrarily, the DenseNet model makes a distinct distinction between information that has already been present and new information. A final classifier uses all of the network's feature maps to generate judgments, allowing for more effective parameter usage and information flow. Thus, it is simpler to train the network.

2.7 Table of Comparison

No	Author	Title	Application
1.	Djebbara Yasmina, Rebai Karima, Azouaoui Ouahiba (2018)	Traffic signs recognition with deep learning	This project uses the convolutional neural network (CNN) application to detect traffic signs. It also presented a technique that extracts a deep representation of traffic signs using a modified LeNet-5 network to carry out the recognition.
2.	Gangyi Wang, GuangHui Ren, Taifan Quan (2013)	A traffic sign detection method with high accuracy and efficiency	This paper presents the provided an innovative method for detecting prohibitory traffic signs. These techniques make use of both colour and form properties, greatly increasing the detection efficiency and accuracy.
3.	Zhiyong Huang, Yuanlong Yu* and Shaozhen Ye (2014)	Extreme learning machine-based traffic sign detection	An extreme learning machine (ELM) was applied in this research for the method to detect traffic signs. This approach primarily uses 5 category ELM classifiers trained using HOG features, as well as 9 category ELM classifiers learned

			using HOG features and colour histograms.
4.	Amol Jayant Kale, Prof.R.C. Mahajan (2015)	A road sign detection and the recognition for driver assistance systems	The paper discusses the development of a road sign detection and recognition system for Driver Assistance Systems (DAS) to help drivers avoid accidents by recognizing road signs in traffic scene images. The system uses a YCBCR color space and an artificial neural network for classification, addressing challenges in constructing characteristic patterns.
5.	Ahmed Hechri, Abdellatif Mtibaa (2012)	Automatic Detection and Recognition of Road Sign for Driver Assistance System.	An effective method for recognizing and detecting road signs has been given forth in this research and tested on the actual video. During the detecting stage, a reliable way of YCBCR colour space is used for colour segmentation. For the shape classification of all prospective road signs, a template-matching technique has been developed.

6.	Zumra Malik and Imran Siddiqi (2014)	Detection and Recognition of Traffic Signs from Road Scene Images	This work provided a traffic sign detection and recognition methodology that is effective and efficient and insensitive to variations in lighting, scale, and viewing angle. The HSV colour space is used for the colour segmentation, and then the Hough transform is used for the shape analysis of potential regions.
7.	Saturnino Maldonado-Bascon, Sergio Lafuente-Arroyo, Pedro Gil-Jimenez, Hilario Gomez-Moreno and Francisco Lopez-Ferreras (2007)	Road Sign Detection and Recognition Based on Support Vector Machines	This article uses two main modules a first module for form classification based on linear SVMs and a second one that was produced using Gaussian kernels for recognizing the inner area. These two elementary modules have been developed based on the capability of SVMs as an innovative method of pattern recognition.
8.	Wang Canyong (2018)	Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning	The fundamental contribution of this paper is that it proposes a new framework for traffic sign detection and recognition based on the SSD

			method, guided by a fully convolutional network.
9.	Wen Jia Kuo and Chien Chung Lin (2003)	Two-stage Road Sign Detection and Recognition	This paper provides a method for recognizing and detecting traffic signs that makes use of a two-stage categorization process. The exact position of the road sign in the image is determined during the detection phase using corner detection, projection, Hough transformation, and the geometric features of road traffic signs.
10.	J.Stallkamp, M.Schlipsing, J.Salmen, C.Igel (2012)	Man vs Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition.	The performance of modern machine learning algorithms and humans at recognizing traffic signs in detail. The best person in the human performance experiment attained a nearly flawless accuracy of 99.22%, but the best machine learning strategy, a committee of convolutional neural networks, outperformed it in this difficult challenge with a 99.46% right classification rate.

11.	Xuehong Mao, Samer Hijazi, Raul Casas, Piyush Kaul, Rishi Kumar, and Chris Rowen (2016)	Hierarchical CNN for Traffic Sign Recognition	This paper used the hierarchical CNN (HCNN) application to solve a complicated problem by partitioning it into multiple easier sub-problem and distributing the effort of solving it according to their difficulty. This work also applies the HCNN to other applications and investigates the recursive multi-level classification.
12.	Erik Bochinski, Tobias Senst, Thomas Sikora (2017)	Hyper-Parameter Optimization for Convolutional Neural Network Committees Based On Evolutionary Algorithms	This experiment uses a convolutional neural network (CNN) to optimize the topology of a CNN based on committee performance rather than individual accuracy. To support the main goal, it also developed an evolutionary algorithm-based hyper-parameter optimization approach and a novel fitness function.
13.	Jianping Wu, Maoxin Si, Fangyong Tan, Caidong Gu (2009)	Real-Time Automatic Road Sign Detection	This project developed and implemented an algorithm to recognize and identify all general-purpose road signs on Chinese highways by employing a real-time

			circle detection algorithm to extract regions of interest (ROI) and implementing it in a Visual C++ application.
14.	Johannes Stallkamp, Marc Schlipsing, Jan Salmen (2011)	The German Traffic Sign Recognition Benchmark: A multi-class classification competition	The project of automatic traffic sign recognition using convolutional neural network (CNN), which is necessary for advanced driver assistance systems challenge. It also represents a tough real-world computer vision and pattern recognition challenge, demonstrating the performance of advanced machine learning algorithms in the difficult task of traffic sign recognition.
15.	Bhogadi Sreeja, Sruthila Bokka, Giddi Shravya, Katari Sri Vidya Vardini (2022)	Traffic Sign Detection Using Transfer Learning and A Comparison Between Different Techniques	This research compares multiple traffic sign prediction methods such as Local Binary Pattern (LBP), Convolutional Neural Network (CNN), and Transfer Learning (TL). This paper also claims that the transfer learning strategy has a high accuracy of 98%, which is a 3% improvement over the CNN

			approach's accuracy of 95%, and that the Local Binary pattern has a high accuracy of 96% when combined with CNN.
16.	Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, Shimin Hu (2016)	Traffic-Sign Detection and Classification in The Wild	This paper shows how a resilient end-to-end convolutional neural network (CNN) application identifies and classifies traffic signs in realistic real-world images. This work also established a new benchmark for simultaneously identifying and classifying traffic signs in images that are more variable and include significantly smaller signs.
17.	Andreas Mogelmose, Mohan Manubhai Trivedi, Thomas B. Moeslund (2012)	Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey	This work analyzes the traffic sign detection literature, focusing on detection systems for traffic sign recognition (TSR) applications for driver assistance. This paper also discusses the next prospects of TSR application research, such as the integration of context and localization in traffic sign detection.

