Design Framework for Road sign and Speed limit Detection Using Deep Learning

A Thesis Presented

By

Gemechis Belay

To

The Faculty of Informatics

Of

St. Mary's University

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Computer Science

ID No:-SGS/0381/2013A

January , 2025

ACCEPTANCE

Design Framework for Road sign and Speed limit Detection Using Deep Learning

By

Gemechis Belay Wakweya

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

_____

Internal Examiner

_____

External Examiner

_____

Dean, Faculty of Informatics

January 2025

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Gemechis Belay Wakweya

_____

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Alembante Mulu (PhD)

Full Name of Advisor

_____

Signature

Addis Ababa

Ethiopia

January 2025

Acknowledgement

Firstly I would like to thanks God for always being there for me and giving me courage and patience in doing this work. It would not have been to this stage without the help of God. Next, I would like to express my deepest gratitude to my thesis advisor Dr. Alembante Mulu for contributing his constructive comments, directions, ideas, and courage during the thesis. I would like to give my special thanks to my Wife Hana Teferi for her encouragement and support all through my studies and throughout my life and My Brother Kena Belay for his support.

Finally, I would like to extend my sincere thanks to all those who have provided support encouragement and understanding ,I extend my sincere thanks, your contribution have played a crucial role in the successful completion of this thesis.

Contents

List of Acronyms

ADAS-Advanced Driver-Assistance Systems

CNNs-Convolutional Neural Networks

ITS-intelligent transport system

IVS -Intelligent Vehicle System

ACC-Adaptive Cruise Control

ML -Machine learning

CUDA -Compute Unit Device Architecture

R-CNN - Region-based Convolutional Neural Networks

R-FCN - Region-based Fully Convolutional Networks

YOLO -You only look once

SSD -Single shot multibox detector

NLP -natural language processing

List of figures

List of tables

Abstract

Road sign and speed limit detection is a crucial aspect of intelligent transportation systems (ITS) and autonomous driving. This paper presents a design framework for an efficient and robust road sign and speed limit detection system using deep learning techniques. The framework integrates convolutional neural networks (CNNs) and advanced object detection models, such as YOLO (You Only Look Once) or Faster R-CNN, to accurately detect and classify road signs and speed limit indicators in real-time. The system is designed to handle varying environmental conditions, including different lighting, weather, and occlusions. A preprocessing pipeline is employed to enhance image quality and improve detection accuracy. The framework is evaluated on benchmark datasets and real-world road scenarios to assess its performance. Experimental results demonstrate high detection accuracy and computational efficiency, making the proposed framework suitable for deployment in advanced driver-assistance systems (ADAS) and autonomous vehicles. Future improvements include integrating multi-modal sensor fusion and reinforcement learning to enhance robustness and adaptability. This investigate contributes to the improvement of a solid framework for Ethiopian street sign and light discovery distinguishing proof. By leveraging profound learning strategies and optimization approaches, we address the confinements of existing frameworks, such as little datasets and lower acknowledgment precision. The discoveries illustrate the potential of utilizing MobileNet with SGD optimization as an successful arrangement for street sign and light acknowledgment in Ethiopia, clearing the way for progressed street sign and light recognizable proof in robotized self-service hardware.

Keywords: Road Sign Detection, Speed Limit Recognition, Deep Learning, CNN, YOLO, Autonomous Driving, ADAS

1. chapter 1 Introduction

1.1. Background

In recent years, the integration of intelligent systems into road safety infrastructure has garnered significant attention, particularly as autonomous driving and Advanced Driver-Assistance Systems (ADAS) continue to evolve. One of the most critical elements of ensuring safe navigation and effective traffic management is the detection and interpretation of road signs, including speed limit signs. Road sign recognition is essential for vehicles to understand traffic rules, adjust their speed accordingly, and prevent accidents caused by violations of traffic regulations. With advancements in machine learning, particularly deep learning, there has been substantial progress in automating the detection and classification of road signs.

Deep learning techniques, specifically Convolutional Neural Networks (CNNs), have emerged as highly effective solutions for tasks related to image recognition, including the detection of road signs. These models can learn complex patterns from large datasets, enabling them to recognize road signs with high accuracy across a variety of environmental conditions. This thesis proposes a design framework for road sign and speed limit detection using deep learning, which aims to provide a robust, real-time solution for both conventional and autonomous vehicles.

An intelligent transport system (ITS) is a global phenomenon, attracting worldwide interest from transportation professionals, the automotive industry, and political decision-makers. ITS provides transport solutions by utilizing state-of-the-art information and telecommunications technologies. It is an integrated system of people, roads, and vehicles, designed to significantly contribute to improving road safety, efficiency, and comfort, as well as environmental conservation through the realization of smoother traffic by relieving traffic congestion according to Andersen and Sutcliffe. Early work in ITS was carried out by the Japanese in the 1980s. At that time, the Japanese had not coined a specific name for ITS as it was considered part of traffic control. They later referred to this work as the Japanese Intelligent Vehicle System (IVS) program as reported by Andersen and Sutcliffe [1]. Sustainable and efficient transport systems are requirements for economic well-being. ITS holds the promise of sustainability which presents the opportunity for better management of existing resources and infrastructure, through the provision of information to travellers and transportation planning professionals and offers new control possibilities. Driver Assistance

System (DAS) is an important module in ITS. The system is developed to alert a driver or to interact directly with the vehicle for safety and better driving. DAS includes Driver drowsiness detection, Adaptive Cruise Control (ACC), Lane departure warning system, Traffic sign recognition, Wrong-way driving warning, automotive navigation system, and so on [2]. Advances in Image and Video Processing; Volume 7, No. 5, October 2019 Society for Science and Education, United Kingdom 31 The purpose of Advanced Driver Assistance Systems (ADAS) is that driver error will be reduced or even eliminated, and efficiency in traffic and transport is enhanced. The benefits of ADAS implementations are potentially considerable because of a significant decrease in human suffering, economical cost and pollution [3]. However, there are also potential problems to be expected, since the task of driving an ordinary motor vehicle is changing in nature, in the direction of supervising an or partly automated moving vehicle [3]. The main focused of this work is on obstacle detection on the road i.e. speed breakers. Speed bumps or speed breakers are designed on road to avoid or reduce over speed. Speed Breakers are traffic handling tool which are used to slow down vehicles speed [4]. Authors stated further that they are called Speed Humps, Speed Bumps, Speed Ramps, Speed Cushions and Speed tables. They are designed to be driven over at a predetermined comfortable speed, while causing exceeding discomfort at higher speeds. The reduction in average vehicular speed significantly improves the safety of people in the neighbouring areas [5]. Speed breakers can have different heights, lengths, spacing, signs, etc. In fact, no particular design is suitable for all the types of vehicles using the road. Where speed breakers are permitted to be installed, provide visual and tactile stimuli which alert drivers and cause them to slow down. Speed breakers are typically placed on residential roads. According to survey, speed breakers have predominantly reduced casualty crash frequencies, fatal crashes and pedestrian crashes [5]. According to Rahayu et al [6] speed breaker detection is important to community in order to save the car and human life. Authors in [2] suggested the need to develop a system that services the end user driver using image processing algorithms such as Gaussian filtering, Median filtering and Connected Component Approach etc. The state-of-art algorithm suggested by Devapriya et al in [2] fails on datasets that includes stop markings. Hence there is a need for an algorithm that provides a solution to the state-of-art algorithm problem.

Additionally significant research attention has recently been paid to the monitoring of irregularities in the road sign and surface, such as potholes and speed bumps. Speed breakers are used to stop excessive driving speed. Speed breakers are traffic control devices intended

to reduce vehicle speed while riding. The New York Times noted on June 7, 1906, that Chatham, New Jersey, was planning to raise its crosswalks five inches (13 cm) above the road level as an early example of what might be considered speed breakers: "This scheme of stopping automobile speeding has been discussed by different municipalities, but Chatham is the first place to put it in practice." At that time, the peak speed of the typical car was about 50 km/h (30 mph), but braking was inadequate by today's standards. Scientist Arthur Holly Compton won the Nobel Prize in physics in 1927 for his findings, which significantly altered electromagnetic theory. He is well-known for his work on the X-ray Compton Effect. In 1953, he also created "traffic control bumps," the precursor to the speed hump. After observing how quickly cars passed Brookings Hall at Washington University in St. Louis, Missouri, where he served as chancellor, Compton started designing the speed bump. The British Transport and Road Research Laboratory examined vehicle behaviour for a wide range of various bump geometries and released a thorough report on its findings in 1973. Back then, public highways were not allowed to have speed bumps, but had been put in place on personal roads. The first speed bump in Europe was constructed in 1970 in the Dutch city of Delft, claims a report by the Institute of Transportation Engineers. In summary, deep learningbased speed breaker recognition is a significant and promising area of research that has already yielded some amazing results. Building on prior research, we can improve the accuracy and efficacy of speed breaker identification systems, which could have a significant impact on traffic safety and the advancement of autonomous driving technology

1.2. Motivations

Machine learning (ML) methods for the industry has opened new gateways for enhancement in the automotive domain, especially for Advanced Driver Assistance Systems (ADAS). In the development of ADAS and self driving vehicles, object detection is a key function. In order to achieve a robust and meticulous object detector, many advanced detection algorithms based on machine learning technology have been proposed in the literature [69, 70, 71].

Deep learning-based Convolutional Neural Networks (CNN) methods have made significant improvements in object classification, object recognition and object detection. Over the past few years, object detection using deep learning has been a research hotspot. Although the deep CNN model provides a very powerful and robust feature representation for object detection and recognition, it is computationally expensive and requires a highend hardware platform for model inference calculations [72, 73]. Reliable real-time detection of traffic sign and light on a computationally limited platforms are a significant concern for the task of

autonomous driving. Therefore, only neural network models with small network scale and low computational complexity are needed. In the development of autonomous driving the issue at hand is, how to transplant the object detection network model to the embedded platform for operation while maintaining acceptable accuracy and stability. Various deep learning frameworks were released to execute deep learning algorithms. Darknet [74] is one of them that is used for object detection. It is recognized for its simple architecture and fast processing speed. But the drawback is that it only supports NVIDIA Compute Unit Device Architecture (CUDA) for accelerating its deep learning calculations. This leads to a restricted system configurations as the users are bound with limited options for graphic card selection.

The current frameworks for object detection task can be categorized into two main types, two-stage detector, and single-stage detector. Two stage detectors have better performance in terms of localization and recognition accuracy. However, for the training and inference purposes the need for immense computational power makes them unacceptable for real time applications often requiring several seconds per image.

Examples are Faster Region-based Convolutional Neural Networks (R-CNN) [75] and Region-based Fully Convolutional Networks (R-FCN) [76]. The other category bridges the generation of region proposal, classification, and regression task as one multi-task learning problem. The most prominent algorithms in this category are: You only look once (YOLO) [77] and Single shot multibox detector (SSD) [10]. They have improved real time detection speed and therefore can be implemented on embedded platforms. However, their efficiency is lower in terms of accuracy when compared with two stage detectors. The main SSD [78] algorithm uses Visual Geometry Group (VGG16) as the base network for feature extraction. Where VGG16 has a very deep network architecture and therefore too large for embedded platforms to achieve real-time processing speed.

One major problem with SSD [10] algorithm is that it is not good at dealing with small objects. The main cause of the problem are the pooling operations that increase the receptive field and as a result reduce the computational effort. However, at the same time it lowers the image resolution leading to difficulties for precise localization of small objects. Based on the fact that even though besides the size of the object, each object type has a different context information requirement and traffic signs, and lights comes in regular shapes like triangle, circle, rectangle and in distinct colors. Therefore, this unique and easy visual appearance enables them to be detected at first glance by the early feature map layers.

The commonly observed speed limit control measures on Ethiopia roads are :-Marked Speed hump , Marked Speed bump ,Rumble strips , Unmarked hump/bump Other speed control measures which we rarely observe are rumble strips and speed tables. Speed humps on minor roads sometimes should span a width of 3.5 meter and height between 10 to 12cm and warning sign need to be present 40m ahead of the speed hump to adjust the speed of the vehicle. Speed humps should be painted and there will have to be road sign to make vehicle commuters visible and illuminated by solar cat eyes to assist road commuters about their presence during night time. Speed humps are designed to bring down the vehicle to safe crossing speeds of around 25km/hr. The Speed bump need to have an abrupt raise in area on the pavement surface usually spans 2 to 4 inches and is designed to maintain crossing speeds of 10mph or less. These are appropriate where the vehicle speeds are lowest to begin with such as parking lot, school entrances, hospitals, garages and private roads. We observe that lot of speed bumps are being constructed on public roads in Ethiopia which are resulting in severe discomfort for the commuters and causing damage to the vehicle. Key contribution of this work is to Design Framework for road sign and Speed limit Detection Using Deep Learning.

1.3. Statement of the Problem

Even though deep learning has given positive results in the recognition of Road sign and speed limit, there are still a lot of problems that need to be overcome. One of the key challenges is that changes in the weather and lighting can make signs, speed breakers, and holes less visible. The design of the road speed breakers may also be covered or distorted as a result of environmental factors like trees, buildings, or other cars. The size and quality of the training dataset can also influence how well deep learning models work. Time is of the essence when it comes to recognizing speed bumps in autonomous driving, so it's important that the recognition system is speedy, accurate, and reliable. To make sure that's the case, deep learning architectures and optimization techniques must be used. So this study looks into making a reliable and efficient AI-based system to detect and recognize road sign and speed limit , holes, and other stuff on roads in different lighting and blocking conditions, while also still performing quickly. By tackling the issues with existing methods, the proposed solution should help improve both the development of self-driving tech and road safety

1.4. Research Question

RQ1. How can we create a robust and diverse dataset for training a road sign detection system? RQ2. What deep learning models are most effective for real-time road sign detection? RQ3. How can we improve the accuracy of speed limit recognition from road signs? RQ 4. How do we integrate the road sign and speed limit detection framework with real-time autonomous driving systems? RQ5 What are the ethical and safety implications of deploying road sign and speed limit detection systems in real-world vehicles? RQ6.Which model is best to use in detecting speed breaker

1.5. Objectives

1.5.1. General Objective

Design and develop an efficient framework for road sign and speed limit detection using deep learning techniques, aimed at enhancing the performance and accuracy of Advanced Driver Assistance Systems (ADAS). The framework will leverage state-of-the-art deep learning models to automatically detect and classify road signs and speed limits, contributing to improved vehicle safety and autonomous driving capabilities.

1.5.2. Specific Objective

- **I**mplement the detection framework in a real-time simulation or test environment to evaluate its practical performance under different real-world driving conditions (e.g., day/night, rain, fog, highway, urban, rural settings).
- Analyze the model's ability to handle challenges such as occlusions, blurred signs, and varying distances from the camera.
- Develop a prototype of the detection framework and integrate it with an ADAS simulation platform or a vehicle-mounted camera system to test its performance in live driving scenarios.
- Evaluate the real-time performance of the system, including detection accuracy, speed, and the ability to provide actionable feedback to the driver or autonomous vehicle system.
- Analyze potential limitations and challenges encountered during real-time testing, such as difficulty detecting small, distant, or obscured signs.
- Propose potential improvements or additional research areas, such as incorporating additional sensor data (e.g., LiDAR, radar) or improving detection under adverse weather conditions.

- Provide recommendations for deploying the detection framework in practical applications, considering hardware requirements, computational resources, and real-time processing constraints.

1.6. Scope/Limitations

When designing a framework for road sign and speed limit detection using deep earning, it's essential to define both the scope and limitations to set clear boundaries on what the project will and will not cover.

Scope

Dataset: standard road sign and speed limit detection datasets are used in this thesis

Type of Road Signs: speed limits, stop signs, and warning signs in Ethiopia.

Model Architecture: Convolutional Neural Networks (CNNs), YOLO, or SSD, and other experiments are used.

Environmental Conditions: daytime, nighttime, various weather conditions, etc.

Limitations

Dataset Limitations: Limited to pre-existing datasets, world wide signs and road sign standards.

Camera Limitations: Performance may vary based on the quality of the camera or images used, which might impact detection accuracy in low-resolution or blurry images.

In general, The goal of this research is to Design Framework for road sign and Speed limit Detection Using Deep Learning and assess the system's performance on common benchmark datasets.

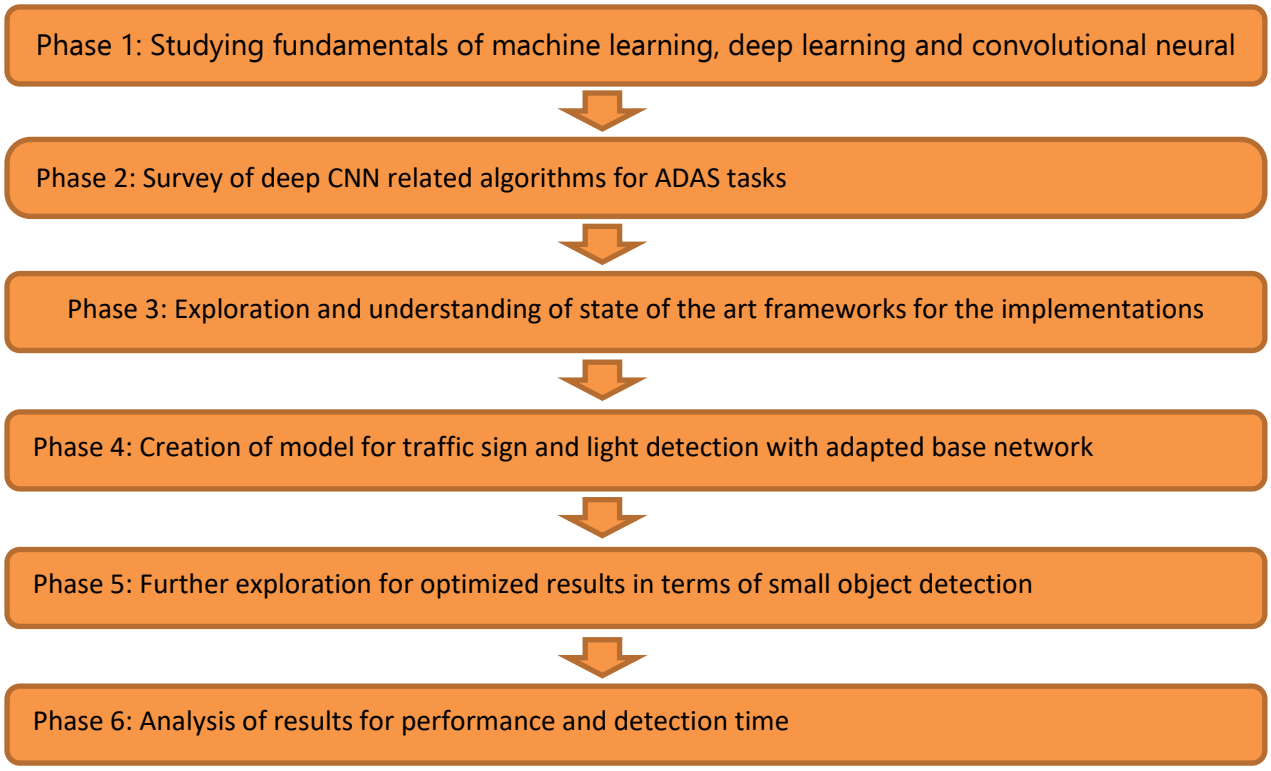
1.7 Methodology

Methodology refers the broad plan and justification for the research effort are referred to as methodology. It entails researching the theories and ideas that underpin the procedures employed in industry in order to create a strategy that is in line with thesis goals. A variety of approaches have been used to achieve the study's general and specific aims.

This study follows an experimental research approach with the objective of designing and developing a Framework for ADAS and also the SSD algorithm is adapted and optimized for improved Road sign and speed limit detection. The main contribution of this thesis is that the proposed model solves the essential problem of small object detection that is of utmost importance in the application domain of Road sign and Speed limit. Through exhaustive experimental investigation obtained not only better results in terms of small object detection but also significant reduction in detection time.

In order to further improve the inference speed since the proposed object detector needs to run in a computationally limited embedded system. Therefore, a small, fast, and lightweight feature extraction network is necessary and the base network of SSD that is originally VGG16 is also replaced by MobileNet_v2 architecture.

The methodology involves the following phases:



8. Significance of the study (contributions)

The importance of this study stems from its potential to increase road safety and advance autonomous driving technologies. Accurate Framework for road sign and Speed limit Detection is critical for guaranteeing safe and efficient traffic flow. This work can help a variety of stakeholders by providing a realistic solution for road sign and Speed limit Detection identification, including: Automotive industry, Government agencies, Traffic safety researchers, General public.

1.9. Organization of the thesis

Chapter 1: Introduction in this chapter Introduction to the research topic and its significance will be presented and Research objectives and research questions, Scope and limitations of the study, Significance of the study and Outline of the thesis chapters will be included.

in Chapter 2: Literature Review ,Overview of frame work for road sign and Speed limit Detection , Review of computer vision techniques and deep learning approaches , Analysis of existing deep learning architectures and image processing techniques for road sign and Speed limit Detection, Discussion of relevant studies and research in the field will be included

Chapter 3 will include Methodology, Description of the dataset used for training and evaluation, Pre-processing techniques for data preparation, Selection of deep learning architecture(s) and model configuration, Training and optimization procedures, Evaluation metrics and performance analysis

And Chapter 4 will include Deep Learningmodel for framework of road sign and Speed limit Detection, Detailed explanation of the chosen deep learning model(s) for Frame work, Discussion of the implemented image processing techniques, Presentation of the model architecture and its components, Explanation of the training process and optimization techniques applied

In Chapter 5: Experimental Results and Analysis, Presentation and analysis of the experimental results obtained from the developed deep learning model, Evaluation of the model's accuracy, robustness, and performance under varying conditions ,Discussion of the strengths, limitations, and potential improvements of the developed, system will be included

And lastly in Chapter 6: Conclusion and Future Work , Summary of the research findings and contributions, Recapitulation of the research objectives and their fulfilment , Discussion of the key insights obtained from the study, Limitations and areas for future research and improvement and Final remarks and conclusion will be included

Chapter 2

LITERATURE REVIEW AND RELATED WORKS

2.1. Overview

This chapter mainly focuses on the literature related to Road sign and speed limit. It contains the background information and a review of literature that are related to the domain of the thesis. The type of Road sing and speed limit are discussed. And also the data related to Speed Breaker Detection is stated briefly. available Technologies which has a contribution in detection of Road sign and Speed Breaker Detection such as digital image processing, Machine Learning, Computer vision and the basic concepts related to deep learning along with the state-of-the-art CNN architectures are illustrated well. Finally, a summary of related work is included in this chapter.

2.2. Importance of traffic sign and light detection as an ADAS feature

Lives can be saved by having ADAS that monitor the environment and warn or intervene in critical situations. The traffic control devices that communicate to drivers include the traffic signs, traffic lights and pavement markings. For this reason, automotive companies are focusing on ADAS research for improving safety.

Progress in machine learning methods has opened new gateways for enhancement in the automotive domain, especially for ADAS where object detection is the key function.

2.3 Basic concepts

In general, when people think of Artificial Intelligence (AI) today, mostly they mean machine learning i.e., training a machine to learn a desired behaviour.

Artificial Neural Network (ANN)

An artificial neural network is a computing system that is comprised of a collection of connected units called neurons, also known as nodes. They are organized in the layers. There are three types of layers in every ANN: input layer, hidden layers, and the output layer.

Deep Neural Network (DNN)

If an ANN has more than one hidden layer the ANN is said to be a deep artificial neural network or deep neural network.

