# A Modern Technique to Detect Potholes by Computer Vision and Deep Learning

by

Muntasir Mahmud Saif
18201021
Tanvir Badsha
17101295
Mohammed Arman Khan
18201014
Sadman Sakib
18301164
Rafeed Bin Akbar
18301160

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

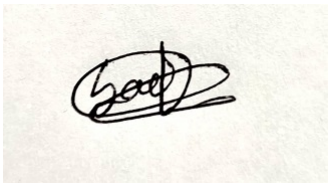
Department of Computer Science and Engineering
Brac University
September 2022

# Declaration

It is hereby declared that

1. The thesis submitted is 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:**

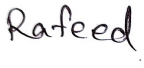


Muntasir Mahmud Saif
18201021

Mohammed Arman Khan
18201014

Tanvir Badsha
10129517

Sadman Sakib
18301164



Rafeed Bin Akbar
18301160

# Approval

The thesis titled "A Modern Technique to Detect Potholes by Computer Vision and Deep Learning" submitted by

1. Muntasir Mahmud Saif (18201021)
2. Tanvir Badsha (17101295)
3. Arnan Khan (18201014)
4. Sadman Sakib (18301164)
5. Rafeed Bin Akbar (18301160)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 24, 2022.

**Examining Committee:**

Supervisor:
(Member)

Dewan Ziaul Karim

Lecturer
Department of Computer Science and Engineering
BRAC University

Program 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 and Associate Professor
Department of Computer Science and Engineering
Brac University

# Ethics Statement

It is imperative that the thesis be written strictly in compliance with the university's rules and regulations, as well as ethical principles for doing research. Original data has been incorporated into our thesis. We double-checked our citations and references. Each of the paper's five co-authors accepts responsibility for any violations of the thesis rule. There were several Questionnaire Free tools, articles, and YouTube videos that helped us address problems. Furthermore, we would want to take this opportunity to express our gratitude to everyone who has helped us along the way. Our thesis was completed without the use of unethical tactics. The BRAC University's code of ethics guides our activities.

# Abstract

Roads are connecting lines between different places and are used in our daily life but anomalies in road surface not only impact road quality but also affect driver safety, mechanic structure of the vehicles, and fuel consumption. Several approaches have been proposed to automatic monitoring of the road surface condition in order to assess road roughness and to detect potholes. Potholes are one of the main reasons behind the occurrence of road accidents. According to a report submitted by The Roads and Highways Department (RHD), around 25% roads of Bangladesh under the RHD across the country are in "poor, bad or very bad" condition. This causes a lot of hassle and issues on the road for both humans and vehicles. Very often because of these potholes road accidents occur. Techniques for detecting potholes on road surfaces are being developed to provide real-time or offline vehicle control (for driver assistance or autonomous driving) as well as offline data collecting for road repair. For these reasons, researchers have looked into ways for detecting potholes on roads all over the world. This paper begins with a quick overview of the area before categorizing developed strategies into various groups. Then, by developing methodologies for automatic pothole detection, we present our contributions to the field. For this reason, we propose a deep learning approach that allows us to automatically identify the different kinds of road surface and to automatically distinguish potholes from destabilizations produced by speed bumps or driver actions. The system can detect potholes in different environments, lighting and weather conditions. We have trained and tested our model with a custom dataset which contains raw 3000 images with 1500 normal road images and 1500 images with potholes using deep learning algorithms. We have augmented these images and turned them into 120000 images so that the model can understand any image input in any scenario. In particular, we have analyzed and applied different deep learning models such as convolutional neural networks (CNN) and Yolov4. With these models we have achieved 97.35% accuracy with the CNN model and 87.6% accuracy with the YOLOv4 model.

# Dedication

This paper is dedicated to our families and fellow team members. The constant support of the family members and the determination of the team members played a great role in making this paper a success. We could not have finished our thesis without the help of our esteemed supervisor, who has been a continual source of guidance and advice. We also dedicate this paper to him.

# Acknowledgement

In the first place, we want to express our gratitude to Allah, the Almighty, for providing us with the resources, opportunities, and guidance necessary to complete this research on schedule.
As a second tribute, we would like to thank our thesis supervisor, Mr. Dewan Ziaul Karim, for his tireless support and guidance as we tackled a difficult topic. To get through the challenges, we had his continuous support and comments. Despite the current pandemic, he made time for us and provided valuable insights to help us enhance our job. For that, we will be eternally thankful.
Final words of thanks go out to the entire faculty and student body for establishing such a welcoming learning environment in which we were able to develop professionally while also doing this research to the best of our abilities.

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$CNN$ Convolutional Neural Network

$NH$ National Highways

$RHD$ Roads and Highways

$YOLO$ You Only Look Once

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background Information

Roads are such things that we use on a daily basis. From moving one place to another we use roads. But if the roads are full of potholes, it becomes a big of a hassle to move on the road. Due to the high cost of keeping the road surface in good condition, some automatic detection systems need to be implemented in Bangladesh. Bangladesh is such a place where we can see roads are full of potholes. According to a recent report on "The Daily Star" published on THURSDAY, December 16, 2021,[1] "the condition of the pothole-riddled roads worsened further due to accumulation of rainwater during this monsoon season. The roads have become impassable for all sorts of vehicles, let alone for pedestrians." Another report published on New Nation on the same day says, "Potholes on Dhaka roads cause damage to vehicles. Commuters face immense suffering." People are being victims of various accidents because of these potholes. [2]

Road networks are an essential resource because of their capacity to facilitate the rapid and uncomplicated movement of both goods and people. Many countries invest only a small percentage of what is required to keep their roads in good condition, despite the fact that road maintenance is vital to the well-being of both society and the economy. Most of the time, the responsibility for the upkeep of road networks falls on parts of territorial or municipal governments that operate with limited funds. People who are made to feel powerless are more inclined to engage in risky behavior such as engaging in illegal street activities, which can end in expensive injuries or even death. The aid in protection that would be provided by the early detection and description of such flaws would lead to decreased costs, increased utility, and improved well-being. Visual inspections are now the most reliable method available for determining a road's overall condition. The ease with which a certain street's assets may be reached as well as the significance of those assets are two factors that determine whether or not that street is feasible. Highways that see a lot of traffic will have open checks every day, and the flow of vehicles on those highways will be continuously monitored to assure their safety. A falling level of interest Even if they are checked every week, rural roads might have damage that isn't discovered for weeks or even months at a time.

As a consequence of this, it is abundantly clear that Bangladesh's roadways are

still plagued by a significant issue and need to be improved. The vast majority of collisions involving buses are caused by the drivers. They drive in a manner that is both hazardous and reckless the entire night. As a direct consequence of this, there have been numerous reports of accidents involving buses in recent news articles. The Roads and Highways Department (RHD) is currently filling in these potholes, but the issue is not going away anytime soon. There have been numerous attempts made in Bangladesh to apply existing pothole identification methods to road surface data; however, none of these attempts have been successful as of yet. In order to construct these systems, a wide range of different deep learning algorithms, frameworks, and architectures are being utilized. Convolutional neural networks (CNN), augmentation processes, long short-term memory (LSTM), yolo v4, solid-state drives (SSD), and other examples are some types of examples. In order to locate and identify potholes in the asphalt that covers the roads in Bangladesh, we will be using a number of detection methods in this research. This is done to achieve the highest level of accuracy and precision possible. Although systems very similar to this one already exist in locations such as Los Angeles and India, this one may prove to be particularly useful in Bangladesh due to the prevalence of potholes in that country. It is a remarkable tool that has the capacity to locate potholes in a dependable and exact manner on a broad variety of road kinds and surfaces.

