Deep Learning-Based Real-Time Pothole Detection for Avoiding Road Accident

by

Rafsan Basher
17301042
Asif Raihan Ayon
17301170
Avijit Gharamy
19101519
Abdullah Al Zayed
17301126
Md Samin Yeasar Ibna Zaman
17101533

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rafsan Basher
17301042

Asif Raihan Ayon
17301170

Avijit Gharamy
19101519

Abdullah Al Zayed
17301126

Md Samin Yeasar Ibna Zaman
17101533

Approval

The thesis/project titled "Deep Learning-Based Real-Time Pothole Detection for Avoiding Road Accident" submitted by

1. Rafsan Basher (17301042)
2. Asif Raihan Ayon (17301170)
3. Avijit Gharamy (19101519)
4. Abdullah Al Zayed (17301126)
5. Md Samin Yeasar Ibna Zaman (17101533)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 18, 2021.

Examining Committee:

Supervisor:
(Member)

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
BRAC University (On Leave)
Assistant Professor(Research Track)
Technology Studies Department
Woosong University, Daejeon, South Korea

Co Supervisor:
(Member)

Faisal Bin Ashraf
Lecturer
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

DR. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
BRAC University

Ethics Statement

For the demonstration of our proposed model, we have researched a variety of articles, took help from websites, journals and publications. Our data has been taken from Kaggle and Github.

Abstract

Bangladesh is a fast-developing country, and the number of roads increasing with it is immense. With the ever-growing amount of road comes the age-old problem of a pothole. This paper represents a model of deep learning-based, real-time pothole detection for finding and avoiding road accidents. Any types of image processing-based detection, in this case, pothole detection, are done through various steps. For example, collecting data sets is one of the most crucial steps to create any recognition system. Labeling an image means pinpointing the subject which we will be trying to find. Training the algorithm through those images to detect the subjects is critical in detecting potholes. In this research paper, to detect potholes from real-time videos, firstly, we collected data sets containing more than 600 images of potholes. After that, we labeled those images through labeling software. Then in chapter-1 we used those images to train the model (MobileNet, Inception-v3) which was detecting potholes from still photos given to it. Next, we used YOLOv5 to detect potholes from real-time feeds. In this proposed system, by using the real-time feed, potholes will be detected. Moreover, this will help the masses to detect potholes on roads to avoid accidents, and it will also help people related to the road works to find the potholes for further road maintenance.

Keywords: Road, Pothole, Deep-learning, Image processing, Real-Time, MobileNet, Inception- v3, YOLOv5.

Acknowledgement

First of all, All gratitude to Allah, for whom our thesis was completed without serious setbacks.
Secondly, This project would not have been possible without the support of many people. Many thanks to our supervisor, Dr. Jia Uddin, who read our numerous revisions and helped make some sense of the confusion. Without his guidance this would have been a difficult ordeal.
Thirdly, also thanks to our Co Supervisor, Faisal Bin Ashraf who offered guidance and support.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration i

Approval ii

Ethics Statement iv

Abstract v

Acknowledgment vi

Table of Contents vii

List of Figures ix

List of Tables xi

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Research Problem . . . 2
- 1.3 Research Objectives . . . 3
- 1.4 Research Orientation . . . 3

2 Literature Review 4

3 Work Plan 7

4 Pothole Detection using MobileNet & Inception-v3 8
- 4.1 Methodology . . . 8
- 4.2 Model Description . . . 9
  - 4.2.1 MobileNet . . . 9
  - 4.2.2 Inception-v3 . . . 12
- 4.3 Data Preliminary Analysis . . . 15
  - 4.3.1 Collecting Dataset . . . 15
  - 4.3.2 Labeling Image . . . 15
  - 4.3.3 Converting Images into Arrays . . . 15
  - 4.3.4 Normalizing the data . . . 16
  - 4.3.5 Training using deep learning (MobileNet, Inception-V3) . . . 16
  - 4.3.6 Implementation of transfer learning . . . 16
  - 4.3.7 Binary classification (pothole/normal) . . . 16
  - 4.3.8 Train and test . . . 17

4.4 Result Analysis 17

5 Detecting Potholes using YOLOv5 **22**
5.1 Methodology 22
5.2 Model Description 23
5.2.1 YOLOv5 23
5.3 Data Preliminary Analysis 25
5.3.1 Collecting Dataset 25
5.3.2 Image Augmentation 26
5.3.3 Labeling Image 26
5.3.4 Converting Images into Arrays 26
5.3.5 Normalizing the Data 27
5.3.6 Training Using Deep learning (YOLOv5) 27
5.3.7 Implementation Details 27
5.3.8 Train and Test 27
5.4 Result Analysis 28

6 Conclusion & Future Scope **32**
6.1 Conclusion 32
6.2 Future Scope 32

Bibliography **36**

List of Figures

3.1 Flowchart for detection of pothole. 7

4.1 flowchart for methodology 8
4.2 Architecture of MobileNet 9
4.3 The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter [32]. 12
4.4 compact-network replacing the 5 × 5 convolutions [30]. 13
4.5 Main Inception CNN model showed in [36]. 13
4.6 Components of inception with increased filter bank outputs [30]. . . . 14
4.7 Auxiliary classifier atop the previous17×17 layer [30]. 14
4.8 While reducing grid size, this Inception module also expands filter banks [30]. 15
4.9 F1 score of mobilenet. 17
4.10 F1 score of inception-v3. 17
4.11 Confusion Matrix of MobileNet 18
4.12 Confusion Matrix of Inception-v3. 18
4.13 Accuracy graph of MobileNet model on the given data set. 19
4.14 Loss graph of MobileNet model on the given data set. 19
4.15 Accuracy Graph of Inception-v3 model on the given data set. 20
4.16 Loss Graph of Inception-v3 model on the given data set. 20

5.1 flowchart for methodology. 22
5.2 Parts of YOLOv5. 23
5.3 Growth rate of k = 4 for 5-layer thick block All previous feature maps are used as input for each layer [40]. 24
5.4 Cross Stage Partial DenseNet (CSPDenseNet)[41]. 24
5.5 On top of the FPN, (b) PANet adds a bottom-up bottom-up pathway; (c) NAS-FPN applies the same block again to find an irregular feature network topology; Finally, (d) is our BiFPN with improved precision and performance trade-offs[42]. 25
5.6 Pothole Dataset 25
5.7 Labeling Pothole from dataset. 26
5.8 Visualization of instances in the data set. 27
5.9 Trained dataset. 28
5.10 Trained dataset. 28
5.11 F1 Graph. 29
5.12 F1 Confusion Matrix. 29
5.13 P Curve. 30

5.14 R Curve. 30
5.15 Output results from live feed. 31

List of Tables

4.1 Output Results of MobileNet and Inception-v3 21

Chapter 1

Introduction

1.1 Overview

Potholes have been a significant cause of road accidents and vehicle damage. Recently, as vehicular traffic and pollution levels have increased, the roads have become increasingly congested. Almost every city in the world has large and small potholes in the street. Pothole detection by hand is a time-consuming and labor-intensive task. A variety of techniques, including vibration-based approaches, 3D reconstruction-based approaches, and vision-based methods, have been employed to detect potholes. Each of these strategies, however, has its own set of limitations [1]. This is why this research aims to see if the present algorithms for detecting potholes can be used to assist drivers or autonomous vehicles to detect potholes [2]. A pothole is a depression in the pavement surface that is bowl-shaped and has a minimum plan dimension of 150mm. A two-dimensional image-based methodology has been confined to pothole identification and is therefore incapable of determining the magnitude of potholes for assessment. To circumvent the limitations of the preceding method, video-based algorithms for detecting potholes and calculating the overall number of potholes across a sequence of frames have been developed [3]. Numerous factors influence pothole patching decisions, including traffic volume, the time remaining until scheduled rehabilitation or overlay, staff availability, equipment, materials, and the traveling public's tolerance. Material, labor, and equipment costs all affect the total cost-effectiveness of the patching procedure. The key to future reconstruction decision-making is the evaluation of damage based on gathered data [4]. Both internal and external factors cause potholes. Internal factors include the deterioration and responsiveness or durability of the pavement material to climate change factors such as heavy rainfall and heat. External factors include a lack of quality control and construction management. Advanced digital inspection trucks acquire pavement photos and video data in contemporary practice, but technicians manually analyze the damage estimation. As a result, it is a time-consuming and costly endeavor. Since it depends on the worker's precision and experience, it is frequently more subjective than objective. In addition, existing applications typically require complex equipment that is somewhat costly and requires specialized maintenance [5]. Keeping accordance with the development of Bangladesh, the number of the road is increasing rapidly. Moreover, in Bangladesh, road communication is crucial. Therefore, it has had remarkable growth in the road transportation sector during the last two decades. The total length of roads in the Bangladesh is around

271,000 kilometers (km), including approximately 21,000 kilometers of major highways [6]. While the number of roadways continues to expand, so does the number of automobiles on the road. Additionally, climate conditions, excessive road use, cars that exceed the road's statutory weight limit, and spontaneous construction work are all significant contributors to road damage in this country. Thus, this proposed system will detect potholes on the road and assist the general public in avoiding accidents while also assisting authorities in locating and fixing them.

1.2 Research Problem

A pothole's detection system is designed to warn drivers of uneven roads and potholes on their track. We study the various ways that the system's goals can be achieved.
The methods we have chosen in these projects are justified. And then, we give details of how the various subsystems work. You can give the problem statement as follows. Access points responsible for storing potholes information in the vicinity, taking feedback from vehicles, updating the information in the depot, and transmitting information to other vehicles. The entire scenario works like this. During the deployment of the access point, we use some initial potholes data. Then it continues to broadcast the data. Finally, the customer-equipped vehicle captures that data.
Nevertheless, due to environment or fatigue, new potholes can always be formed. This means that the customer device also acts as a sensor and finds the newly formed potholes on the road. If it discovers any new potholes, it gives the feedback data of the new Access Pothole. Access points updates this information to its data storage and adds it to the data transmitted [7].
In most developing countries, poorly maintained roads are a fact of life. Therefore, a well-managed road network is a must for every country's well-being and development. A practical road surface monitoring system must therefore be created. Automated detection of potholes is our system focus. The aim is to develop an image processing system for detecting road potholes uploaded to the server, notify all users, and update the information when required [8].
The significant challenges in the road transport sector are potholes, not good quality, and low funding for road maintenance. The handling capacity of road vehicles is also low, but the vehicle density is increasing on the road. As a result, roads deteriorate, and transport costs increase. This resulted in road accidents. The other reason for jams and accidents is the poor road conditions [9].
The main problem facing developing countries is road maintenance. Due to poor maintenance and maintenance of the roads, potholes have been created. According to an automatic survey, potholes are one of the main reasons for road accidents. Moreover, when the vehicle's speed slows down, there are high chances of a collision with the vehicle. We, therefore, believe that sharing information is an essential element to prevent potholes and reduce accidents [10].
Potholes can damage different parts of the vehicle. Tires are affected by the pothole strikes and are susceptible to damages such as bulging sidewalls, tread splits, and flats. A pothole's harsh, sharp edges push the tire beyond its limit, splits, stretches, or shrinks the rubber. Even if you do not have a flat tire after hitting a pothole, you should check the wheel for damage to the safety or other problems. Hit a sufficient pothole, and the force can buckle and perhaps crack the wheel, leading to a range

of costly problems to solve. Sudden jolts on the road can damage the suspension, knock it out and cause poor steering performance. The tire's walls may contact the road if you drive into a deep pothole, causing severe exhaust damage [11].
Although it is possible to get over small pieces of debris without any consequences quickly, large sumps and holes can cause severe damage to your car and cause you to lose control [12].