Convolutional neural network (CNN)

CNN is a special class of deep neural network that is used for Computer Vision(CV) for analyzing visual imagery and text analysis for natural language processing (NLP). A simple CNN architecture is shown in Figure 2.1.

Its architecture is analogous to that of the connectivity pattern of neurons in the human brain. The neurons have learnable weights and biases. A simple CNN is a sequence of layers, and every layer of a CNN transforms one volume of activations to another through a differentiable function. The key role of the CNN is to minimize the images into a form that's easier to process, without losing critical features which are necessary for getting a good prediction. This is important when to design an architecture to be not only good at learning features but also scalable to massive datasets. The main advantage of CNN compared to its predecessors where the filters were hand engineered is that it learns directly from the input data with the automatic generation of feature maps. Another advantage of CNNs is by using 'transfer learning', that not only helps in case of smaller dataset but also decreases the training time significantly by converging faster.

Where the idea behind transfer learning is taking a model trained on one task and applying it to another similar task. The idea is that a model has already some or all of the weights for the second task trained so no need to train from scratch and the model can be implemented much quicker. There are two ways to use transfer learning:

• Fine tuning a CNN

• Using the CNN as a fixed feature extractor

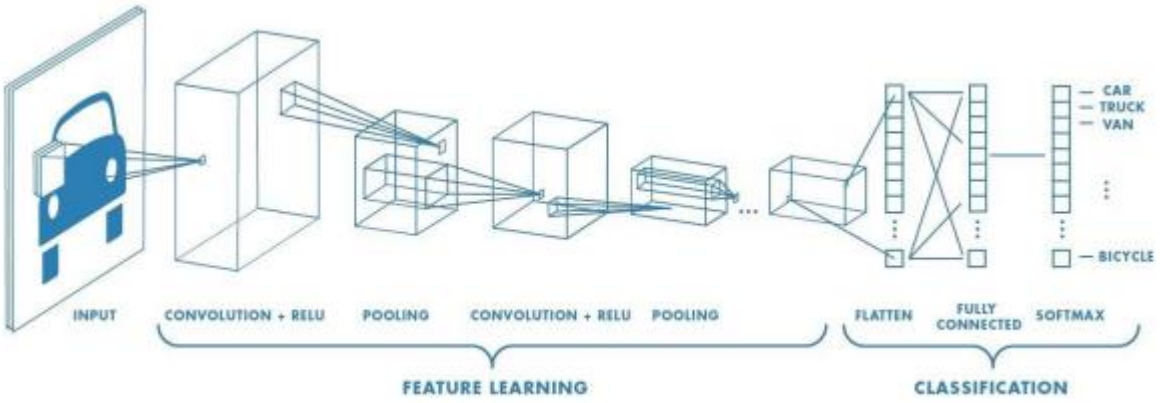


Figure 1 A convolutional neural networ 1 architecture and its different layers [11]

CNNs need not be limited to only one Convolutional Layer. It consists of many stacked layers, as shown in Figure 2.1, where each layer learns unique features from the input images. Conventionally, the first convolution layer is responsible for capturing the low-level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the high-level features as well, render a network, which has the wholesome understanding of images in the dataset.

There are three main types of layers to build a CNN architecture:

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

Some other layers with a specific objective are as follows:

- ReLU (Rectified Linear Unit) layer
- Normalization layer
- L2 regularization

These layers are stacked to form a full CNN architecture.

Convolutional layers

The convolutional layers are the main building blocks of a CNN model. Automatically detecting meaningful features given only an image and a label is not an easy task. Convolution is a mathematical operation to combine two sets of information [12]. Convolution is applied on the input data using a convolution filter or kernel to produce a feature map, as shown in Figure 2.2. Multiple convolutions are performed on an input image each using a different filter and resulting in a distinct feature map, as shown in Figure 2.3. All these feature maps are then stacked together that becomes the final output of the convolutional layer.

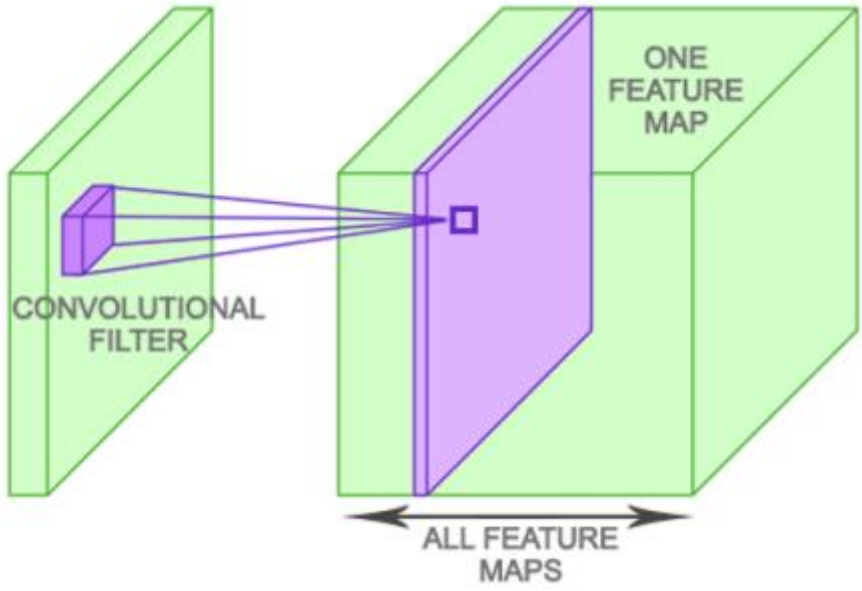


Figure 2 Convolutional operation between a kernel and input data to produce a feature map

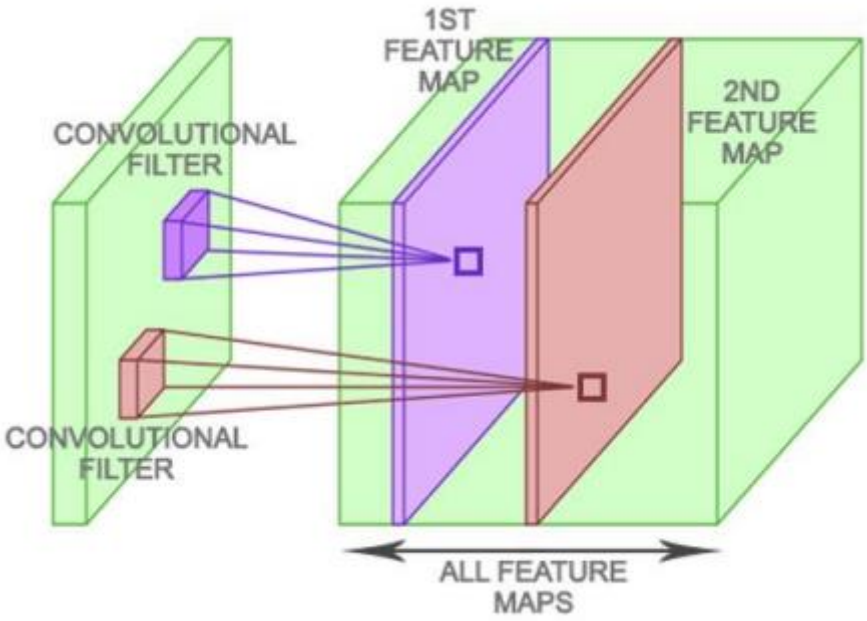


Figure 3 Result of a convolution operation with two kernels performed independently and feature maps stacked along the depth dimension

Pooling layer

While the objective of the convolutional layer is to detect local conjunctions of features from the previous layer, the role of the pooling layer is to amalgamate semantically similar features into one.

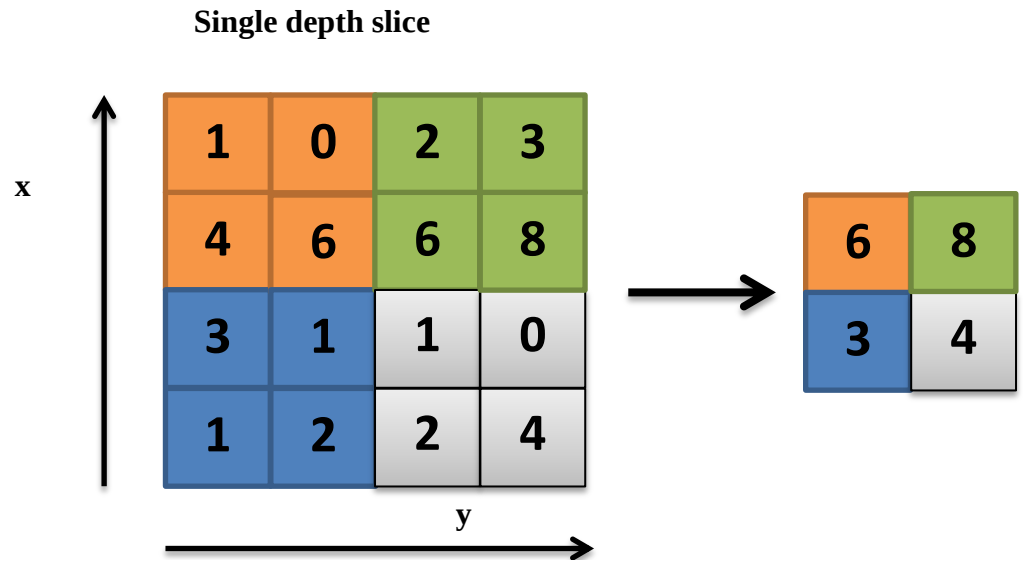


Figure 4 pooling operation

Pooling helps to minimize the number of parameters that not only shortens the training time but also controls overfitting [14]. They are usually placed after the convolutional layer with a down sampling factor of 2. Contrary to the convolution, pooling has no parameters. There are two types of pooling, as shown in Figure 2.5, where ***max pooling*** is the most common type.

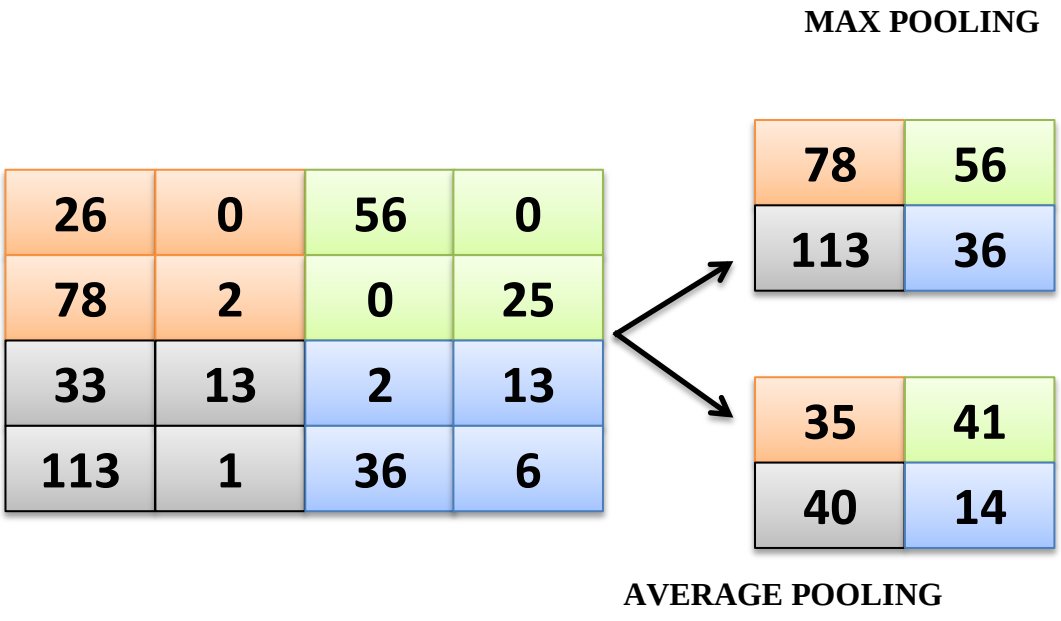


Figure 5 Two types of pooling operations

Fully connected layer

The neurons in a fully connected layer have full connections to all activations in the previous layer [14]. The output of both convolution and pooling layers are 3D volumes while a fully connected layer expects a 1D vector of numbers. That is why the output of the last pooling layer is flattened to a 1D vector that becomes the input to a fully connected layer, as shown in Figure 2.6.

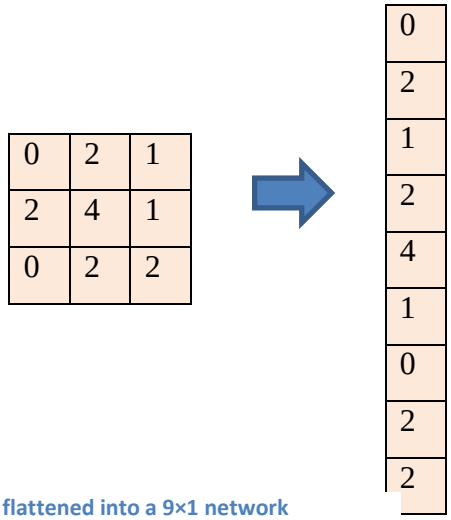


Figure 6 A 3×3 image matrix flattened into a 9×1 network

ReLU layer

For any kind of a neural network to be robust enough it must contain non-linearity. It makes it easy for the model to adapt or generalize with variety of data and to differentiate between the output [12]. The ReLU layer applies a non-linear thresholding function where the negative values are set to zero and the positive values have no variation [15], to each element of a feature map , as shown in Figure 2.7.

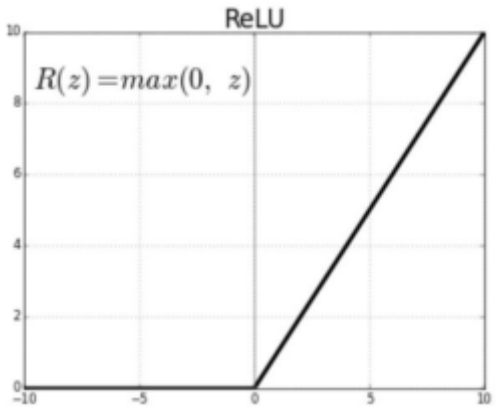


Figure 7 ReLU layer with associated thresholding function applied to the input data[15]

Batch normalization layer

Batch normalization is a technique proposed by Ioffe et al. [16]. The fact that, during training, the distribution of each layer's inputs changes as the parameters of the previous layer's changes. This phenomenon is referred to as internal covariate shift.

Batch normalization is a technique that addresses this complication by performing the normalization for each training mini batch. This way not only permits to use higher learning rates and be less mindful about initialization but also works as a regularizer, minimizing the need for Dropout layers [17] that are typically used to reduce overfitting.

L2 Regularization

To best map inputs to outputs the neural networks learn a set of weights. An unstable network is one where small change in the input can lead to large output changes.It happens when the network has large weights and its a sign that the network has overfit the training dataset [18].

Regularization is a technique used to control overfitting by adding a penalty term to the loss function. The penalty term also known as regularization parameter or weight decay determines the amount of penalty that is the square magnitude of all parameters, added to the weights of the model.

Softmax Classifier

Softmax classifier is a popular multi-class classifier. It uses Softmax function as activation function and described below in equation:

$$fj(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \qquad (2.1)$$

Here the Softmax function takes an input vector z with arbitrary scores for each class j and outputs a vector with values in the range 0 and 1 for each class j. The output vector has the property that the sum of all its elements is equal to 1.0, making it suitable for a probabilistic interpretation that is very useful in machine learning. Softmax normalization is a way of reducing the influence of outliers or extreme values in the data without removing data points from the set.

2.4 Overview of Road sign and Speed limit Detection

Road surface monitoring is currently performed by humans, but the road surface condition is one of the main indicators of road quality, and it may drastically affect fuel consumption and the safety of both drivers and pedestrians [8]. Authors further suggested solution that uses a gyro, an accelerometer, and a GPS sensor mounted on a car. The methodology they suggested had the potentials to detect speed bumps in quasi real-time conditions, and can be used to construct a real-time surface monitoring system. Due to the bad roads that have caused damages to vehicles and also lead to accident, authors in [9], proposes a system that provides a cost-effective solution to timely alerting to the driver regarding potholes or hump. The proposed system was divided into three sub-units: the sensing sub-unit, the server sub-unit and the user sub-unit. At the sensing sub-unit, an ultrasonic sensor is used to detect potholes and humps, whose location co-ordinates are retrieved by the GPS receiver. This data is stored to the database, which is the sever sub-unit. At the user sub-unit, a hardware module is set-up that provides timely alerts to the drivers regarding potholes and humps.

In [10] a system that detects speed bumps using colour segmentation and knowledge base object detection was proposed. The work done by Afrin et al [11] was to help in preventing accident caused by unmarked speed breakers. They then came up with a system that facilitates autonomous speed breaker data collection, dynamic speed breaker detection and warning generation for the on-road drivers. Their system also incorporates real-time tracking of driver, vehicle and timing information for speed breaker rule violations. The system they proposed outperformed the state-of-the-art works in terms of response time and accuracy.

As reported in [12] a summarized paper on road condition detection using dedicated sensors and smart phones was presented. In the same paper authors highlighted the disadvantage of using smart phones and GPS for road sign and speed bump detection. The drawbacks of the using smartphone system are vibration patterns of sensor data, benign events, GPS error, network overload, delay and battery draining. One of the most common methodologies for speed bump detection is using smartphone; the problem arises because of its hard code nature. It is so called, since the detection of speed bump is based on the previous history not based on current scenario so it is unfit for real time scenario.

BLOB analysis technique was used to detect the speed breaker in a given image [13]. Camera fixed in the vehicle was used to capture the image of the road and it was analyzed in real time to detect the presence of speed breaker. This technique demonstrated on painted speed breaker. The technique used by authors proved to be effective because it was designed to alert the driver before the vehicle hit the speed breaker.

Authors in [14] used image processing techniques to detect t h e speed breakers that lie along the roads and to hint the driver to carry on with velocity that makes for vehicle comfort. They came up with a conclusion that using image processing proves to be more effective and precise than using a sensor to detect the speed breakers.

According to Rahayu et al [6], speed breaker detection is important to community in order to save the car and human life. The most critical thing is preventing remedy leading to the cause of accident. Because many speed bumps are constructed without proper permission. Not notifying of speed bump over high speed is harmful for patients in transit, pregnant women, rapid wear and tear and damage to vehicles [2]. They went further to suggest the need to develop a system that services the end user driver using image processing algorithms such as Gaussian filtering, Median filtering and Connected Component Approach. The state-of-art algorithm suggested by Author works fails on datasets that includes stop signs markings. Hence there is a need for an algorithm that provides a solution to the state-of-art algorithm problem. We are basing our research on the methodology proposed by the authors; however, we went further by performing classification of the detected bump and included an OCR algorithm as a way of providing an alternative marking for speed reduction.

2.4.1 Optical Character Recognition

Optical character recognition (OCR) is the mechanical or electronic translation of images of handwritten or printed text into machine-editable text [15]. According to Mithe et al [16] it is a technology that enables you to convert different types of documents such as scanned paper

documents, PDF files or images captured by a digital camera into editable and searchable form. Character recognition system has received considerable attention in recent years due to the tremendous need for digitization of printed documents [17]. The OCR based system consists of image acquisition, image pre-processing, image segmentation, feature extraction, post-processing, and template-matching and correlation techniques [18] [19]. Images captured by a digital camera differ from scanned documents or image. They often have defects such as distortion at the edges and dimmed light, making it difficult for most OCR applications, to correctly recognize the text [16]. In light of that, [20] used OCR to detect text-based traffic signs but in other to improve the accuracy of recognition, the OCR results from several frames were combined together by matching individual words through frames and using a weighted histogram of results, Similarly, authors in [16] used "tesseract" an open source OCR engine to perform the character recognition because of its widespread approbation, its extensibility and flexibility, its community of active developers, and the fact that it "just works" out of the box. Due to the inconsistencies on the fonts used in the marks, OCR template-matching is picked for the proposed system methodology.

2. 4.2 Local Binary Pattern

Local Binary Pattern (LBP) is a non-parametric descriptor, which efficiently labels the pixels of an image with decimal numbers that encode the local structure around each pixel of an image [21] [22]. Perhaps the most important property of the LBP operator in real-world applications is its invariance against monotonic gray level changes caused, for example, by illumination variations. Another equally important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings [23

2. 4.3 Support Vector Machine

According to Pupale [24] Support Vector Machine (SVM) is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. SVM can be used for binary and multi classification tasks [22]. The idea of SVM is simple: The algorithm creates a line or a hyper plane which separates the data into classes. SVM [25] were originally designed for binary classification. How to effectively extend it for multiclass classification is still an ongoing research issue. Currently there are two types of approaches for multiclass SVM. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. Up to now there are still no comparisons which cover most of these methods. The formulation to solve multiclass SVM problems in one step has variables proportional to

the number of classes. Therefore, for multiclass SVM methods, either several binary classifiers have to be constructed or a larger optimization problem is needed [26]. SVM classifier was picked in light of its popularity and ease of usage.

2. 4.4 3D Reconstruction-Based Detection

In previous work [37] the road detection is detected using 3D LiDAR sensor based on a Markov random field (MRF) framework in unstructured and complex road environments. Gaisser and Jonker [43] developed an application for radar detection with the classification of visual data to evaluate different types of road sections using radar and camera. In [44] frequency modulated continuous wave (FMCW) radar is used to interferometric processing for height detection to obtain elevation information of terrain that is used to produce high-resolution topographic maps. Vo et al. [38] used aerial laser scanning and imagery data fusion to form coloured point clouds through the vector, local surface roughness and HSL (Hue Saturation Lightness). Multi-feature comparison based on 2D LIDAR [45] is used to classify different of obstacle and compare the width data, signal intensity data, range data, and signal intensity data variation in the database for an autonomous vehicle for path generation. Mobile mapping system with equipped laser scanning [46] is used to acquired densely sampled 3D measurements of the underlying surface for surface modelling framework. In other related work, speed breakers are detected using 3-axis accelerator meter, and Global Positioning System (GPS) and some of the researchers used the embedded accelerator meter sensor and GPS device. In the work of [47], the authors developed a road infrastructure monitoring system. The system is built through the integration of acceleration and angular rate sensor and GPS sensor embedded at the centre of gravity of the vehicles to transmit data to a server thru Wi-Fi as the training data from taxicab fleet to supervise classification machine learning for identifying the road surface. In [48], real-time is implemented using a Raspberry Pi with connected to several sensors like gyro, accelerometer and GPS sensor mounted in a car. In addition, [49] the speed breaker detection system is done with a collision sensor, three-axis accelerometer, GSM and GPS sensor. The system can detect the type of accident from the accelerometer, posture after crashing of vehicle and GPS ground speed. This application cannot predict the ensuing accident but can alert other road users via alarm message and position of an accident after the accident occurs.

The drawback of using lidar is the misclassification of speed bump. Lidar may not penetrate to the ground surface in densely vegetated areas, producing an anomalous elevation at those points that may be significantly higher than the actual elevation and lead to potential

misclassification of environment but can use to avoid road congestion. Despite this, there is some valuable information like detailed and accurate elevation measurements that is not collected by machines but monitored by humans.