## 1.2 Problem statement

In Bangladesh, a country that is still in the process of development, the population is about 170 million. The dismal condition of the roads is a significant obstacle. The majority of people's daily activities include the use of roads in some capacity. The transportation industry incurs enormous annual costs as a result of the poor state of the roads. Roadways in Bangladesh are considerably more congested and confined compared to those in neighboring countries. The standard configuration for streets in urban areas is two lanes in each direction. Additionally, it is effective for roads and highways. As a result of environmental variables, poor construction, and the overloading of vehicles, the roads are deteriorating and accumulating potholes. Accidents on the roads that are caused by poor maintenance are on the rise, and the number of persons who are killed in these kinds of occurrences is continuously climbing higher. When traveling at the high speeds that are typical of a driveway, it can be difficult to see potholes in advance. It's a risky move that could end in the vehicle being totaled out. In addition to putting drivers and pedestrians in risk, potholes cost the government millions of dollars each year and cost businesses even more. Meanwhile, the poor road infrastructure in Bangladesh is responsible for roughly $65 million in annual expenses related to traffic congestion.

Potholes in the road are one of the most common sources of frustration on the road. Potholes have become more widespread in unusual weather, such as heavy rain in the summer and snowfall, posing a threat to road safety and causing damage to the street. It causes social problems such as vehicle breakdowns and incidents resulting in social expenses. In this way, pre-programmed pothole detection tactics are read for effective pothole repair and asphalting by the executives. [3]

As a reaction to these issues, this research proposes a novel dataset that can be used to analyze photos of potholes and regular roads in Bangladesh. This dataset can be found in the following sentence: In a wide range of illuminance and climatic conditions, our system will be able to differentiate between a road in good condition and one that is in poor condition due to the presence of potholes. Around 3,000 photographs depicting a diverse variety of road situations and conditions have been gathered by us in order to achieve this goal (including those with and without potholes). We are the sole creators of these photographic works. The photographs show potholes in a variety of states, including those that are filled with water and those that have been dry as a result of recent precipitation. Through the application of binarization and segmentation algorithms to photographs of potholes gathered from across Bangladesh, the purpose of this study is to compile our very own dataset. After that, we built a filter so that the background noise would no longer be audible. As soon as that is done, we will be able to proceed to the subsequent steps, which are candidate extraction, region refinement, and finally closing. The mechanism for monitoring potholes has at long last been upgraded.

# Chapter 2

# Research

## 2.1 Research Motivation

One of the most significant ways in which metropolitan areas communicate with one another is through their distinct transportation networks. Additionally, it plays a significant role in the world of business. Regarding this particular issue, the unpredictability of the driving conditions presents a challenge. The majority of Bangladesh's asphalt roads were constructed without any form of planning in place, which is surprising given the country's extremely high population density. The vast majority of the roads are quite winding and constricted in their width. In addition to this, both state highways and municipal roads are riddled with potholes, which further impedes the movement of vehicles on the roads. Even though there are no other vehicles on the road, the vehicles are unable to go any quicker than they already do since the surface is riddled with countless potholes. In 2017 [4], there were 7,397 deaths that were attributable to occurrences involving motor vehicles, which accounted for 4,979 of those fatalities. Accidents are responsible for an increasing number of fatalities and injuries each year, in addition to the loss of a great number of other people's possessions and the ruin of their own. The Dhaka Tribune forecasts that there would be an average of 18 people killed per day due to road accidents in the year 2020. As a direct consequence of the subpar road building concepts that were used, significant portions of the roadway have deteriorated and turned into a patchwork of potholes. In this broad region, there are an abnormally high amount of potholes and other road defects. As a direct result of this, there is always the possibility that a very minor accident will take place. This is because of the fact that. Because of this, a considerable amount of suffering and hardship will be inflicted upon the human population.

## 2.2 Research Objective

During this project, we will be attempting to identify potholes in photographs by employing a Convolutional Neural Network, more commonly referred to as a CNN. We are going to use a dataset that is solely comprised of photographs of asphalt roads in Bangladesh for this very first attempt of its sort. This will be the first time that something like this has been attempted. In the course of our research, we make use of various object detection strategies, such as YOLOV3 and Single Shot Detec-

tor, with the goal of identifying problematic road conditions in photographs (SSD). During the course of our inquiry, we plan to make use of a dataset of pictures that is far more comprehensive than the ones that have been used in previous publications; the total number of these photographs will be somewhere in the vicinity of 12,000. The convolutional neural network, more commonly referred to by its acronym CNN, is a crucial component of neural networks due to the fact that it employs visual identification and classification in order to locate the things of interest. The CNN will undergo improvements that will allow it to properly and swiftly recognize potholes in images. These improvements will take place in the near future. Our computer program is also able to identify emergency situations, such as photographs that are blurry or potholes that are filled with water.

# Chapter 3

# Literature Review

## 3.1 Pothole Identification Approaches

Three developed methods for pothole identification have already been published in conference papers by the authors. The materials reported in that paper are expanded upon in this paper. This paper begins with a review of pothole identification approaches, building on brief notes on related literature from prior conference talks. This work will also present (with more material) the three previously published approaches, introduce the fourth method, and compare the four methods. This is the first time they've used a more diversified group of data for this assessment. Potholes create a variety of challenges depending on the weather, lighting, road layout, and traffic. Because there is no online benchmark dataset for pothole detection, we will gather data from a variety of sources and will recommend that future discussions of progress in this subject of pothole detection use those five datasets, which will be recorded under various weather circumstances. [5] [6]

## 3.2 Automated Pothole Detection

Koch and Brilakis presented a method for automated pothole detection in asphalt pavement images[7]. The image is initially segmented into the defect and non-defect regions using the proposed method. The geometric properties of a fault zone are then used to approximate the possible pothole shape. The texture of a prospective region is then retrieved and compared to the texture of the non-defect region surrounding it. The defect zone is deemed to be a pothole if the texture of the defect region is rougher and grainier than the surrounding surface texture. To evaluate the suggested method, it was implemented in MATLAB using the Image Processing Toolbox, and images were cropped from video files acquired using a remote-controlled robot vehicle prototype equipped with an HP Elite Autofocus Webcam positioned at a height of about 2 feet, as shown in Fig 1. A total of 120 photos were acquired, with 50 being utilized for training and the rest for testing. As a result, The accuracy was 86% with 82% precision and 86% recall.

## 3.3 Vision-based Strategy

Buza et al proposed a new model. a vision-based strategy that does not require supervision [8] Expensive equipment, extra filtration, and a training period Image processing and other techniques are used in their procedure. Identification and rough estimation using spectral clustering Potholes abound. The proposed approach is broken down into three parts. stages like picture segmentation and form extraction Identification and extraction, as well as spectral clustering In MATLAB, the proposed method was implemented. 50 photos of potholes were chosen and evaluated. Image compilation from Google. The precision with which a value is estimated The surface area of potholes was approximately 81%. As a result, this strategy can be used to get a rough estimate for pavement repairs and restoration.

Vision-based approaches are suitable for accurately detecting potholes across a large region at a low cost. Many methods based on 2D pictures and video data have been investigated. Koch and Brilakis [15] were the first to propose employing 2D photos to detect potholes. Searching for specific pothole traits and establishing pothole zones was part of their strategy. They employed a remote-controlled robot vehicle prototype with a webcam (an HP Elite Autofocus) mounted at a height of around 60 cm. Buza et al. proposed a new unsupervised vision-based method that does not necessitate the purchase of expensive equipment.

Koch and Brilakis' method [4] was confined to single frames and hence could not estimate the extent of potholes in a video-based pavement assessment frame. Koch et al. introduced an upgraded pothole-recognition approach that updates the texture signature for intact pavement portions and uses vision tracking to follow detected potholes across a sequence of frames to complement and improve the earlier method [9]. The proposed approach was evaluated on 39 pavement videos with a total of 10,180 frames using MATLAB. Total identification precision and recall were 75 percent and 84 percent, respectively, as a consequence of the study.As a result, when compared to the previous method, the texture-comparison performance was improved by 53% and the computation time was cut in half. They believed that only one pothole enters the viewpoint at a time, hence considering several potholes in the viewport requires more work.