18.	Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, Michael Unser (2017)	Deep Convolutional Neural Network for Inverse Problems in Imaging	A particular deep convolutional neural network (CNN)-based technique for addressing ill-posed inverse problems was proposed in this research. It also presented a deep convolutional network for inverse problems with a focus on biomedical imaging that uses the FBPConvNet approach, which combines the filtered back projection (FBP) with a multiresolution CNN with a U-net structure and residual learning.
19.	Rabia Malik, Javaid Khurshid, Sana Nazir Ahmad (2007)	Road Design Detection and Recognition Using Colour Segmentation, Shape Analysis and Template Matching	The paper proposes the creation of a system for detecting and recognising road signs. It also includes a revised version of the fuzzy shape detector and recognition module, which use template matching to recognise rotated and affine distorted road signs. This study claims an overall system accuracy of 86% after testing 100 photos, with an additional accuracy of 96% for colour

			segmentation with filtration and 94% for shape detection.
20.	Mucahit Karaduman, Haluk Eren (2017)	Deep Learning based Traffic Direction Sign Detection and Determining Driving Style	The proposed methods aim to contribute to the development of advanced driver assistance systems (ADAS) by combining two concurrently running algorithms that determine driver manoeuvres and a deep learning-based algorithm that detects traffic direction signs using a convolutional neural network (CNN) application. Furthermore, it employs a smartphone sensor application for data collection to evaluate the driving style of drivers.
21.	Hossain M, Hassan M, Ameer Ali M, Kabir M, Shawkat Ali A (2010)	Automatic Detection and Recognition of Traffic Signs	The paper presents a new algorithm for automatic road sign detection and recognition, utilizing color segmentation in RGB and Hu moment invariants, and a neural network for classification. The results show superior performance with low computational time complexity, highlighting the

			importance of safe travel and the model's stages.
22.	Megalingam R, Thanigundala K, Musani S, Nidamanuru H, Gadde L (2023)	Indian traffic sign detection and recognition using deep learning	In this paper presents a deep-learning-based autonomous scheme for recognizing traffic signs on Indian roads. The authors propose a refined Mask R-CNN model, which achieved a precision of 97.08% using an innovative dataset of 6480 images. The model outperformed conventional deep neural network architectures and improved accuracy with a miss rate of 3.25% and false positive rate of 2.92%. The study also addresses challenges of optical character recognition.
23.	Chauhan A, Rastogi A, Gaur A, Singh A, Gupta M (2020)	Traffic sign detection using deep learning	This paper discusses the importance of accurate traffic sign detection for road safety and the challenges associated with it. It proposes a method using Convolutional Neural Networks (CNNs) for reliable detection in varying environmental conditions. The authors also discuss the development of self-driving cars

			and intelligent transportation systems for traffic sign detection and identification. The system uses the German Traffic Sign Recognition Benchmark dataset and various machine learning algorithms.
24.	Tabernik D, Skocaj D (2019)	Deep Learning for large-Scale Traffic-Sign Detection and Recognition	The paper discusses the use of Mask R-CNN for automatic detection and recognition of traffic signs for efficient inventory management. The approach detects 200 traffic-sign categories in a novel dataset, with improvements proposed for improved recall rate and augmentation techniques. The study serves as a benchmark for complex traffic signs with high intra-category appearance variability.
25.	Carlos Filipe Paulo, Paulo Lobato Correla (2007)	Automatic Detection and Classification of Traffic Signs	This paper presents algorithms for automatic detection and classification of traffic signs from images, aiming to provide a driver alert system. The algorithms analyze color information, particularly red

			and blue, and classify signs into danger, information, obligation, or prohibition classes. The paper includes innovative components to improve system performance and prevent accidents.
26.	Kiran C, Prabhu L, V. A Rajeev K (2009)	Traffic Sign Detection and Pattern Recognition Using Support Vector Machine	The paper discusses the detection and recognition of traffic signs using color information in image sequences. Researchers use color-based segmentation techniques, linear SVM, and multi-classifier non-linear SVM for training and testing. They use red, blue, yellow, and white traffic signs for training and testing, and use SVM classifiers for shape classification and pattern recognition.
27.	Alessandro Giusti, Dan C.Ciresan, Jonathan Masci, Luca M. Gambardella, Jurgen Schmidhuber (2013)	Fast Image Scanning with Deep Max-Pooling Convolutional Neural Networks	This paper discusses the optimization of Deep Max-Pooling Convolutional Neural Networks (DNNs) with convolutional and max-pooling layers, focusing on convolutional and max-pooling layers. It suggests dynamic

			programming can speed up the process by ensuring each fragment contains independent information. The approach can handle various architectures and extends to the whole image level, ensuring no redundant computation.

2.8 Summary

In this chapter, there are 27 journals and articles research that can be references for this project. The journal and articles research more about the Convolutional Neural Network (CNN). Many methods can be used to detect road signs, one of the methods is the Convolutional Neural Network (CNN). This method is a type of deep learning neural network architecture specifically designed for the processing and analysis of visual data, such as images and videos.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will discuss the processes and methods used in this project, as well as how the final result is acquired to meet the project's goals. This chapter will also go over the project in detail, including the software development for a 'Road Sign Detection using Deep Learning'. It includes detailed descriptions of the dataset to be used and software material selection and analysis procedures. The entire system follows a specified flow path, which can be broken down into multiple steps depending on the various parts' functionalities.

3.2 Project Workflow

This chapter will explain the flow of the project. This chapter presents the general block diagram and flowchart of this project. Besides, this chapter also presents the expected result and material used in this project. Figure 3.1 shows the project workflow of this project.

Figure 3.1 Project workflow

3.2.1 General Block Diagram

This diagram shows planning to ensure the project may be done smoothly. First identify the problem, at this stage problem of road sign detection identified. Problem is the efficieny of the system is low. Objective of this project to develop a deep-learning system that is capable of reliably detecting and recognising various traffic signs, such as warning, yield, and speed limit signs. Data from this project will be collect an analyze to get the result. Then demostrate the detection sign using the website that had for this project. Finally, the effectiveness road sign detection using deep learning can be determined.