2.4.5 Vibration-Based Detection

Carlos et al. [39] used mobile Smartphone to evaluate road anomalies. It detected pothole, metal bump, crack and uneven using accelerometer sensor inbuilt in a smartphone. Dynamic Time Warping (DTW) algorithm [50] used data from accelerometer and GPS receiver of the smartphone to classify the speed breaker and store the location in the database as well as an early warning to another driver toward the location. Q-parameterization approach [40, 51–53] used the Android OS smartphone to operate the GPS, accelerometer and magnetometer sensor for analysing road quality and Ghat complexity as well as transmitting the data to the central server. This approach is also called as Nericell systematizes, in which the sensor in the smartphones are used to alert and transmit data back to a server for collection. An unsupervised method, i.e., K-means clustering technique is used to cluster the data collected from smartphone gravity sensor and GPS sensor to identify speeds where the bump is present [54].

Seraj et al. [55] used smartphones equipped with GPS and inertial sensors to detecting and classify road surface. They implemented wavelet transform analysis and an approach using envelope techniques to remove velocity, slope and drift effects from sensor signals. The vector of characteristics with considers maximum and minimum information of 3 axes in accelerometer reading was evaluated in [56] for the identification of road surface. Most procedures are based on heuristic methods where the increase of the z-axis signal is measured with respect to a previously defined threshold. Evidence has been found that the other axes of the accelerometer also suffer disturbances and its characterisation can be used to improve the percentages of classification accuracy.

The work carried out in [35, 36] provided a summary of the detection of road surface using dedicated sensors and smartphones. It was reported that the function of the smartphone as an ad hoc tool to accumulate road information, some of the researcher established an application where inbuilt inside the smartphone to evaluate the data from the vehicle such as road surface condition like potholes, bumps, hump, uneven road, cracks as well as smooth road. Moreover, in the [50, 54] the early warning systems are implemented to react to imminent frontal collisions using smartphone and accelerometer.

The drawbacks of the using vibration-based detection are the readings are less accurate and cannot differentiate the type of road surface condition. In most of the vibration-based detection methods for speed breakers detection used the inbuilt sensor in a smartphone because it is easy to implement and hard code nature. The detection of the speed breaker is based on the previous database but not the current situation, so it is not suitable for the real-time scenario. Also striking is the lack of public datasets that allow comparing and contrasting solution procedures. To date, all authors have demonstrated the usefulness of methods in their datasets, without the real capacity to compare the scope and limitations of the results. It is necessary to open to the community these experiments, to allow for validation of the proposed methods.

2.4.6 Computer Vision-Based Detection

Devapriya et al. [57] proposed an approach to detect speed bump using RGB to grayscale conversion, grey to binary conversion, morphological image processing and to compute projections. Nonetheless, such a simple detection method is only suitable to detect speed bumps designed with zebra crossing pattern. In [42], the authors proposed road anomaly detection method to sense of speed bump/road anomaly in real time. Their focus is on analysing camera motion and infers the existence of suspect road anomalies by implement vehicle shake detection, pavement region segmentation, road surface saliency extraction, and speed bump/road anomaly classification.

Gaussian Filtering and Connected Component Approach [58] are used to detect the speed bump in real time. The methodology begins with simple preprocessing with resize and RGB to Grayscale conversion followed by Gaussian and median filtering. The resultant image is subtracted with highlight the edge variation and applied thresholding technique for classified the speed bump through the neural network. In the [59], the hump is detected from the camera fixed at the low height of the vehicle using dilation and erosion process. The edge detection technique is used to establish vanishing point which identified by straight lines for road segmentation. The authors in [60], proposed vision-based detection of road bump using camcorder. The road bump detection processing steps include time-sliced image generation, vertical motion analysis, and road bump locating by means of Open Source Computer Vision (OpenCV). Mean Square Error is used to categorise the bump by comparing the time sliced image vertical displacement and the previous slice in term of the pixels.

In [61], a novel detection method is proposed for identifying pothole using a black box camera in real time. The proposal method algorithm differentiated potholes in the region of

interest (ROI) by inspecting several features like dimension, variance, range and trajectory. Jeong-Kyun and Kuk-Jin [62], used free space estimation, digital elevation map (DEM) estimation, and road surface profile (RSP) estimation to estimate drive area for vehicles. Ordinarily, the RSP estimation is inconsistent while the speed bump is observed through DEM reference grid that generated from fitted road model. However, recalculated the reference grid using camera height, pitch, and roll angle errors data in 3D geometric for predicted the DEM. This temporally consistent elevation value from DEM is considered detection of speed bump where the pitch angle of the vehicle is sudden changes.

The work carried out in [25, 36] presented a summary on the detection of speed breaker and pothole using camera/video recorder/smartphone via computer vision technique. In the summary, some of the researchers also trained in machine learning to map with the control system of the vehicle for alerted and assisted the driver during the journey. In preview works [42, 60, 61], researchers have proposed to automatically detect and send the road anomalies information to a respective government department for rapid the road surface maintenance. In computer vision, different approaches can be implemented, for instance, shape segmentation, and differences of the texture of regular pavement have been exploited for in identifying potholes [61]. Similar methods by applying a shape, edges, contour information and template model fitting in OpenCV can detect another type of road infrastructure. The drawbacks of the using vision-based detection depend on expensive camera/video recorder and the processing time with common problems related to light intensity and the statistics that anomalies because do not have a proper prototypical design.

2.5 Machine Learning

Machine learning is a method of data analysis that provides a meaningful interpretation of the data in which conventional statistical methods are unable to provide. It is a field of artificial intelligence that employs mathematical models to provide classification or prediction based on the data provided to the model. Different machine learning models have been reported in the literature in facilitating the identification as well as classification of speed breakers.

2.3.1 Support Vector Machine (SVM)

Support vector machine is a supervised learning algorithm which can be used for both classification and regression analysis. SVM is merely the identification of hyperplanes that best segregates the data into different classes [35, 37, 39, 42, 47, 55]. The 3D point cloud data [37] are used to classify various road environment like flat road, uphill, downhill, and sloped road by means of SVM, its classification performance is compared to that of the radial basis

function model. In was shown that the proposed method shows good classification accuracy (approximately 91%), error rate, precision rate, and recall rate compared with other research work [63-65]. In [47], the inertial sensor data filtered by Fast Fourier Transformation wavelets for train new vehicles based on the trained SVM trajectory. The data from vertical acceleration and pitch rate of the new vehicles under controlled condition illustrate that the calculation time is faster compare with other calculation time with different distance algorithms [66]. SVM classifier is used to identify the section of acceleration data for detection of road anomalies [39]. The best F-measure score (Test's accuracy which can interpret the harmonic of precision and sensitivity) on average in the overall comparison, outstanding performance in this proposed algorithm to detect the anomalies on the road. The average of the F-measure score for all detectors is shown the correct detection on the real road scenarios. Seraj et al. [55] also used SVM to classify road anomaly. The radial basis function (RBF) kernel-based SVM is was shown to provide an 88.78% classification accuracy. Support Vector Machine [42] is implemented for the classification of speed bumps and road anomalies (non-speed bumps) in visual dictionary generation. There is different road anomaly type detected during the experiment such as a pothole, bump, road, patch, manhole, uneven road and the proposed method has the outstanding performance compare with shake detection with a classification accuracy of bump detection [60].

Support Vector Machine is the most common machine learning methods used the identification of speed breakers as well as road anomalies. Nevertheless, it is worth noting that SVM does not accommodate word structure as it will lose sequentially information and leads to a poor performance for the Bag-of-Words cases.

2.5.2 Artificial Neural Networks (ANN)

The concept of Neural Network (NN) is brain-inspired algorithm capable of learning based on training and environment instead of programming (if-else). NN have emerged as the primary means of implementing deep learning in the autonomous driving system, with specialised hardware accelerators running in vehicle NN to classify object like pedestrians and road sign. In [43], convolution neural network (CNN) used vision-based road user detection with the contrastive loss for classification of road section. This method is trained on a dataset to learn fully visible pedestrians and road sign. Consequently, the classification accuracy of the CNN based system is 95.1% while driving on the public road which is an increase of 7.6% reported in [67]. Moohyun et al. [45] employed ANN to classify obstacle type. It was demonstrated in the study that the ANN approach provided a higher classification

accuracy against an existing conventional The comparison of classification between the proposed approach and the existing approach [68]. Five hundred different Gaussian filtered sample images were trained with NN to detect speed bumps in [58]. It was shown that the NN model is able to recognize accurately 90% of the painted speed bumps through their pattern, colour and dimension variety information. Logistic Regression (LR) model was compared with NN model in classifying irregularity events on roads and highways [56]. It was shown from the investigation that the NN model is better at classifying the irregularities in comparison to the LR model.

Artificial Neural Network has demonstrated its capability in identifying or classifying speed breakers and road irregularities. This is primarily due to its ability to learn the non-linear behavior of the data better than other machine learning models, in the event more data is supplied to the model. Nevertheless, other variation of NN models has yet been investigated in classifying speed breakers for instance Generative Adversarial Networks, that is worth looking into.

2.5.3 Other Machine Learning Models

Apart from SVM and ANN, other machine learning models have been investigated in autonomous driving, albeit not as extensive as the former models. For instance, decision tree classifier has been used to identify speed breakers through comparing the differences caused by various ways of representing normal vector and colour [38]. In [48], genetic algorithm optimised logistic model has been applied to detect speed bumps accurately based on accelerometric features.

González et al. [36] applied and compared seven popular machine learning techniques to classify the roadway surface disruption. The models evaluated are ANN, SVM, Decision Trees (DT), Random Forrest (RF), Naive Bayes Classifier (NB), K-nearest neighbours (KNN) and Kernel Ridge (KR). It was shown from the study that the ANN model provided the highest classification accuracy. Nonetheless, it is worth noting that the SVM and KR models provided reasonably good predictions too. Although different machine learning models exits, however, the selection of the suitable model is dependent on the nature of the data itself, which evidently augurs well with the “no free lunch theorem” [9].

Types of speed breaker

Type	picture
Rubber Speed Hump	
(Asphalt Speed Bump)	
Rumble Strips	

Bump and Pothole	

Figure 8 types of speed breaker

2.6 Object detection

In the field of computer vision, object detection occupies an important position and is one of the main research areas. The main goal of object detection is to find objects in an image (object localization) while object classification is used to determine the class of the object among a predefined set of categories [27].

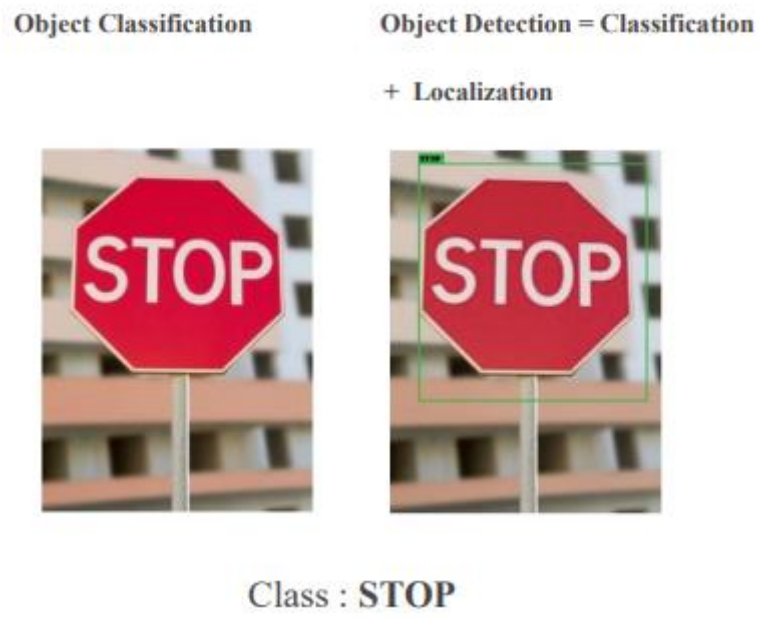


Figure 9 Difference between object classification (left) and object detection (right)

2.6.1 Traditional object

detection Traditional object detection algorithms follow a typical pipeline with following procedures,

Region selection

Traditional methods scan the entire image to find objects in an image by applying sliding windows. The sliding windows are of distinct scales and sizes and generate smaller image crops that are analyzed individually later to determine if there is an object inside it. The process is computationally expensive due to the enormous number of analyzed candidates [27].

Feature extraction

The candidates generated during the sliding windows process are analyzed using the visual features. The visual features are a way to know about the image as they give meaningful information. Popular visual features include scale invariant feature transform (SIFT) features [28] that have the property of being invariant to image scale and rotation, histogram of oriented gradient (HOG) features [29] used in human detection and Haar-like features [30] used in face recognition. Although, to detect specific type of objects most of feature descriptors are designed but their performance could be affected by illumination conditions.

Classification

After obtaining the feature descriptor vector of each sliding window the next step is classification. The image crops are classified in a background and target object class. The most commonly used algorithm for classification purposes is support vector machine (SVM) [31].

Non-maximum suppression (NMS)

Many candidates are generated during the sliding window process. For the filteration of most significant results, NMS is used. NMS selects the candidates with the highest scores as the result of the object detector. This is discussed later in Chapter 4.

2.6.2 CNN based object detection

In recent years, object detectors based on CNNs have become immensely popular due to the success of the application of CNN architectures (for example, VGGNet [23], ResNet [24]) as feature extractors. The choice of feature extractor is important as the number of parameters and types of layers directly affect memory, speed, and performance of the detector. While

some of the object detectors based on CNNs with a different architecture have achieved state of the art performances in terms of accuracy and good detection speed to be deployed in mobile devices [32]. CNN networks with deep architectures are capable to learn more sophisticated features by finding complex patterns in images. That's why as compared to manually designed features, features learned by CNNs are more robust. This enables CNN architectures to be more suitable for a variety of applications allowing a model to be trained with different datasets [27].

Generic object detectors

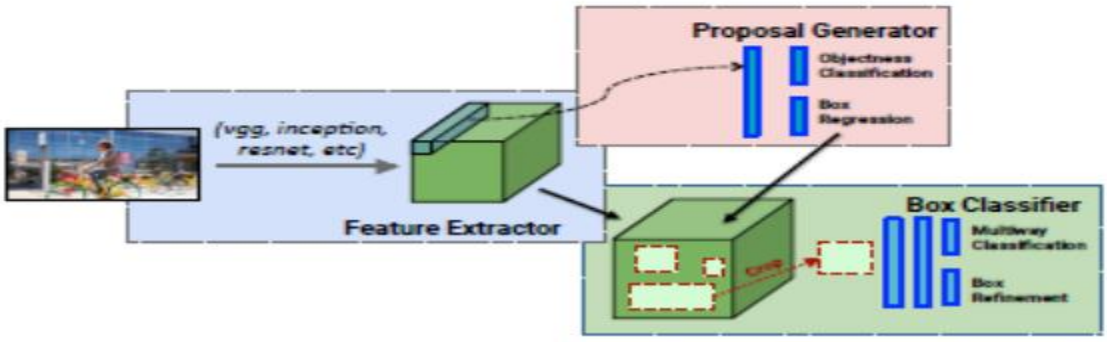
Generic object detectors based on CNNs have the objective of identifying the objects in an image and show the position of the objects by drawing a rectangular bounding box around them. Generic CNN based object detection methods can be classified into two main groups: two-stage detectors and single-stage detectors.

Two-stage detectors

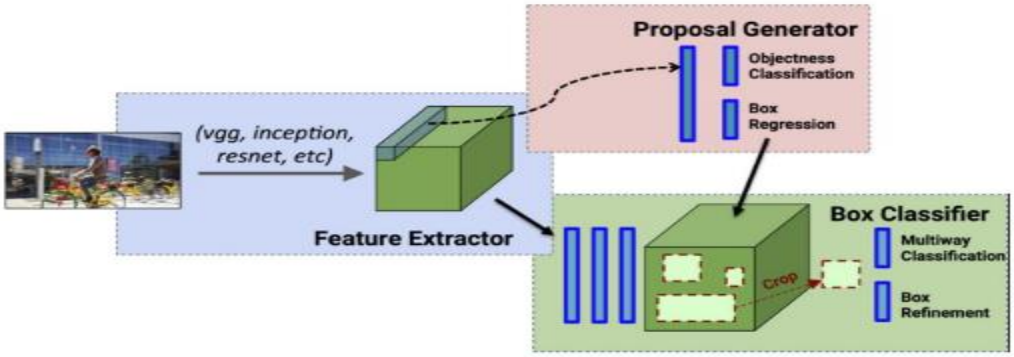
Two-stage detector frameworks consists of a two-step process. In the first stage, the algorithm focuses on generating region of interest or proposals and in the second stage classifying each region of interest into predefined object classes, (Figure 3.2(a) & (b)). Successful algorithms in this category are Region-based fully convolutional networks (RFCN) [8] and Faster region-based convolutional neural networks (R-CNN) [7].

In Faster R-CNN, a base network (VGGNet) is used as a feature extractor and features from an intermediate feature map are selected for proposal generation (300 as mentioned in the paper [7]). In the second stage, these feature proposals combined with the output of the base network are used to predict object classes and bounding box coordinates.

R-FCN [8] follows a similar process as in Faster R-CNN except that the regions of interest or proposals are generated from the output of the base network instead of an intermediate feature map. This change brings a decrease in computational power needed while increasing the speed of detection and training. It also achieves similar accuracy as Faster R-CNN.



(a) Faster R-CNN



(b) R- FCN

Figure 3.2(a) & (b) High-level diagram for two-stage detectors [32]

Single-stage detectors

Single-stage detectors use a CNN to directly predict the object class and location as a regression problem in a single pass without a second stage per-proposal classification

operation (Figure 3.3). The focal point of single-stage detectors is to improve the detection speed; however, their accuracy is lower as compared to two-stage detectors.

Popular algorithms in single-stage category are, You only look once (YOLO) [9] and Single shot multibox detector (SSD) [10]. In case of YOLO, the input image is divided in an $S \times S$ grid, where S denotes an integer. For each grid, the network predicts B bounding boxes and confidence scores. Therefore, the number of grids S and bounding boxes B are the hyperparameters of the algorithm. SSD uses default anchor or prior boxes with different aspect ratios and scales to make predictions in six feature maps, with the objective to predict objects with different scales and shapes. The output generated by the network is in the form of scores for the presence of objects and bounding box coordinates.

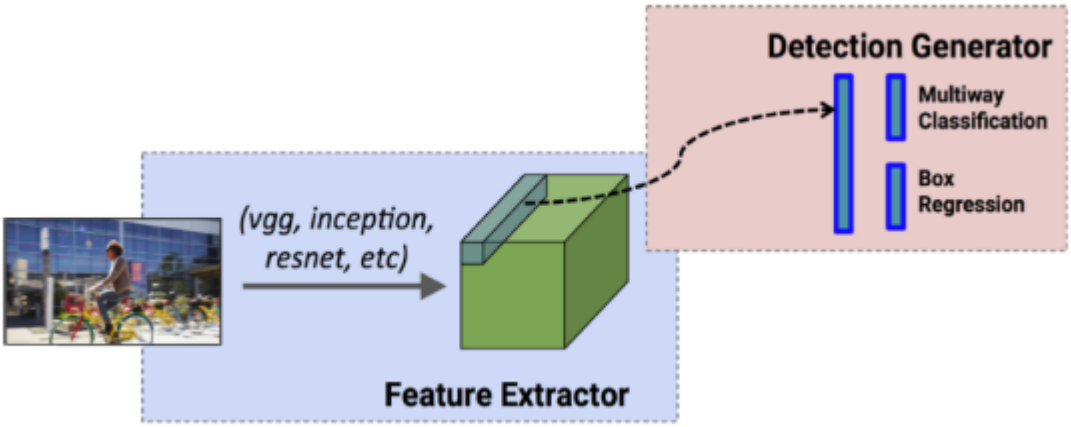


Figure 10 High level diagram for single stage detectors [32]

2.6.3 Evaluation metrics

To measure the performance of a model, metrics like accuracy, recall, precision, mean average precision (mAP) are used. The performance of the classification task is evaluated depending on, whether the image contain any instances of a certain object class. And in case of detection task depends on, where in the image, are the instances of a particular object class (if any)?

$$Precision = \frac{\text{True positive}}{\text{True positive+False positive}} \tag{3.1}$$

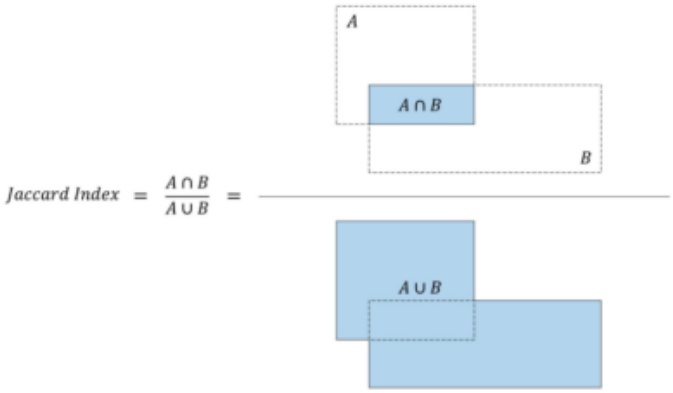
$$Recall = \frac{\text{True positive}}{\text{True positive+False negative}} \tag{3.2}$$

Here, True Positive is the number of predictions that has intersection over union (IoU) [explained below] greater than 0.5 with ground truth boxes, False Positive is the number of predictions with IoU less than 0.5 with ground truth boxes and False Negative is the number of ground truth boxes that are not detected by the model.

Intersection over union

Intersection over union also known as the Jacard index is a metric used for the bounding boxes evaluation. It is a ratio between the intersection and the union of the area of predicted boxes (Apred) and the area of the ground truth boxes (Agt), as shown in Figure 3.4. IoU is the metric to measure the overlap between two boundaries to determine if a default bounding box corresponds to the ground truth box or the background. An IoU threshold (like 0.5) is predefined in some datasets to classify whether the prediction is a true positive or a false positive.

$$IoU = \frac{Apred \cap Agt}{Apred \cup Agt} \tag{3.3}$$



Mean Average Precision

Mean average precision is the primary metric of evaluation for object detectors. The predictions are sorted by their confidence score from highest to lowest. The eleven different confidence thresholds called ranks are selected such that the recall at those confidence values have eleven values ranged from 0 to 1 by an interval of 0.1. Average precision (AP) is computed as the average of maximum precision values at the chosen eleven recall values. Therefore, AP for class c is defined as by equation (3.4) where P(r) is the precision value for one of the 11 recalls r,

$$AP_C = \frac{1}{11}\sum_{r \in \{0.0,\ldots\ldots\ldots\ldots 0.1\}}^{R} \max(p(r)) \qquad (3.4)$$

$$mAP = \frac{1}{C}\sum_{c}^{C} AP_C \qquad (3.5)$$

In case of the Pascal Visual object classes (VOC) dataset, the metric is calculated for an IoU threshold of 0.5. For the Common Objects in Context (COCO) dataset the mAP is calculated as an average of 10 different IoU thresholds from 0.5 to 0.95 in intervals of 0.05. Taking an average of 10 IoU thresholds, this method complements models that have better localization precision. We use mean average precision (mAP) as performance of model.

2.6.4 State-of-the art in traffic sign and traffic light detection

Most of the related work is either for the detection of traffic sign or traffic light, but not both. In retrospect, each of the state of the art for traffic sign and traffic light detection methods are

separated by two approaches. One is conventional and merges the distinct features of traffic signs and traffic lights. The other one is characterized by deep learning that can self-learn different features.

2.6.4.1 Conventional approach

Conventional detection methods solely depend on the feature extraction algorithms. Let it be a classification or detection problem, these methods mandatorily require low-level features, like color intensities, edge detail and shape features.