1.3 Research Objectives

Our aim in this research paper is to build a Pothole Detection System using image processing. However, this system for pothole detection is designed to collect road images with an optical instrument modified in the vehicle and detect a pothole using the algorithm from the date it collects.

1. Build an image dataset to train our algorithm for detecting potholes so that the detecting instrument can collect data from the road and give a result instantly.

2. Build a system for detecting potholes in real-time from real-time videos or feeds.

3. To avoid accidents on the road by detecting potholes.

1.4 Research Orientation

1. We collected all our datasets and videos from different online sources since we could not go out due to Covid-19.

2. We have used many pre-trained layers in the trained model for transfer learning implementation, and we have manipulated the model for better accuracy.

3. We implemented two methods which are Inception-v3 and MobileNet to detect potholes in images. After implementing MobileNet and Inception-v3 we get the model accuracy of MobileNet is 95% and 93% Inception-v3.

4. We applied YOLOv5 to detect potholes in real time video feed to avoid road accidents.

Chapter 2

Literature Review

There are many reasons in the world for a road accident. Such as Distracted driving, over speeding, Inexperience, using a phone during driving, Overtaking tendency. However, bad road condition is also a notable cause of an accident. Lack of maintenance, heavy vehicle, flood, or rain can damage the road. Texture changes, crack, and the pothole on road surface causes road damage.
To avoid such accidents, a technological approach can be used to detect potholes. That process basically indicates pothole and notify the driver before the vehicle approach towards that. Various methods are there to show potholes. Among these are vibration-based methods, 3D laser scanner methods, 2D image-based approaches, image-based integrated processing, and GPR datasets for mechanized pothole detection.
Camera observation and vibration detection are the most common technologies used by mobile devices to monitor road conditions. Vision-based systems utilize smart phones installed in a vehicle to collect shots of the road surface and automatically analyze the surface data included in the images using picture analysis techniques. For instance, the Road surface Distress Detection Method [13] detects road damage in film footage via an automated data processing procedure. However, in this scenario, the system is just capable of deciding the position and estimated amount of suspected crack or distress characteristics inside video image frames. Additionally, it is unable of providing structural dimensions (crack width/depth/height) or classifying the damage or distress features. Maeda et al. [14] and Ochoa-Ruiz et al. [15] built extensive image data sets which are used for the precise detection of potholes in roads by using deep-learning methods. But there is no publicly available standard road damage dataset.
Hence there is no standard for detecting road damage. Although data augmentation is suitable for training generic object detectors, its performance can be limited by the poor image quality. The most often utilized sensors in vibration systems are accelerometers and gyroscopes, thus they are susceptible to vibrations induced by road irregularities, such as potholes and dips. Numerous strategies, classified into three groups, have been applied: (1) target level based methods, (2) intelligent timeshare , and (3) machine learning, in conjunction with feature-based engineering. Requiring proper road network is challenging because of numerous reasons, including inclement weather, unexpected traffic volumes, and unpredictable wear and tear. Due to the unpredictable nature of[4] [16] [17]. Although modern digital inspection trucks record road pictures and video data, professionals manually analyze damage

estimations. As a result, it is a lengthy and costly process. It is frequently more subjective than objective because it is dependent on the precision and experience of the workers.

Numerous users have developed a successful post-processing approach for utilizing clustering data, significantly increasing the percentage of analysis and identification and decreasing the rate of wrongful convictions [18].

An anomaly is found if its amplitude or other signal characteristics (like root mean square (RMS) and crest factor) are above a certain level. Nericell [19] proposed a system for identifying potholes and bumps using two vehicle speed detectors. Artificial readjustment is performed in order to bring the misplaced rangefinder up to the same standard with the vehicle coordination system for street damage detection. At high speeds, a spike that exceeds the specified limit (or the detection of a change point) is labeled as a potential irregularity. At the same time, the detector searches for an ongoing dip of az produced by the tires entering the pothole at a low speed. Over multiple observations, the sensor maintained a low error rate (5-10%) but a significant falsified rate (20-30%). Mednis et al. [20] suggested that while the car was temporarily free-falling during entry or leave, all multiple velocity profile were near zero-g. They compare the G-ZERO algorithm to three minimum level approaches for detecting potholes: Z-THRESH, Z-DIFF, and STDEW (Z). TERM is a road monitoring system that employs dynamic thresholds and crowdsourcing techniques [21]. This algorithm detects a potential bump when the X- (lateral) and Z-axes (vertical) acceleration values are more than the set limit, and a speed bump when the Y- (longitudinal) and Z-axes acceleration values are greater than the stated limit. If more than five data samples indicate the same location as a prospective abnormality (bump in the road or potholes), the location is tagged on the map as a true irregularity. Even though the methods are simple to detect, the results are encouraging, with 90 percent of bumps and 85 percent of potholes detected successfully, owing largely to the use of crowdsourced data. Despite the excellent results, the false alarm rate of the method has not been disclosed.

DTW is a method approach for determining the correlation between different datasets that may be period and space-dependent [19]. But in this case, due to the limited availability of the stated algorithms and data sets, it isn't easy to compare the accuracy and performance of alternative techniques when it comes to the performance of the linked algorithms. Singh et al. [22] said, DTW was utilized to detect and classify road irregularities based on sensor data. Reference templates for potholes and bumps were created manually using sensor readings and kept in a template database. The detection was accomplished by determining the degree of similarity between the input data and the reference templates. Potholes and bumps were detected at an 88.7 percent and 88.9 percent rate, respectively.

Road inspection techniques based on machine learning have been widely implemented and installed in cars traversing long distances across the city of Boston. Pothole Patrol [20] used portable methods to gather information (accelerometer, GPS). To eliminate one or more non-pothole event types, a variety of filters (speed, high-pass, z-peak, xz, and speed vs. z-ratio) were used to its pothole detecting algorithm. Each filter's specific characteristics are used to optimize the system's detection accuracy, which itself is compatible with the fundamental concept of machine learning. By clustering the findings by location, the algorithm attained a final pothole detection accuracy of 92.4 percent in the labeling data. In comparison to

this Pothole Patrol (p2) system, the majority of machine-learning approaches for road inspection typically employ the following processing steps. To begin, functions from diverse aspects are retrieved from data set using a variety of different ways. The classification of these features enables the identification of road problems and the differentiation of various types of defects [21]. But due to both technical restrictions and human timing constraints, a human operator activating the system during the test drive is prone to many errors. Perttunen et al. [23] removed time and frequency domain features with the fast Fourier transformation (FFT) and used selection algorithm for selecting optimum set of features. Because speed characteristics have an impact, Perttunen et al. fitted a line to each of the data features to remove linear speed dependency. Seraj et al. [24] did investigate time and frequency domains al well as took account of characteristics resulting from wavelet transformation through using the stationary wavelet transformation (DWT). The support vector machine (SVM) was used as the classifying model for Perttunen et al. [23] and Seraj et al. [24]. But, for a smooth road, the detector must have a low percentage of false negatives. Because most roads are smooth, even a tiny portion of smooth roads being flagged as road abnormalities will result in an unreasonable amount of false positives. The analysis and assessment of different classifications were carried out by Silva et al. [25], Varona et al. [26], and Basavaraju et al. [27]. Using the concepts from the field of computer vision, Varona et al. extended and reduced the sequence of various signals to add to the data sets. Basavaraju et al [27] analyzed the distinction between the use of characteristics on all y-axis axis and the use of data on a particular axis. Li et al. [28] were already using the Wavelet process to detect potholes, and they provided a new method for calculating the diameter of potholes.

In comparison to limiting methods and the DWT, machine learning method is a more complex and comprehensive method that may extract more meaningful information from the data. While machine learning techniques have developed and generate excellent outcomes, this strategy is still based on thresholds; in other words, the process of determining appropriate benchmarks is employed to transmit knowledge from humans to machines. However, studies have repeatedly neglected the fact that the majority of original data is generated by flatter stretches, which may be removed using simple thresholds. Additionally, the majority of research utilized machine-learning classification methods directly, without the need of any filters. The energy and data consumption increase – as the client uploaded redundant data – and the server load for the classification of machine learning increased. Another drawback is that the pavement condition was not considered while identifying potholes. A model is trained on a dataset of a certain type of road (such as a road) cannot be used to another type of road, as potholes vary in frequency, shape, and dimension.

Chapter 3

Work Plan

The proposed system utilizes an optical instrument to capture road images. To do so, the model needs the creation of a process that takes image data in an optical device as an input, processes input data in a systematic manner, and generates the desired result. The model (Figure-3.1) of detecting potholes is created with

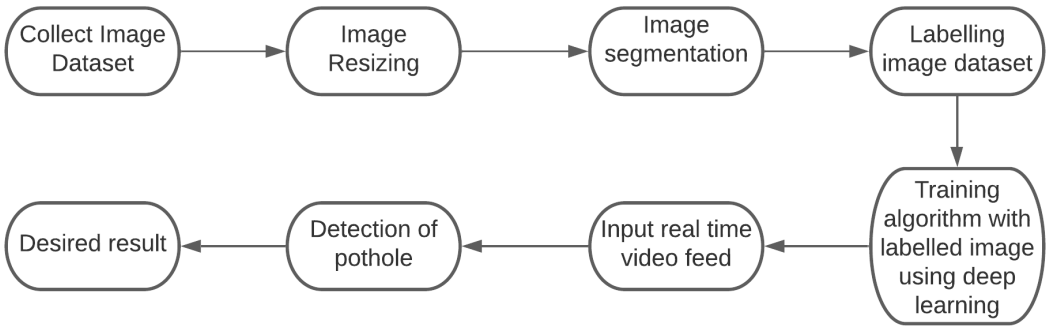


Figure 3.1: Flowchart for detection of pothole.

the help of a dataset that is applied in an algorithmic way with image processing techniques. Image labeling is the process of recognizing and highlighting various aspects of an image. In image labeling, when creating metadata, it's helpful to automate the process. A known deep learning method will be used to make the algorithm understand how to recognize potholes. To label the images, first upload all of the raw images to the system. After that, image labeling software is installed to annotate such photos using specific techniques according to the customized criteria. Video data is a common asset that is used every day, whether it is a live stream in security camera or dash-cam, live video broadcasts are quite simple to access, and we only have to merge some efficient amount of code for live video access with our pothole detection.