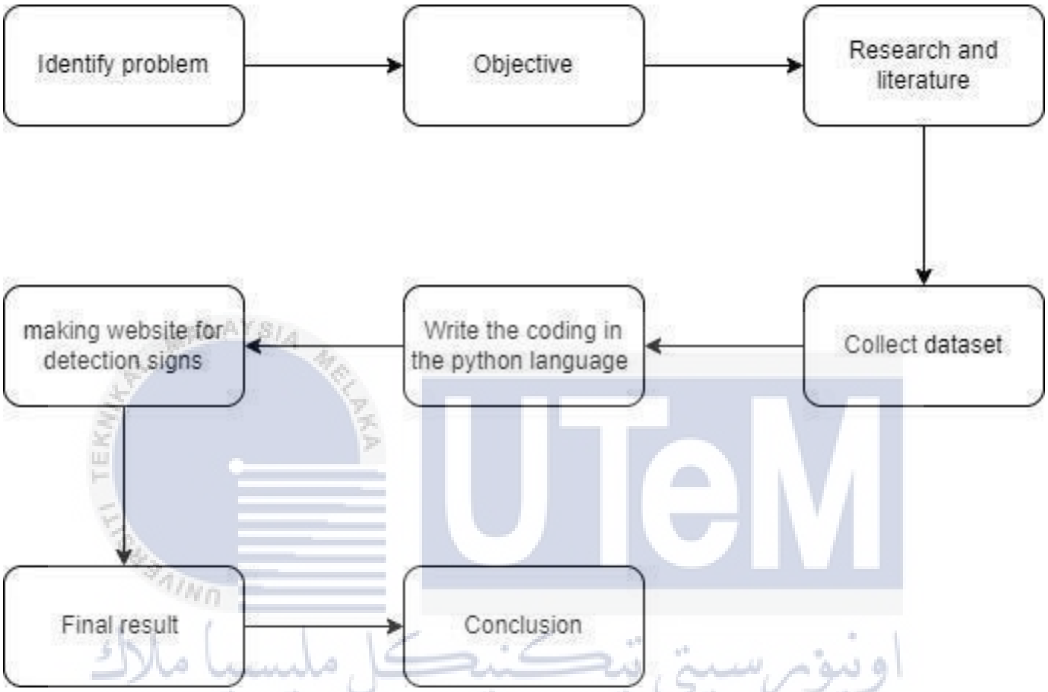


Figure 3.2 Block diagram of the development of road sign detection using deep learning.

3.2.2 Project Flow Chart

As shown in Figure 3.1, the project starts with Data collection where the input data read image from “myData” directory and extract class labels from directory structure. Next, the prosess is data preprocessing where the the images will be resize to a consistent resolution and normalize pixe values to a standard scale. After that, data splitting into training data and validation data. The next process the model creation where the model define as a Convolutional Neural Network (CNN) model. Next process is image preprocessing where the process convert images to grayscale and normalize pixel values into the range [0,1]. The coding set to show the plot loss and accuracy over the training epochs. Lastly, the test score and accuracy will be display the results.

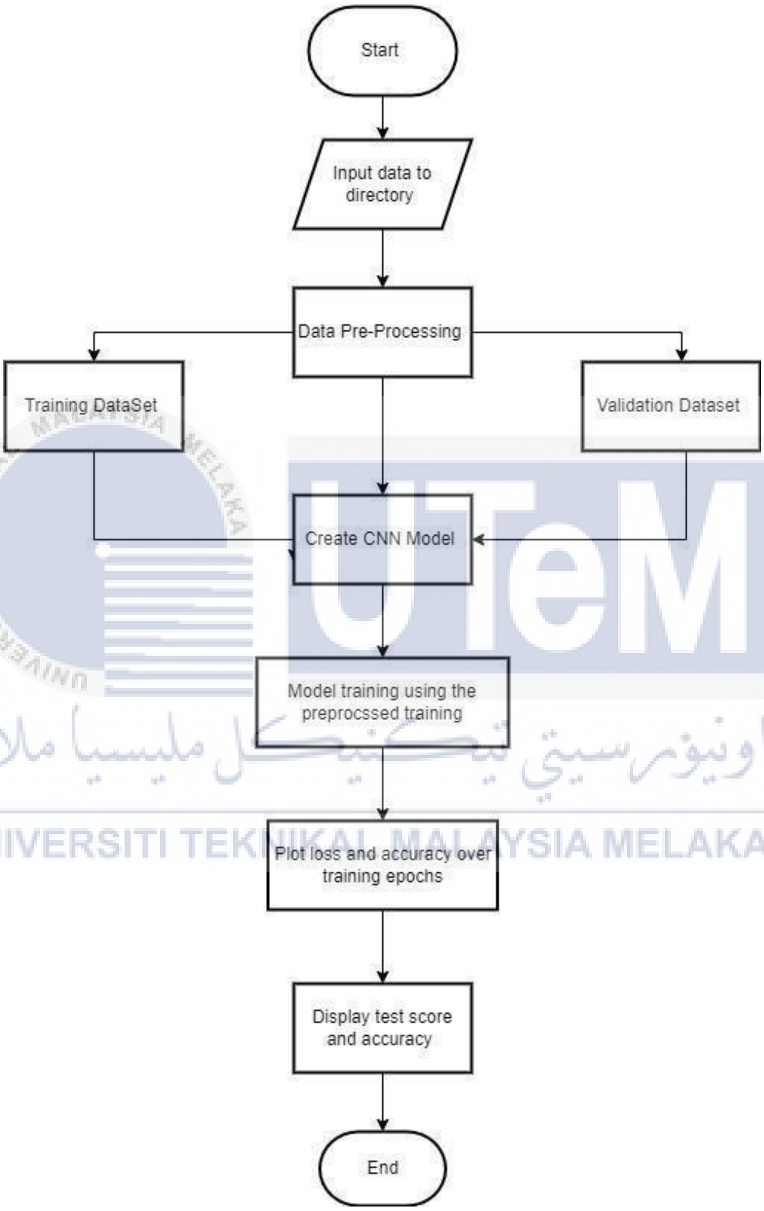


Figure 3.3: Flowchart Project

3.3 Defining Design Target

The purpose of this project is to detect the road sign using the dataset that had been inserted into the programs. The collected dataset was by resizing the images to a consistent size, normalizing the pixel values, and applying any necessary augmentation techniques. The model selection for this dataset is an appropriate deep-learning model for road sign detection. Convolutional Neural Networks (CNNs) are commonly used for this task due to their ability to learn hierarchical features from images. The graph plot of loss and accuracy over training epochs will appear. The demostrate for this road sign detection was also provided on the website that was created to predict the signs.

This program also can help autonomous vehicles to detect road signs easily and may reduce road accidents. 43 classes of datasets are used in this project such as speed limit signs, no passing signs, stop signs, yield signs, no entry signs, and many more. The software involved in this project is Visual Studio Code which uses Python language coding.

3.4 Software Specfications

This research is focused on the Python language and CSS language to make the application of road detection using deep learning. This project also focuses on the accuracy training, accuracy validation, loss training and loss validation of the epoch of the system. This project also links to the website that uses the HTML method.