Traffic light detection

The methods are mainly focused on hand crafted features like color, brightness, and shape of the traffic light bulb. Various publications on traffic light detection depending on color are [33], [34], [35], [36], [37]. Mostly, they use a simple thresholding in different color spaces [Red green blue (RGB), Hue Saturation Value (HSV)]. De Charette et al. [38] proposed a popular approach that uses the white top-hat operator to extract blobs from grayscale images. The methods using the shape of the traffic light bulb for proposal generation are [33], [39], [40]. [41] used stereo vision as an additional source.

Traffic sign detection

Color based techniques use the color of the traffic signs as the main feature to locate on the image [42]. Supreeth et al. [43] used color clues, to be specific, the traffic sign's color detail as the main feature to locate the traffic signs on the input road scenes. For this purpose, the images were transformed to gray scale from RGB color space then the algorithms selected region candidates in the image applying shape and size constraints, cropped and saved the selected regions, and passed them through a neural network for classification. Nguwi et al. [44] used the Hue-Saturation-Intensity (HSI) color space to locate road signs and neural networks for classification. Romdhane et al. [45] used HueSaturation-Value color space to generate candidate regions. HOG features were used as feature descriptors, and an SVM classifier was used to classify the traffic sign category.

Shape based segmentation relies on the assumption that traffic signs come in a regular shape (triangle, rectangle, circle, octagon). Nguyen et al. [46] used Hough transform to detect general shapes (rectangles, circles, etc.) and is complemented with edge detection to detect speed limit and warning signs. Yang et al. [47] used a color probability model to generate

areas of interest, alongside with HOG features as a feature descriptor, while a CNN perform the classification.

Pros and cons of conventional approach

The conventional approaches predominantly depend on the feature extraction algorithms or descriptors. Although such hand-crafted features have achieved a higher precision for traffic signs and traffic lights, they lack robustness as a whole system. For instance, the models perform efficiently on linear features as compared to non-linear features. So, the models are feature specific and not generalize well.

2.6.4.2 Deep learning approach

Recently, deep learning methods has been popular among computer vision research. As mentioned earlier, the main categories for generic object detectors based on deep learning are two-stage detectors [7, 8] and single-stage detectors [9, 10]. They use variants of the generic object detectors such as SPP-Net [48], Fast R-CNN [49], and Faster R-CNN [7].As an advanced solutions for the problem of traffic sign and light detection, researchers have focussed on building deep neural networks (DNNs).

Traffic light detection

Weber et al. [50] implemented a deep architecture, coined as DeepTLR for traffic lights detection in 2016. AlexNet was used for feature extraction. The network returns a pixel-segmented image and then they applied a bounding box regressor for each class of the traffic light. Results evaluated with IoU threshold greater than 0.25. Behrendt et al. [51] introduced a complete system for detection, tracking, and classification using a deep convolutional neural network (DCNN). By utilizing the general object detector, the SSD, Muller et al. [52] introduced a model only for traffic light detection. They used a deeper base network and modified prior boxes to increase accuracy for traffic lights detection. Evaluation of results done in terms of False Positives Per Image (FPPI). Following the work of [51], Yudin and Slavioglo [53] proposed another traffic light detector that is built based on fully convolutional network (FCN) for image segmentation. They used the FCN to get a heat map highlighting plausible areas of traffic lights, then employ a high speed clustering algorithm to obtain traffic light bounding boxes. Besides being a transfer learning approach, it has a very low precision of detection as compared to SSD based solutions, like in [52].

Traffic sign detection

On the other hand, for traffic sign detection, Zhang et al. [54] used a modified version of YOLOv2 [55] to detect traffic signs. They manipulated the size of filters, and the number of layers to obtain a balance between detection accuracy and speed. Results evaluated in terms of precision and recall separately. Zhu et al. [56] designed a custombuilt CNN architecture to target more on the smaller size objects, i.e., the objects like traffic-signs that occupy only a small fraction of a road scene. Such target-specific implementation also faces lack of generalization performance.

Pros and cons of deep learning approach

The deep learning-based solutions can achieve good robustness compared to the conventional counterparts, as they self learn the feature correspondence between the raw inputs and targets. However, they face the criticism for being hungry for data and compute power. The two stage detectors have advantages in classification precision and localization accuracy. However, high computational power is required for training and inference and real time implementation is hard to get with two stage detectors. While single stage methods are much faster because of the unified network structures but the process precision decrease. Another drawback with single stage detectors especially with SSD is the detection of small objects.

2.7. Digital Image processing

Image processing is the process of utilizing a digital computer to process a digital image, removing noise and any other anomalies [10]. Digital computers have become widely available and used for a variety of purposes throughout the previous four to five decades. Digital image processing is always a fascinating area since it improves pictorial information for human interpretation and image data processing for storage, transport, and representation for machine perception [11, 12].

Image processing is a technique used to improve row images received from cameras or sensors installed on aircrafts, satellites, or space crafts, or pictures taken in everyday life for a variety of applications. This field of image processing has advanced significantly in recent years and has spread to various fields of science and technology. Image processing mainly deals with image acquisition, image pre-processing, image segmentation, feature extraction, image detection and classification [13, 12]. In the following sections, we will see some image

processing techniques that include image acquisition, image pre-processing and feature extraction.

1. Image Acquisition. This is the first step or process of the fundamental steps of digital image processing. The initial stage in digital image processing is to acquire or obtain digital images using devices such as cameras or scanners. Real-world scenes are turned into digital image representations at this step. Scanners digitize actual prints or documents, whereas cameras acquire images by sensing light through sensors. To construct a discrete representation of the image, the analogue signals from the sensors are sampled and quantized. Image acquisition is affected by a variety of elements such as sensor qualities, lighting conditions, and camera settings. To capture accurate and high-quality photos, it is critical to consider aspects like as resolution, colour depth, and sensor sensitivity. The camera or scanner used has an impact on the total picture acquisition process. Understanding the principles behind sensors, sampling, and quantization is critical for obtaining accurate and reliable digital images in image capture. This stage lays the groundwork for later processing steps like pre-processing, augmentation, and analysis.
2. Image Enhancement:-This means improving an image's quality, such as by removing noise or changing the brightness and contrast. is a crucial stage in digital image processing that focuses on improving an image's visual look. It seeks to improve overall perception by improving image quality, highlighting vital elements, and highlighting important information. Image enhancing techniques include histogram equalization and spatial filtering. Histogram equalization is a technique that redistributes an image's pixel intensities to improve contrast. It expands the image's histogram to use the entire range of intensity values, resulting in increased visibility of details and overall image quality. Another approach often employed in image enhancement is spatial filtering. It involves employing filters to change pixel values based on their spatial relationships in the image. Spatial filters can be used to do smoothing, sharpening, and edge detection. These filters help in the enhancement of image features and the overall visual appearance. Histogram quantization and spatial filtering are both effective picture enhancement methods that allow image properties to be adjusted to produce

desired visual enhancements. Images can be made visually appealing, more clearly and easier to interpret by employing these strategies.

3. Image Restoration:- Image Restoration is the process of recovering an image that has been degraded by using a priori knowledge of the degradation phenomenon. These techniques are oriented toward modelling the degradation and applying the inverse process to recover the original image. Restoration improves image in some predefined sense. Image enhancement techniques are subjective processes, whereas image restoration techniques are objective processes.
4. Colour Image Processing In colour image processing, an abstract mathematical model known as colour space is used to characterize the colours in terms of intensity values. This colour space uses a three-dimensional coordinate system. For different types of applications, a number of different colour spaces exists.
5. Wavelets and Multi-resolution Processing Wavelet transform is used to analyze a signal (image) into different frequency components at different resolution scales (i.e. multiresolution). This allows revealing image's spatial and frequency attributes simultaneously. In addition, features that might go undetected at one resolution may be easy to spot at another. The multiresolution theory incorporates image pyramid and subband coding techniques.
6. Compression Image compression is a process applied to a graphics file to minimize its size in bytes without degrading image quality below an acceptable threshold. By reducing the file size, more images can be stored in a given amount of disk or memory space.
7. Morphological Processing Morphological processing is a collection of operations related to the shape or morphology of features in an image. There are four operations for morphological processing which are dilation, erosion, opening, and closing.
8. Segmentation This involves dividing an image into different regions, such as by Colour or texture
9. Representation & Description:- A digital image is an image composed of picture elements, also known as pixels, each with finite, discrete quantities of numeric representation for its intensity or gray level that is an output from its two-dimensional functions fed as input by its spatial coordinates denoted with x, y on the x-axis and y-axis, respectively.[1] Depending on whether the image resolution

is fixed, it may be of vector or raster type. By itself, the term "digital image" usually refers to raster images or bitmapped images (as opposed to vector images).

10. Knowledge base Knowledge is the last stage in DIP. In this stage, important information of the image is located, which limits the searching processes. The knowledge base is very complex when the image database has a high-resolution satellite.
11. Object recognition This involves identifying objects or scenes in an image.

2.7.1. Image Pre-processing

Image preprocessing is the major procedures to extract appropriate features from image. Regardless matter which image acquisition devices are utilized, the images with noise are always unsatisfactory. For instance, there are noises in the image, the image's region of interest is not clear, environmental elements or other objects' interference are present in the image, and so on. For different picture applications, different preprocessing methods will be chosen [15]. Picture resizing, image enhancement, segmentation, noise removal filtering, gray scale conversion, image histogram, and thresholding are all employed in this application.

2.7.2. Image Segmentation

The quality of the result is determined by picture segmentation, which is an important part of the image analysis technique. It is the technique of separating or grouping different portions of an image. The extent to which this subdivision is carried out is determined by the problem at hand [11, 16]. The image is frequently segmented during segmentation until the objects of interest are isolated from their backdrop. Picture segmentation can be accomplished in a variety of ways, ranging from simple thresholding to complex color image segmentation approaches. Normally, these components correspond to something that people can easily separate and view as independent things. Because computers lack the ability to recognize objects intelligently, numerous different methods for segmenting photos have been devised [12, 14].

The segmentation process is based on the image's numerous attributes. This could be color information, image boundaries, or a segment. Two approaches for segmenting gray level values are used in segmentation algorithms. The first is based on gray level discontinuity, which partitions a picture based on unexpected changes in gray level, while the second is based on gray level similarity, which employs thresholding and region growth. Based on the

findings of the histogram analysis, the threshold value was generated, which is expected to be constant in all photographs with the same lighting conditions [17, 14].

2.7.3. Feature Extraction

The technique of extracting relevant information that can characterize an image is known as feature extraction. It can also be defined as the process of identifying an image's basic qualities or attributes. The extraction of sufficient information, which leads to the description of the kind and character of image, is one of the essential ideas in image analysis [14]. Using a combination of color and texture features to detect and classify diseases is more accurate than using each component separately.

2.8. Deep learning

Deep learning as a subfield of machine learning that uses algorithms inspired by the structure and function of the brains [18, 19]. Artificial neural networks are the models used in deep learning that are based on neural networks (ANN). Simply described, artificial neural networks are computing systems inspired by the brain's neural networks. These networks are made up of a collection of connected units known as artificial neurons or simply neurons. Each connection between these neurons can send a signal from one neuron to another, which is subsequently processed by the receiving neuron. Then it sends messages to downstream neurons attached to it, which are normally grouped in layers. Different layers may apply various transformations to their inputs and outputs. Travel from the first layer, known as the input layer, to the final layer, known as the output layer. Any layers between the input and output layers are called hidden layers. Because of numerous powerful computers (inexpensive processing units such as GPU) and a significant amount of data, neural network applications have grown quicker than ever in the recent decade. There are one or more processing levels in an ANN. Depending on the problem we want to solve the number of layers we use in the network defers. The network shallow architecture is defined as having a small number of layers, such as two or three. The network is referred regarded as deep architecture when it comprises a significant number of layers, and deep learning refers to this deep architecture of NN [19].

Deep learning entails layer-by-layer analysis of the input, with each layer extracting higherlevel information about the input. Let's look at a simple picture analysis scenario. Assume that the pixels in your input image are divided into a rectangular grid. The pixels are now abstracted by the first layer. The image's edges are understood by the second layer. The

following layer creates nodes from the edges. The following step would be to find branches from the nodes. The output layer will finally identify and classify the entire item. The feature extraction procedure goes from one layer's output to the next layer's input in this case. We can process a large number of features using this method, making deep learning a very powerful tool. Deep learning algorithms try to leverage the unknown structure in the input distribution to find good representations, generally at several levels, with higher level learnt features defined in terms of lower level features [18].

For decades, developing a pattern recognition or machine learning system necessitated extensive domain knowledge and meticulous hand engineering to develop a feature extractor that converted raw data (such as image pixel values) into a suitable internal representation or feature vector from which the learning system, such as a classifier, could detect or classify patterns in the input. Deep learning allows inputting the raw data (pixels in case of image data) to the learning algorithm without first extracting features or defining a feature vector. The traditional machine learning algorithm needs separate hand-tuned feature extraction before the machine learning phase. Deep learning has only one neural network phase. At the beginning of the neural network, the layers are learning to recognize the basic features of the data and that data feedforward to the other layers in the network for additional computation of the network [20, 18].

2.8.1 Artificial Neural Networks

The most popular and primary approach of deep learning is using “Artificial neural network” (ANN). They are brain-inspired systems that are designed to mimic how we humans learn [21]. An ANN is made up of artificial neurons, which are a collection of connected units or nodes that loosely replicate the neurons in a human brain (see Figure 3). Neurons are cells found in the central nervous system of humans. Axons and dendrites are the connecting regions between axons and dendrites, and synapses are the connecting regions between axons and dendrites. Each link allows neurons to receive data and do mathematical calculations, as well as transmit synapses represented by weights. In 1958, psychologist Frank Rosenblatt built the first Artificial Neural Network (ANN) based on this paradigm [22].

1. ANNs are comparable to neurons in that they are made up of numerous nodes. The nodes are firmly linked and grouped into several hidden layers. The input layer receives the data, which is then passed via one or more hidden layers in a sequential order before being predicted by the output layer

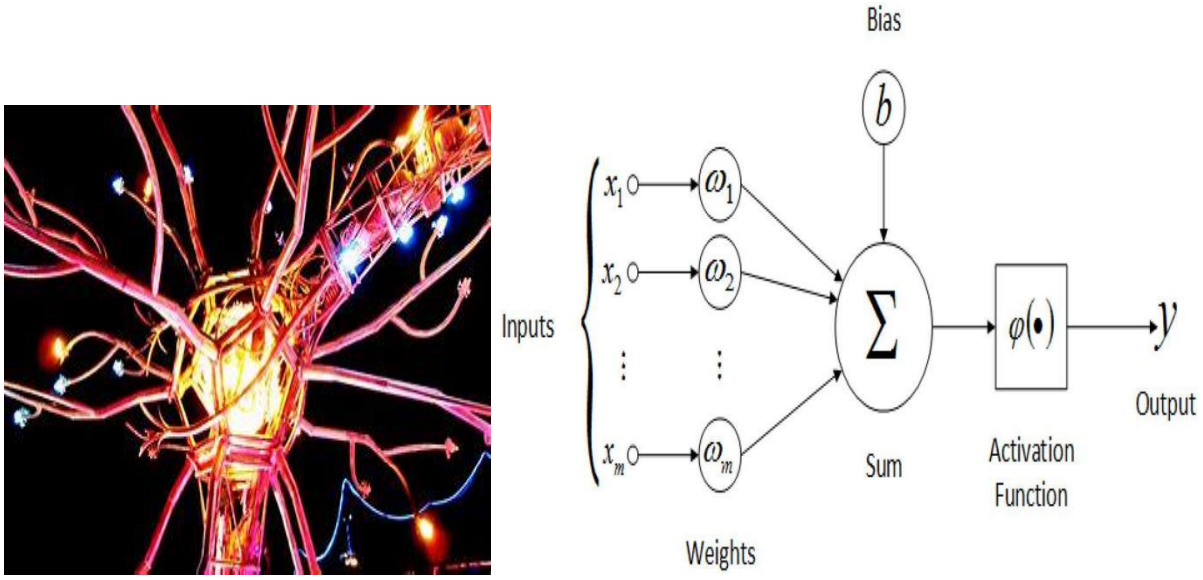


Figure 11 Example of Artificial Neurons [23]

The ANNs ability to learn non-linear complex relationship automatically from data made them a model of choice in many application areas. In addition, advances in computing power and techniques for building and training ANNs have allowed these models to achieve state-of-the art results in image classification [24, 25, 26]. This breakthrough in image classification is attributed to a special type of ANN called CNN.

An artificial neural network computes a function of the inputs by propagating the computed values from the input neurons to the output neuron(s) and using the weights as intermediate parameters. Learning occurs by changing the weights connecting the neurons. Just as external stimuli are needed for learning in biological organisms, the external stimulus in artificial neural networks (ANN) is provided by the training data containing examples of input-output pairs of the function to be learned. For example, in thesis the training data might contain pixel representations of images (input) and their annotated labels e.g., the outputs are Type_1, Type_2, and Type_3. These training data pairs are supplied into the neural network, which makes predictions about the output labels based on the input representations. Based on how closely the projected output for a specific input matches the annotated output label in the training data, the training data offers feedback on the validity of the weights in the neural network. The errors caused by the neural network in the computation of a function might be viewed as a form of unpleasant feedback in a biological organism, causing synaptic strengths to be adjusted. In a neural network, the weights between neurons are modified in response to prediction mistakes. The purpose of modifying the weights is to modify the computed

function such that future iterations' predictions are more accurate. As a result, the weights are carefully adjusted in a mathematically justified manner to minimize the computation error in that example. The function generated by the neural network is refined over time by successively altering the weights between neurons over numerous input-output pairs, resulting in more accurate predictions. As a result, if the neural network is trained with a large number of different photos of cervical cancer, it will eventually be able to correctly detect a cervical cancer in a new image. The capacity of all machine learning models to generalize their learning from observed training data to unseen samples is their major benefit. The input is modified by the weights and functions employed in the nodes as it passes through each hidden layer, eventually producing an output. When the created output is compared to the real value, the error propagates back to change the weights, learning occurs. The back propagation algorithm is what it's called. The process is repeated for a predetermined number of iterations (epochs) or until the error falls below the desired level.

2.8.2 Feedforward neural network

A feedforward neural network is a type of artificial neural network in which nodes' connections do not form a loop [27, 28]. As a result, it differs from its offspring, recurrent neural networks.

The feedforward neural network was the first and most basic artificial neural network to be created [19].

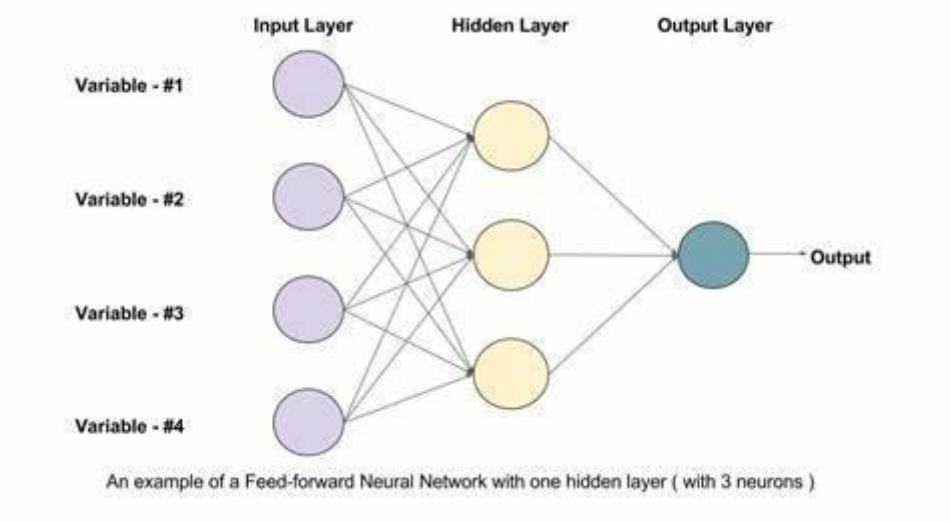


Figure 12 feedforward neural network

Multi-layer perceptron (MLP)

The most basic type of ANN is the multi-layer perceptron. It has a single input layer, one or more hidden layers, and an output layer at the end [21]. A layer is made up of a group of perceptrons. One or more aspects of the input data make up the input layer. Every hidden layer contains one or more neurons that process a specific feature component before sending the processed data to the next hidden layer. The output layer procedure receives data from the last hidden layer and outputs the result. In the 1940s, Warren McCulloch and Walter Pitts described a similar neuron. As long as the threshold value is between the activated and deactivated states, any value for the activated and deactivated states can be used to generate a perceptron. To train perceptrons, a basic learning algorithm known as the delta rule can be used. It evaluates the differences between computed and sample output data and uses this information to alter the weights, resulting in a gradient descent algorithm.

Single-layer perceptron networks can only learn linearly separable patterns; Marvin Minsky and Seymour Papert demonstrated in their famous monograph Perceptron in 1969 that a singlelayer perceptron network could not learn an XOR function (however, multi-layer perceptron networks can produce any possible Boolean function). Four layers

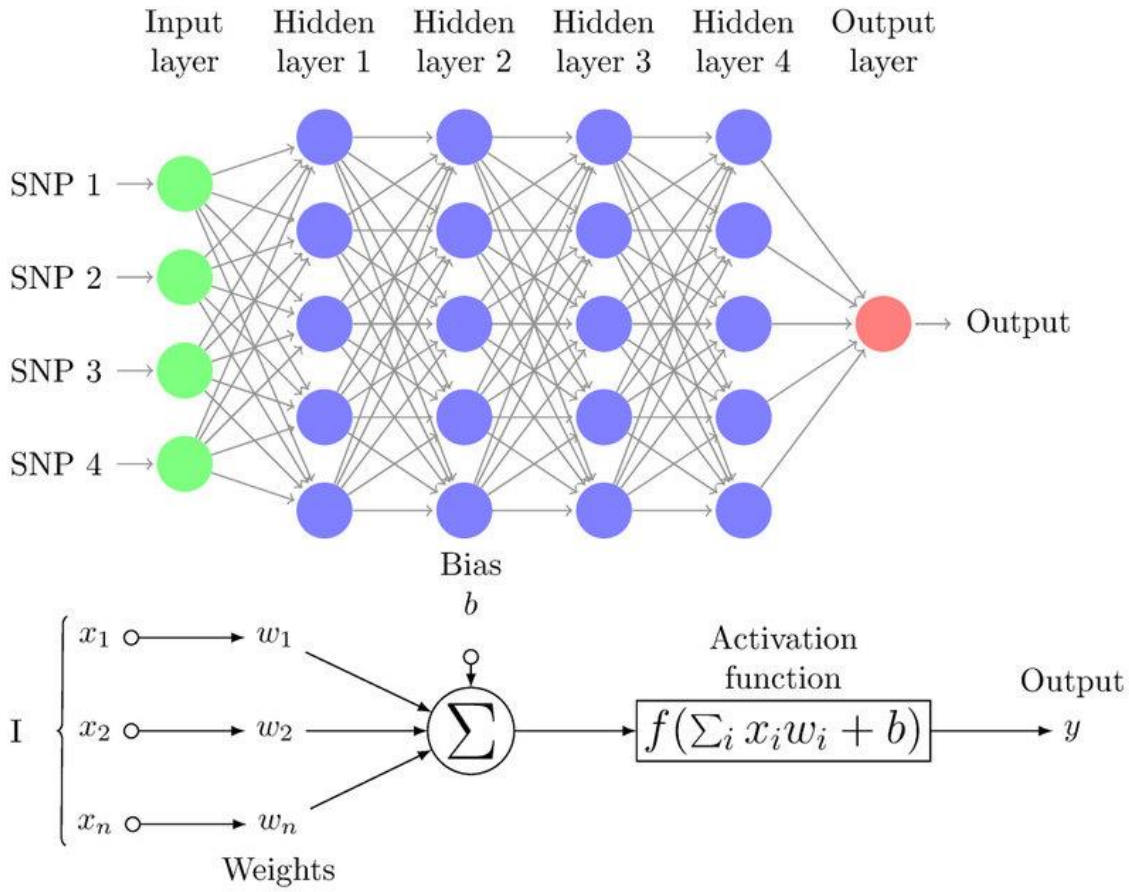


Figure 13 Multi-layer perceptron

2.8.2.1. Activation function

It's an important aspect of a neural network's design. The activation function of the hidden layer determines how well the network model learns the training dataset. The activation function used in the output layer determines the type of predictions the model can make [30]. Each node in the neural network receives a large number of inputs and generates a single output based on the weighted sum of those inputs. This capability is achieved with a rather simple model. Frank Rosenblatt first invented the perceptron in 1957 [22]. The perceptron adds all of the inputs and their weights together, compares the result to a threshold, and then outputs a discrete value based on the comparison. Activation functions represent a key factor in ANN implementation [31].