## 3.4 Convolution Neural Network

In [10] Pereira, V. et al. employed Convolution Neural Network and compared the performance of their model to SVM, discovering that their model outperformed SVM by 99.80%. They used CNN, pooling, the ReLU activation function, the Adam Optimizer, and the Sigmoid function to deploy the model. Convolution and pooling were employed to extract features in this case. The Adam optimizer is used to lower the cost function and sigmoid function for projected output values ranging from 0 to 1.

## 3.5 Accelerometer-based Model

In [11] Artis M. et al. suggested an accelerometer-based model that employs the Z-THRESH, Z-DIFF, STDEV(Z), and G-ZERO algorithms that can be implemented on any Android OS-based smartphone with low hardware and software resources. They examine the model's performance with a 90 percent true positive value.

## 3.6 Crack-net

For identifying road cracks, a novel neural network called Crack-net [12] has been developed. The distinction between this neural model and others is that it does not include any pooling layers. This approach proved particularly effective in detecting fractures and uneven road surfaces.

For crack damage detection from concrete photographs, a deep learning-based convolutional neural network was utilized as a classifier.[13] They create a classifier that is less affected by noise caused by illumination, shadow casting, and other factors. When opposed to traditional methods, the advantages of this experiment are that it automatically learns the feature without requiring any feature extraction or computation.

## 3.7 Detecting Road Damage

Hiroya Maeda et al. developed a method for detecting road damage using deep neural networks and photos collected with a smartphone. They created a new large-scale dataset for road damage identification and applied a deep learning-based end-to-end object detection method to the road surface damage detection problem, verifying its detection accuracy and processing speed for road damage detection and classification.[14]

## 3.8 Vibration Method

The vibration method is a low-cost approach for determining the severity of potholes. However, this technology may be hazardous to the car because it is unable to distinguish between potholes and other road artifacts. The laser scanning approach in 3D reconstruction can evaluate pothole size and severity, however, this device is expensive and has a limited detection range. The stereo vision approach is quite inexpensive and can estimate pothole size, but it cannot assess pothole severity. The cameras must be well aligned and have a limited detection range. The Kinect approach is still a new way of detecting potholes and assessing their size and severity, however, it does not work in direct sunlight and has a limited detection range. It is less expensive than most of the previously described approaches and can detect potholes ahead of time. It can measure the magnitude of potholes but not their severity. A pothole dataset for Lebanese and other nations' roads is developed in order to design an efficient pothole detecting model. The detecting model has been upgraded to work in real-world settings. The system has a strong processing capability and can detect potholes in real-time at a high frame per second (FPS). Our

technology is reliable, with adequate precision, sensitivity, and recall.This work can be used to report potholes on roadways in real time to responsible agencies, enhance driver safety by assisting them in detecting potholes ahead of time, and improve the performance of self-driving cars in the future to ensure safe rides for passengers.[16]

## 3.9 Recognizing Potholes with 3D Asphalt Data

Tsai and Chatterjee (2018) offer a method for recognizing potholes that makes use of 3D asphalt data and a watershed approach. They carried out tests making use of the 3D data collected on tenth Street in Atlanta, Georgia, and 6 meters of street in Savannah, Georgia, on U.S. 80. The results showed an exactness of 94.97 percent, an accuracy of 90.80 percent, and a review of 98.75 percent. It has been shown that the suggested approach is effective for pothole locating and may provide a reliable method to pothole identification. This is especially true in situations when 3D asphalt information has been acquired for break discovery and is already available [17].

## 3.10 Identify Road Damage

In order to identify road damage.Hiroya Maeda, et al. [18] built a system employing CNN algorithms on photos obtained by smartphones. A real-time pothole detecting system for Android handsets was developed by Tedeschi A, et al[25].

# Chapter 4

# Approach

## 4.1 Data Preprocessing Plan

### 4.1.1 Data Acquisition

There are approximately 1500 pictures total that show roads that have been damaged by potholes, and another 1500 pictures total that show roads that are completely undamaged. These images were taken in a number of sites in Dhaka and Dinajpur, which are considered to be the two most significant cities in Bangladesh. These two cities can both be found in the country of Bangladesh. We compiled a photo album that was completely comprised of photographs that were taken in the open areas of Bangladesh. The compiled collection of photographs features the work of each and every member of the crew, who all contributed one or more photographs to the project. The dataset is divided into training data and test data with an 8:2 split between the two categories of data.

### 4.1.2 Image Augmentation

In order to expand our dataset, we performed image augmentation. Rotation, scaling, cropping, flipping, translation, affine transformation zooming, and picture sharing were also done with image augmentation parameters. This was employed as a result of generating more photos and modifying the positions of the images in order to improve our trained models' recognition.

### 4.1.3 Image Annotation for YOLOv4

The task of annotating an image with labels is known as image annotation. The labels are used to provide information about the images that are utilized as input data to a computer vision model. We used the **Labelme** annotation tool to annotate our training and testing photographs. The **Labelme** is used to segment our images (training and testing), as well as to build bounding boxes and generate coordinates for each training image.

### 4.1.4 Final Dataset (YOLOv4)

In order to train our YoloV4 model, we create two more files with the names "dataset.names" and "dataset.data". The class name of our model is found in

the "dataset.names" file. In our situation, we divide everything into two categories: "Normal" and "Pothole." The location of the train and validation picture (in our case, the location of train.txt and test.txt), the location of the "dataset.names" file, and the location of where to store newly created weights are all included in the "dataset.data" file. "dataset.data" also contains the number of classes, which in our case is two. We also move the picturename.txt file into the "dataset" folder. Now, our dataset is ready to train YoloV4.

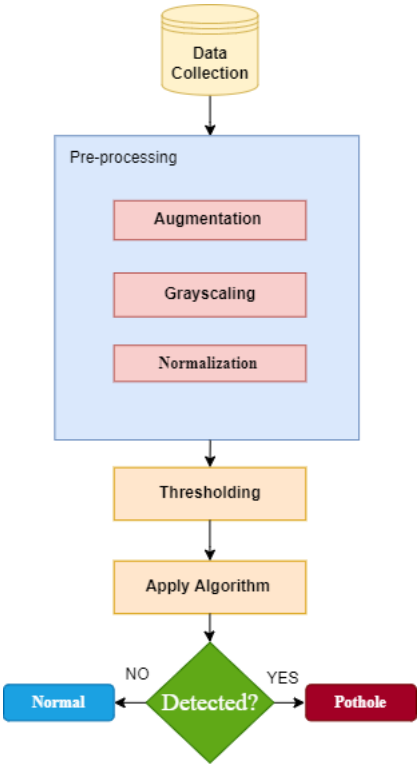



Figure 4.1: Full Working Plan

## 4.2 Architecture and Frameworks

After the cleaning and preparation of the data set has been finished, we will go on to the next step of training two separate models using it. We feed the data about the potholes into a computer program called YOLOv4 (You Only Look Once), as well as a CNN that has been taught to create predictions of this kind. Both of these systems are designed to analyze the data and produce forecasts. Because we created our own model, not only were we able to construct a model that was more accurate than the model that CNN typically uses, but we were also able to design a model that was more efficient.

YOLO (You Only Look Once) is an object detection algorithm that is popular for detecting objects in images. This algorithm uses a single neural network to predict the vector of the bounding boxes and potholes[17,18]. It works by splitting images into a grid with the size of S*S. Every cell in the grid can predict N possible bounding boxes and the level of probability (i.e. confidence score) of it being the object

which in our case is a pothole. This gave us S * S * N boxes. Figure X demonstrates the architecture of YOLO. In this paper, we are using YOLO V4, the training of this model is done on full images and the probability of the class in the bounding boxes. The YOLO V4 is better than the YOLO v3 model in terms of speed and accuracy. Because YOLO V4 treats object detection as a regression problem, no complicated pipeline is required. Every time we need to make a prediction, we run the neural network on a new picture. Before detecting and making predictions, YOLO V4 looks over the image as a whole and one time only.