3.4.1 The Dataset

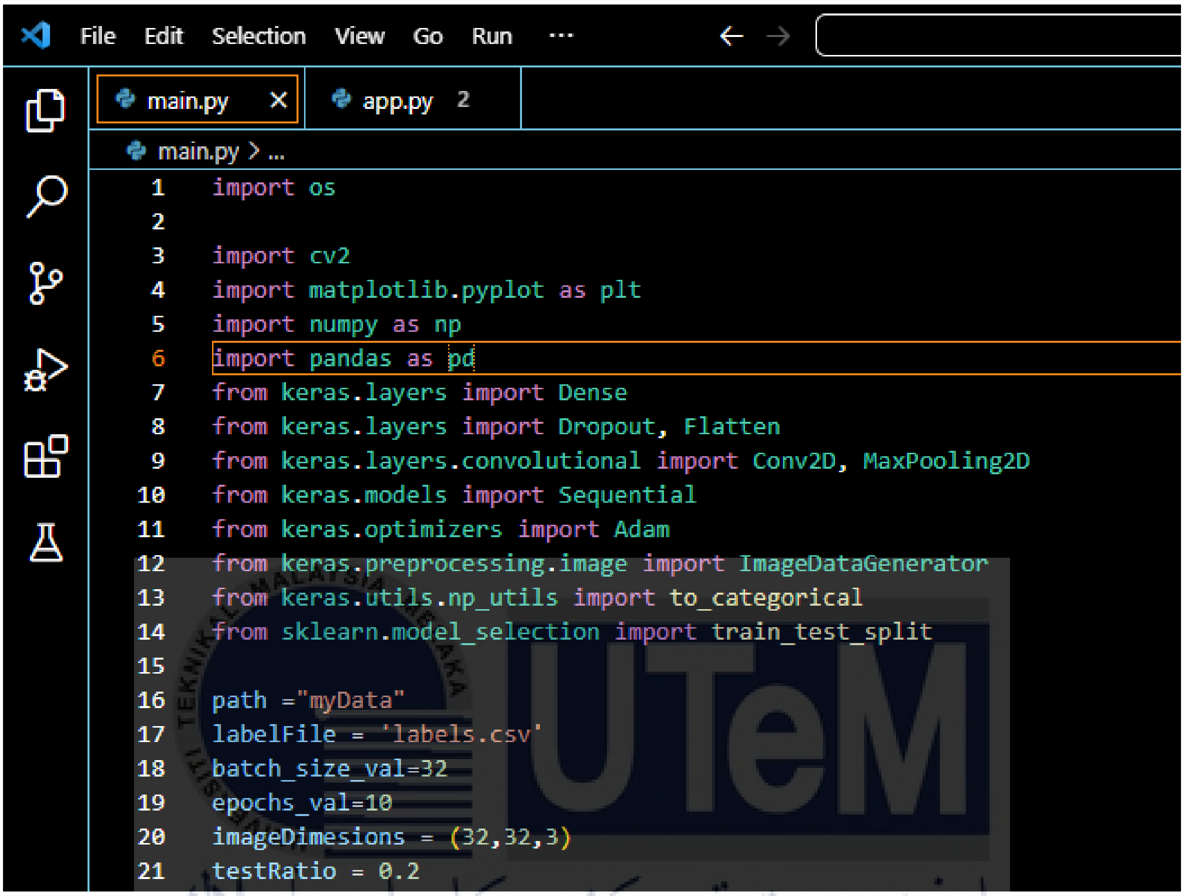
The German traffic sign detection benchmark (GTSDB) dataset and additionally, the German traffic sign recognition benchmark (GTSRB) dataset are mainly used to statistically train and evaluate the provided approach. There are 600 training images and 300 test images in the GTSDB dataset. Each image has one to multiple traffic signs from various categories and is a scene of traffic with a resolution of 1360 x 800 pixels. Prohibitive signs, danger signs, and necessary signs represent the majority of the categories. Other traffic indicators exist as well that do not fit into any of these categories. There are 43 different classes and 4 different categories of traffic signs overall [28].

Figure 3.4 Example of the dataset used in this project

3.4.2 Visual Studio Code

The application that is used in this project is Visual Studio Code and the language is Python. Visual Studio Code combines robust developer tools like IntelliSense code completion and debugging with the simplicity of using a source code editor. Figure 3.5 shows the main.py coding where the dataset is imported as “myData”. The accuracy training, accuracy validation, loss training and loss validation of epochs are also in this main.py.

Figure 3.6 shows the app.py coding that settings the system to predict sign detection to manage the accuracy the system and it coding link with the website that uses in HTML.




```python
import os

import cv2
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from keras.layers import Dense
from keras.layers import Dropout, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.utils.np_utils import to_categorical
from sklearn.model_selection import train_test_split

path ="myData"
labelFile = 'labels.csv'
batch_size_val=32
epochs_val=10
imageDimesions = (32,32,3)
testRatio = 0.2
```