Their major goal is to change a node's input signal to an output signal, which is why they're called transformation functions. The output of network layers without an activation function is turned into a linear function, which has several limitations due to their limited complexity and capacity to understand complicated and intrinsic correlations among dataset variables. In

realworld datasets, non-linearity is common. Images, voice recordings, and videos, for example, cannot be described using this simple approach since they are made up of numerous dimensions that are not represented by linear transformations. Differentiability is another quality of activation functions that is required for a back propagation optimization technique to be successful [31]. The gradient descent optimization algorithm, which modifies the weight of neurons by computing the gradient of the loss function, uses back-propagation in the context of learning. After a training observation has propagated through the neural network, the loss function measures the difference between the network output and its predicted output. Almost all activation functions suffer with the vanishing gradient problem, which is a wellknown difficulty. As more layers with a specific activation function are added to a neural network, the gradients of the loss function approach 0 and the network training is terminated. As a result, resolving the problem becomes extremely difficult. Because back-propagation finds the network's derivatives by going layer by layer from the last to the first, the derivatives of each layer are multiplied down the network according to the chain rule statements. A small gradient (near zero) indicates that the weights and biases from the first layers will not be effectively updated while training the network, which can result in a loss of accuracy because the model is unable to recognize core elements from the input data, which occurs frequently at network entry.

Although perceptrons can theoretically be used to compute any function, they are rarely employed in practice, owing to their discontinuous nature: a little change in the weights can lead the perceptron to generate a significantly different output, which is undesirable for learning [32]. As a result, instead of using a perceptron, a sigmoid activation function is employed in this neural network implementation, which is relatively comparable to the former but eliminates its flaw. The formula for the sigmoid function is the following:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

2.8.2.2 Loss function

The neural network used in this thesis is designed to learn how to classify photos. To learn something, one must be given feedback on his current performance. The loss function's job is to evaluate the neural network's accuracy. If the loss is small, the neural network is correctly classifying the images; if the network is incorrectly classifying the images, the loss will be

large. The output of the neural network must first be translated as class scores before the loss for a specific estimate can be calculated. The score function does this by calculating the chance that a given input is a specific class using the values from the output layer nodes.

2.8.2.3 Gradient descent

We defined a loss function in the previous section, which essentially tells us how accurately the neural network can classify data. Now we need a way to act on this information and somehow minimize network loss during training by changing the weights of different nodes. The gradient descent algorithm is used to accomplish this.

Gradient descent works by determining the gradient of the loss function with respect to the network weights, which tell us which direction the function increases the most. To minimize the loss, a fraction of the gradient must be subtracted from the corresponding weight vector.

2.8.2.4. Back propagation

Back propagation is a practical realization of the gradient descent algorithm in multilayered neural networks. It calculates the gradient of the loss function with respect to all the weights in the network by iteratively applying the multivariable chain rule.

Applying the back propagation algorithm to a neural network is a two way process: we first propagate the input values through the network and calculate the errors, and then we back propagate the errors through the network backwards to adjust the connection weights in

order to minimize the error. The algorithm calculates the gradient of the loss function with respect to the weights between the hidden layer and output layer nodes, and then it proceeds to calculate the gradient of the loss function with respect to the weights between the input layer and hidden layer nodes. After calculating the gradients, it subtracts them from the corresponding weight vectors to get the new weights for the connections. This process is repeated until the network produces the desired outputs

2.9. Convolutional Neural Network (CNN)

Convolutional neural network also known as ConvNet or CNN is one of the most popular ANN [18, 33]. CNNs are a subtype of neural network that is used to process data with a known, gridlike topology, such as time series and image data [34]. Convolutional neural networks are biologically inspired networks that are used in computer vision for image classification and object detection [35]. It is a multi-layered deep learning and neural network technique. It is widely employed in image and video recognition. Its foundation is the mathematical concept of convolution. It is similar to a multi-layer perceptron in that it has a

series of convolution layers and a pooling layer before the fully connected hidden neuron layer. CNN's architecture and training procedure A CNN is composed of several building blocks, which are detailed in Section 2.9.1. Convolution layers, pooling layers (e.g., max pooling), and fully connected (FC) layers are examples. The performance of a model with specific kernels and weights is calculated with a loss function on a training dataset using forward propagation, and learnable parameters, i.e., kernels and weights, are updated according to the loss value using the gradient descent optimization algorithm described in section 2.8.3. In computer vision, images can be filtered by using convolution operation to produce different visible effects. CNN has convolutional filters that are used to detect some objects in a given image such as edges which is the same as the biological receptive field. Since the late 1980s and in 1990 CNN gives interesting results in handwritten digit classification and face recognition. The following figure (Figure--) [36] illustrates CNN architecture.

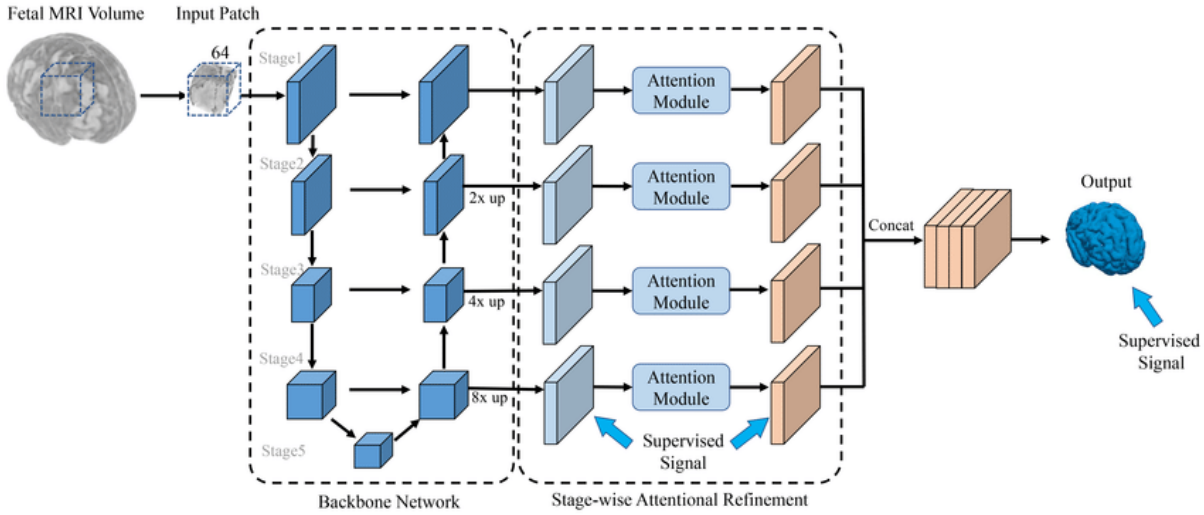


Figure 14 CNN architecture.

2.9.1. Building blocks of CNN architecture

As shown in Figure 10, the CNN architecture consists of several building blocks, including convolution layers, pooling layers, and fully connected layers. A typical architecture consists of several convolution layers, followed by a pooling layer that is repeated several times, and one or more fully connected layers. The process of transforming input data into output data using these layers is referred to as forward propagation. Although the convolution and pooling operations described in this section are for two-dimensional (2D) CNN, they also apply to three-dimensional (3D) CNN.

In this thesis, CNN is used to detect and classify Type_1, Type_2, and Type_3 cervical cancer by giving cervical cancer image as an input. For the process of detection and classification, CNN is used which is composed of various sequential layers and every layer of the algorithm transforms one volume of activation to another using different functions. The basic and commonly used layers of CNN are the convolution layer, the pooling layers, and the fully connected Layer.

I. Convolution layer

It is the fundamental building block that performs computational tasks using the convolution function. The convolution layer of a CNN architecture is a critical component that performs feature extraction. It is usually made up of a mix of linear and nonlinear operations, such as convolution and activation functions [26] Convolution is a type of linear operation used for feature extraction that involves applying a small array of numbers known as a kernel across a tensor array of numbers as input.

A feature map is obtained by calculating an element-wise product between each element of the kernel and the input tensor at each location of the tensor and summing it to obtain the output value in the corresponding position of the output tensor, as shown in Figure 6. This procedure is repeated with an arbitrary number of kernels to produce an arbitrary number of feature maps representing different properties of the input tensors; different kernels can thus be viewed as different feature extractors. The size and number of kernels are two key hyper parameters that define the convolution operation. The former is usually 3 × 3, but it can also be 5 × 5 or 7 ×7. The depth of the output feature maps is determined by the latter, which is arbitrary. The following figure (Figure 6) [37] illustrates example of convolution layer operation.

$$p_{new} = \sum_{i \in s} p_i . w_i$$

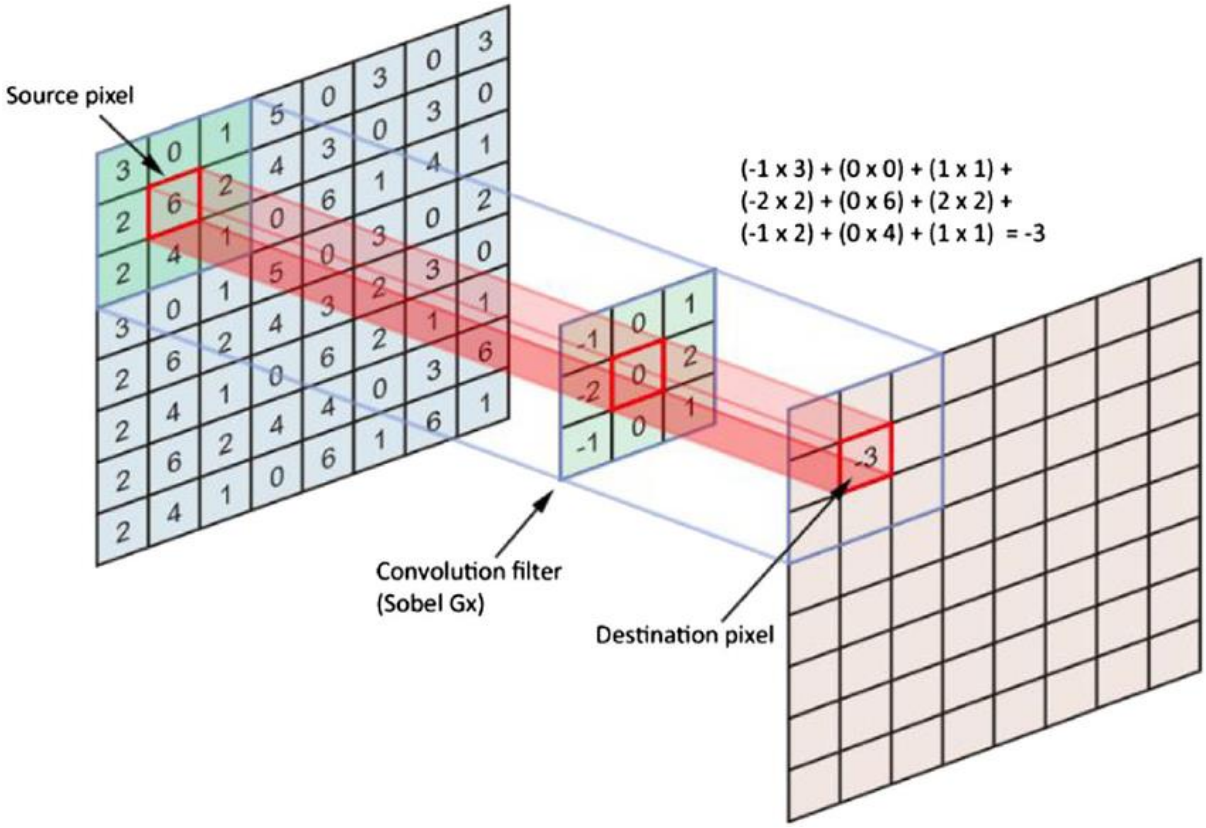


Figure 15 An example of Convolution layer operation

Figure shows a convolution operation with a kernel size of three, no padding, and a stride of one. A kernel is applied across the input tensor, and an element-wise product between each element of the kernel and the input tensor is calculated and summed at each location to obtain the output value in the corresponding position of the output tensor, which is referred to as a feature map.

II. Pooling layer

It is placed next to the convolution layer and is used to reduce the size of inputs by removing unnecessary information so that computation can be performed more quickly. It implements a standard down sampling operation that reduces the in-plane dimensionality of the feature maps in order to introduce translation invariance to small shifts and distortions and reduce the number of subsequent learnable parameters. It is worth noting that none of the pooling layers have learnable parameters, whereas filter size, stride, and padding are hyper parameters in pooling operations, similar to convolution operations [26]. Figure 7 shows an example of a

max pooling operation with a filter size of 2×2, no padding, and a stride of 2, which extracts 2× 2 patches from the input tensors, outputs the maximum value in each patch, and discards all other values, resulting in a factor of 2 down sampling of an input tensor's in-plane dimension. The following figure (Figure 10) [37] illustrates example of max pooling operation.

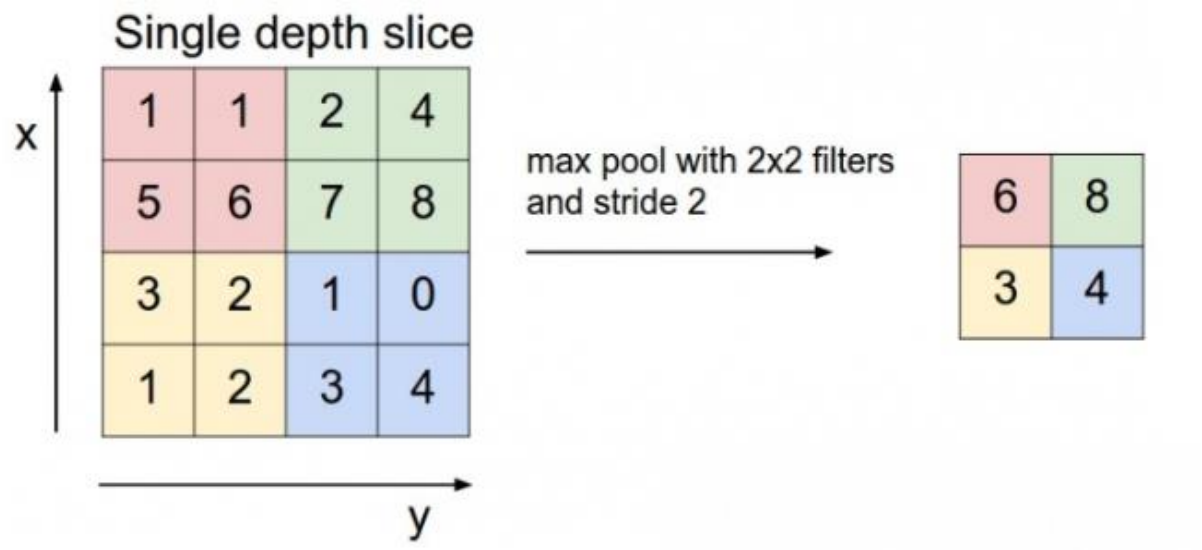


Figure 16 An example of max pooling operation

Different types of pooling methods are available for use in various pooling layers. These methods include tree pooling, gated pooling, average pooling, min pooling, max pooling, global average pooling (GAP), and global max pooling. As illustrated in Figure 11, the most well-known and widely used pooling methods are max, min, and GAP pooling. The following figure (Figure 8) [36] illustrates example of three type of pooling operations.

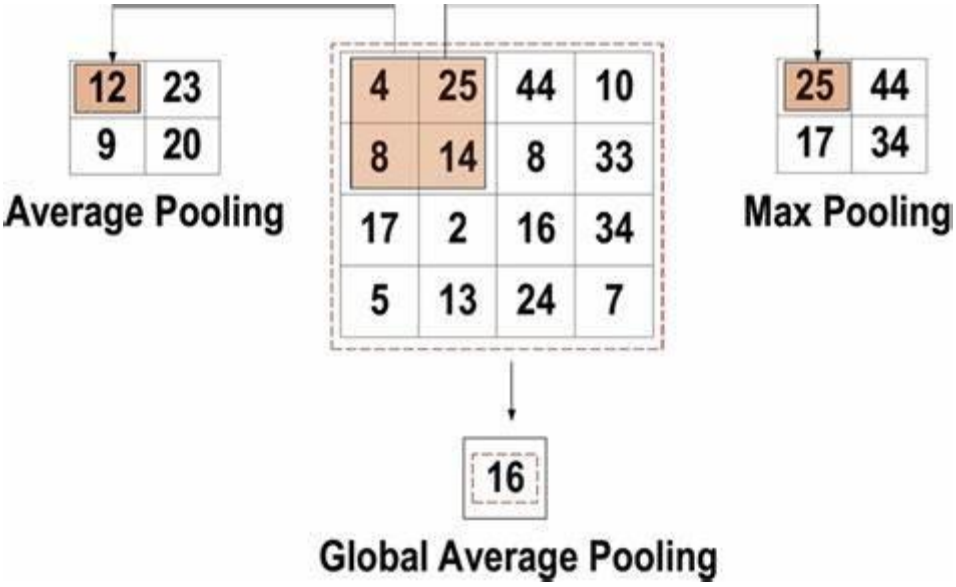


Figure 17 type of pooling operations

III. Fully connected layer

The final pooling layer's output is flattened and fed into one of several fully connected layers that act as classifiers. CNN training is similar to fully connected neural network training that employs the back propagation algorithm. It divides input into groups and is arranged next to a series of convolution and pooling layers. The final convolution or pooling layer's output feature maps are typically flattened, i.e. transformed into a one-dimensional (1D) array of numbers (or vector), and connected to one or more fully connected layers, also known as dense layers, in which every input is connected to every output by a learnable weight. In classification tasks, a subset of fully connected layers maps the features extracted by the convolution layers and down sampled by the pooling layers to the network's final outputs, such as the probabilities for each class. Typically, the final fully connected layer has the same number of output nodes as he number of classes. Following each fully connected layer is a nonlinear function, such as ReLU.

2.9.2. Evolution of Convolutional Neural Network Models

AlexNet: The first breakthrough came in 2012 when the convolutional model which was named AlexNet significantly outperformed all other conventional methods in ImageNet LargeScale Visual Recognition Competition (ILSVRC) 2012 that featured the Image Net dataset. The AlexNet brought down classification error rate from 26 to 15%, a significant improvement at that time. AlexNet was simple but much more efficient than LeNet [38]. The improvements of AlexNet is Large labeled image database (ImageNet), which contained around 15 million labeled images from a total of over 22,000 categories, was used and The

model was trained on high-speed GTX 580 GPUs for 5 to 6 days. The other improvements are ReLU (Rectified Linear Unit) f (x) _max(x, 0) activation function was used. This activation function is several times faster than the conventional activation functions like sigmoid and tanh. The ReLU activation function does not experience the vanishing gradient problem. AlexNet consists of five convolutional layers, three pooling layers, three fully connected layers, and a 1000-way softmax classifier [18].

ZFNet: In 2013, an improved version of CNN architecture called ZFNet was introduced [39]. ZFNet reduced the filter size in the first layer from 11 × 11 to 7 × 7 and used a stride of 2 instead of 4 which resulted in more distinctive features and fewer dead features. ZFNet turned out to be the winner of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2013.

VGG: introduced in 2014, Winner of the ImageNet ILSVRC-2014 competition, VGGNet was invented by Oxford's Visual Geometry Group used increased depth of the network for improving the results. The depth of the network was made 19 layers by adding more convolutional layers with 3 × 3 filters, along with 2 × 2 max-pooling layers with stride and padding of 1 in all layers. Reducing filter size and increasing the depth of the network resulted in CNN architecture that produced more accurate results. VGGNet achieved an error rate of 7.32% in ILSVRC 2014 and was the runner-up model in ILSVRC 2014 [18].

GoogLeNet: Google developed a ConvNet model called GoogLeNet in 2015. GoogLeNet proposed a novel concept referred to as inception architecture [35]. The model has 22 layers and was the winner of ILSVRC 2015 for having the error rate of 6.7%. The previous ConvNet models had convolution, and pooling layers stacked on top of each other but the GoogLeNet architecture is a little different. It uses an inception module which helps in reducing the number of parameters in the network. The inception module is actually a concatenated layer of convolutions (3 × 3 and 5 × 5 convolutions) and pooling sub-layers at different scales with their output filter banks concatenated into a single output vector making the input for the succeeding stage. These sub-layers are not stacked sequentially but the sub-layers are connected in parallel. To compensate for the additional computational complexity caused by extra convolutional operations, 1 1 convolution is used, resulting in fewer computations before expensive 3 3 and 5 5 convolutions are performed. Two convolutional layers, four max-pooling layers, nine inception layers, and a softmax layer comprise the GoogLeNet model. Because of the use of this unique inception architecture, GoogLeNet has 12 times fewer parameters than AlexNet [18].

ResNet: In 2015, Microsoft Research Asia proposed a 152-layer deep CNN architecture called ResNet [40]. ResNet introduced residual connections, which add the output of a conv-relu-conv series to the original input and then pass it through a Rectified Linear Unit (ReLU). In this way, the information is carried from the previous layer to the next layer and during backpropagation, the gradient flows easily because of the addition operations, which distributes the gradient. ResNet outperformed humans in classification, detection, and localization, winning the ILSVRC 2015 with an error rate of 3.6%, which is lower than the human error rate of 5–10% [18].

Inception-ResNet: A hybrid inception model which uses residual connections, as in ResNet, was proposed in 2017 [24]. This hybrid model, known as Inception-ResNet, significantly improved the training speed of the inception model and outperformed the pure ResNet model by a narrow margin.

Xception: is a convolutional neural network architecture based on depth-wise separable convolution layers [24]. The Xception is said to perform slightly better than the Inception V3 on ImageNet [18].

2.10. Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNN) is beneficial for addressing flaws in other artificial neural network (ANN) models. In training, the majority of the ANN didn't remember the steps from earlier cases and instead learnt to make decisions based on context. Meanwhile, RNN saves historical data and bases all of its choices on what it has learned in the past.

This method is most beneficial when it comes to image classification. In order to fix the past, we may need to reach into the future. Bidirectional RNN is useful in this scenario for learning from the past and predicting the future. We have handwritten examples in several inputs, for example. If one of the inputs is confusing, we must review the other inputs again to identify the proper context, which is based on a decision made in the past.