Chapter 4

Pothole Detection using MobileNet & Inception-v3

4.1 Methodology

The purpose of the proposed pothole detection model (Figure-4.) is to identify pothole using Inception-v3 and MobileNet. In order to do so, the model requires planning a process that takes information from image data set as an input, efficiently process input data, and deliver predictions of two folds; "pothole" or "plain road". The purpose of this study was to determine the occurrence of potholes. As a result, it was an issue of binary classification.

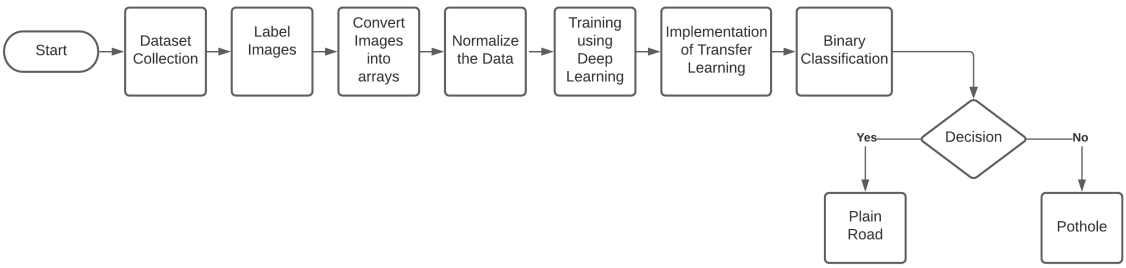


Figure 4.1: flowchart for methodology

The pothole detection process method consists of three major steps-

1. Image pre-processing: The study began with the collecting of data and labeling of pothole and non-pothole incidences. This period is concerned with organizing the input data which is the whole dataset of images so that the method (Pothole detection) can easily process it.

2. Training Images: This period is concerned with converting the image dataset into arrays, normalizing the dataset, training using deep learning methods (Inception-v3, MobileNet), implementing of transfer learning and binary classification to provide prediction

3. Testing: this stage is concerned with taking random images to test to see if there is any pothole in that particular image.

After normalizing the data, we have split the 100% dataset into three parts that are one part (70%) is for training, one part (15%) for testing and other part (15%) for validation.

4.2 Model Description

4.2.1 MobileNet

Convolutional neural networks are constructed utilizing depthwise separable convolutions with MobileNet's simplified architecture. It is a compact framework that is well-suited for mobile and embedded vision applications.
As seen in Figure 4.2, MobileNet's structure is based on depth-separable filters [29].

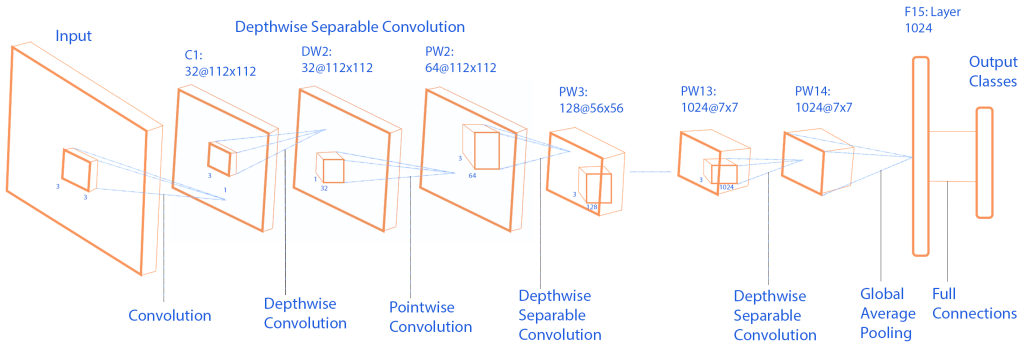


Figure 4.2: Architecture of MobileNet

1. **Depthwise Seperable Convolution:** Using depthwise separable convolutions, a type of factorized convolution, the MobileNet model may be broken down into depthwise and invertible convolutions (1 x 1), the latter of which is a sort of pointwise convolution. Depthwise convolution is used in MobileNet, each input channel has a single filter applied. Following that, the depth-wise convolution outputs are combined using an 1 x 1 convolution in the pointwise convolution. Standard convolution classifies and combines data in a single step to produce new outcomes. The filtering layer and the combining layer are separated by depthwise separable convolution. The duration and size of the model are greatly reduced as a result of computation. Conventional convolution's factoring is shown in Figure-4.3, 4(a) into a depthwise convolution 4(b) and a 1×1 pointwise convolution 4(c). A conventional convolutional layer accepts as input a $D_F \times D_F \times M$. F feature map and generates a $D_F \times D_F \times N$ where G is a map of attributes D_F denotes the geographical length and width of a square input feature map, M denotes the input's channel count (input depth), DG denotes the spatial width and height of a square output feature map, and N denotes the number of output channels (output depth).
The fundamental convolutional layer is specified by the convolution kernel's size, K. $D_k \times D_k \times M \times N$ where D_k is the geometric dimension of the kernel, which is presumed to be square, and M and N are the previously given unit

values for the input and output, accordingly. The following is the extracted feature pattern for standard convolution, considering phase one and buffering:

$$G_{\mathrm{k},\,\mathrm{l},\,\mathrm{n}} = \sum K_i, j, m, n.F_k + i - 1, l + j - 1, m \tag{4.1}$$

The computing cost of standard convolutions is as follows:

$$D_{\mathrm{k}}.D_{\mathrm{k}}.M.N.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.2}$$

where the processing cost is cumulative in nature and related to the number of input, the number of output , the kernel size D_k x D_k, and the feature map size D_F x D_F. MobileNet models handle each of these topics and their interactions. To begin, it uses depth - wise separable convolution operation to decouple the number of output from the kernel size. The standard convolution process filters and merges data using convolutional kernels. By adopting depth-wise factorized convolutions, the screening and pairing stages can be divided into two processes.

Distinguishable convolutions for huge cost savings. Distinct and separate depthwise convolutions are built of two layers: depthwise and pointwise convolutions. For each input channel, we use depth-wise convolutions to apply a single filter (input depth). Convolution in the order of the points, a standard 1×1 combination, which will then be applied to build a linear model of the depth - wise separable layer's output. Non-linearities in the batch norm and ReLU are used in MobileNet at both layers:

$$\hat{G}_{\mathrm{k},\,\mathrm{l},\,\mathrm{m}} = \sum \hat{K}_i, j, m.F_k + i - 1, l + j - 1, m \tag{4.3}$$

Where $\hat{\mathrm{K}}$ is the size of the depthwise convolutional kernel $D_k \times D_k \times M$ where the m_{th} filter in $\hat{\mathrm{K}}$ is implemented to the m_{th} channel in F to generate the m_{th} channel of the final feature map's processed output $\hat{\mathrm{K}}$. Convolution by depth incurs a computational cost of:

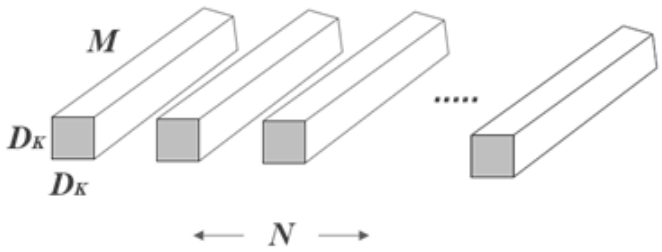
$$D_{\mathrm{k}}.D_{\mathrm{k}}.M.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.4}$$

Convolution by depth is more efficient than convolution by width. Rather than mixing or filtering the input channels, it introduces completely new characteristics. To implement these new characteristics, an additional function is required that computes a linear combination of the depth-wise result of eleven convolutions. Convolution that is depthwise separable is a concept that refers to the result of merging depthwise and 11 (pointwise) convolution. This concept was introduced for the first time in the text of [30]. Cost of depth wise separable convolutions:

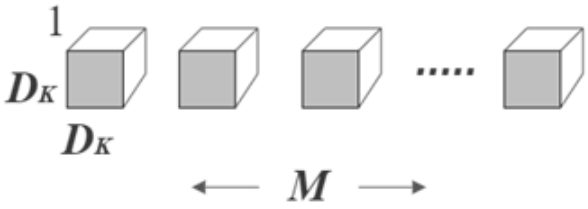
$$D_{\mathrm{k}}.D_{\mathrm{k}}.M.D_{\mathrm{F}}.D_{\mathrm{F}} + M.N.D_{\mathrm{F}}.D_{\mathrm{F}} \tag{4.5}$$

This is the sum of the 1×1 pointwise and depthwise convolutions. By describing convolution as a two-step filtering and combining operation, we obtain a computation reduction of:

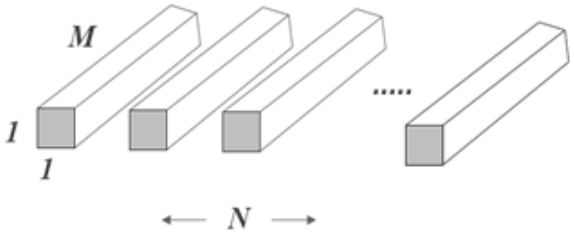
MobileNet employs 3×3 When compared to conventional convolutions, depth-wise separable convolutions demand as low as 8 to 9 times as much computation as conventional convolutions. For example, the addition of spatial factorization. [31] [30], saves little additional time because depthwise convolutions consume relatively little compute.



(a) Standard Convolution Filters



(b) Depth wise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context Of depth wise Separable Convolution

Figure 4.3: The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter [32].

4.2.2 Inception-v3

Inception-v3 is a more advanced variant of the inception-v1 and inception-v2 architectures developed by GoogleNet. It was designed to be more efficient than prior incarnations of the architecture. It was named the first runner-up for picture classification in the 2015 ILSVRS (ImageNet Large Scale Visual Recognition Competition) due to its 42-layer architecture's decreased error rate [33]. Inception-v3 is pre-trained with over a million images from the ImageNet collection. As a result, it is capable of classifying photos into 1,000 object categories, such as pencils, vehicles, persons, and animals [34]. As a result, this architecture has developed the ability to recognize a wide variety of objects in images. The structure of the Inception-v3 network is constructed step-by-step. A brief discussion of this is given below:

1. **Factorized Convolutions:** In this case, the convolution size is quite small in comparison to older systems. And this contributes to the model's faster

training.

2. **Smaller Sized Convolutions:** In this case, the convolution size is quite small in comparison to older systems. And this contributes to the model's faster training. Figure-4.4 below shows that a compact-network is used in place of the 5 x 5 convolutions.