Figure 3.5 The coding of main.py

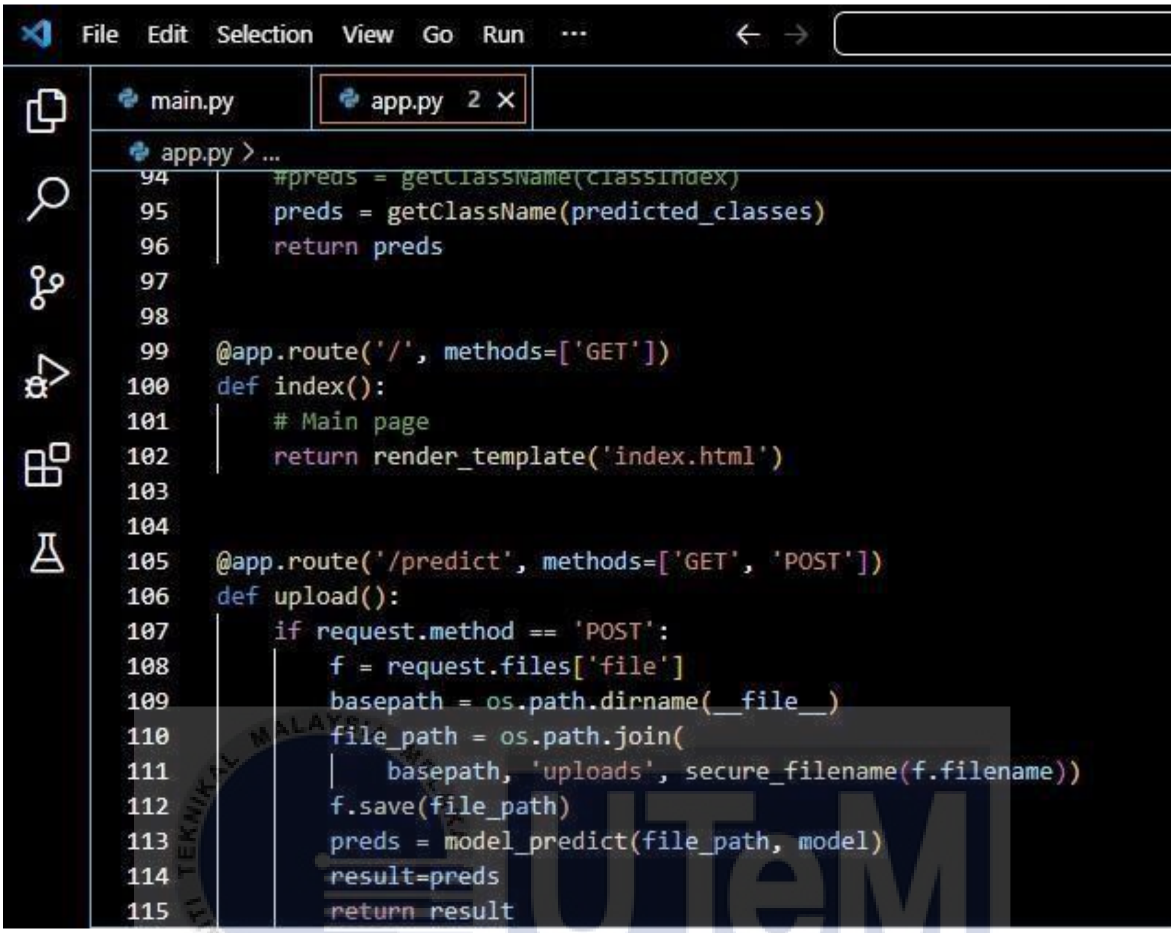


Figure 3.6 The coding of app.py

3.5 Website HTML

Figure 3.7 shows the interface of the website designed to train for recognizing road sign, it would typically involve functionalities and components that facilitate the process of collecting, annotating, and managing data for training the model. Users can upload images containing road signs and the interface may allow users to organize and manage datasets, grouping images and annotations into different categories or projects.

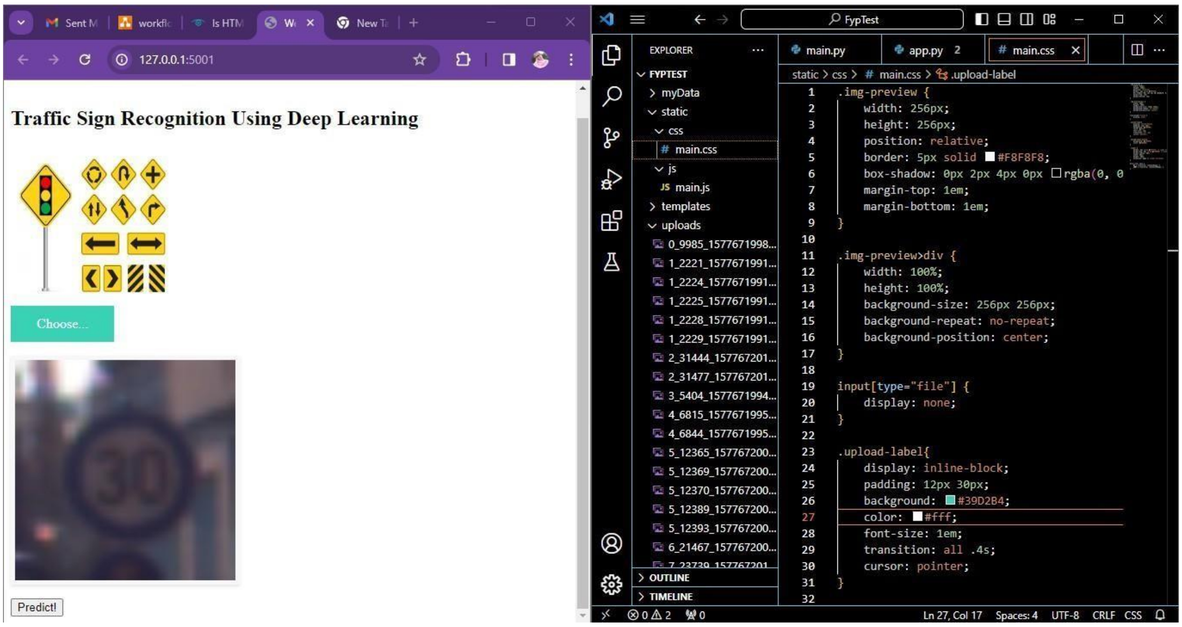


Figure 3.7 The interface and coding for the website

3.6 Gantt Chart

GANTT CHART BDP 1

PROJECT PLANNING														
List down the main activity for the project proposal. State the time frame needed for each activity.														
	2023													
Project Activity	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
BDP 1 Briefing by JK, PSM, FTKEE	**Briefing**							**Mid Term Break**						
Submit project title														
Chapter 1: Introduction														
Study about project														
Chapter 2: Literature Review														
Research for articles and journals														
Chapter 3: Methodology														
Editing of project report														
Testing & analysis														
PSM 1 Documentation														
PSM 1 Presentation & Report Submission														

GANTT CHART BDP 2

PROJECT PLANNING														
List down the main activity for the project proposal. State the time frame needed for each activity.														
	2024													
Project Activity	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
BDP 2 Briefing by JK, PSM, FTKEE														
Making Coding														
Train the project														
Chapter 4: Result & Analysis								**Mid Term Break**						
Editing of project report														
Chapter 5: Conclusion														
PSM 2 Documentation														
PSM 2 Presentation & Report Submission														

3.7 Summary

The road sign detection process employing deep learning, as expounded in this chapter, revolves around a singular concept the identification of input and dataset images for the recognition of specific signs. Within the broader context of project management, the significance of a well-defined project methodology is paramount. This methodology serves as a guiding framework, ensuring the systematic completion of the project by adhering to a precise sequence of techniques. At each stage of the method, encompassing project operation development, project determination, and ultimate project integration, the project developer formulates a comprehensive strategy for project development. These distinct stages collectively operate as a structured schedule, providing a roadmap for the project's progression. Subsequently, the entire project integration undergoes rigorous testing and troubleshooting, aiming to fulfil the project's overarching goals. This approach ensures a methodical and coherent development process for road sign detection through deep learning, aligning with best practices in project management.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The culminating chapter of this exhaustive project report serves as the definitive showcase of a year-long endeavor, from the initial stages of rigorous research and comprehensive literature review to the intricate processes that brought the project to fruition. In this concluding section, readers will be immersed in the tangible outcomes of relentless effort and meticulous planning. The focal point lies in unveiling the final result of the project implementation, providing an immersive exploration of the developed software. Through detailed descriptions, accompanied by carefully curated screenshots, videos, and possibly live demonstrations, the chapter meticulously illustrates the features and functionalities encapsulated within the software.