RNN is one of the first deep learning architecture which gives a road map to develop other deep learning algorithms. It is commonly used in speech recognition and natural language processing [41]. RNN is designed to recognize the sequential characteristics (remembers previous entries) of the data. When we analyze time serious data, the network has memory (hidden state) to store previously analyzed data. To perform the present task RNN needs to look at the present information (short term dependency) and this is the main drawback. RNN differs from a neural network is that RNN takes a sequence of data defined over time [41]. LSTM is a special type of RNN which is explicitly designed to overcome the problem of

long-term dependencies by making the model remember values over arbitrarily time interval. The main problems of RNN are vanishing gradient and exploding gradients. The gradient is the change of weight with regard to the change in error. It is well suited to process and predicts time series given time lags of unspecified duration. For example, RNN forgets the model if we want to predict a sequence of one thousand intervals instead of ten, but LSTM remembers such kind of activities. The main reason that LSTM can remember its input in a long period of time is that it has a memory that is like memory on a computer which allows the LSTM to read, write and delete information [42]. It is mostly applied to natural language text compression, speech recognition, handwritten recognition, gesture recognition, and image captioning. Recurrent neural networks have improved long short-term memory (RNNs). In order to solve the vanishing and exploding gradient problem, LSTM suggests memory blocks instead of traditional RNN units. The key difference between it and RNNs is that it adds a cell state to save long-term states. An LSTM network can remember and connect data from the past with data from the present. The LSTM is made up of three gates: an input gate, a "forget" gate, and an output gate, with xt denoting the current input, Ct and Ct-1 denoting the new and previous cell states, and ht and ht-1 denoting the current and prior outputs, respectively.

CHAPTER THREE

3.1 Methodology

3.1.1. Base network selection: The SSD [10] design is initially built upon the visual classification VGG16 profound CNN, whereby an ImageNet pretrained VGG16 is misused as a include extractor. In spite of the fact that VGG16 has great include extraction capabilities, it is very a huge arrange engineering for implanted stages. For this reason, when executed on an implanted framework, VGG16 may surpass the most extreme framework memory making it troublesome to attain genuine time execution.

In this work, MobileNet_v2 [26] was chosen as a base arrange, as appeared in Figure 4.2, to construct a dependable activity sign and light detector when the computational fetched may be a restriction. They both have a comparable precision 70.6% and 72% for MobileNet_v1 and MobileNet_v2 individually, as VGG16 has 71.5% on ImageNet dataset. They both have as it were 1/30 of the computational taken a toll and demonstrate measure compared to VGG16. It was chosen as the base network after experimenting using SSD with both MobileNet_v1 and MobileNet_v2.

The essential component that creates MobileNet a light design is the depth-wise separable convolution, which isolates the conventional convolutional layer into two parts:

depth-wise convolution and point-wise convolution, as appeared in Figure 4.10. The primary portion, the depth-wise convolution, applies a single 3x3 channel to each input channel creating yield picture that too has 3 channels where each channel gets its possess set of weights. The moment portion, pointwise convolution is connected that's same as a standard convolution but with a 1 x 1 part. Pointwise convolution basically includes up all the yield channels of the depthwise convolution as a weighted sum in arrange to make modern highlights. In comparison, a ordinary convolution layer, channels and combines the inputs into a set of yields in one step, as appeared in Figure 4.9. In spite of the fact that, the conclusion comes about obtained by both customary and depth-wise divisible convolution are the same, a regular convolution must perform numerous more scientific computations and ought to learn more weights compared to the last mentioned one.

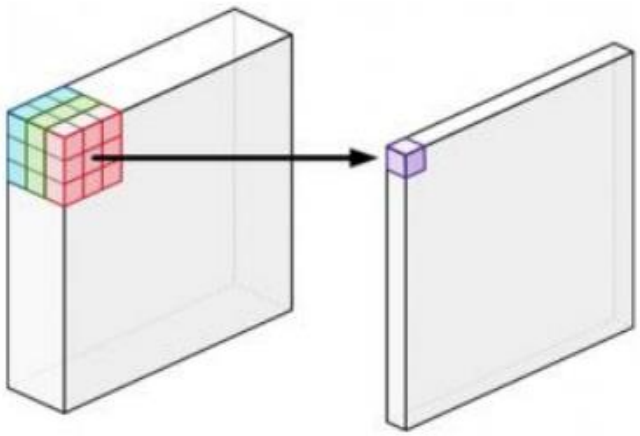


Figure 4.9. Regular convolution[58]

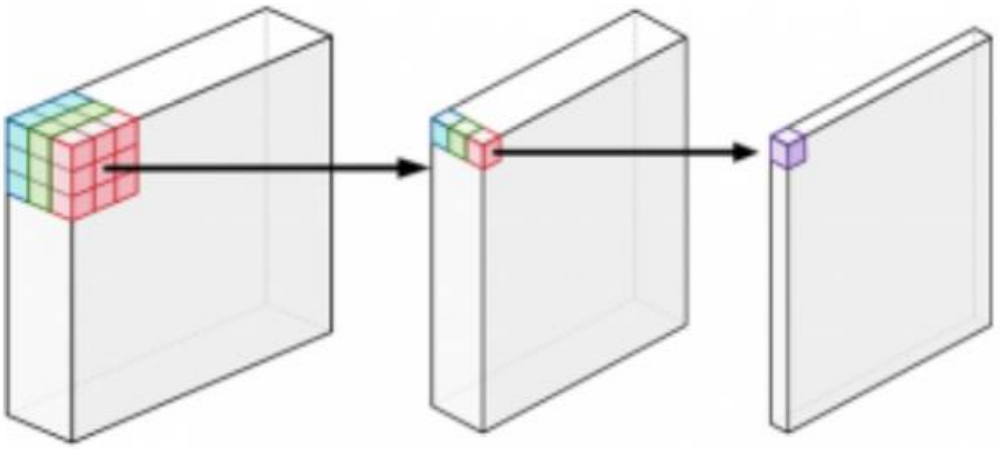


Figure 18 Depth-wise separable convolution[58]

MobileNet-v1

Table 4.1 shows the MobileNet-v1 architecture which consists of a regular convolutional layer followed by 13 depth-wise separable convolutional blocks. Each one of these blocks contains a batch normalization layer along with a ReLU activation function. The last three layers used for classification task consists of an average pool layer followed by a fully connected layer and linked to a softmax layer. To use MobileNet-v1 as a feature extractor for the task of object detection the last three layers responsible for the classification were removed following the same procedure used in the SSD paper [10].

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Table 1 MobileNet- v1 architecture [25]

MobileNet–v2

The basic building block of MobileNet-v2 is an inverted residual structure with the shortcut connections between the thin bottleneck layers. The architecture of MobileNet-v2 consists of an initial fully convolutional layer with 32 filters, followed by 19 residual bottleneck layers

[26]. Table 4.2 shows the MobileNet-v2 overall architecture, where t is the expansion factor, c denotes the number of output channels, n denotes repeating numbers and s represents stride 3 x 3 kernels used for spatial convolution.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 2 MobileNet-v2 architecture [26]

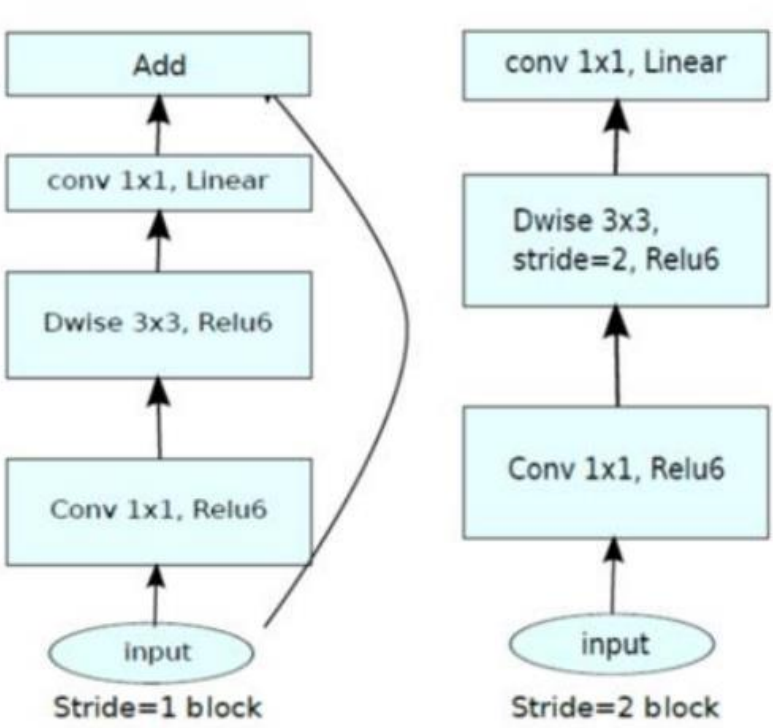


Figure 19 MobileNet-v2 basic building blocks[26]

MobileNetv2 has two types of blocks, as shown in Figure 4.11. One is residual block with stride of 1. The other one with stride of 2 for downsizing.

There are 3 layers for both types of blocks.

• The first layer is 1×1 convolution with ReLU6.

• The second layer is the 3×3 depthwise convolution.

• The third layer is another 1×1 convolution but without any non-linearity.

Experiments were performed to evaluate and compare the performance of MobileNet-v2 and MobileNet-v1 as feature extractors for the traffic sign and light detection with a modified version of the SSD on Road sign detection dataset.

3.1.2. Feature map layer selection

The problem with small object detection

Because a typical CNN progressively shrinks the feature map scale and increases the depth as it goes to the deeper layers. The deeper layers cover larger receptive fields and therefore construct more abstract representation while the shallow layers cover smaller receptive fields. By making use of this information, we can use deeper layers to predict big objects and shallow layers to predict small objects. In SSD[8] the six feature maps from 19× 19 to 1 × 1 have empirically calculated prior boxes with a scale of 0.1, 0.2, 0.375, 0.55, 0.725 and 0.9. Therefore, larger feature maps have prior boxes with smaller scales, so they are ideal for detecting smaller objects.

Thus, the proposed object detector is built with a modified structure of the standard SSD. It makes predictions at two different scales (19×19 and 10×10) as shown in Table 4.1and illustrated in Figure 4.2, by a box with black broken line. The number of feature map layers are selected depending on the fact that traffic sign and light comes in regular shapes and in defined colors therefore they could be detected with simpler CNN architectures. And experimentation helped in choosing the option with the best balance between accuracy and detection time.

Multiscale feature maps

The main SSD algorithm uses six feature map layers as shown in Figure 4.1. They extract features from images at multiple scales detecting objects of various sizes.

Table 4.3 shows the dimensionality comparison of the multiscale feature maps extracted from the SSD with VGG16, SSD with MobileNet_v2 and the proposed model.

Feature map layer	SSD/ VGG16	SSD/ MobileNet_v2	Proposed model
Layer 1	38 × 38 ×512	19 × 19 × 96	19 × 19 × 96
Layer 2	19 × 19 ×1024	10 × 10 × 1280	10 × 10 × 1280
Layer 3	10 × 10 ×512	5 × 5 × 512	___
Layer 4	5 × 5 ×256	3 × 3 × 256	___
Layer 5	3 × 3 ×256	2 × 2 × 256	___
Layer 6	1 × 1 ×256	1 × 1 × 256	___

Table 3Dimensions of feature maps used in SSD/VGG16, SSD/MobileNet_v2 and proposed model

.

SSD with MobileNet_v2 uses feature maps with scales 19×19, 10×10, 5×5, 3×3, 2×2 and 1×1, which is different from VGG16 based SSD. The 38×38 feature map provided by the MobileNet_v2 is a shallow feature and it is not easy to extract an effective image feature map using this. Therefore, it is not used due to poor results achieved. To use MobileNet (both versions) as a feature extractor for object detection, the last three layers (responsible for classification) were removed, following a similar process used in the SSD paper [10]. As part of the experiments for this thesis, we tried using different number of feature map layers, each with a different scale. The effects of these experiments on the accuracy and detection time were recorded.

Chapter 4

PROPOSED TRAFFIC SIGN AND SPEED LIMIT DETECTOR

This chapter starts with a comparison between the two sorts of single organize locators, YOLO(you as it were see once) and SSD(single shot locator). This is often taken after by a brief survey of the single shot multibox locator calculation that's utilized as a pattern to send the show. Afterward, the strategy utilized to build the activity sign and light detector is displayed together with the exploratory assessment comes about. The ultimate exploratory comes about are compared against the bland protest locator SSD with VGG16 over the dataset.

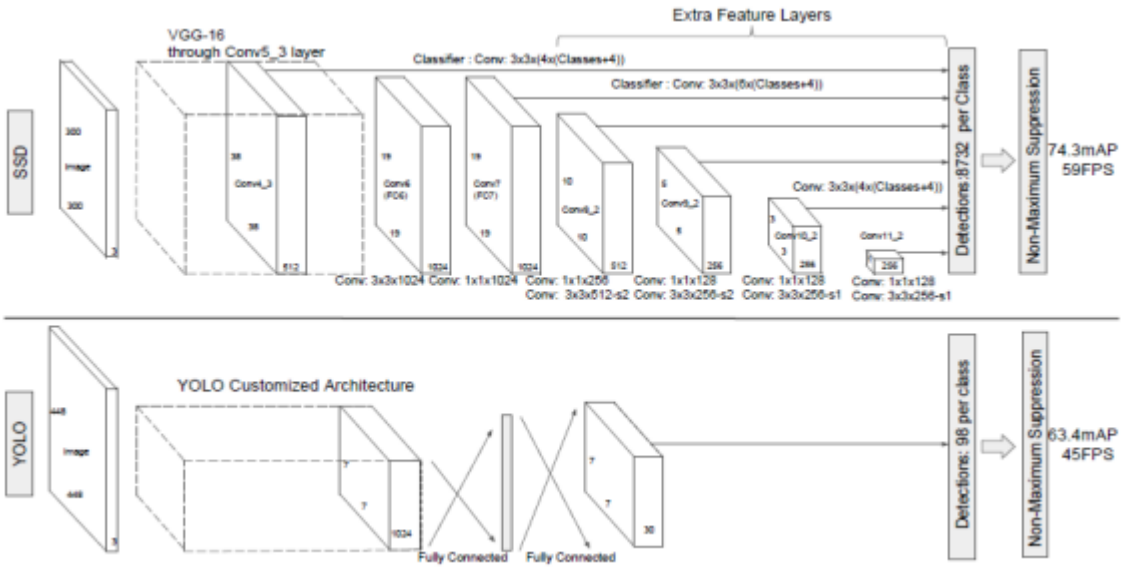


Figure 20 Comparison between the two SSD models, SSD[10] and YOLO [9].

SSD adds feature map layers at the end of a base network and, uses six feature maps to make predictions. While YOLO uses one feature map to make predictions.

4.1 SSD architecture review

Prior designs utilized for question location comprised of two unmistakable stages – a locale proposition organize that performs protest localization and a classifier for identifying the sorts of objects within the proposed locales. SSD is outlined to typify both localization and discovery errands in a single forward clear. The yield is communicated in terms of a set of default bounding boxes and a esteem of certainty of the nearness of the target question course for each default bounding box. SSD calculation, in an endeavor to distinguish objects of diverse sizes and shapes, makes forecasts in six distinctive scale highlight maps.

SSD framework is composed of six major modules or algorithms and each of them plays a crucial role in providing the final output.

- Base network for feature extraction
- Convolutional predictor box
- Prior boxes generation
- Matching prior boxes to ground-truth boxes
- Hard negative mining
- Post-processing – NMS

Base network for feature extraction

SSD architecture is mainly based on a deep CNN that works as the feature extractor. Remarkable performance is shown by deep CNNs in image classification on many benchmark datasets, which indicates that the features extracted by these networks can be very useful in performing other more sophisticated computer vision tasks like object detection as well. Original SSD has VGG16 as the base feature extractor as shown in Figure 4.1.

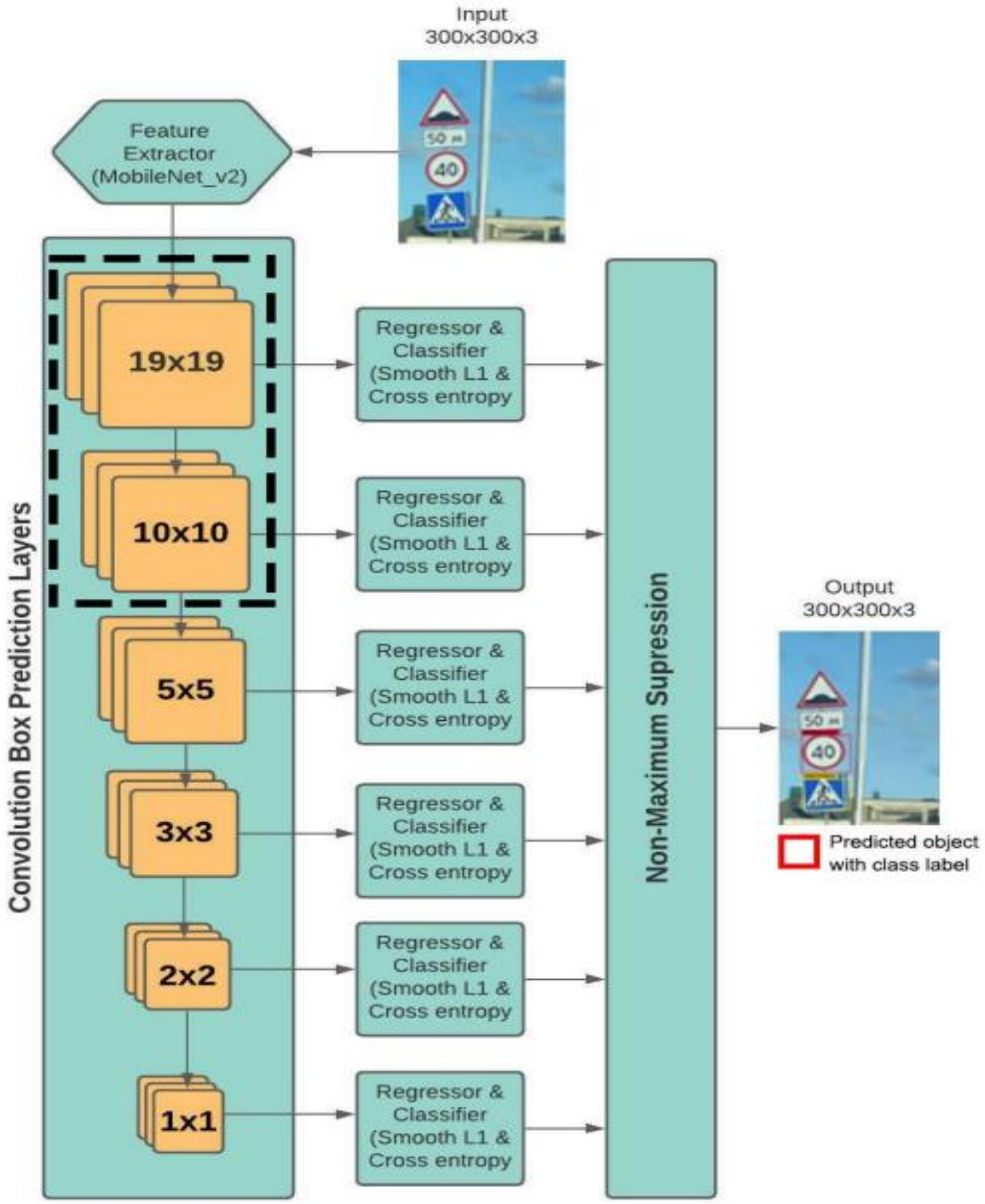


Figure 21 The proposed SSD-based Traffic Sign and Light Detection framework

From the original sixbox prediction layers, the proposed model utilizes only top-2 layers for efficient detection performance as marked by black broken line

Convolutional Predictor Box

For object detection, a base CNN-based feature extractor is extended to a larger network by eradicating top classification layers of the base network and adding some successive layers. At the end of the base network, SSD algorithm adds extra convolutional feature map layers. These feature map layers decrease in size progressively to make predictions on six different layers with sizes of (19×19, 10×10, 5×5, 3×3, 2× 2, 1×1) as shown in Figure 4.2. The successive layers are connected to two main heads, one regressor to predict bounding boxes and one classifier to classify each of the detected boxes as shown in Figure 4.2.

Prior boxes generation

Prior boxes are a combination of precalculated, predefined boxes. They are manually but carefully selected based on the sizes and shapes of ground truth objects in our dataset. Their importance is inevitable in terms of providing a strong starting point as opposed to starting with completely random coordinates for the prediction of ground truth boxes using bounding box regression algorithm. In defining the prior boxes, as paper [10] suggested that the largest feature map, conv4_3, have prior boxes with a scale of 0.1, i.e., 10% of image's dimensions, while the rest have prior boxes with scales linearly increasing from 0.2 to 0.9. That is why larger feature maps with prior boxes of smaller scales are therefore ideal for detecting smaller objects. Each of the prior boxes predict exactly one bounding box.

Prior boxes and aspect ratios

SSD algorithm applies prior boxes to all the six feature layers used for prediction. The algorithm makes four predictions with the 1st, 5th, and 6th layers and six predictions with the remaining three layers. SSD associates each cell in the feature maps used for prediction with a set of prior boxes. Figure 4.3 shows what the prior boxes look like at the central tile of the 5 x 5 feature map.

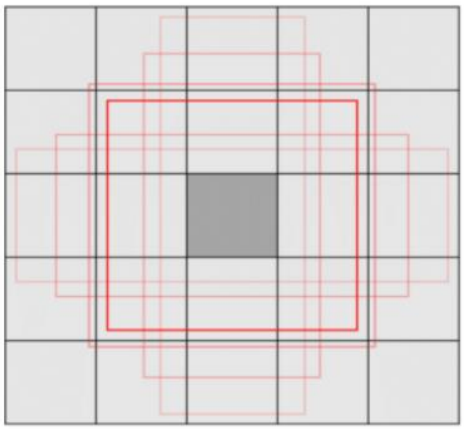


Figure 22 5×5 feature map with prior boxes

The algorithm predicts the offsets relative to the prior box in the cell and a confidence score that expresses the presence of a target object class inside the prior box, as shown in Figure 4.4(c). For t given locations in a feature map, the algorithm computes c class scores by applying (c + 4)t filters for each feature map cell. For a g × h feature map, the output will have a size of (c + 4)tgh. These prior boxes have different scales and aspect ratios to efficiently predict objects of different shapes and sizes as shown in Figure 4.4.

As shown in Figure 4.4(b) & (c), the larger feature map with a smaller scale prior boxes are suitable for small object detection while smaller feature map with big scale prior boxes suitable for big object detection.