In order for us to accomplish this goal, we are relying on a wide range of different frameworks, such as but not limited to, Tensorflow and OpenCV, amongst others. These frameworks allow us to accomplish the task at hand. Because of the high degree of performance it provides and the vast number of individuals who use it, Tensorflow is a well-known framework for the detection of objects in computer vision. This is due to the fact that it is utilized by a big number of people. Because it comes with a big library of pre-trained models that have been trained on a broad variety of datasets, we were able to use it to easily create our own one-of-a-kind classifiers. This was made possible by the fact that it was pre-loaded with the models. Because of this, we were able to cut down on our overall time commitment. This was made possible by the fact that it comes with pre-trained models that have been trained on a wide variety of datasets. This made it possible for it to do what it set out to do. This was made possible as a result of the fact that it comes with models that have already been pre-trained. TensorFlow's provided structure was followed religiously throughout the entirety of the procedure in order to ensure consistent results. Because of this, every stage, including preprocessing, model development, and model training, was carried out in an accurate manner.

OpenCV is a library that can be used on a variety of different platforms and is mostly aimed toward the recognition of objects via the application of image processing. This is the core focus of OpenCV, which can be used to recognize items through the application of image processing. We turned to the aforementioned method in order to convert our color photographs into black and white versions of themselves. Grayscale photos have fewer channels than their color equivalents do, which is the reason why this is the case. Color images are the reason why this is the case. After the photographs have been converted to grayscale, morphological techniques such as erosion and dilation are applied in order to improve the picture's quality and repair any data issues that may have occurred.

# Chapter 5

# Dataset and Model

It is common knowledge that the process of accumulating data can on occasion provide some challenges, and it is also well known that doing so can be somewhat taxing. If a model is provided with access to more data, then it has the ability to generate findings that are more accurate. After the activation of the lockdown procedures and the subsequent shutdown of the schools, we were given additional baggage to transfer and we were also directed to remain in our places. In addition, we were given instructions to remain in our locations. It has come to light that approximately three thousand images of highways in Bangladesh's three principal regions have been acquired with a high resolution. These pictures show parts of roadways that have been destroyed as well as parts of the same routes that have been unaffected by the storm. The complete lengths of both undamaged and ruined segments of highway are captured in their entirety by these images. Every single person who calls Bangladesh home provided us with some information for our database. This comprises those who live in other parts of the country in addition to those who live in the cities of Dhaka, Dinajpur, and Narsingdi. In order to develop CNN and YOLOv4, respectively, it was necessary to make use of two distinct model architectures because doing so was necessary in order to obtain the results that were desired. This was essential in order to achieve the goals that had been set for the project. To get a better idea of how well our bespoke models are performing, we compared them to both of these and to a large number of other pre-trained CNN models. This allowed us to determine whether or not our bespoke models are effective. Because of this, we are in a position to assess how well our bespoke models function and determine whether or not they are successful. Because we wanted our model to be able to process the information more effectively, we annotated each and every photo that we took in the numerous different scenarios in which we gathered data. This was done because we wanted our model to be able to properly digest the information. This was done because we wanted to ensure that our model could correctly handle the information that was given to it.

Following figure shows images that are used in our dataset which are taken only from Bangladesh area and further modified for the best outcome.


Figure 5.1: Dataset Image

## 5.1 Data Preprocessing

### 5.1.1 CNN Model

The data must first be preprocessed before the Convolutional Neural Network (CNN) method can be applied to the data; this step is required. During this phase, the data that will later be utilized by the CNN technique are prepared. At this point in the process, any components of the CNN model that do not directly contribute to an improvement in the overall performance are deleted. This ensures that the performance of the CNN model as a whole is optimized. The term "decoupling" refers to this particular stage of the process. Alternately, this stage provides us with the opportunity to make any necessary modifications to the raw data in order to enhance both the operational capabilities of the CNN model and the output quality that it generates. The chance to make any necessary alterations to the raw data is utilized in order to attain this goal. We used a raw data snapshot that included 3000 points as the basis for our state-of-the-art algorithm that we suggested for the detection of potholes. Because of this, we were able to construct a more solid basis for our algorithm. We brought the total number of images utilized in the model up to 12,000 so that it would be able to generate reliable forecasts regardless of the conditions. We altered the image by zooming it, panning it, squinting it, panning it, rotating it, flipping it vertically and horizontally, and altering the levels of color saturation and brightness in the image. After that, the items contained within were separated into their respective groups. Participating in frequent practice and conducting evaluations are both absolutely necessary. The training dataset contained a total of 10512 pictures, and 80% of those photographs were applied to the final product. On the other hand, in the validation set, there was only a 20% overlap between the photographs that were used in the training dataset and the photographs that were used (2001 images).

The training dataset that we used has undergone some modification. The following changes have been made in regards to the parameters. Most likely, these modifications were brought about directly as a result of the work we accomplished. Table I includes a record of every change made to the dataset both before and after the model was trained. It also includes a log of every change made while the model was being trained. Table I thus provides a record of each update to the dataset that was made at any moment while the model was being trained.

| Properties | Values |
|---|---|
| Target size | 64 X 64 |
| Batch size | 32 |
| Rescalling size | 1 / 255 |
| Zoom range | 0.2 |
| Horizontal Flip | enabled |
| Shear range | 0.2 |

Table 5.1: Properties used for CNN model preprocessing

We picked the 'binary' class mode since our categorization results fall into one of two categories, either it will be 'pothole' or 'Normal'. We used the same image and batch size to our validation dataset as well. We've also preserved the binary class mode.

### 5.1.2 YOLOv4

#### 5.1.2.1 Normalizing the Data

Obviously, the pictures were in different resolutions and in different aspect ratios. We resize the picture using free software called "Caesium" into a fixed resolution and aspect ratio. The fixed resolution was "416 * 416" and the aspect ratio was 1:1.

#### 5.1.2.2 Labeling Image

To feed data into yoloV4 for training we need to annotate the training images into such a structure that yoloV4 accepts. And that is a vector consisting of 7 rows.

Here, Pc is the probability of a class that is either 1(Pothole) or 0(Normal) in our case. Bx, By are the coordinates of the center of the bounding box. By, Bh is the width and height of the bounding box. C1, and C2 are class1(Pothole) and class2 (Normal).

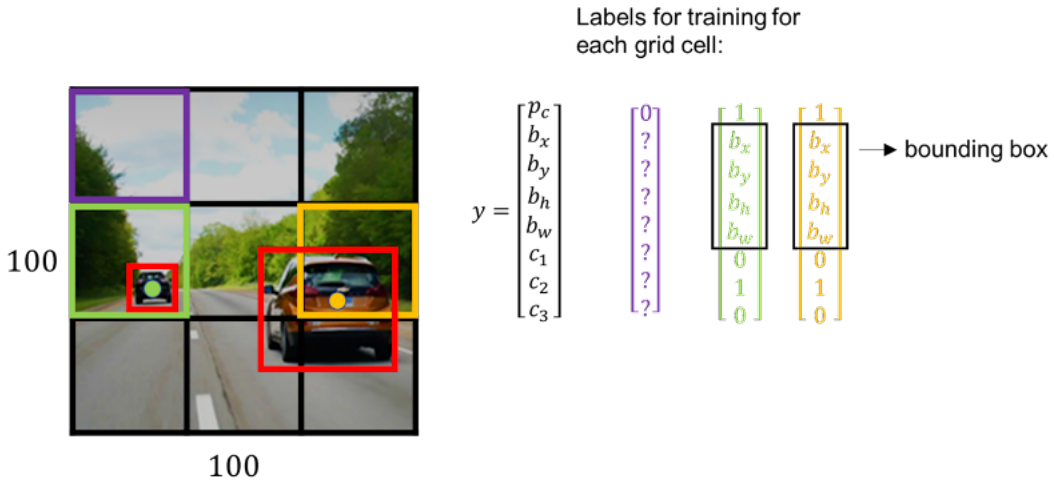



Figure 5.2: Vector of bounding box to give input from image

We used an open-source software called "Labelmg" to annotate our image.