The content of the final project is dissected, offering an in-depth analysis of its components, unique features, and innovative aspects. This narrative aims not just to present the end product but to convey the depth of understanding and creativity embedded in every line of code. The symbiotic relationship between the coding intricacies and the overarching project objectives is elucidated, unraveling the narrative of how each line of code contributes to the realization of specific goals and functionalities. This exploration of the coding landscape underscores the technical prowess and strategic thinking behind the project. Visual aids such as flowcharts, graphs, and figures serve as critical elements in elucidating complex processes. Flowcharts articulate the logical pathways embedded in the software, graphs depict performance metrics, and figures encapsulate key conceptual frameworks, enriching the reader's understanding.

4.2 Data Analysis

In Figure 4.1, the results of the analysis are presented, showcasing images from 43 distinct classes. The training set, denoted as (22271, 32, 32, 3), comprises 22,271 samples, each represented as a 32x32 image with three colour channels (RGB). This set serves as the foundation for training the model. Similarly, the validation set, with dimensions (5568, 32, 32, 3), consists of 5,568 samples, mirroring the characteristics of the training set. The validation set is crucial for assessing the model's performance on unseen data. Finally, the test set, indicated as (6960, 32, 32, 3), encompasses 6,960 samples, maintaining the same image dimensions and colour channels. The test set is utilized to evaluate the model's generalization capabilities. These numerical specifications offer detailed insights into the dataset's composition, providing essential information for understanding the scale and characteristics of the images used in the deep learning or computer vision project.

```
PS D:\FypTest> & C:/Users/Asus/AppData/Local/Programs/Python/Python37/python.exe d:/FypTest/main.py
Total Classes Detected: 43
Importing Classes.....
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
Data Shapes
Train(22271, 32, 32, 3) (22271,)
Validation(5568, 32, 32, 3) (5568,)
Test(6960, 32, 32, 3) (6960,)
data shape  (43, 2) <class 'pandas.core.frame.DataFrame'>
```

Figure 4.1 The result of importing classes

In the training of a machine learning model, an epoch signifies a complete iteration through the entire training dataset. Within each epoch, the model's parameters are dynamically adjusted based on the calculated error or loss derived from the training data. Figure 4.2 provides a visual representation of the accuracy and loss values for both the training and validation datasets throughout ten epochs. Examining the loss epoch graph reveals a consistent decrease in both training and validation loss, indicating an improvement in the model's predictive performance. Concurrently, the accuracy values for both training and validation sets demonstrate a progressive increase. Notably, this upward trend in accuracy, coupled with a simultaneous decline in loss, is indicative of a successful training process. Such a scenario suggests that the model is effectively learning from the training data and successfully generalizing its knowledge to new, unseen data. The higher accuracy value relative to the loss value in the graph further reinforces the notion of a well-performing model, emphasizing its capacity to make accurate predictions and effectively capture patterns within the dataset.

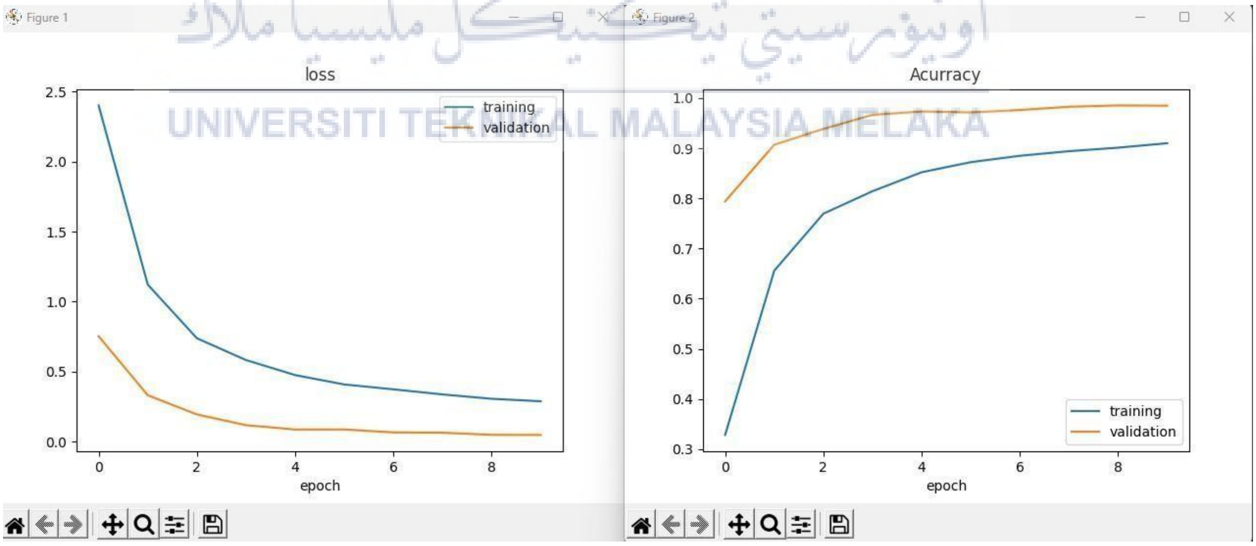


Figure 4.2 The graph of accuracy and loss of training and validation across epoch =10

```
Epoch 9/10
695/695 [==============================] - 130s 188ms/step - loss: 0.3071 - accuracy: 0.9012 - val_loss: 0.0498 - val_accuracy: 0.9855
Epoch 10/10
695/695 [==============================] - 135s 195ms/step - loss: 0.2894 - accuracy: 0.9101 - val_loss: 0.0487 - val_accuracy: 0.9849
Test Score: 0.052910543978214264
Test Accuracy: 0.9843390583992004
```

Figure 4.3 The test accuracy of epoch =10

In Figure 4.4, a comprehensive depiction is offered, revealing the nuanced dynamics of accuracy and loss within the training and validation datasets, specifically when the epoch value is set to 20. This visual representation serves as a crucial checkpoint in the model training process, encapsulating the evolution of performance metrics over a more extended training period. Analyzing the loss epoch graph divulges a consistent downward trend in both training and validation loss, signifying an ongoing refinement in the model's predictive capabilities. Concurrently, the accuracy values for both the training and validation sets demonstrate a discernible upward trajectory, underscoring the model's proficiency in correctly classifying data points. Noteworthy is the comparative analysis with epoch=10, indicating an enhanced test accuracy at epoch=20. This improvement aligns with the general expectation that as the number of epochs increases, the model refines its understanding of the underlying patterns in the data, leading to more accurate predictions. The discernible correlation between the epoch value and the model's accuracy reinforces the iterative nature of the training process, suggesting that a more extended training period contributes to heightened accuracy and robust generalization. The multifaceted insights gleaned from Figure 4.4 underscore the iterative nature of model refinement, shedding light on the dynamic interplay between accuracy, loss, and the progression of epochs in achieving increasingly accurate and reliable outcomes.