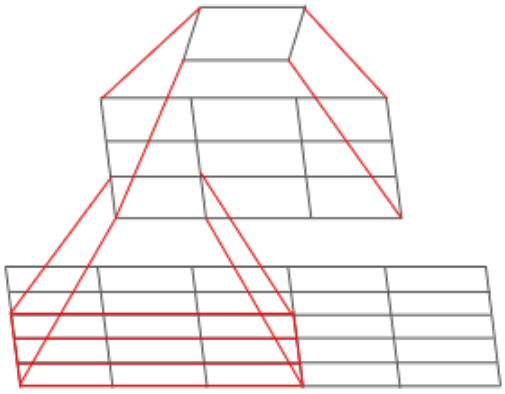


Figure 4.4: compact-network replacing the 5 × 5 convolutions [30].

3. **Asymmetric Convolution:** Firstly, a 3 x 3 convolution could be substituted for a 1 x 3 convolution, followed by a 3 x 1 convolution. If we substitute a 2 x 2 convolution for a 3 x 3 convolution, the number of convolutions rises somewhat [35]. Figure-4.5 below, it is specified as the main Inception CNN model [36]. Also Figure-4.6 showing the Inception modules with enlarged outputs from the filter bank [30].

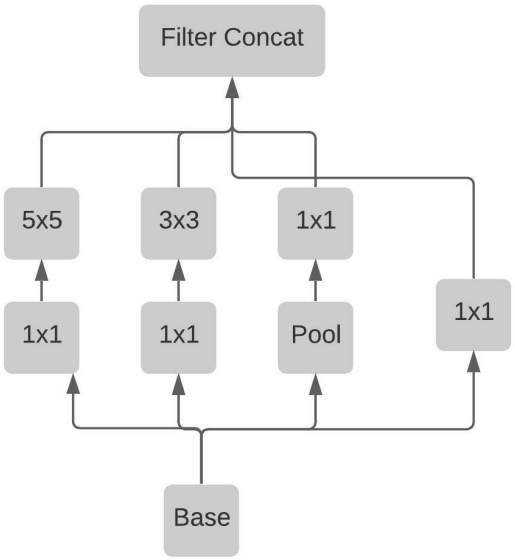


Figure 4.5: Main Inception CNN model showed in [36].

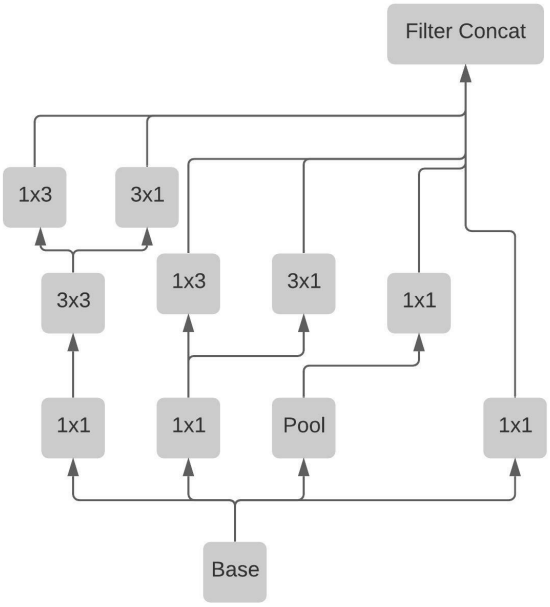


Figure 4.6: Components of inception with increased filter bank outputs [30].

4. **Auxiliary Classifier:** Auxiliary classifiers are miniature CNNs that are added across segments while the model is being trained. Typically, aside from the primary network damage, there is also an accidental loss. While GoogleNet made use of auxiliary classifiers to create a denser network, Inception-v3 makes use of them as a regularizer [35]. Figure-4.7 below shows the auxiliary classifier atop the previous 17×17 layer.

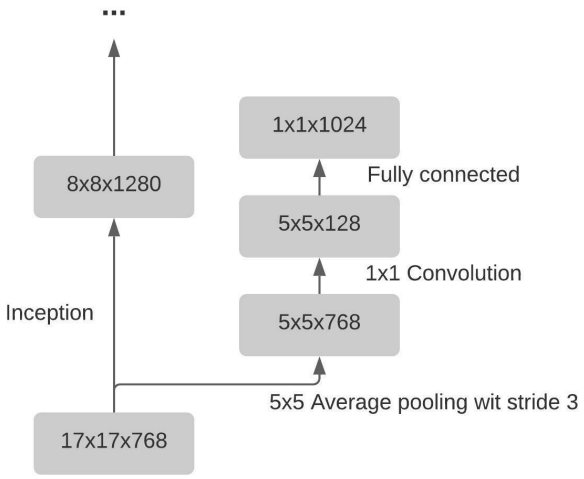


Figure 4.7: Auxiliary classifier atop the previous17×17 layer [30].

5. **Grid Size Reduction:** Pooling activities are frequently employed to lower the size of the grid. However, in order to overcome the constraints imposed by computational cost, a more efficient structure is recommended in Figure-4.8:

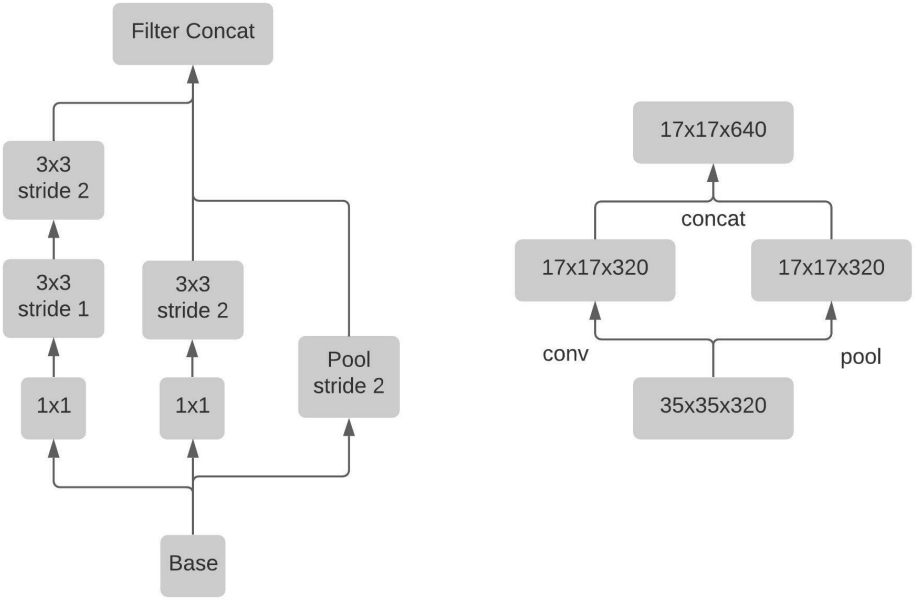


Figure 4.8: While reducing grid size, this Inception module also expands filter banks [30].

All these above concepts are combined to create the final architecture.

4.3 Data Preliminary Analysis

4.3.1 Collecting Dataset

This is a collection of 680 images of roads that have been labeled with potholes. The dataset is available in a variety of formats for use with a variety of popular deep learning models. This dataset will be used for instructional purposes and research.

4.3.2 Labeling Image

The process of image labeling entails identifying and marking various details within an image. This process can detect details in images automatically by utilizing on-device and cloud-based technology.
In computer vision, the most often used type of image labeling is bounding boxes. Bounding boxes are rectangle boxes used to specify an object's location. The x and y axis coordinates in the top-left corner of the rectangle and the x and y axis coordinates in the bottom-right corner can be used to determine them. When it comes to locating and detecting objects, the use of bounding boxes is widespread.
Bounding boxes are commonly expressed by two coordinates (x1, y1) and (x2, y2), or by a single vector (x1, y1) and the width (w) and height (h) of the bounding box.

4.3.3 Converting Images into Arrays

NumPy is Python's most popular and commonly used package for image data processing. NumPy uses the as array() method to transform PIL images to NumPy

arrays. Additionally, the np.array function yields the identical output. The class of an image is returned by the type function. The Image from array() function can be used to reverse the process. Numpy.ndarray image data can be saved as a PNG or JPEG file later using this function.
Print(data) returns the pixel value from the NumPy array image. We've Utilized Matplotlib, OpenCV, and the Keras API to load and display an image. The retrieved images were then transformed to and from the NumPy array and NumPy libraries were utilized to perform basic image manipulation and save it to our local system. In the Keras API and OpenCV, images are read as arrays.

4.3.4 Normalizing the data

We reshape and resize the images of the whole dataset into one particular size for our ease of usage.

4.3.5 Training using deep learning (MobileNet, Inception-V3)

We imported necessary DL libraries and downloaded pre-trained weights and printed out the model of MobileNet Inception V3. First, we used MobileNet then we went for Inception-v3.

4.3.6 Implementation of transfer learning

For implementing transfer learning, we have taken base input and output. We have many pre-trained layers in the trained model and for base input we have taken the very first layer from the pre-trained model and for base output we have taken the fourth last layer from the pre-trained model. We have manipulated the model for better accuracy. The model we called actually maximum layers of them are pre-trained. So, we don't want to train them again as that would take much time. The main benefit of transfer learning is that it decreases the training time.

4.3.7 Binary classification (pothole/normal)

The binary classification indicates whether or not there is a pothole on the road. To begin, we'll create a library. We added some enhancements where image numbers are multiplied by this method, which results in multiple forms for a single image and also creates new memory for load images. After that, we implement batch size, loading 20 images at a time. We have included a link to our training data set. We resize the target size to produce a higher-quality output. In conclusion, we use Epochs because they consistently produce valid results. We made use of Adam's optimizer. For the loss function, binary cross entropy is being used. These are the primary techniques for determining accuracy. Then we ran and obtained 30 distinct epochs. Each epoch produces unique results that demonstrate our accuracy and our valid accuracy.

4.3.8 Train and test

For training and testing, we used two models which were Inception-v3 and MobileNet. We split the data set into 3 parts, 70% of data is used for training, 15% for testing, and 15% for validation.

4.4 Result Analysis

Here in Figure-4.9, 4.10 Gives us the F1 score of MobileNet and Inception-v3 respectively:

Classification Report of mobilenet_v2

	precision	recall	f1-score	support
normal	1.00	0.97	0.99	36
potholes	0.97	1.00	0.99	34
accuracy			0.99	70
macro avg	0.99	0.99	0.99	70
weighted avg	0.99	0.99	0.99	70

Figure 4.9: F1 score of mobilenet.

Classification Report of inception_v3

	precision	recall	f1-score	support
normal	0.97	0.97	0.97	36
potholes	0.97	0.97	0.97	34
accuracy			0.97	70
macro avg	0.97	0.97	0.97	70
weighted avg	0.97	0.97	0.97	70

Figure 4.10: F1 score of inception-v3.