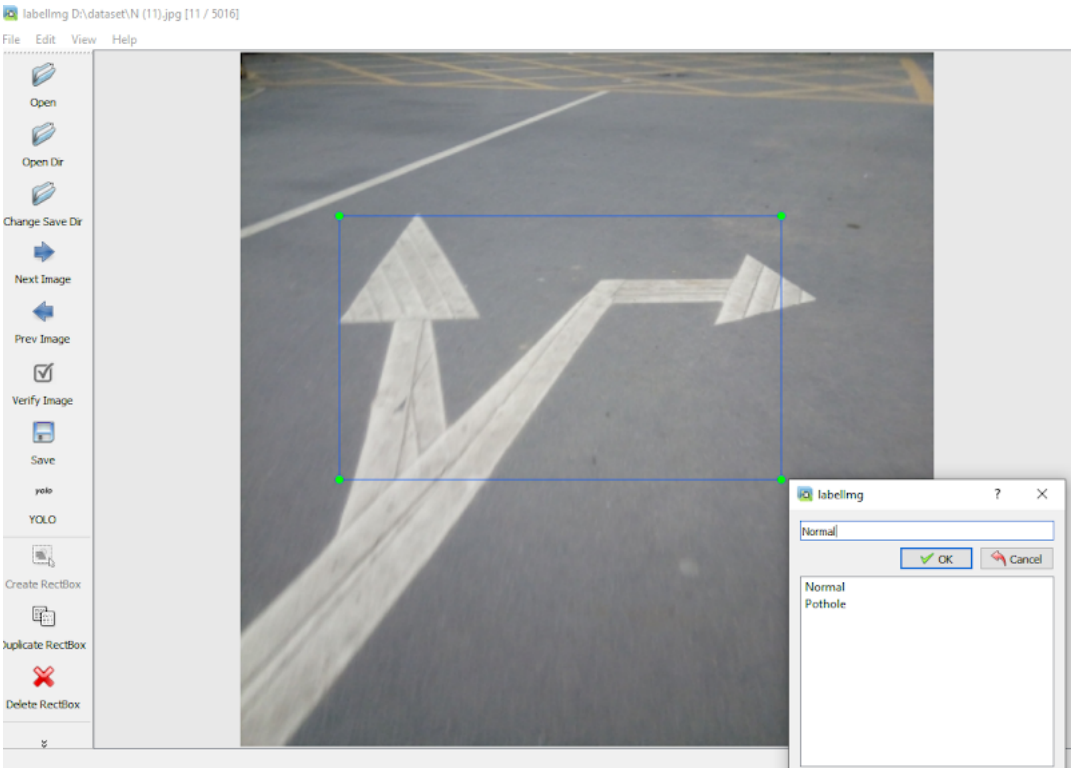



Figure 5.3: Annotation of images using LabelIng software

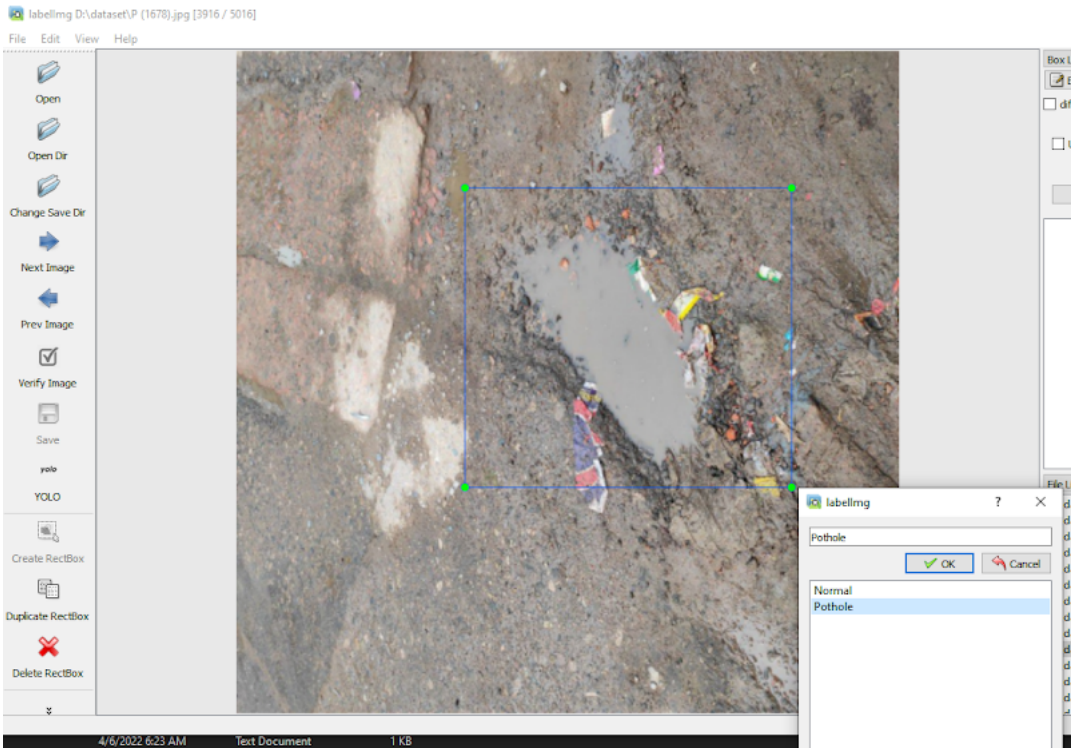



Figure 5.4: Annotation of images using LabelIng software

After completing the annotation, we get a picturename.txt file where we get the structured coordinates of the bounding boxes for yoloV4.

#### 5.1.2.3 Creating Final Dataset

Then we move all the pictures from the "Pothole" and "Normal" folders into one single folder named "dataset". We also move the picturename.txt file into the "dataset" folder. Now, our dataset is ready to train YoloV4.

We use another open-source framework called "Darknet" to train our model on yoloV4. Darknet is the most famous framework to work with yoloV4 in the world.

We took 10% of our data for testing (total 501) and 90% of our data (total 4515) for training randomly generated by a python script. This script generates two text files named "train.txt" and "test.txt". "train.txt" contains a list of randomly chosen pictures from our dataset for training. "test.txt" contains a list of randomly chosen pictures from our dataset for testing purposes.

We also need to create two other files named "dataset.names" and "dataset.data" to train our YoloV4 model. "dataset.names" contains the class name of our model. In our case, we have two classes: "Normal" and "Pothole". "dataset.data" contains the number of classes (2 in our case), location path of train and validation picture (in our case, location of train.txt and test.txt) , location of "dataset.names" file and location of where to store newly created weights.

## 5.2 Model Architecture

### 5.2.1 CNN(Convolutional Neural Network)

As a component of the approach known as "Deep Learning," hierarchical neural networks are utilized to perform an analysis on the gathered data. This method takes as its inspiration the way in which the human brain performs its functions. In order to achieve its overarching aim of enhanced decision-making, its primary focus is on the production of algorithms for data analysis and pattern production. This is done in order to meet its goal of improving decision-making. Because of this, it will be able to complete its primary objective. The Convolutional Neural Network is one of the algorithms for deep learning that is used the most frequently, and it is typically used during the process of processing a wide variety of forms of visual input. This is because the CNN is able to "convolute" or "convert" one form of visual input into another. This is due to the fact that CNN is one of the algorithms for deep learning that is utilized by most people. In the past ten years, it has delivered insights that have fundamentally altered the course of play in a wide variety of disciplines, ranging from picture identification to speech recognition [24]. This has been the case in a number of different fields. CNN's programming may be broken down into three distinct categories. The first layer is a convolutional layer, the second layer is a pooling layer, and the third and final layer is a fully connected layer. In all, there are three layers. The first layer is a convolutional layer (FC). Each of these tiers is an integral part of CNN's infrastructure and makes up the network as a whole. In addition to these core features, the design includes the Activation function as well as the dropout layer in its composition. The process of gaining a more in-depth understanding of an image calls for the utilization of both convolution layers and pooling layers. It is necessary for them to work together. After the process of convolution has been finished, the data that was produced are subjected to an evaluation through a method called feature extraction. This makes it possible to place a photograph into one of several different categories. This procedure is commonly known as "Classification," and for the purpose of elucidating it, we shall refer to it by that term throughout this discussion.