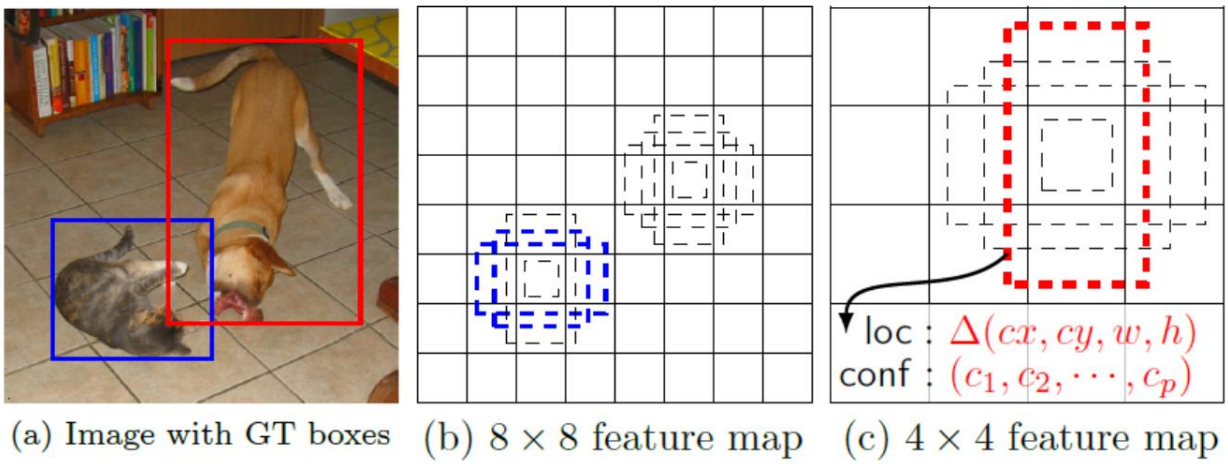


Figure 23 SSD uses lower resolution layers to detect larger scale objects and higher resolution layers for small object detection [10].

Predictions versus Prior boxes The prior boxes are the approximate representation of the possibilities for predictions. It means that each prior box is used as an approximate beginning point and later try to find out how much it needs to be fine-tuned to obtain a more close and exact prediction for a bounding box. A way is needed to measure the deviation of each predicted bounding box from a prior box.

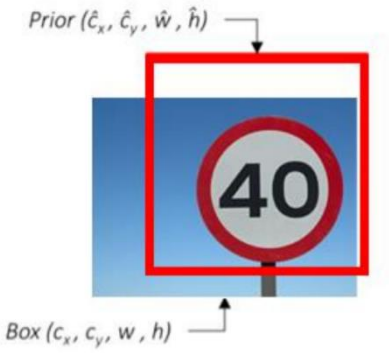


Figure 24 Center size coordinates of the ground truth box and the prior box's center size coordinates.

For each prior box to be adjusted to obtain a more precise prediction, these four offsets (gcx, gcy, gw, gh) called the geometric center size coordinates are the form in which bounding boxes coordinates are regressed and are calculated as follows. Where , Cy represents the center offsets of the ground truth box and $\hat{C}x$, $\hat{C}y$ represents the center offsets of the prior box compared to the bounds of the whole image. Similarly, w , h and $\hat{w}$, $\hat{h}$ represents the width and height of ground truth box and prior box respectively relative to the image.

$$g_{c_x} = \frac{c_x - \hat{c}_x}{\hat{w}} \qquad (4.1)$$

$$g_{c_y} = \frac{c_y - \hat{c}_y}{\hat{h}} \qquad (4.2)$$

$$g_w = \log\left(\frac{w}{\hat{w}}\right) \qquad (4.3)$$

$$g_h = \log\left(\frac{h}{\hat{h}}\right) \qquad (4.4)$$

Prediction convolutions

Two convolutional layers are needed for the prediction for each feature map.

• Localization prediction: A convolutional layer with 4 filters for a prior box calculate the four encoded offsets (gcx , gcy , gw, gh) for the bounding box predicted from that prior box.

• Class prediction: A convolutional layer with n-classes filters for each prior box present at the location.

All the filters are applied with a kernel size of (3, 3). They need not be the same shapes as the prior boxes because the different filters will learn to make predictions with respect to the different prior box shapes. The same is done for all the layers and stacked together.

Matching prior boxes to ground truth boxes

Each ground truth box is matched to prior box for the best IoU. Usually, the IoU threshold is 0.5 and index values below are labelled as background and higher ones as target object. This method simplifies the learning process and gives the network flexibility. Now each prior box or prediction has a match either positive or negative. Positively matched predictions have ground truth coordinates that will serve as targets for localization, i.e., in the regression task. Similarly, all predictions have a ground truth label which is either the type of object class if it's a positive match or a background class in case of a negative match. They are used as targets for class prediction, i.e., the classification task.

Localization loss

The localization loss is computed based on how accurately positively matched predicted boxes are regressed to the corresponding ground truth box coordinates. Therefore, it is the averaged Smooth L1 loss between the encoded offsets of positively matched localization boxes and their ground truths.

$$L_{loc} = \frac{1}{n_{positives}} \left(\sum_{positives} smooth\ L\ loss \right) \qquad (4.5)$$

Confidence loss

Every prediction, either negative or positive has a ground truth label associated with it. A greater number of the thousands of predictions contain no object or background. The solution is to confine the number of negative matches that will be evaluated in the loss function.

Hard negative mining

Choosing the predictions where the model found it hardest to recognize that there are no objects is called Hard negative mining. To avoid the class imbalance between the large number of negative and small number of positive prior boxes, they are sorted by using the 3:1 ratio. This helps in a stable training process. Therefore, the confidence loss is simply the sum of the Cross Entropy losses among the positive and hard negative matches.

$$L_{conf} = \frac{1}{n_{positives}} \left(\sum_{positives} CE\ Loss + \sum_{hard\ negatives} CE\ Loss \right) \quad (4.6)$$

Multibox loss It is the aggregate of the two losses from both types of predictions – bounding box localizations and class scores combined in a ratio α.

$$L = L_{conf} + \propto \cdot\ L_{loc} \quad (4.7)$$

where $\propto$ is set to 1, following [5].

Post processing: NMS(Non Maximum Suppression)

During training and inference, final detection results are obtained by conducting NMS. It is a greedy algorithm that filters out duplicate and redundant predictions for the same object.

Algorithmically, it is carried out as follows:

- First the candidates for each non-background class are selected.
- Arrange the candidates for a class in order of decreasing likelihood.
- Consider the highest score candidate. Eliminate all candidates with lesser scores that have an IoU of more than 0.5 with this candidate.
- Consider the next highest score candidate in the pool. Eliminate all candidates with lesser scores that have an IoU of more than 0.5 with this candidate.
- Repeat till the end of entire sequence of candidates.

The result is a single box, for each object in the image.

4.2 Implementation flow of the traffic sign and light detector

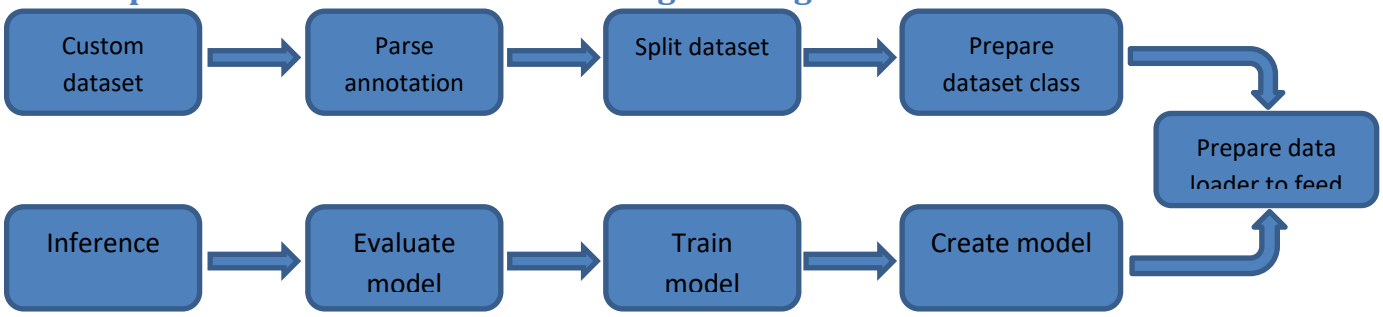


Figure 25 Overall flow diagram of the traffic sign and light detector algorithm

The training, validation and inference modules follow a sequence of functions that are shown in Figure 4.6

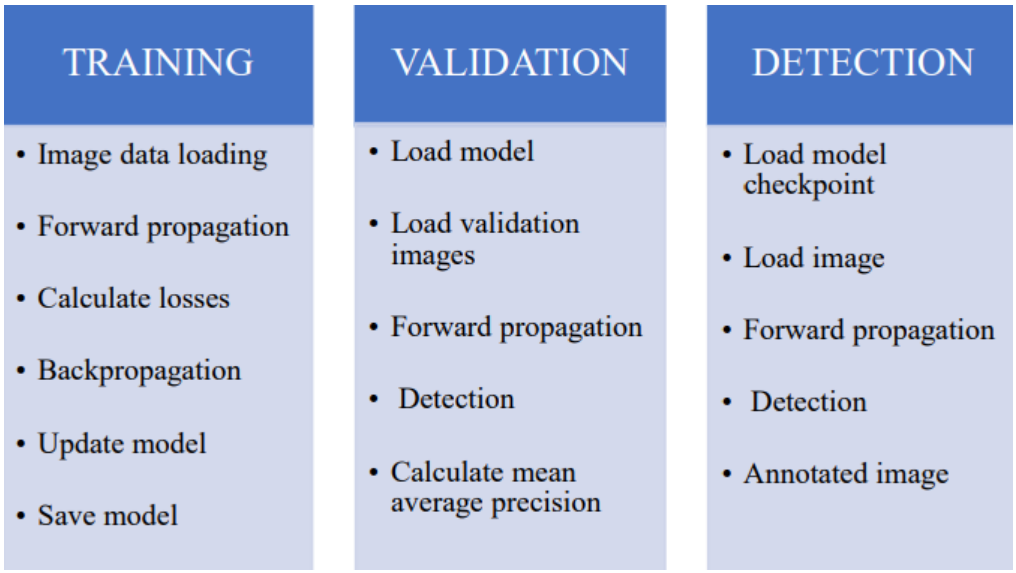


Figure 26 Functions used in training, validation, and detection modules

4.3 Dataset and data preprocessing

In this thesis, the dataset used is the Road sign detection dataset [57]. This dataset provides the images of traffic signs and light both, that serves the purpose of this application. The size of dataset is also manageable with the current computational platform. Currently, there are no research results on this newly emerging dataset. It consists of 877 images. The four classes include: speed limit, stop, crosswalk and traffic light. Figure 4.8 shows a few sample images from the dataset.

Figure 27 Sample images from the dataset

Each image has an annotation file that contains the coordinates of the ground truth bounding box and the class ID of the target object. Other related information include size, segmented, occluded, difficult, truncated and pose. One or more traffic signs and lights can be included in a sample image.

For evaluation purposes, the dataset was split into 80% for the training set and 20% for the test set. The training set consists of 701 training images containing a total of 2103 objects. Evaluation set consists of 175 test images containing a total of 525 objects.

The dataset class module was used to define the training and validation/test datasets. It has two methods; one method returns the size of the dataset, and the other method returns the image along with the bounding boxes of the objects and labels for the objects in this image. It also returns the perceived detection of each of these objects which is required only in the evaluation stage and not required for the training purposes.

4.4. Computational Platform and Training Setup

The experimental evaluation of the model was carried out on a computational platform through Google Colab Pro with the following specifications. A 2.20GHz Intel(R) Xeon(R) CPU with a 12 GB memory, and a 1.59 GHz NVIDIA T4 GPU. The entire program was written in PyTorch 1.9.0+cu102 with Python 3.7.10. The source code is based on an open-source repository [59]. The model was trained using the parameters shown in Table 4.4.

Batch size	16
Optimizer	SGD (Stochastic Gradient Descent)
Learning rate	0.001
Learning rate decay policy	Drop by a factor of 10
Momentum	0.9
Weight decay	0.0005

Table 4 Configuration Parameters

4.5. Experiments

Experiment I was performed to evaluate and compare the performance of MobileNet-v2 (MB_v2) and MobileNet-v1 (MB_v1) as feature extractors with SSD for the traffic sign and light detection using the Road sign detection dataset. Experiment II was performed to evaluate the accuracy and detection time of the proposed model. Experiment III was performed to do a comparative analysis on the three models, that is SSD with VGG16, SSD with MobileNet-v2 and SSD with MobileNet-v2 with two layers.

4.5.1. Experiment I Evaluate accuracy of SSD with different base networks over the Road sign detection dataset

Experiment I was performed to compare and select between the two MobileNet versions. We also evaluated performance of SSD using VGG16 as the base network over the same dataset. The results were evaluated for 10 epochs over the Road sign detection dataset. Table 4.5 shows the results for accuracy in terms of mAP @ IoU threshold of 0.5 with the three base networks.

SSD300 w/Base network	mAP @ IoU 0.5	No. of Parameters
SSD/VGG16	0.489	24,146,894
SSD/MB_v1	0.010	6,869,966
SSD/MB_v2	0.307	7,222,518

Table 5 Results (mAP) for SSD with three networks

Here, MobileNet-v2 (MB_v2) and VGG16 are using pretrained models on ImageNet available in PyTorch framework for the purpose of transfer learning . While MobileNet-v1 (MB_v1) pretrained model is not available in PyTorch and therefore trained from scratch. That's the reason for low accuracy in case of SSD with MB_v1 (SSD/MB_v1). Since SSD/MB_v2 provides better accuracy results therefore MobileNet_v2 was used for further experimentation.

4.5.2. Experiment II Evaluate accuracy with variation in models using different number of layers

While dealing with the convolutional networks, there are two ways to know what a model sees. First are the filters (weights) and second are the feature maps (activation map). These feature maps are the result of applying filters to input images and filters are what detects the patterns. Patterns means edges, shapes, textures, curves, objects, and colors.

At the start of the CNN, we come across these simple and kind of geometric filters to detect edges, corners, circles, and squares. Therefore, first layer detects edges, next combines them to detect shapes. The deeper the network goes the filters become more sophisticated to detect higher level features.

As shown in Figure 4.1, the SSD algorithm uses six feature map layers. Feature maps of size 19×19 performs better for small objects detection while the later layers work better for larger objects detection, especially 3×3, as mentioned in [10].

Based on the above observations, we experimented with different number of feature map layers each with a different scale. Table 4.6 shows the first model SSD with MB_v2 (SSD/MB_v2) with all the six feature map layers. Second entry of the table is the model SSD with MB_v2 and one feature map layer (SSD/MBv2_Onelayer) created with only 19 × 19 feature map layer exclusively while the other five layers were removed. Then the next model (SSD/MBv2_Twolayer) was created that includes only the 19 × 19 and 10 × 10 feature map layers and the last model (SSD/MBv2_Threelayer) created with three feature map layers. Table 4.6 shows the results in terms of mAP of the above-mentioned models carried out for 10 epochs.

SSD300 w/Model	mAP @IoU 0.5	No. of Parameters	No. of predictions per class
SSD/MB_v2	0.307	7,222,518	2,268
SSD/MBv2_Onelayer	0.391	3,536,524	1,444
SSD/MBv2_Twolayer	0.433	4,158,146	2,044
SSD/MBv2_Threelayer	0.370	6,932,088	2,194

Table 6 Results of models created with different no. of feature map layers.

The analysis of results from Experiment II shows that the model SSD/MBv2_Twolayer is giving better results than the rest of the models and the accuracy started deteriorating from the model comprised of three layers. To further validate these results, the segregated objects areas were analyzed based on the size of their corresponding bounding boxes with respect to the full image to calculate percent coverage. Figure 4.12 shows the results for the validation dataset.

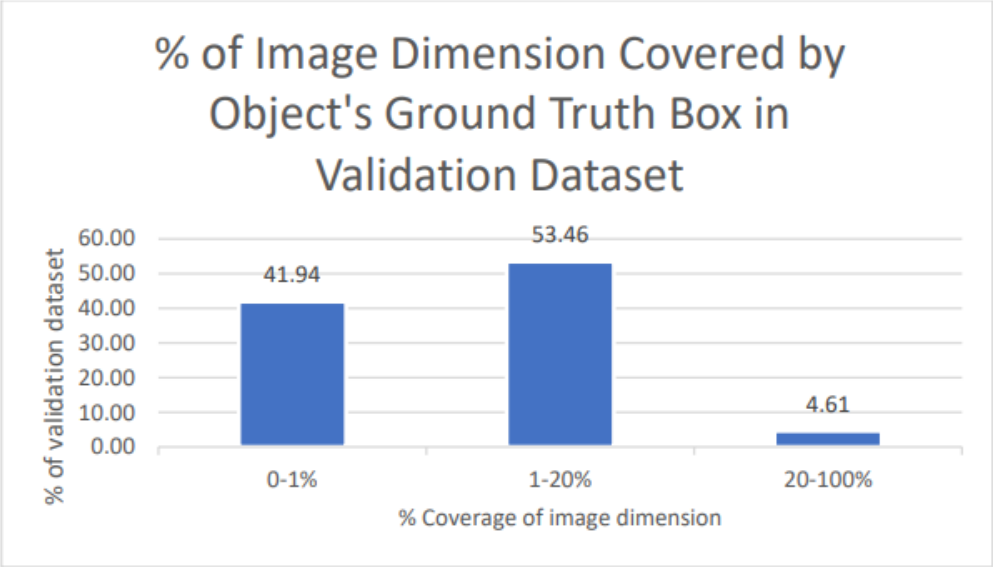


Figure 28 Percentage coverage of image dimension by the object's ground truth box

Results shows that the validation dataset comprised of 95.4 % of objects having maximum 20% of image dimension coverage which means that the objects come under the range of small/ medium size objects and that's the reason for increased accuracy of SSD/MBv2_Twolayer as compared to SSD/MB_v2.

In terms of the number of parameters shown in Table 4.6, there is less number of parameters involved in SSD/MBv2_Twolayer as compared to SSD/MB_v2 with all the six feature map layers. Less number of parameters means a lighter architecture with less computational complexity and that makes the model more suitable for real time detection. Also, the table shows that the model SSD/MBv2_Twolayer involves 224 lesser number of predictions per class as compared to the model with all the six layers (SSD/MB_v2) leading to a smaller number of computations resulting in decreased training time.

4.5.3. Experiment III Comparative analysis

Using the experimental results tabulated in Table 4.6 as a proof of concept, we further conducted an ablation study by training our two-layer model (SSD/MBv2_Twolayer) with the four augmentation techniques: photometric distortions, expand image (zoom out), randomly cropped image, and horizontal flip to get better generalization performance. Where photometric distortions include randomly adjusting brightness, contrast, saturation, and hue, each with a 50% chance. While zoom out image operation range 1 - 4 times as large as the original image and randomly cropped image dimensions between 0.3 and 1 times the original dimensions and the aspect ratio between 0.5 and 2. Figure 4.13 shows the training time performance of the model on the validation dataset.

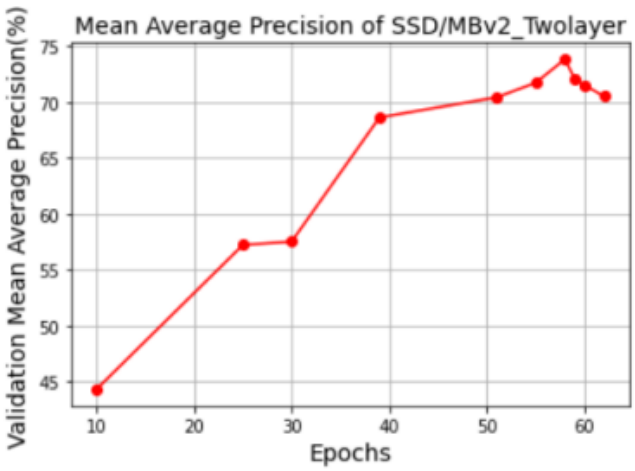


Figure 29 Training performance of SSD/MB-v2_Twolayer up to 62 epochs

Note that in Figure 4.13, the maximum mAP is obtained at the 58th epoch. Therefore, the other two models: SSD/VGG16 and SSD/MB_v2 were trained for the same no of epochs. Next the performance of the three models was also evaluated on the training dataset. The results are shown in Figure 4.14.

It shows the accuracy results in terms of mAP on training and validation dataset of the three models.

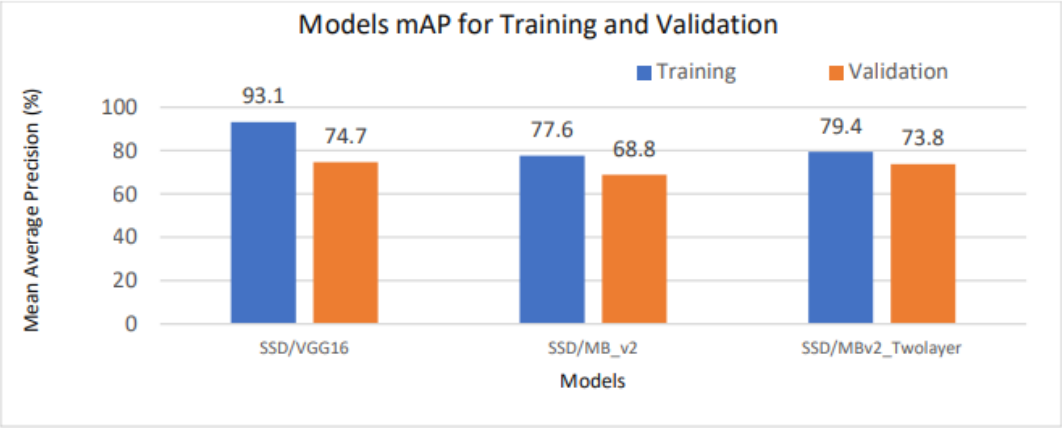


Figure 30 Three models mAP (%) on training & validation dataset

It can be observed that in case of all the three models the difference between the validation accuracy and the training accuracy is not that big (no overfitting).

As part of ablation study the average detection time of an image on a CPU and a GPU is compared of all the three models as shown in Figure 4.15.

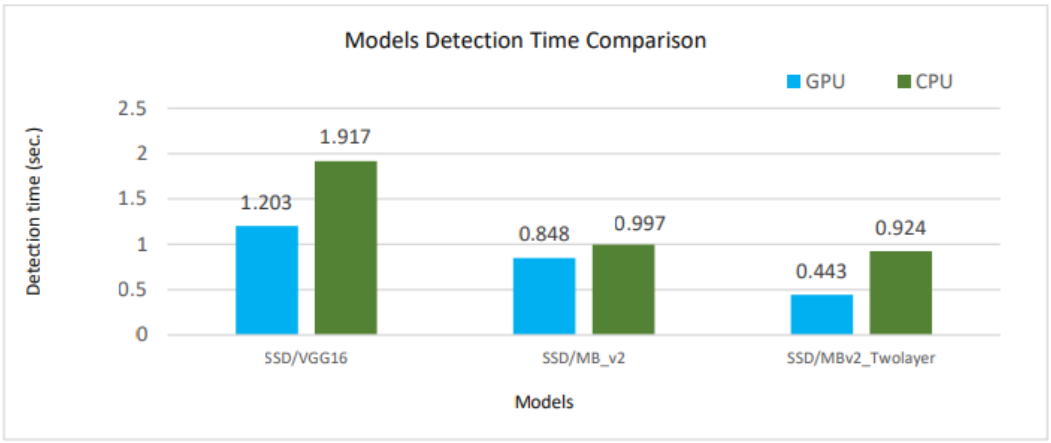


Figure 31 Detection time comparison

Figure 4.15 shows the three models detection time comparison. Detailed analysis on Figure 4.15 is provided in Section 4.7.

The individual class average precision of the three models was also evaluated for a better comparison as shown in Figure 4.16.