Here in Figure-4.11, 4.12 Gives us the confusion matrix of MobileNet and Inception-v3 respectively:

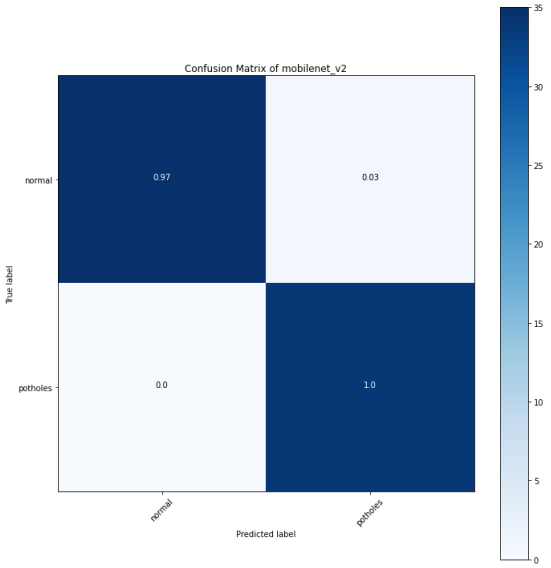


Figure 4.11: Confusion Matrix of MobileNet

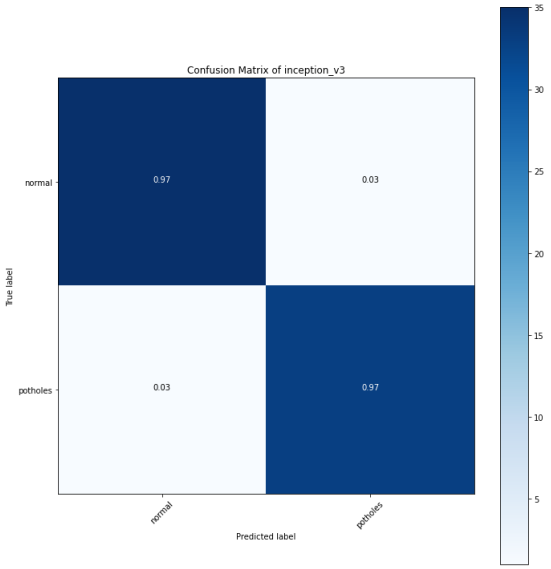


Figure 4.12: Confusion Matrix of Inception-v3.

After running both MobileNet and Inception-v3 on our data set we got results for model accuracy and loss. Here, Figure-4.13 and Figure-4.15 show the accuracy and the loss of the model respectively. And Figure-4.14 and Figure-4.16 is showing the accuracy and loss of the model for our data set respectively.

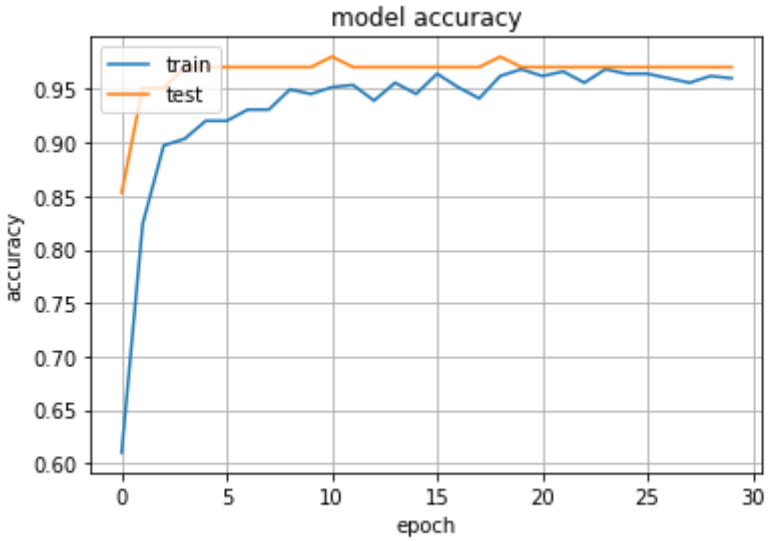


Figure 4.13: Accuracy graph of MobileNet model on the given data set.

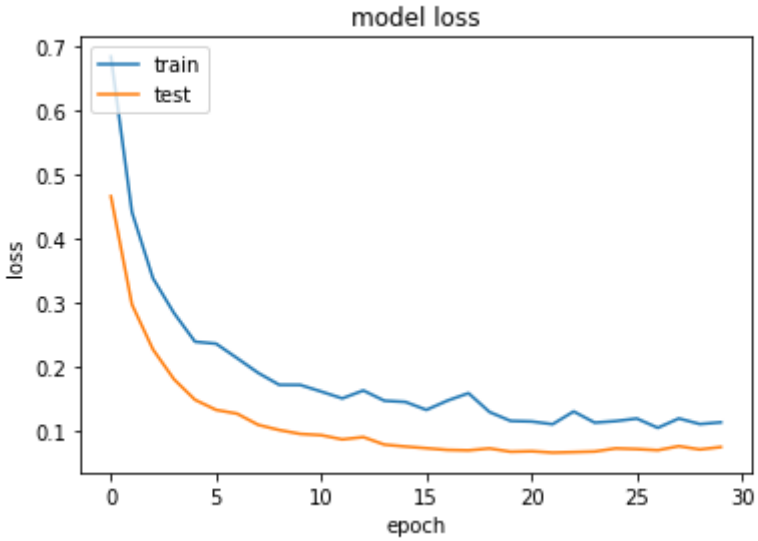


Figure 4.14: Loss graph of MobileNet model on the given data set.

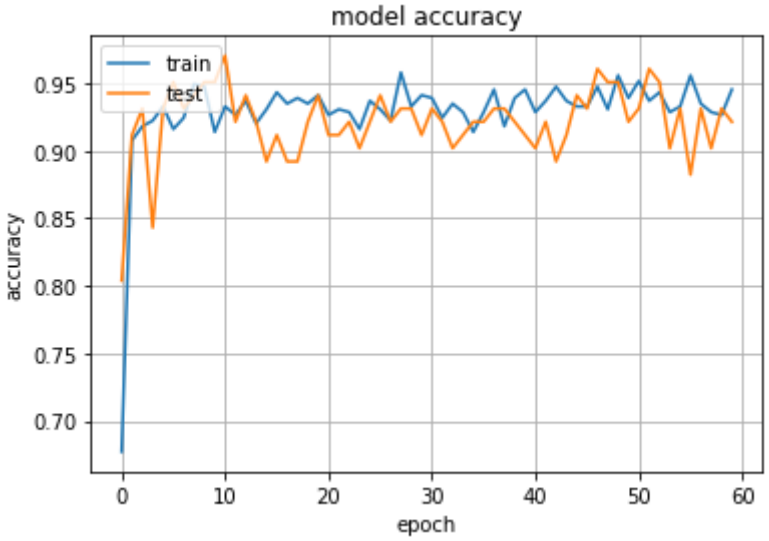


Figure 4.15: Accuracy Graph of Inception-v3 model on the given data set.

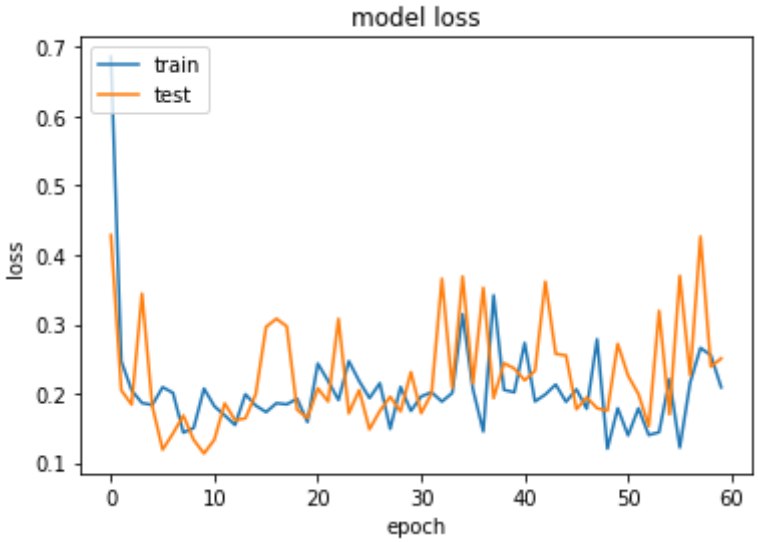


Figure 4.16: Loss Graph of Inception-v3 model on the given data set.

The output results of our model are given below:

Model	**Peak Accuracy**	**Loss**
MobileNet	95%	5%
Inception-v3	93%	7%

Table 4.1: Output Results of MobileNet and Inception-v3

Chapter 5

Detecting Potholes using YOLOv5

In this chapter we will describe the process through which we detected potholes from real-time video feeds via YOLOv5.

5.1 Methodology

The purpose of the proposed pothole detection model is to identify pothole using YOLO V5. To do so, the model requires planning a process that takes information from image data set as an input, efficiently process input data, and detects potholes. This study focused on identifying pothole event.

Figure 5.1: flowchart for methodology.

The pothole detection process method consists of three major steps-

1. The majority of the first stage of the project's development was spent gathering and classifying data on pothole and non-pothole occurrences. This period is responsible for organizing the input data, which is the whole dataset of photographs, so that the method (Pothole detection) can handle it fast and efficiently.

2. Transfer learning and binary classification to provide prediction are the primary concerns of this period, which also includes converting the image dataset into arrays, normalizing the dataset, training with deep learning methods (YOLO V5), implementing transfer learning and binary classification to provide prediction.

3. The testing phase is focused with taking a live feed in order to determine whether or not there is a pothole on the road.

5.2 Model Description

5.2.1 YOLOv5

The object detection method YOLO, which stands for "You Only Look Once," separates images into a grid structure. Each grid cell is in charge of detecting items within itself. Because of its speed and precision, YOLO is one of the most well-known object recognition algorithms. Glenn Jocher introduced YOLOv5 utilizing the Pytorch framework shortly after the release of YOLOv4. On GitHub, you can find the open-source code [37]. It performed exceptionally well on two official object detection datasets: Pascal VOC (visual object classes) and Microsoft COCO (generic components in context), thanks to continuous improvements. Figure 4 depicts the YOLOv5 network architecture. YOLOv5 was chosen as our initial learner for three reasons.
Firstly, YOLOv5 incorporated CSPNet into Darknet, forming CSPDarknet as its backbone. CSPNet incorporates gradient shifts into the feature space, hence decreasing the model's parameters and FLOPS (floating-point operations per second), while maintaining inference speed and accuracy.
Secondly, the YOLOv5 utilized PANet. PANet uses a novel FPN structure with enhanced path propagation. Adaptive feature pooling allows relevant information from each feature level to directly flow into the next subnetwork. PANet increases the utilization of trustworthy localization signals in lower levels.
Thirdly, YOLO has the capability to handle the detection of tiny, medium and large objects [38].
The YOLO network is comprised of three essential parts:

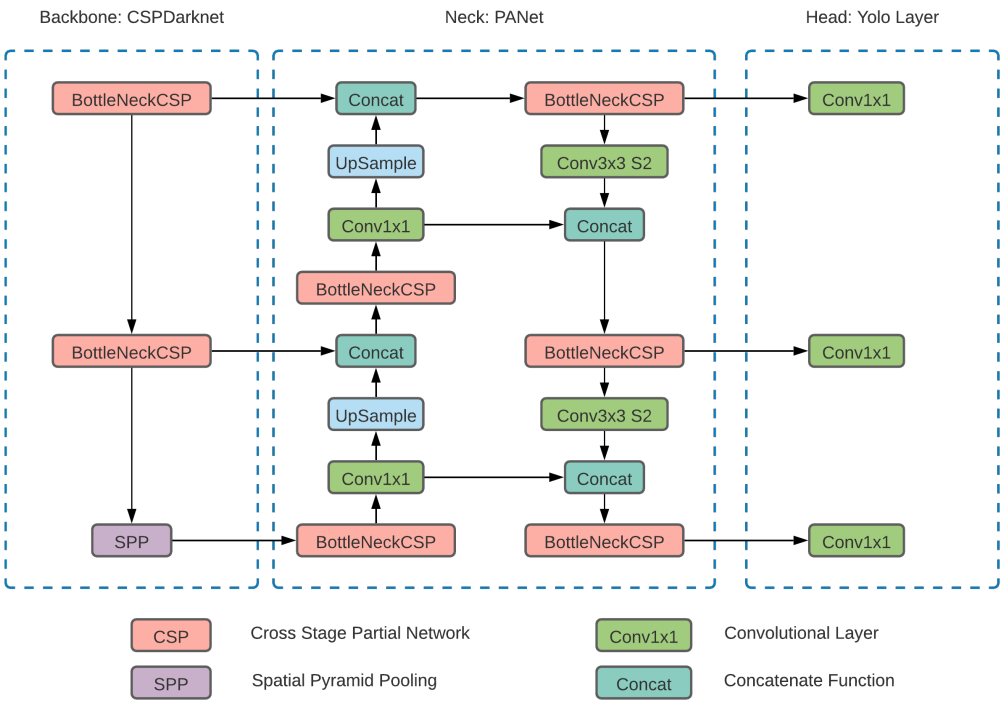


Figure 5.2: Parts of YOLOv5.

1. **Backbone:** A convolutional neural network capable of combining and producing visual features at multiple levels of resolution. Both YOLOv4 and YOLOv5 generate image characteristics using the CSP Bottleneck. The CSP eliminates redundant gradient concerns in other larger ConvNet backbones, resulting in fewer parameters and FLOPS for tasks of comparable priority. DenseNet forms the basis of the CSP models. Using DenseNet, CNN can alleviate the problem of vanishing gradients, improve feature expansion, stimulate network recurrence, and reduce the quantity of network weights [39].

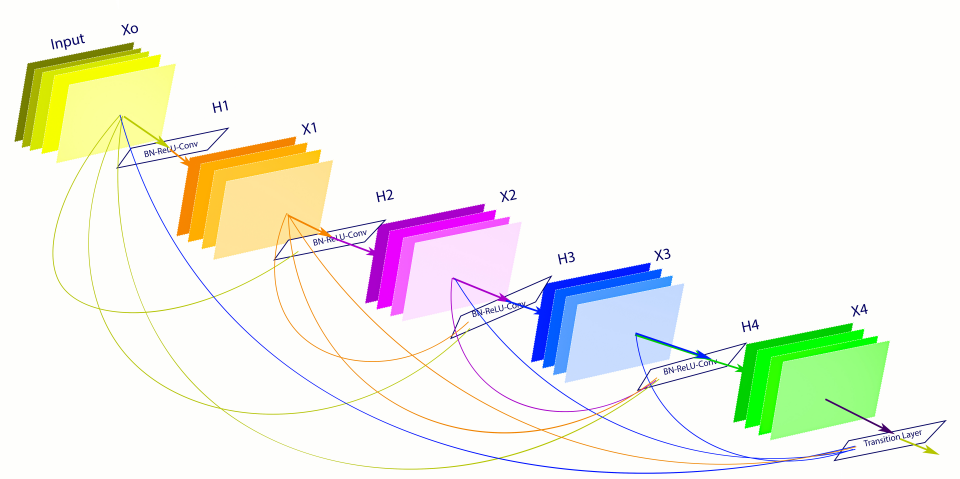


Figure 5.3: Growth rate of k = 4 for 5-layer thick block All previous feature maps are used as input for each layer [40].

This is how the DenseNet worked in CSPResNext50 and CSPDarknet53: It changed how it worked with the feature map of a base layer. The DenseNet was changed so that one copy of the map was sent through a dense block and another directly to the next step. The CSPResNext50 and CSPDarknet53 are meant to help DenseNet processing speed up and improve learning by giving users an unaltered version of the feature space [39].

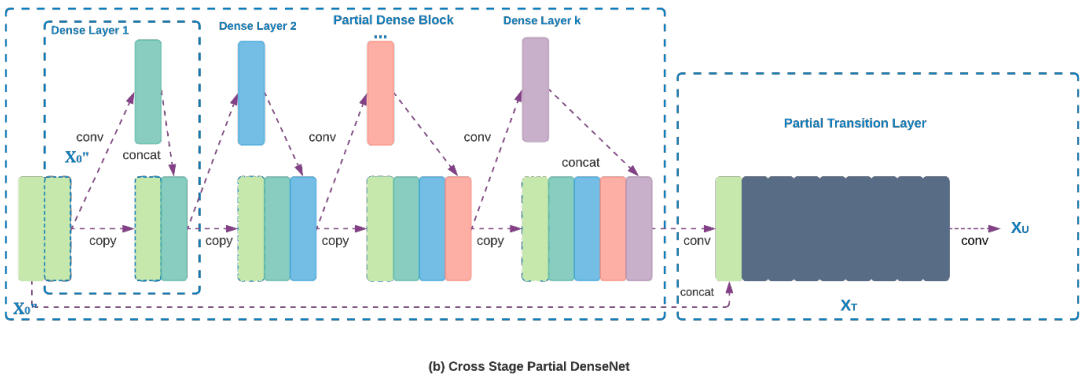


Figure 5.4: Cross Stage Partial DenseNet (CSPDenseNet)[41].

2. **Neck:** A collection of layers that combine and mix visual characteristics prior

to submitting them to estimation. YOLOv4 and YOLOv5 provide feature aggregation via the PA-NET neck.

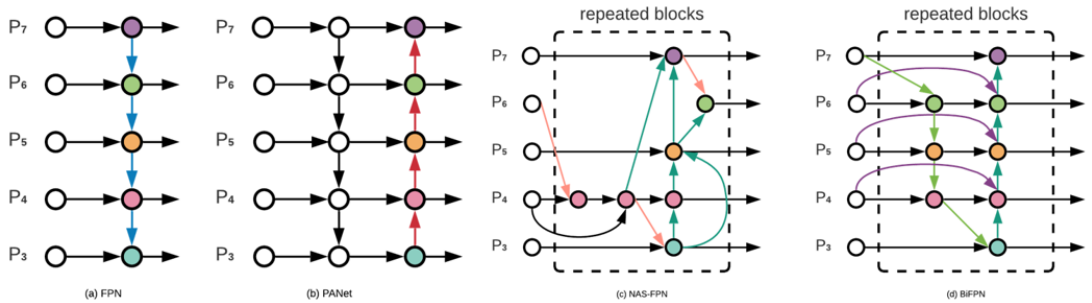


Figure 5.5: On top of the FPN, (b) PANet adds a bottom-up bottom-up pathway; (c) NAS-FPN applies the same block again to find an irregular feature network topology; Finally, (d) is our BiFPN with improved precision and performance trade-offs[42].

In the CSP backbone, each of the P_i entries represents a feature layer. A rendering of Google Brain's EfficientDet object identification architecture can be seen above. This may be an area of further exploration for YOLOv4 and YOLOv5 with different implementations here, as BiFPN was found to be the best option by the authors of EfficientDet [39].

3. **Head:** This function consumes neck features and performs box and class prediction operations [39].

5.3 Data Preliminary Analysis

5.3.1 Collecting Dataset

This is a set of 680 images of roads with potholes that have been labeled. The dataset is available in a number of forms and can be used with a number of different deep learning models. This dataset will be used for educational and research purposes.

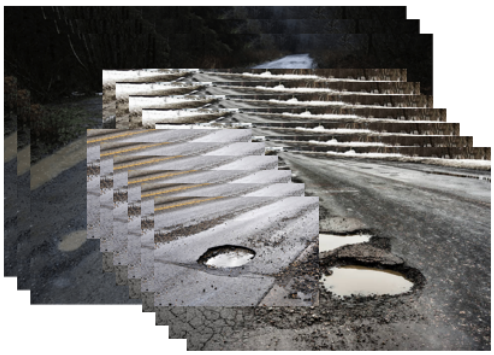


Figure 5.6: Pothole Dataset

5.3.2 Image Augmentation

The term "image augmentation" refers to the process of changing existing photographs in order to provide additional data for the model training process. Simply put, it is a technique for artificially expanding the dataset available for training a deep learning model.
This strategy was also used in our research to increase the amount of photos in our dataset. Initially, the collection had approximately 400 photos. The number increased to 680 after augmentation.

5.3.3 Labeling Image

Bounding boxes are the most widespread technique of image labeling in object recognition. Bounding boxes are rectangular boxes used to indicate the location of an object. Those coordinates can be found by looking at the top-left corner of the rectangle, as well as the bottom-right corner, using the x, and y axes, respectively. Bounding boxes are frequently used to locate and detect objects.

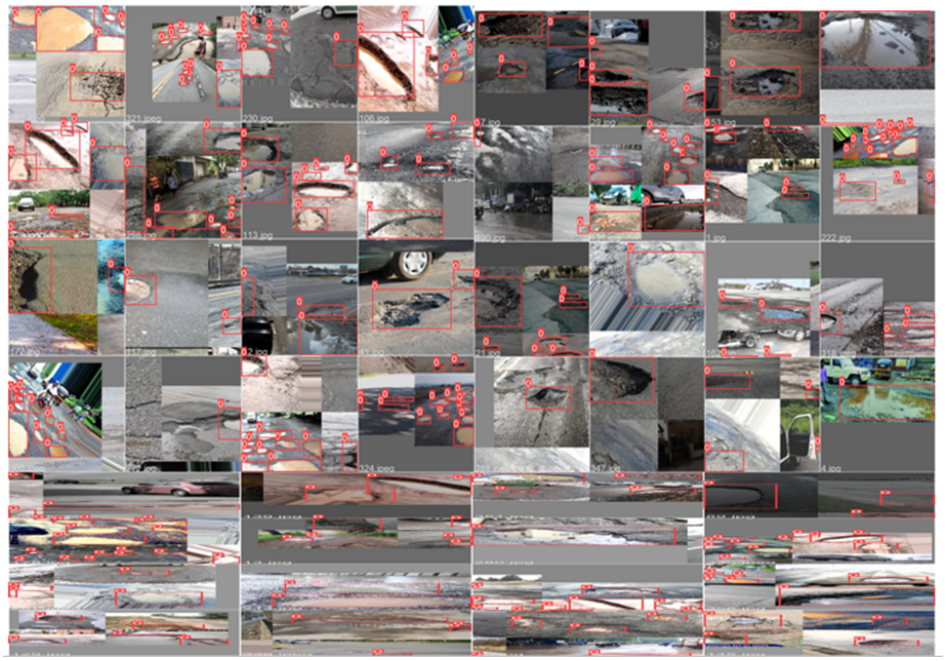


Figure 5.7: Labeling Pothole from dataset.