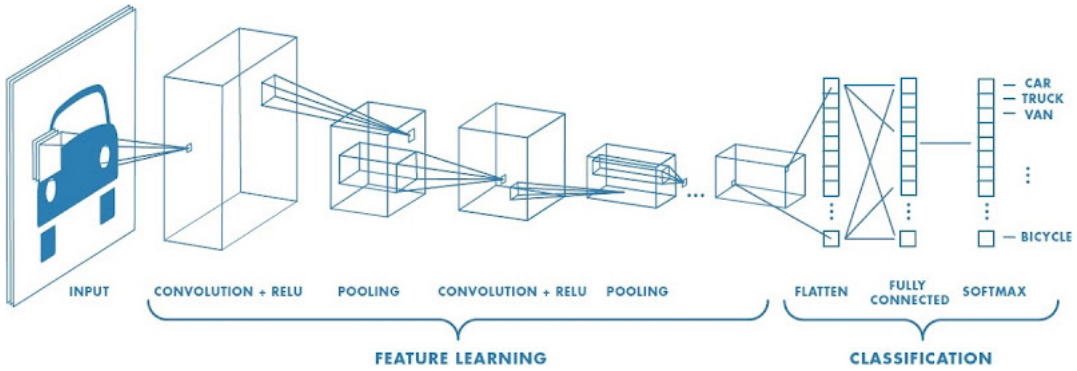



Figure 5.5: CNN architecture

#### 5.2.1.1 Convolutional Layer

For our proposed system,a total of five 2D convolutional layers are used. For the first layer, we specified an input size of (224,224,3) to match the dimensions of the input

photos This layer's kernel was set to 3, 'relu' activation was utilized. To determine the learnable parameters in this case, The only thing we need to do is multiply by the dimensions of the current layer's filters (k), the preceding layer's filters (d), and the form of width m and height n. [23]

**((m * n * d)+1)* k)**

#### 5.2.1.2 Pooling Layer

Convolutional and Fully Connected layers are frequently connected via a Pooling Layer. We employed max pooling in our model. The filter of the first layer is 32 and for the second and third layer it is 64 and for the fourth layer it is 128 and for the fifth layer it is 256. Then we flattened the layers. Then, the 2-Dimensional arrays which are generated from pooled feature maps are flattened into a single, long continuous linear vector.[24]

#### 5.2.1.3 Second Convolutional Layer

We have added a second layer with the same config so that it does not conflict with the other one. Then we flattened the layers. Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

#### 5.2.1.4 Full connection

To establish a full connection (FC), all the layers need to be connected with each other. The weights and biases, as well as the neurons, are all part of the FC layer. The two levels are connected by it. This layer is mostly just a way to help the classification process work. We used two Dense layers and one Flatten layer in this step with a unit of 256. We employed the sigmoid' activation function for the output layer.

**((current layer neurons c * previous layer neurons p)+1*c)**

#### 5.2.1.5 Output Layer

For the output layer we have taken unit = 1 and used the '**sigmoid**' function. The main reason why we use the sigmoid function is that it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since the probability of anything exists only between the range of 0 and 1, sigmoid is the right choice for our model.

#### 5.2.1.6 CNN Model Summary

The total number of "learnable" (if such a term exists) elements for a filter, commonly known as the filter's parameters, in a particular layer is the number of parameters in that layer.[25]. We used a neural network toolkit called Keras to develop a sequential CNN model for the proposed system after first separating the dataset into train data and validation data. This was done in order to test the accuracy of the model. Within our model, we included a total of five 2D convolutional layers in addition to the same number of max pooling layers. Following that, we applied two levels of dense, followed by a single layer of flatten. In the end, we were successful in obtaining an exact total of 493,569 trainable params for the model to use in training the images.

| **Layer Type** | **Output Shape** | **Param #** |
|---|---|---|
| conv2d (Conv2D) | (None, 62, 62, 32) | 896 |
| batch_normalization | (None, 62, 62, 32) | 128 |
| max_pooling2d | (None, 31, 31, 32) | 0 |
| conv2d_1 | (None, 29, 29, 64) | 18496 |
| batch_normalization_1 | (None, 29, 29, 64) | 256 |
| max_pooling2d_1 | (None, 14, 14, 64) | 0 |
| conv2d_2 | (None, 12, 12, 64) | 36928 |
| batch_normalization_2 | (None, 12, 12, 64) | 256 |
| max_pooling2d_2 | (None, 6, 6, 64) | 0 |
| conv2d_3 | (None, 4, 4, 128) | 73856 |
| batch_normalization_3 | (None, 4, 4, 128) | 512 |
| max_pooling2d_3 | (None, 2, 2, 128) | 0 |
| conv2d_4 | (None, 2, 2, 256) | 295168 |
| batch_normalization_4 | (None, 2, 2, 256) | 1024 |
| max_pooling2d_4 | (None, 1, 1, 256) | 0 |
| flatten | (None, 256) | 0 |
| dense | (None, 256) | 65792 |
| dense_1 | (None, 1) | 257 |
| Total params: 493,569<br>Trainable params: 492,481<br>Non-trainable params: 1,088 | | |

Table 5.2: CNN model Summary

### 5.2.2 Yolo V4

#### 5.2.2.1 Input Data

In order to take these images, a total of four distinct mobile cameras were deployed. After that, the data was split into two separate files, one of which was designated as the "Pothole" file, while the other was designated as the "Normal" file. The image of the crater was saved in the first file, and the image of the road with its typical appearance was saved in the second file. The collection comprised a total of 12500 photographs in its entirety. This image was captured in the year 2022, and it serves as a good illustration of the genre as a whole. There were a total of 6250 pictures that included craters in some shape or another.

YOLOv4 is a real-time Object Detection model that is SOTA (state-of-the-art). Alexey Bochkovsky published it in April 2020, it is the fourth edition in the YOLO series. On the COCO dataset, which contains 80 different object classes, it achieved SOTA performance.

The YOLO detector is only capable of handling one stage of the detection process when it is used on its own. When it comes to the process of identifying objects, there are two primary approaches that are generally acknowledged as representing the state of the art. One of these approaches is known as the one-stage procedure. Since this strategy is considered to be one of the two primary approaches, it is frequently referred to as the most cutting-edge approach. It is one of the reasons why this method is considered to be one of the most cutting-edge ways since it places a great emphasis on how rapidly conclusions can be formed from the data. When used in formal writing, the alternative method is typically referred to as the two-stage approach. [Case in point] [This is a prime example] The only predictions that one-stage detector models are able to generate are those for the classes and bounding boxes of the entire image. There is no indication of the ROI, which is an abbreviation that stands for the Region of Interest. They are able to identify chemicals in a shorter amount of time than detectors that have two stages because they only have one stage, which is also the reason why they are able to identify chemicals in a shorter amount of time than detectors that have two stages. This is due to the fact that there is only one stage in their progression.

It divides the object-detection problem into two parts: regression and classification. In a single run, regression predicts classes and bounding boxes for the entire image, assisting in the identification of item position. The class of an object is determined via classification.