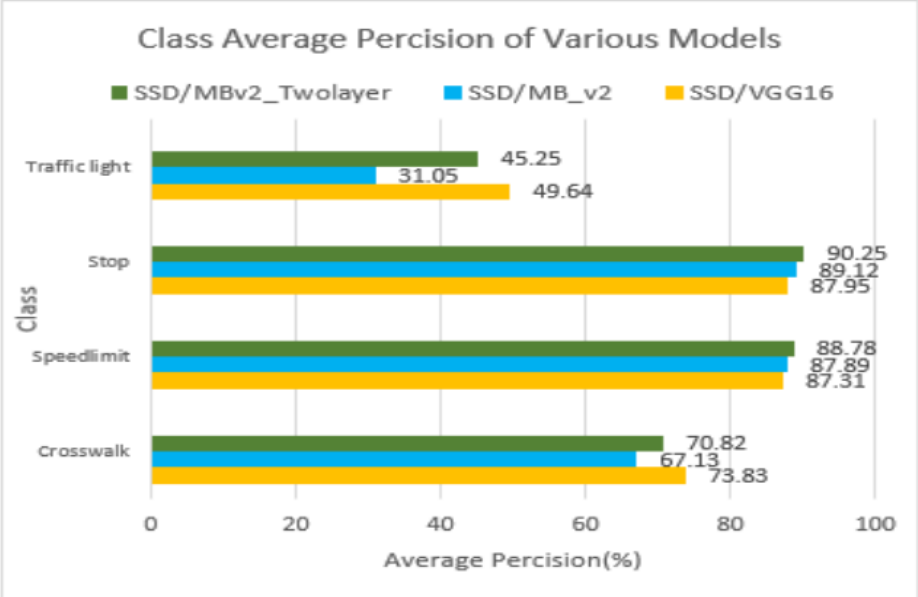


Figure 32 Class average precision of the 3 models

It can be noticed in Figure 4.16 that the AP of the traffic light class is low as compared to the AP of the other traffic sign classes in all the three models. The main reason for it is that the traffic lights need more contextual information because of its orientation layout to be detected by the model. This problem could be solved by training the model with a variety of traffic light layouts and a bigger dataset. While the traffic signs have a very specific shape and a layout position in all the images and therefore give better precision.

4.6. Quantitative Analysis

Finally, based on the results shown in Table 4.7, the object detector model (SSD/MBv2_Twolayer) is selected as the proposed model for the application. This model proved to be faster and of comparable accuracy as SSD/ VGG16.

Models	mAP(%) @ 0.5 IoU	Detection time		No. of Parameters (in millions)
		GPU (sec.)	CPU (sec.)	
SSD/ VGG16	76.7	1.203	1.917	24.1
SSD/ MB_v2	68.8	0.848	0.997	7.2
SSD/ MBv2_Twolayer (Proposed)	73.8	0.443	0.924	4.1

Table 7 Comparison of results of the proposed model against other base models

As expected, SSD/VGG16 gives a better mAP value when compared with the other two models because of the use of a deeper base network. But the proposed adapted version is rendering a comparable accuracy as SSD/ VGG16 and a 5% increase in mAP when compared to SSD/ MB_v2 (73.8% vs. 68.8% respectively). As noticed in Figure 4.15, in terms of detection time, SSD/VGG16 is slower when compared to the proposed model (1.2 sec. vs. 0.44 sec. respectively on the GPU) that is a 63% decrease in average detection time and 48% decrease when compared with SSD/ MB_v2. Also, the results are noticeable in case of CPU when compared the proposed model with SSD/VGG16 (0.92 sec. vs. 1.91 sec. respectively). These results shows that the proposed model can be executed on the embedded platform in real time.

4.7. Qualitative Analysis

Figure 4.17 shows the visual results for the comparison of the three models. We can clearly notice in Figure 4.17, that the proposed model's performance is much better as compared to the two models, especially in case of image ids: road807, road748, road716 and road213 not only able to accurately detect different types of object classes, as well as small object detection is much better as compared to the model SSD/VGG16. It is also noticeable in image id: road821 that the proposed model can detect two extra traffic lights that the other two models were not able to detect.

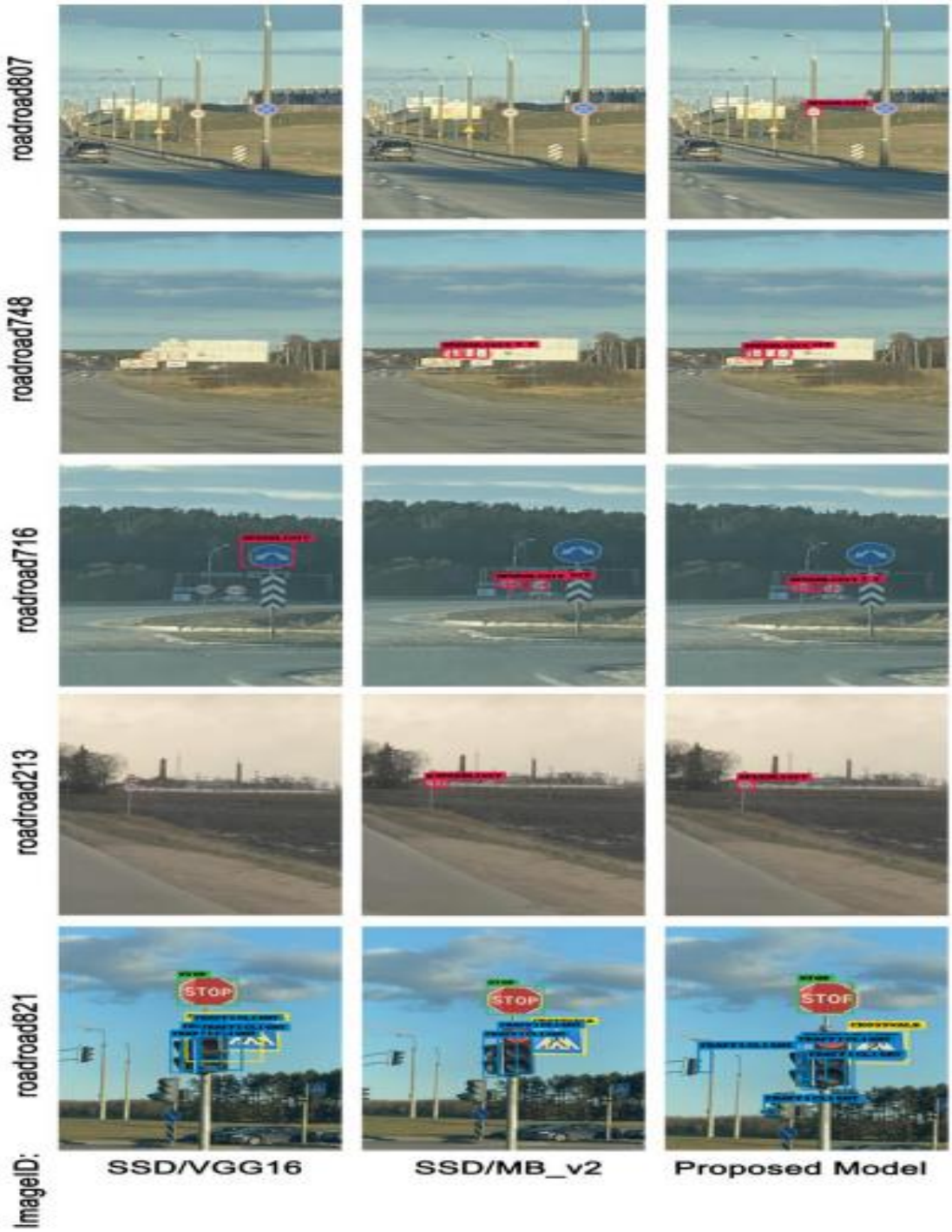


Figure 33 Visual results for SSD/VGG16(1st col.), SSD/MB_v2(2nd col), Proposed model (3rd col.)

Chapter 5.

CONCLUSIONS AND FUTURE WORK

This thesis presents an adaptation of a single shot detection model for a traffic sign and light detection and small object detection in general. Through exhaustive experimental study, this work shows that a modification of the standard SSD object detection architecture by replacing the backbone, VGG16 with a lighter and faster MobileNet_v2 and using only the top-two feature map layers of SSD can provide real-time detection. Such, modifications are proved to be essential not only for small object detection but also faster detection time.

The proposed model has comparable performance of 73.8% mAP with faster detection time than the baseline SSD models. It provides a 63% and a 48% reduction in the detection time on a GPU when compared against the baseline models SSD with VGG16 and SSD with MobileNet_v2 respectively. The results for the proposed model can be further improved by training the model on a bigger dataset. Apart from the intended application, traffic sign and light detection, the proposed model turned out to be beneficial for the application domains, where the detection of small objects plays an important role. The process followed in this paper can be used to design object detectors for specific applications with specific characteristics in terms of shape, for example, streetlight detection, waste bin detection for automatic waste collection, etc.

REFERENCES

[1] Andersen, J., & Sutcliffe, S. (2000). Intelligent Transport Systems (ITS) - An Overview. IFAC Proceedings Volumes, 33(18), 99-106.

[2] Devapriya, W., Nelson Kennedy Babu, C. and Srihari, T. (2016) Real Time Speed Bump Detection Using Gaussian Filtering and Connected Component Approach. Circuits and Systems, 7, 2168-2175.

[3] Brookhuis, K. A., Waard, D. D., & Janssen, W. H. (2001). Behavioural impacts of Advanced Driver
Assistance Systems–an overview. EJTIR, 3(1), 245-253.

[4] Devapriya, W., Babu, C. N., & Srihari, T. (2015). Advance Driver Assistance System (ADAS) - Speed bump detection. 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 478-483.

[5] Reddy, V. K., & Nagesh, B. S. (2016). Smart Phone Based Speed Breaker Early Warning System. International Journal of Computer Science and Information Security (IJCSIS), 14, 20-25.

[6] Rahayu, E., Faizal, M., & Ani, Z. C. (2016). A Vision Based Speed Breaker Detection for Early Warning Notification. 43-47.

[7] Optical Character Recognition (OCR) - File Exchange - MATLAB Central. Retrieved October 18, 2018, from https://www.mathworks.com/matlabcentral/fileexchange/18169-optical-character-recognition-ocr

[8] Celaya-Padilla, J., Galván-Tejada, C., López-Monteagudo, F., Alonso-González, O., Moreno-Báez, A., Martínez-Torteya, A., . . . Gamboa-Rosales, H. (2018). Speed Bump Detection Using Accelerometric Features: A Genetic Algorithm Approach. Sensors, 18(2), 443

P, M., Singh, S., Shukla, S., & Krishnan, U. (2017). detection of humps and potholes on roads and notifying
the same to the drivers. International Journal of Management and Applied Science, 3(1), 130-133.

[10] Danti, A., Dr, Kulkarni, J. Y., Smt, & Hiremath, P. S., Dr. (2013). A Technique for Bump Detection in Indian Road Images Using Color Segmentation and Knowledge Base Object Detection. International Journal of Scientific & Engineering Research, 4(8).

[11] Afrin, M., Mahmud, M. R., & Razzaque, M. A. (2015). Real time detection of speed breakers and warning system for on-road drivers. 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), 495-498. Chugh, G., Bansal, D. and Sofat, S. (2014) Road Condition Detection Using Smartphone Sensors: A Survey. International Journal of Electronic and Electrical Engineering, 7, 595-601.

[13] Manikandan , B., & Bharathi, M. (2018). SPEED BREAKER DETECTION USING BLOB ANALYSIS. International Journal of Pure and Applied Mathematics, 118(20), 3671–3677.

[14] Anusha, K. & Sirisha, K. (2011) "Breaking the Speed Breakers Using Image Processing" Retrieved from https://www.scribd.com/document/54957257/Breaking-the-Speed-Breakers-Using-Image-Processing- 1-2

[15] Sagar, B. M., G, Shobha., & P. Ramakanth Kumar. (2008). OCR for printed Kannada text to Machine editable format using Database approach. WSEAS Transactions on Computers, 7(6), 766–769. [16] Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical Character Recognition. International Journal of Recent Technology and Engineering, 2(1), 72–75.

[17] Sadasivan, A. K., & Senthilkumar, T. (2012). Automatic Character Recognition in Complex
Images. Procedia Engineering, 30, 218–225.

[18] Tiwari, S., Mishra, S., Bhatia, P., & Yadav, P. K. (2013). Optical Character Recognition using
MATLAB. International Journal of Advanced Research in Electronics and Communication Engineering, 2(5), 579–582.

[19] Singla, S., & Yadav, R. (2014). Optical Character Recognition Based Speech Synthesis System Using LabVIEW. Journal of Applied Research and Technology, 12(5), 919–926.

[20] Greenhalgh, J., & Mirmehdi, M. (2015). Recognizing Text-Based Traffic Signs. IEEE Transactions on Intelligent Transportation Systems, 16(3), 1360–1369.

[21] Huang, D., Shan, C., Ardabilian, M., Wang, Y., & Chen, L. (2011). Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 41(6), 765–781.

[22] Irhebhude, M. E., Nawahda, A., & Edirisinghe, E. A. (2016). View invariant vehicle type recognition and counting system using multiple features. International Journal of Computer Vision and Signal Processing, 6(1), 20-32.

[23] Brahnam, S., Jain, L. C., Lumini, A., & Nanni, L. (2013). Introduction to Local Binary Patterns: New Variants and Applications. Local Binary Patterns: New Variants and Applications Studies in Computational Intelligence, 1–13.

[24] Pupale, R. (2019, February 11). Support Vector Machines(SVM) - An Overview. Retrieved August 9, 2019, from https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

[25] Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20, 273–297.

[26] Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 13(2), 415–425.

[27] Gonzalez, C. & Woods, R.E. (2013) Digital Image Processing, Person. 3rd Edition, Person, New Delhi, 635,738.

[28] Marita, T., Negru, M., Danescu, R., & Nedevschi, S. (2011). Stop-line detection and localization method for intersection scenarios. 2011 IEEE 7th International Conference on Intelligent Computer
Communication and Processing, 293–298.

[29] Andersen, J., & Sutcliffe, S. (2000). Intelligent Transport Systems (ITS) - An Overview. IFAC Proceedings Volumes, 33(18), 99-106.

[30] Devapriya, W., Nelson Kennedy Babu, C. and Srihari, T. (2016) Real Time Speed Bump Detection Using Gaussian Filtering and Connected Component Approach. Circuits and Systems, 7, 2168-2175. [3] Brookhuis, K. A., Waard, D. D., & Janssen, W. H. (2001). Behavioural impacts of Advanced Driver Assistance Systems–an overview. EJTIR, 3(1), 245-253.

[31] Devapriya, W., Babu, C. N., & Srihari, T. (2015). Advance Driver Assistance System (ADAS) - Speed bump detection. 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 478-483.

[32] Reddy, V. K., & Nagesh, B. S. (2016). Smart Phone Based Speed Breaker Early Warning System. International Journal of Computer Science and Information Security (IJCSIS), 14, 20-25.

[33] Rahayu, E., Faizal, M., & Ani, Z. C. (2016). A Vision Based Speed Breaker Detection for Early Warning Notification. 43-47.

[34] Devapriya, W., Babu, C.N.K., Srihari, T.: Real time speed bump detection using Gaussian filtering and connected component approach. In: Futuristic Trends in Research and Innovation for Social Welfare (Start up Conclave), World Conference on. pp. 1–5. IEEE (2016)

[35]. Moujahid A, Hina MD, Soukane A, Ortalda A (2018) Machine learning techniques in ADAS:

a review, pp 235–242

[36]. Viswanathan V, Hussein R (2017) Applications of image processing and real-time embedded

systems in autonomous cars: a short review. Int J Image Process 11:35

[37]. Byun J, Seo B-S, Lee J (2015) Toward accurate road detection in challenging environments

using 3D point clouds. ETRI J 37:606–616

[38]. Vo AV, Truong-Hong L, Laefer DF (2015) Aerial laser scanning and imagery data fusion

for road detection in city scale. In: 2015 IEEE International Geoscience and remote sensing

symposium, pp 4177–4180

[39]. Carlos MR, Aragon ME, Gonzalez LC, Escalante HJ, Martinez F (2018) Evaluation of detection

approaches for road anomalies based on accelerometer readings–addressing who's who. IEEE

Trans Intell Transp Syst 1–10

[40]. Tonde PVP, Jadhav A, Shinde S, Dhoka A, Bablade S (2015) Road quality and Ghats complexity

analysis using Android sensors. Int J Adv Res Comput Commun Eng 4:101–104

[41]. Priyanka Ramnath C (2015) Advanced driver assistance system (ADAS). Int J Adv Res Electron

Commun Eng 4:2278–2909

[42]. Chen HT, Lai CY, Shih CA (2016) Toward community sensing of road anomalies using monocular vision. IEEE Sens J 16:2380–2388

[43]. Gaisser F, Jonker PP(2017) Road user detection with convolutional neural networks: an application to the autonomous shuttle WEpod. In: Proceedings of 15th IAPR international conference

on machine vision applications (MVA), pp 101–104

[44]. Melo S, Marchetti E, Cassidy S, Hoare E, Gashinova M, Cherniakov M (2018) 24 GHz interferometric radar for road hump detections in front of a vehicle. In: 2018 19th international

radar symposium, pp 1–9

[45]. Lee M, Hur S, Park Y (2015) An obstacle classification method using multi-feature comparison based on 2D LIDAR database. In: Proceedings of the 12th international conference on

information technology—new generations ITNG 2015, pp 674–679

[46]. Craciun D, Deschaud J-E, Goulette F (2017) Automatic Ground Surface Reconstruction from

mobile laser systems for driving simulation engines. Simulation 93:201–211

[47]. Masino J, Thumm J, Frey M, Gauterin F (2017) Learning from the crowd: road infrastructure

monitoring system. J Traffic Transp Eng (English edn) 4:451–463

[48]. Celaya-Padilla JM, Galván-Tejada CE, López-Monteagudo FE, Alonso-González O, MorenoBáez A, Martínez-Torteya A, Galván-Tejada JI, Arceo-Olague JG, Luna-García H, GamboaRosales H (2018) Speed bump detection using accelerometric features: a genetic algorithm

approach. Sensors (Switzerland). 18:1–13

[49]. Goregaonkar RK (2014) Assistance to driver and monitoring the accidents on road by using

three axis accelerometer and GPS system. Int J Electron Commun Comput Eng 5:260–264

[50]. Reddy VK, Engineering RB (2016) Smart phone based speed breaker early warning system.

Spec Issue Int J Comput Sci Inf Secur 14:20–25

[51]. Shelke V, Kalbhor T, Khanekar S, Shitole B, Kadam YV (2017) Study of estimation of road roughness condition and ghat complexity analysis using smartphone sensors

[52]. Dange T, Mahajan DV (2015) Analysis of road smoothness based on smartphones. Int J Innov

Res Comput Commun Eng 3:5201–5206

[53]. Mahajan DV, Dange T (2015) Estimation of road roughness condition by using sensors in smartphones. Int J Comput Eng Technol 6:41–49

[54]. Daraghmi Y-A, Daadoo M (2016) Intelligent smartphone based system for detecting speed

bumps and reducing car speed. In: MATEC web conference, vol 77, p 09006

[55]. Seraj F, van der Zwaag BJ, Dilo A, Luarasi T, Havinga P (2014) RoADS: a road pavement monitoring system for anomaly detection using smart phones. In: Proceedings of the 1st international work. Machine learning urban sensor data, SenseML, pp 1–16

[56] Gónzalez LC, Martínez F, Carlos MR (2014) Identifying roadway surface disruptions based on accelerometer patterns. IEEE Lat Am Trans 12:455–461

[57] Devapriya W, Babu CNK, Srihari T (2016) Advance driver assistance system (ADAS)—speed

bump detection. In: 2015 IEEE international conference on computational intelligence and

computing research (ICCIC 2015)

[58]. Devapriya W, Babu CNK, Srihari T (2016) Real-time speed bump detection using Gaussian

filtering and connected component approach. In: IEEE WCTFTR 2016—proceedings of 2016

world conference futuristic trends in research and innovation for social welfare

[59]. Geetha Kiran A, Murali S (2014) Automatic hump detection and 3D view generation from a

single road image. In: Proceedings of 2014 international conference on advances in computing,

communications and informatics, ICACCI 2014, pp 2232–2238

[60]. Chen HT, Lai CY, Hsu CC, Lee SY, Lin BSP, Ho CP (2014) Vision-based road bump detection using a front-mounted car camcorder. In: Proceedings—international conference pattern

recognition, pp 4537–4542

[61]. Jo Y, Ryu S (2015) Pothole detection system using a black-box camera. Sensors (Switzerland)

15:29316–29331

[62]. Lee J, Yoon K (2018) Temporally consistent road surface profile estimation using stereo vision

63. Himmelsbach M, von Hundelshausen F, Wuensche H (2010) Fast segmentation of 3D point

clouds for ground vehicles. Iv, pp 560–565

[64]. Guo C, Sato W, Han L, Mita S, McAllester D (2011) Graph-based 2D road representation of

3D point clouds for intelligent vehicles. In: IEEE Intelligent Vehicles symposium proceedings,

pp 715–721

[65]. Choi S, Park J, Byun J, Yu W (2014) Robust ground plane detection from 3D point clouds. In:

Proceedings of 2014 14th international conference control, automation, and systems (ICCAS

2014), pp 1076–1081 (2014)

[66]. Zhang Z, Huang K, Tan T (2006) Comparison of similarity measures for trajectory clustering in

outdoor surveillance scenes. In: Proceedings—International Conference on pattern recognition,

vol 3, pp 1135–1138

[67]. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite

[68]. Habermann D, Hata A, Wolf D, Osório FS (2013) Artificial neural nets object recognition for

3D point clouds, pp 101–106

[69] B.Wu, F.Iandola, P.H.Jin and K.Keutzer, “SqueezeDet: Unified, small, low power fully

Convolutional neural networks for real time object detection for autonomous driving”, in Proc. IEEE Conference on Computer Vision Pattern Recognition Workshops (CVPRW), Jul. 2017, pp. 446454.

[70] B.Reddy, Y.H.Kim, S.Sun, C.Seo and J.Jang, “Real time driver dowsiness detection for

embedded system using model compression of deep neural networks", in Proc. IEEE Conference on Computer Vision Pattern Recognition Workshops (CVPRW), Jul. 2017, pp. 438445.

[71] A. Kozlov and D. Osokin, " Development of real time ADAS object detector for deployment

on CPU", in Proc. SAI Intelligent System Conference, 2019, pp. 740750.

[72] G. Brilli, P. Burgio, and M. Bertogna, ``Convolutional neural networks on embedded

automotive platforms: A qualitative comparison," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS),* Jul. 2018, pp. 496_499.

[73] B. Meus, T. Kryjak, and M. Gorgon, ``Embedded vision system for pedestrian detection

based on HOGCSVM and use of motion information implemented in Zynq heterogeneous device," in *Proc. Signal Process.,Algorithms, Archit., Arrangements, Appl. (SPA),* Sep. 2017, pp. 406_411.

[74] Yongbon Koo, Chayoung You and SungHoon Kim. OpenCL-Darknet: An OpenCL

Implementation for Object Detection. IEEE International Conference on Big Data and Smart Computing, 2018.

[75] ShaoqingRen, KaimingHe, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time

object detection with region proposal networks.IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 39, Issue:6, June 1 2017

[76] JifengDai, Yi Li, KaimingHe, and Jian Sun. R-FCN: object detection via region-based fully

convolutional networks. CoRR, abs/1605.06409, 2016.

[77] Santosh Redmon, Joseph and Divvala, Ross Girshick, and Ali Farhadi. You only look once:

Unified, real-time object detection. arXivpreprint arXiv:1506.02640, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788 2016

[78] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multiboxdetector. ECCV, European Conference on Computer Vision, pp 21-37, 2016.