5.3.4 Converting Images into Arrays

Here we followed the same procedure as chapter-1 to convert the Images. In the figure below shows the graphical representations of instances in the data set.

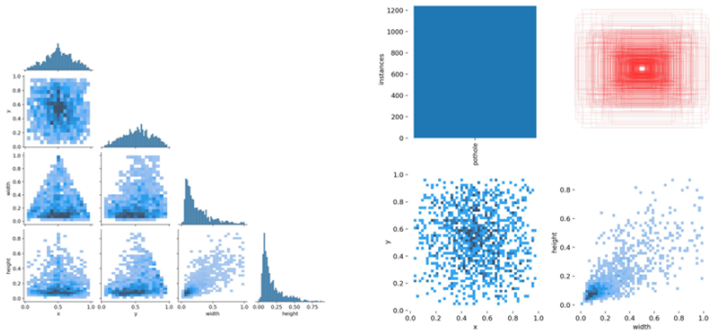


Figure 5.8: Visualization of instances in the data set.

5.3.5 Normalizing the Data

For convenience of use, we restructure and resize the entire dataset's images into a single size which is 640.

5.3.6 Training Using Deep learning (YOLOv5)

We imported pre-existing repository from github and installed requirements for YOLOv5.Next we imported utils library from YOLOv5 to initiate notebook. Then we used YOLOv5 model to train and test our dataset.

5.3.7 Implementation Details

We implemented YOLOv5 on Pytorch. We collected video containing potholes and upload it to the notebook. From data, we changed a file named "COCO128.yaml" which has pre trained models to detect, we deleted all the other pre trained model and added pothole class only which means it will only detect potholes. Then we inserted the folder's directory path which has training data to the COCO128.yaml file and renamed the file to "custom data.yaml". We set the image dimension to 640. The batch was 32 and the epoch was 300. The file we were taking information was from "custom data.yaml". Then we started the training on the video that was uploaded at the first place. We got a video as an output in which we could see the potholes being detected in the resulted video.

5.3.8 Train and Test

We used YOLOv5 for training and testing purposes. We divided the data set into two parts: 80% for training and 20% for testing.

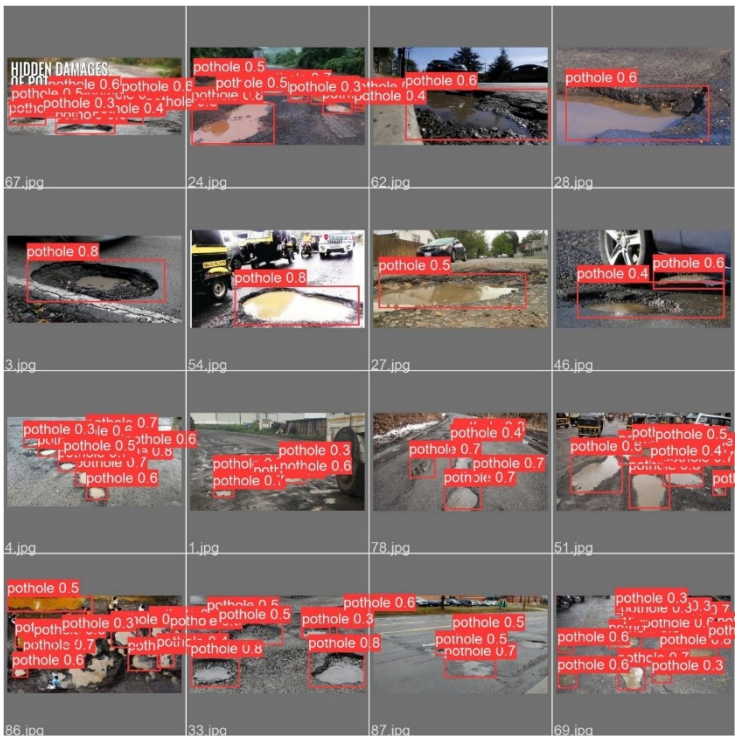


Figure 5.9: Trained dataset.

5.4 Result Analysis

Here figure-5.10 demonstrates the visual depiction of Precision, Recall, and mAP values of our dataset after running YOLOv5.

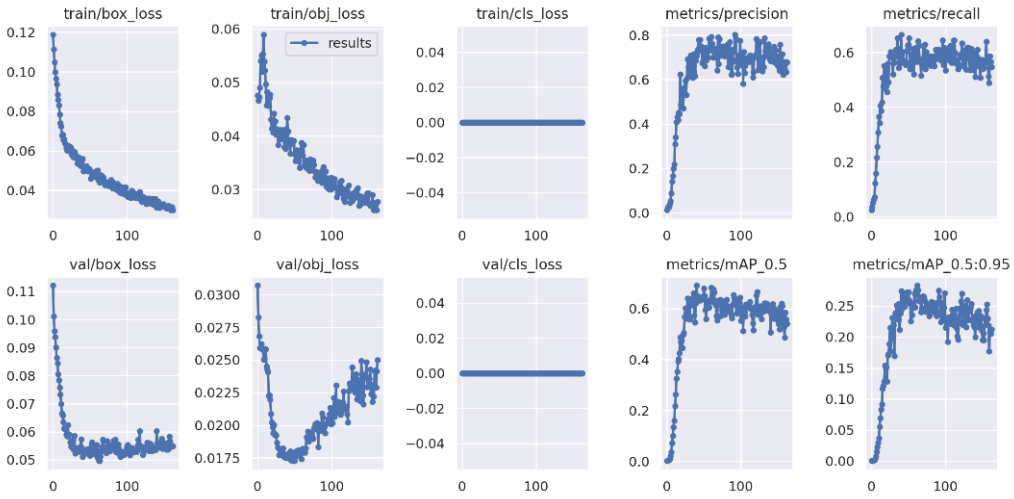


Figure 5.10: Trained dataset.

Below are figure-5.11, 5.12, 5.13, 5.14 which represents the F1 graph, the confusion matrix visual representation, precision and recall graph respectively.

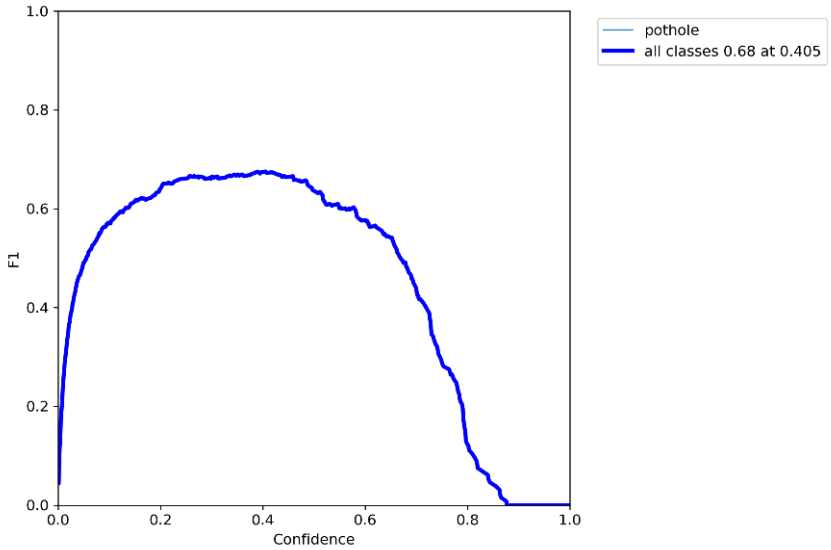


Figure 5.11: F1 Graph.

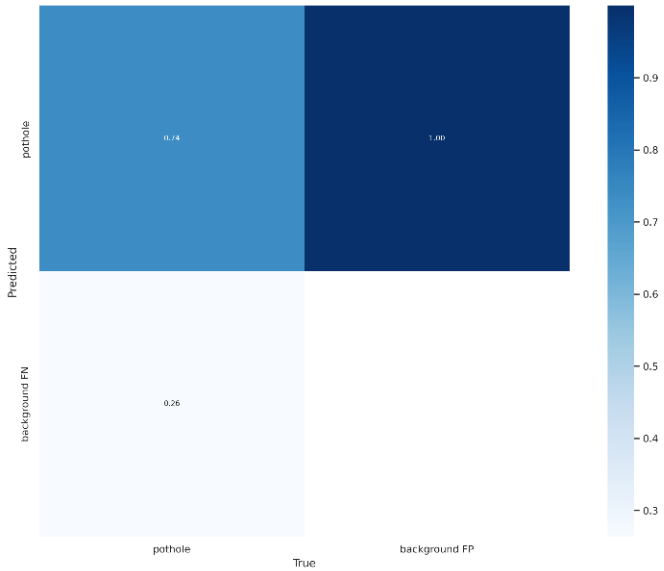


Figure 5.12: F1 Confusion Matrix.

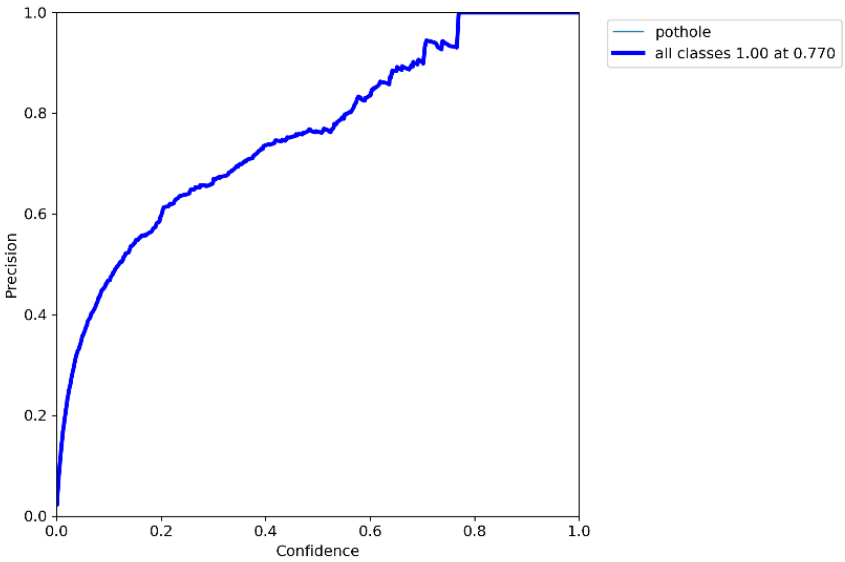


Figure 5.13: P Curve.