#### 5.2.2.2 Algorithm Overview

The architecture is made up of several pieces, the first of which is the input, is basically what we have as our collection of training images that will be supplied to the network - they are processed in batches by the GPU in parallel. The Backbone and Neck are the next two components, and they are responsible for feature extraction and aggregation. The Object Detector is made up of the Detection Neck and Detection Head. [26]

Finally, the head is in charge of detection and prediction. The Head is primarily in charge of detection (both localization and classification).

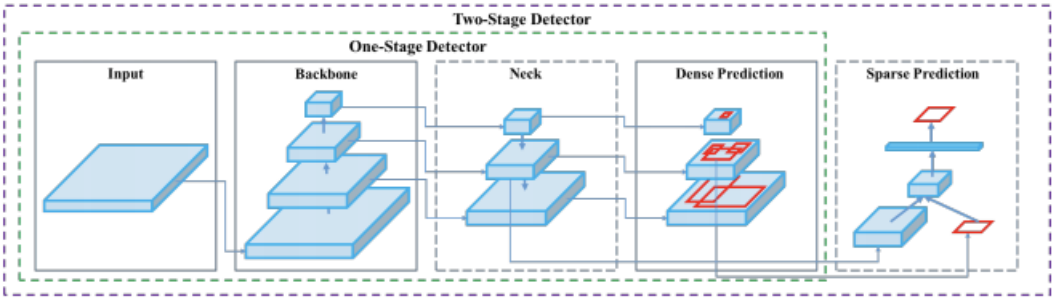



Figure 5.6: Architecture of YOLOv4

Because YOLO is a one-stage detector it does both of them simultaneously (also known as Dense Detection). Whereas, a two-stage detector does them separately and aggregates the results (Sparse Detection)

The Figure is as follows:

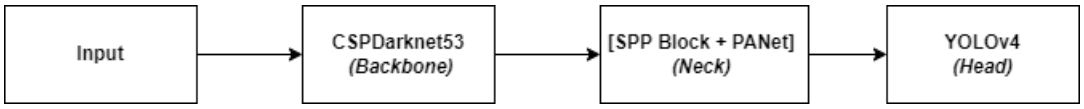



Figure 5.7: Workflow of YOLOv4

#### 5.2.2.3 Backbone Network

The backbone network of yoloV4 is CSPDarknet53 CNN. [27]

CSPDarkNet53 is based on the DenseNet design. It concatenates the previous inputs with the current input before proceeding into the dense layers - this is referred to as the Dense connectivity pattern.

CSPDarkNet53 consists of two blocks

- Convolutional Base Layer
- Cross Stage Partial (CSP) Block

#### 5.2.2.4 PANet (Path Aggregation Network)

A redesigned path aggregation network is used in YOLOv4, mostly as a design change to make it more appropriate for training on a single GPU.

PANet's primary objective is to improve instance segmentation by preserving spatial information, which is essential for accurate pixel localization in mask prediction. This will allow PANet to achieve its goal of improving instance segmentation. PANet

will be able to accomplish its goal of enhancing instance segmentation as a result of this. In order to accomplish this objective in a fruitful manner, PANet is deployed. [28] Bottom-up Three essential components—Path Augmentation, Adaptive Feature Pooling, and Fully-Connected Fusion—play a significant role in the high accuracy of their mask prediction. The whole process of prediction is significantly impacted by the contributions made by each of these components.

#### 5.2.2.5 SPP – Additional Block

The CSPDarkNet53 brain and the feature aggregator network are separated by the SPP block, which is also known as the spatial pyramid pooling block (PANet) (PANet). In spite of this, the action almost has no impact on the amount of data that can be transferred across the network. This is because it is done in order to extend the receptive field and get rid of components of the context that aren't necessary. The reason for this is because it is done in order to expand the receptive field. It is connected to the most complex convolutional layers of the CSPDarkNet, which allows it to communicate through those layers.

#### 5.2.2.6 Neck

You'll find a collection of features that characterize the individual in the region that's near to the neck, specifically. It takes the feature maps from each of the several previous iterations of the backbone and merges and mixes them so that they are prepared for the upcoming iteration of the backbone. In most cases, a neck is composed of a number of top-down pathways as well as a number of bottom-up pathways. These two types of pathways work in opposite directions from one another. Both of these trails head in different directions relative to one another. The functions that are served by these two different sorts of passageways couldn't be more different from one another. These two routes diverge dramatically from one another and go in completely different directions.

#### 5.2.2.7 Head

The main purpose of this function is to locate bounding boxes and do categorization. The head of yoloV4 is the same as the head in yoloV3. [29]

The x, y, height, and width of the bounding box, as well as the scores, are detected. The x and y coordinates are the center of the b-box stated in relation to the grid cell's boundary. Width and height are calculated in relation to the entire image.

$$
\begin{aligned}
b_x &= \sigma(t_x) + c_x \\
b_y &= \sigma(t_y) + c_y \\
b_w &= p_w e^{t_w} \\
b_h &= p_h e^{t_h}
\end{aligned}
$$

#### 5.2.2.8 Bag of Freebies

A "Bag of Freebies" is used in YOLOv4 to improve network performance without necessitating that the production environment's schedule compensate for an increase in the amount of time spent on inference processing. This was made possible by the use of a "Bag of Freebies." For the purpose of completing this task in a timely and efficient manner, a "Bag of Freebies" was used. The majority of the information that is provided in the Bag of Freebies is devoted to discussing freebies that are associated with data augmentation. This is because these freebies are the most prevalent. This is due to the fact that the principal topic of discussion in Bag of Freebies is data augmentation. The creators of YOLOv4 made use of data augmentation so that they could broaden the scope of their training set and provide the model with fresh semantic contexts to learn from. Because of this, they were able to make the model's predictions with greater precision.

#### 5.2.2.9 Bag of Specials

YOLOv4 employs "Bag of Specials" techniques, so named because they add little delays to inference time but dramatically improve performance, making them worthwhile.

#### 5.2.2.10 Activation Function

Alexey Bochkovsky[26] tested a variety of activation functions. As features move across the network, activation functions change them. YoloV4 uses Mish as the activation function because it can push signals to the left and right, something that functions like ReLU cannot do..

To separate predicted boundary boxes, the authors employ DIoU NMS. Over a single item, the network may predict numerous bounding boxes, and it would be nice to choose the best one quickly.

The author(Alexey Bochkovsky) employs Cross mini-Batch Normalization (CmBN) for batch normalization, with the goal of being able to run it on any GPU. Many batch normalization techniques necessitate the use of many GPUs in parallel.

DropBlock regularization is used in YOLOv4. Sections of the image are hidden from the initial layer in DropBlock. DropBlock is a strategy for forcing the network to learn features it wouldn't have learned otherwise.

# Chapter 6

# Result and comparison

## 6.1 CNN

Following the development of the CNN model that would be utilized by the system that we were proposing, we went on to the process of training and verifying the model by making use of the data that had been gathered. This was accomplished by applying the CNN model. After completing the step that came before this one, we went on to the next one in the process. This happened after we had finished the phase in the process that came before it with flying colors. Following the completion of our work on the CNN model, our team moved on to the subsequent stage, which was also carried out by our group. After we finished the previous step, we moved on to the next one. The students needed to have completed 25 epochs before moving on to the next topic. This was a requirement of the curriculum that they were working through, therefore it was a prerequisite that they meet before moving on. We were able to achieve an accuracy level of 97.58% on the dataset that was utilized for the purpose of training. This is a statistic that can be used to judge how successful our efforts have been. After finishing all of the training, the model was then applied to the test set, and it had a val-accuracy of 97.35% when it was done so. This was the best possible score that could be achieved. After finishing all of the training that had been offered, this problem surfaced as a result of the situation that had been created. The findings are very reassuring, and they suggest that there are a great many opportunities simply waiting to be taken advantage of. Because we are the first individuals to attempt to build this model using data relating only to the roads in Bangladesh.