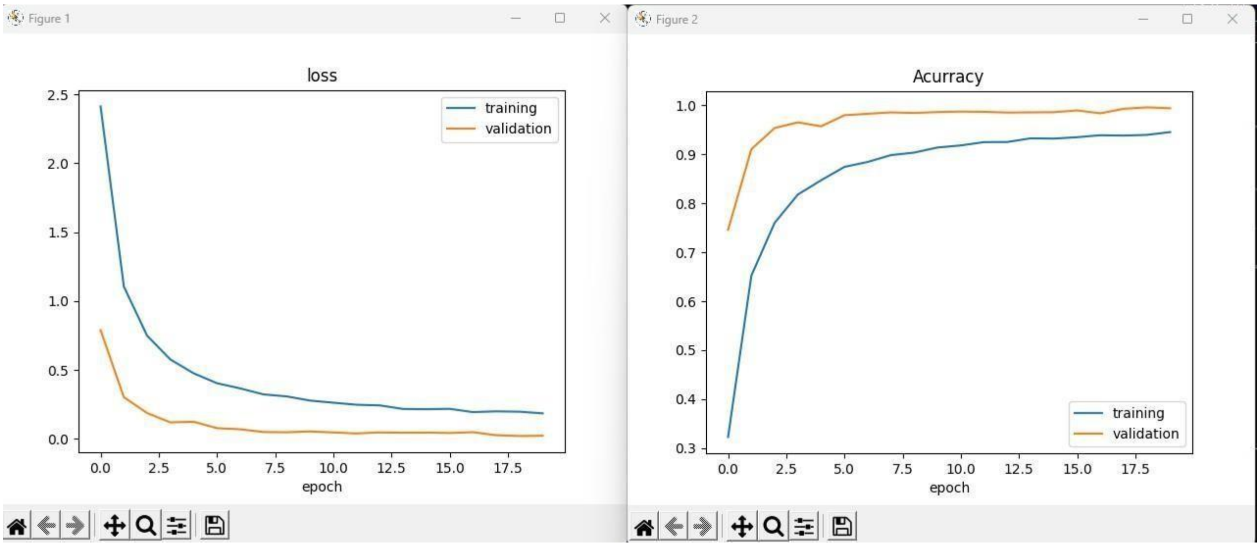


Figure 4.4 The graph of accuracy and loss of training and validation across epoch =20

```
Epoch 19/20
695/695 [==============================] - 111s 160ms/step - loss: 0.1967 - accuracy: 0.9397 - val_loss: 0.0203 - val_accuracy: 0.9959
Epoch 20/20
695/695 [==============================] - 98s 141ms/step - loss: 0.1846 - accuracy: 0.9454 - val_loss: 0.0220 - val_accuracy: 0.9941
Test Score: 0.024589868262410164
Test Accuracy: 0.9933907985687256
```

Figure 4.3 The test accuracy of epoch =20

4.3 Result

The program will recognize and predict the road sign. Below are the dataset that had been taken from the training set along with the results.

Figure 4.4 The result of road sign detection

4.4 Summary

In this chapter, the outcomes of the road sign detection tasks are systematically unveiled. Typically, a model is trained to adeptly recognize and categorize various types of road signs. A pivotal aspect of this process lies in the meticulous division of the dataset into training, validation, and test sets, each playing a distinct role in gauging the model's performance. The delineation of dataset sizes carries significant weight, offering insights into the abundance of data allocated for both training and evaluation purposes. These numerical indicators serve as valuable metrics, providing a practical understanding of the scale of information that the model is exposed to during its training phase and subsequently evaluated during testing. Such meticulous dataset management forms the bedrock for robust model development and evaluation in road sign detection applications.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The project aims to develop a system for road sign detection using deep learning techniques, specifically, Convolutional Neural Networks (CNNs), to address the impact of road sign detection on society and global issues. The project's objective is to detect road signs using a dataset inserted into the programs, with a focus on improving road safety and potentially aiding autonomous vehicles in sign detection to reduce road accidents. The literature review in Chapter 2 provides an overview of traditional traffic sign detection methods, including feature extraction and machine learning, as well as an in-depth exploration of CNNs and their application in visual data processing and analysis, particularly for road sign detection. The processes and methods used in the project, emphasise the development of a 'Road Sign Detection using Deep Learning' software. The methodology includes dataset preparation, software material selection, and the use of CNNs for road sign detection. The project's design target is to develop a deep-learning model for road sign detection, with potential applications in enhancing road safety and supporting autonomous vehicle technology. In summary, the project focuses on leveraging deep learning, specifically CNNs, to develop a road sign detection system with the potential to contribute to road safety and autonomous vehicle technology. The project's methodology and objectives align with the broader societal and technological implications of road sign detection, as outlined in the literature review and project introduction.

5.2 Potential for Commercialization

The automotive industry is undergoing a transformative shift, with the rise of advanced driver assistance systems (ADAS) and the development of autonomous vehicles. At the forefront of this revolution is the integration of deep learning techniques for road sign detection. The potential for commercialization in this domain is substantial, driven by the increasing demand for enhanced road safety, the evolution of smart cities, and the imperative for autonomous vehicles to navigate complex urban environments.

As an example, The dynamic nature of deep learning allows for continuous innovation and upgrades. Companies can offer subscription-based services or periodic software updates to keep their road sign detection systems up-to-date with the latest advancements and regulatory changes. This not only ensures the longevity of their solutions but also fosters customer loyalty and trust. Other than that, Road sign detection systems, powered by deep learning algorithms, play a pivotal role in enhancing road safety. By providing real-time information to drivers about speed limits, traffic regulations, and potential hazards, these systems contribute to a significant reduction in accidents. This safety improvement becomes a compelling proposition for consumers, automotive manufacturers, and regulatory bodies alike, creating a ripe market for commercialization.

In conclusion, the commercial potential of road sign detection using deep learning is vast and multifaceted. From improving road safety to facilitating the evolution of smart cities and supporting the advent of autonomous vehicles, the applications are diverse and promising. As technology continues to advance, companies at the forefront of developing robust, adaptable, and compliant road sign detection systems are poised to capitalize on the growing demand for safer and more intelligent transportation solutions.