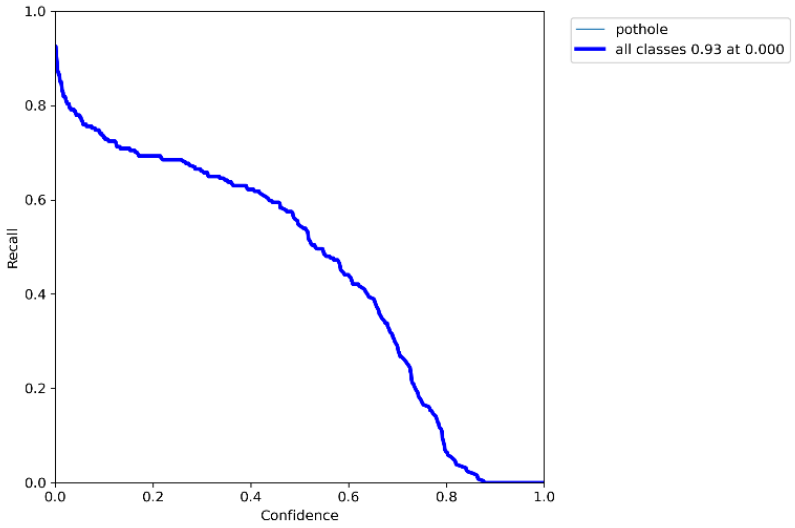


Figure 5.14: R Curve.

After this we used the YOLOv5 to detect potholes from live feeds. Below are some visual representations figure-32 of the results we found through the process.

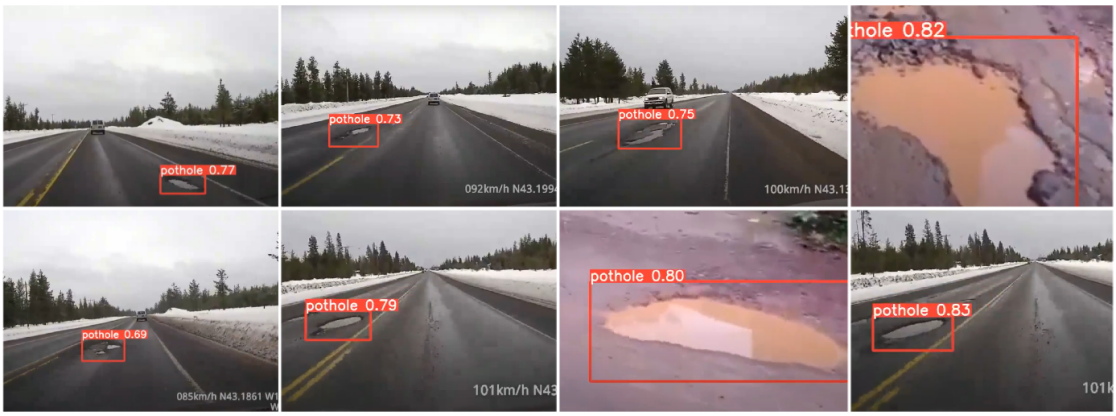


Figure 5.15: Output results from live feed.

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

Potholes will always be there as long as there will be roads. So creating new ways to avoid accidents and other problems created by those will always be in development. As mentioned in this paper, there are various methods to detect potholes from the road surface. But, this paper focuses on a deep learning-based real-time detection system which will find potholes from given input of video feeds. And to conduct this research we used 3 models which are Inception-V3, MobileNet and YOLOv5. In this system, the algorithm was trained with a large number of photos of potholes to increase the accuracy. When the full implementation is done, this system can be used by people, in general, to spot and avoid potholes on the road. Moreover, people related to road constructions, development, and maintenance will find potholes from the ever-growing length of roads in this country. They will also be able to repair them at a faster pace.

6.2 Future Scope

In our research paper we used available online resources to collect our data set because of Covid-19 situation. When everything will be normal again we will create our own data set by taking pictures of the roads of Bangladesh. On the other hand sometimes potholes can't be detected because of camera angle of that vehicle. In future we will try to overcome this issues and will make this proposed system much more precise.

Bibliography

[1] Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, A. Akula, *et al.*, "Convolutional neural networks based potholes detection using thermal imaging," *Journal of King Saud University-Computer and Information Sciences*, p. 1, 2019.

[2] *Definition of 'road transport'*, 2018. [Online]. Available: https://economictimes.indiatimes.com/definition/Road-transport..

[3] S.-K. Ryu, T. Kim, and Y.-R. Kim, "Image-based pothole detection system for its service and road management system," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–2, 2015.

[4] E. Buza, S. Omanovic, and A. Huseinovic, "Pothole detection with image processing and spectral clustering," in *Proceedings of the 2nd International Conference on Information Technology and Computer Networks*, vol. 810, 2013, p. 1.

[5] G. Jog, C. Koch, M. Golparvar-Fard, and I. Brilakis, "Pothole properties measurement through visual 2d recognition and 3d reconstruction," in *Computing in Civil Engineering (2012)*, 2012, pp. 553–560.

[6] *Transportation systems and its problems in bangladesh*, 2019. [Online]. Available: https://www.lawyersnjurists.com/article/transportation-systems-and-its-problems-in-bangladesh/..

[7] S. S. Rode, S. Vijay, P. Goyal, P. Kulkarni, and K. Arya, "Pothole detection and warning system: Infrastructure support and system design," in *2009 International Conference on electronic computer technology*, IEEE, 2009, pp. 1–2.

[8] P. Kadale, S. Barde, and A. Pawar, "Automatic detection of potholes and humps on road," *International Journal of Scientific Engineering and Research (IJSER) Volume*, vol. 6, p. 39, 2018.

[9] *The transport sector in india. world bank*, 2021. [Online]. Available: https://www.worldbank.org/en/country/india/brief/transport-sector-india..

[10] S. Gayathri, P. Menita, R. Mamatha, B. Manasa, and B. Sanjana, "Automatic pothole detection system," *Int J Eng Res Technol*, vol. 7, pp. 1–5, 2019.

[11] *Pothole problems: Do you know how to report one?* 2021. [Online]. Available: https://www.holtsauto.com/holts/news/pothole-problems-know-report-one/..

[12] *A pothole can damage your vehicle, your passengers, and you*, 2021. [Online]. Available: https://www.attorneystevelee.com/our-library/a-pothole-can-damage-your-vehicle-your-passengers-and-you/..

[13] W. Y. Yan and X.-X. Yuan, "A low-cost video-based pavement distress screening system for low-volume roads," *Journal of Intelligent Transportation Systems*, vol. 22, no. 5, pp. 376–389, 2018.

[14] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," *arXiv preprint arXiv:1801.09454*, pp. 1127–1141, 2018.

[15] G. Ochoa-Ruiz, A. A. Angulo-Murillo, A. Ochoa-Zezzatti, L. M. Aguilar-Lobo, J. A. Vega-Fernández, and S. Natraj, "An asphalt damage dataset and detection system based on retinanet for road conditions assessment," *Applied Sciences*, vol. 10, no. 11, p. 1, 2020.

[16] S. Vijay and K. Arya, "Low cost–fpga based system for pothole detection on indian roads," *M-Tech Thesis, Indian Institute of Technology Bombay*, 2006.

[17] M. Strutu, G. Stamatescu, and D. Popescu, "A mobile sensor network based road surface monitoring system," in *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*, IEEE, 2013, pp. 630–634.

[18] S. Sattar, S. Li, and M. Chapman, "Road surface monitoring using smartphone sensors: A review," *Sensors*, vol. 18, no. 11, p. 3845, 2018.

[19] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Using mobile smartphones for rich monitoring of road and traffic conditions," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 357–358.

[20] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," in *2011 International conference on distributed computing in sensor systems and workshops (DCOSS)*, IEEE, 2011, pp. 1–6.

[21] N. Sabir, A. A. Memon, and F. K. Shaikh, "Threshold based efficient road monitoring system using crowdsourcing approach," *Wireless Personal Communications*, vol. 106, no. 4, pp. 2407–2425, 2019.

[22] G. Singh, D. Bansal, S. Sofat, and N. Aggarwal, "Smart patrolling: An efficient road surface monitoring using smartphone sensors and crowdsourcing," *Pervasive and Mobile Computing*, vol. 40, pp. 71–88, 2017.

[23] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, 2008, pp. 29–39.

[24] F. Seraj, B. J. Van Der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Big data analytics in the social and ubiquitous context*, Springer, 2015, pp. 128–146.

[25] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues, "Anomaly detection in roads with a data mining approach," *Procedia computer science*, vol. 121, pp. 415–422, 2017.

[26] B. Varona, A. Monteserin, and A. Teyseyre, "A deep learning approach to automatic road surface monitoring and pothole detection," *Personal and Ubiquitous Computing*, vol. 24, no. 4, pp. 519–534, 2020.

[27] A. Basavaraju, J. Du, F. Zhou, and J. Ji, "A machine learning approach to road surface anomaly assessment using smartphone sensors," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2635–2647, 2019.

[28] X. Li, D. Huo, D. W. Goldberg, T. Chu, Z. Yin, and T. Hammond, "Embracing crowdsensing: An enhanced mobile sensing solution for road anomaly detection," *ISPRS International Journal of Geo-Information*, vol. 8, no. 9, p. 412, 2019.

[29] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-mobilenet models," *Mobile Information Systems*, vol. 2020, 2020.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[31] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," *arXiv preprint arXiv:1412.5474*, 2014.

[32] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," *arXiv preprint arXiv:1403.1687*, 2014.

[33] S.-H. Tsang, "Review: Inception-v3—1st runner up (image classification) in ilsvrc 2015," *linea]. Disponible en https://bit. ly/2MKWI5k*, 2018.

[34] *Inception-v3 convolutional neural network - matlab inceptionv3*, 2021. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/inceptionv3.html;jsessionid=6317c74f4ea682a26269b22524d7#mw_3bb6ba60-19b1-4bb0-b6d1-2810b754e2e4_sep_mw_6dc28e13-2f10-44a4-9632-9b8d43b376fe.

[35] V. Kurama, "A review of popular deep learning architectures: Resnet, inceptionv3, and squeezenet," *Consult. August*, vol. 30, 2020.

[36] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[37] G. Jocher, *Yolov5 documentation*, 2020. [Online]. Available: https://docs.ultralytics.com/#yolov5.

[38] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, 2021.

[39] J. Solawetz, *Yolov5 new version - improvements and evaluation*, 2020. [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/?fbclid=IwAR1die8PPTN0OeL9N-ITnnG9wG-fQAdnyfBR_bQ4CbbswENRuxsxKmE65rw.

[40] ——, *Yolov5 new version - improvements and evaluation*, 2020. [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/?fbclid=IwAR1die8PPTN0OeL9N-ITnnG9wG-fQAdnyfBR_bQ4CbbswENRuxsxKmE65rw.

[41] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[42] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.