5.3 Future Works

Improving road sign detection is crucial for enhancing road safety and enabling advanced driver assistance systems (ADAS) and autonomous vehicles. Potential future improvements are:

i. Utilize more advanced deep learning architectures, such as transformer-based models or custom-designed neural networks, to improve the accuracy and efficiency of road sign detection.

ii. Implement transfer learning techniques to leverage pre-trained models on large datasets. This can help the model generalize better to diverse road sign scenarios and improve performance with a smaller amount of labeled data.

iii. Combine object detection with semantic segmentation to better understand the context of road signs within the entire scene. This can improve the accuracy of detection in complex environments and ensure better localization.

iv. Integrate data from various sensors, such as cameras, LiDAR, and radar, to provide a more comprehensive perception of the environment. Multi-sensor fusion can enhance the robustness of road sign detection under different weather conditions and lighting scenarios.

v. Optimize algorithms for real-time processing to ensure timely and accurate detection of road signs. This is critical for applications in autonomous vehicles and ADAS where quick decision-making is essential for safety.

vi. Implement domain adaptation techniques to make the model more robust across different geographic locations, considering variations in road sign design, colors, and layouts.

vii. Explore edge computing solutions to perform road sign detection directly on-board vehicles. This can reduce the dependence on cloud services and improve real-time responsiveness.

viii. Use advanced data augmentation techniques to artificially increase the diversity of the training dataset, making the model more resilient to variations in illumination, weather, and other environmental factors.

ix. Enable incremental learning capabilities to allow the model to adapt to changes in road sign designs or new types of signs without retraining the entire model.

x. Implement systems that involve human feedback to continuously improve the road sign detection algorithm. This can include mechanisms for users to correct misclassifications or provide additional labeled data.

xi. Address privacy concerns by developing road sign detection systems that prioritize the protection of sensitive information, such as license plates or facial features, to comply with privacy regulations.

xii. Work towards standardization of road sign designs and colors to create a more uniform environment for detection algorithms, reducing the complexity of adapting models to different regions.

.

REFERENCES

[1] Kale, A. J., & Mahajan, R. C. (2015). A road sign detection and the recognition for Driver Assistance Systems. *2015 International Conference on Energy Systems and Applications*, 69–74. https://doi.org/10.1109/ICESA.2015.7503315

[2] Hechri, A., & Mtibaa, A. (2012). Automatic detection and recognition of road signs for driver assistance system. *2012 16th IEEE Mediterranean Electrotechnical Conference,* 888–891. https://doi.org/10.1109/MELCON.2012.6196571

[3] de la Escalera, A., Moreno, L. E., Salichs, M. A., & Armingol, J. M. (1997). Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, *44*(6), 848–859. https://doi.org/10.1109/41.649946

[4] Yanjun Fan, & Weigong Zhang. (2015). Traffic sign detection and classification for Advanced Driver Assistant Systems. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),* 1335–1339. https://doi.org/10.1109/FSKD.2015.7382137

[5] Wang, C. (2018). Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning. *2018 International Conference on Robots & Intelligent System (ICRIS)*, 150–152. https://doi.org/10.1109/ICRIS.2018.00047

[6] Karaduman, M., & Eren, H. (2017). Deep learning based traffic direction sign detection and determining driving style. *2017 International Conference on Computer Science and Engineering (UBMK)*, 1046–1050. https://doi.org/10.1109/UBMK.2017.8093453

[7] Xuehong Mao, Hijazi, S., Casas, R., Kaul, P., Kumar, R., & Rowen, C. (2016). Hierarchical CNN for traffic sign recognition. *2016 IEEE Intelligent Vehicles Symposium (IV)*, 130–135. https://doi.org/10.1109/IVS.2016.7535376

[8] Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-Sign Detection and Classification in the Wild. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2110–2118. https://doi.org/10.1109/CVPR.2016.232

[9] Wang, G., Ren, G., & Quan, T. (2013). A Traffic Sign Detection Method with High Accuracy and Efficiency. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. https://doi.org/10.2991/iccsee.2013.359

[10] Malik, R., Khurshid, J., & Ahmad, S. N. (2007). Road Sign Detection and Recognition using Colour Segmentation, Shape Analysis and Template Matching. *2007 International Conference on Machine Learning and Cybernetics*, 3556–3560. https://doi.org/10.1109/ICMLC.2007.4370763

[11] Huang, Z., Yu, Y., Ye, S., & Liu, H. (2014). Extreme learning machine-based traffic sign detection. *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 1–6. https://doi.org/10.1109/MFI.2014.6997672

[12] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition. *The 2011 International Joint Conference on Neural Networks*, 1453–1460. https://doi.org/10.1109/IJCNN.2011.6033395

[13] Mogelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1484–1497. https://doi.org/10.1109/TITS.2012.2209421

[14] Glavtchev, V., Muyan-Ozcelik, P., Ota, J. M., & Owens, J. D. (2011). Feature-based speed limit sign detection using a graphics processing unit. *2011 IEEE Intelligent Vehicles Symposium (IV)*, 195–200. https://doi.org/10.1109/IVS.2011.5940539

[15] Jin, K. H., McCann, M. T., Froustey, E., & Unser, M. (2017). Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Transactions on Image Processing*, *26*(9), 4509–4522. https://doi.org/10.1109/TIP.2017.2713099

[16] Lin, K., & Wang, Z. (2022). Traffic Sign Classification by Using Learning Methods: Deep Learning and SIFT Based Learning Algorithm. *2022 14th International Conference on Computer Research and Development (ICCRD)*, 239–243. https://doi.org/10.1109/ICCRD54409.2022.9730126

[17] Ming Liang, & Xiaolin Hu. (2015). Recurrent convolutional neural network for object recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3367–3375. https://doi.org/10.1109/CVPR.2015.7298958

[18] Bochinski, E., Senst, T., & Sikora, T. (2017). Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. *2017 IEEE International Conference on Image Processing (ICIP)*, 3924–3928. https://doi.org/10.1109/ICIP.2017.8297018

[19] Giusti, A., Ciresan, D. C., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2013). Fast image scanning with deep max-pooling convolutional neural networks. *2013 IEEE International Conference on Image Processing*, 4034–4038. https://doi.org/10.1109/ICIP.2013.6738831

[20] Hussain, S., Abualkibash, M., & Tout, S. (2018). A Survey of Traffic Sign Recognition Systems Based on Convolutional Neural Networks. *2018 IEEE International Conference on Electro/Information Technology (EIT)*, 0570–0573. https://doi.org/10.1109/EIT.2018.8500182

[21] Suárez-Paniagua, V., & Segura-Bedmar, I. (2018). Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction. *BMC Bioinformatics*, *19*(S8), 209. https://doi.org/10.1186/s12859-018-2195-1

[22] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, *32*, 323–332. https://doi.org/10.1016/j.neunet.2012.02.016

[23] Gomez-Moreno, H., Maldonado-Bascon, S., Gil-Jimenez, P., & Lafuente-Arroyo, S. (2010). Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition. *IEEE Transactions on Intelligent Transportation Systems*, *11*(4), 917–930. https://doi.org/10.1109/TITS.2010.2054084

[24] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2016). *Aggregated Residual Transformations for Deep Neural Networks*.

[25] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*.

[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

[27] Cetinkaya, M., & Acarman, T. (2021). Traffic Sign Detection by Image Preprocessing and Deep Learning. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS),* 1165–1170. https://doi.org/10.1109/ICICCS51141.2021.9432088

[28] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., & Igel, C. (2013). Detection of traffic signs in real-world images: The German traffic sign detection benchmark. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2013.6706807