For better understanding we generated an acc vs val_acc graph in Figure 6 where we saw till 40 epoch the model was spiking a lot. But after crossing 40 epochs, the model became very stable. The graph of the total training is given here for better understanding:

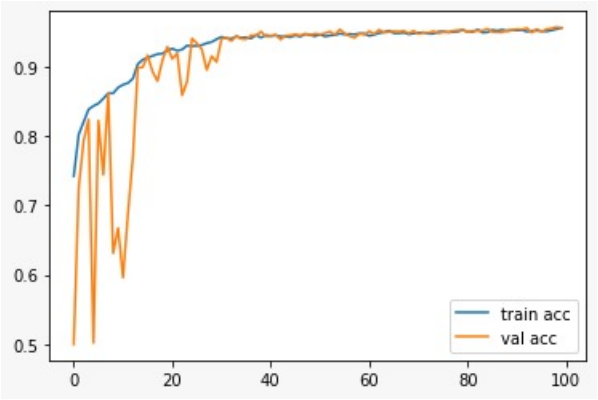



Figure 6.1: Train-acc vs val-acc of CNN

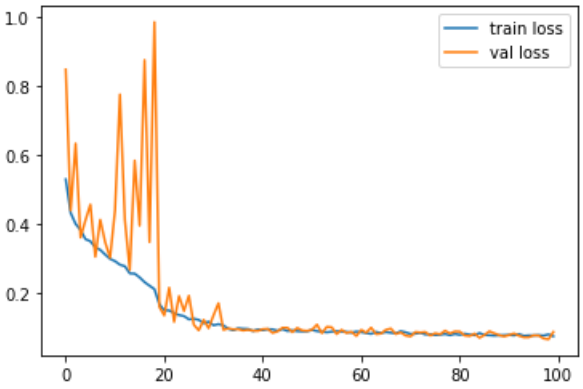



Figure 6.2: Train-loss vs val-loss of CNN

It has been demonstrated that our model achieves an accuracy that is approximately 1.2% higher than that achieved by modern pre-trained CNN models such as VGG16, VGG19, ResNet50, EfficientNetB0, EfficientNetB6, and InceptionV3. All of these CNN models are good examples of those that have been trained in times past. Nevertheless, in spite of the fact that, at first glance, the two models appear to have a great lot of similarities with one another, the customized model is in a position to make more accurate predictions. We are able to discern whether or not an image was taken of a pothole or a standard road thanks to the ability of the one-of-a-kind CNN algorithm to accurately assess the surface in any given lighting condition. Because of this, we are able to establish whether or not the puddle was visible in the shot. Our software can now generate an output that is capable of providing an incredibly accurate prediction of how each image that is used as input will look after it is processed.

Following table explains how the model is performing compared to the other prebuilt models like vgg16, vgg19 and more. It is clear that SZR5 has performed much better with all the modifications and works done to it.

| Model name | Training accuracy | Validation accuracy |
|---|---|---|
| VGG16 | 95.75% | 94.23% |
| VGG19 | 95.35% | 95.32% |
| ResNet50 | 94.28% | 91.32% |
| EfficientNet B0 | 93.04% | 92.24% |
| EfficientNet B6 | 94.28% | 93.66% |
| InceptionV3 | 95.87% | 95.66% |
| Custom CNN | 97.58% | 97.35% |

Table 6.1: All model comparison table

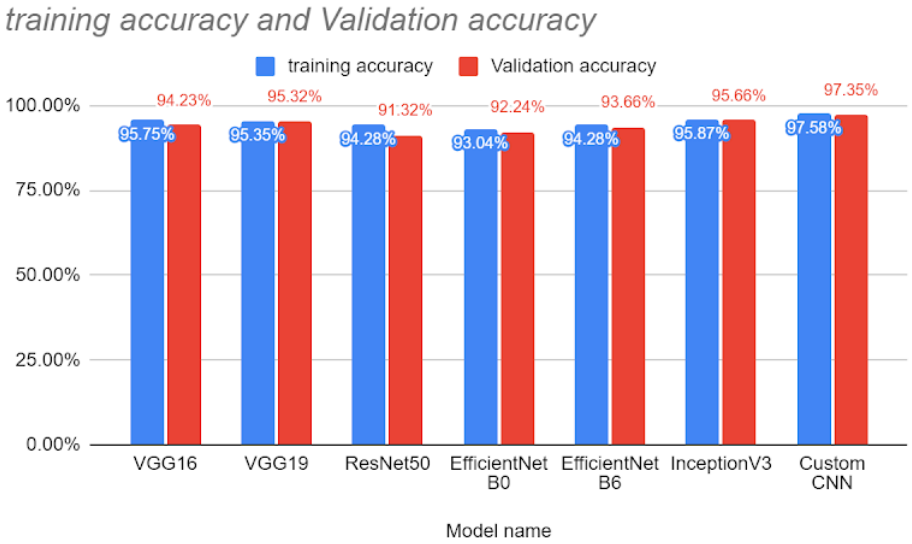



Figure 6.3: Bar chart showing accuracy and loss of all CNN models,

On the other hand, we plan to improve our performance in the future by making the required adjustments to both the model and the dataset in the future. These changes will contribute to an improvement in the accuracy of the training model. Recall, precision, and f1-score are displayed in Table V of the classification report below where 2000 photos were used to evaluate the model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| *normal* | 0.98 | 0.95 | 0.96 | 1000 |
| *pothole* | 0.98 | 0.94 | 0.95 | |
| *accuracy* | | 0.95 | | 2000 |
| *Macro avg* | 0.96 | 0.95 | 0.95 | |
| *Weighted avg* | 0.96 | 0.95 | 0.95 | |

Table 6.2: Classification report of CNN

## 6.2 YOLOv4

As for the YOLOv4, we conducted a search for material in the Darknet library so that we could train our yoloV4 model. [30] You can get Darknet, a free and open-source framework for CUDA and C neural networks, off the internet and download it to your computer. It can be carried out with either the central processing unit or the graphics processing unit, is quick to operate, does not require a significant amount of mental work to comprehend, and offers two different processing options. In the beginning, we set up a server and installed Darknet in addition to any other software that was required to get things rolling. Following that, we updated our own configuration file with the necessary settings. The image resolution, the number of epochs, the learning rate, and other criteria were among these settings (yolov4-custom.cfg). The following is a list of various variables that were susceptible to shifts over time:

| **Properties** | **Values** |
|---|---|
| Dimensions | 416 * 416 |
| Batch | 64 |
| Subdivisions | 32 |
| Learning_rate | 0.00261 |
| Policy | Step |
| Max_batches | 7000 |
| Steps | 4800,5400 |
| Class | 2 |

Table 6.3: Properties of Yolov4 custom model

It was agreed that 6000 epochs would be the highest amount of time that could ever be taken into consideration for any one particular circumstance. After much thought and consideration, we came to this conclusion. At this current juncture, there is a limit placed on the greatest amount of time that can be utilized in its entirety. We arrived at the conclusion that this is the most effective way to go after engaging in a considerable amount of introspection and conversation about the matter.The first "maP" computation was 35% as it should have been, as shown in the graph.When 4300 epochs have passed, the "maP" computation has reached a situation that we can call "close to saturation", at which time it will be maintained in a stable state until additional testing has been carried out (at approximately 70 percent). After a total of 6000 iterations, the problem was finally solved at a degree of precision that was 88.00 percent successful. Before drawing any conclusions regarding the level of volatility exhibited by the output, we lengthen the training phase so that it lasts for a total of 7000 iterations. We go through this process so that we can make intelligent choices. The number did not change from 88.00 percent even after 6000 epochs; rather, it remained the same for the whole of the era that was being discussed. After that, we were no longer required to honor the prior promise that we had made to our training in terms of the amount of time that we had been giving up to that point.

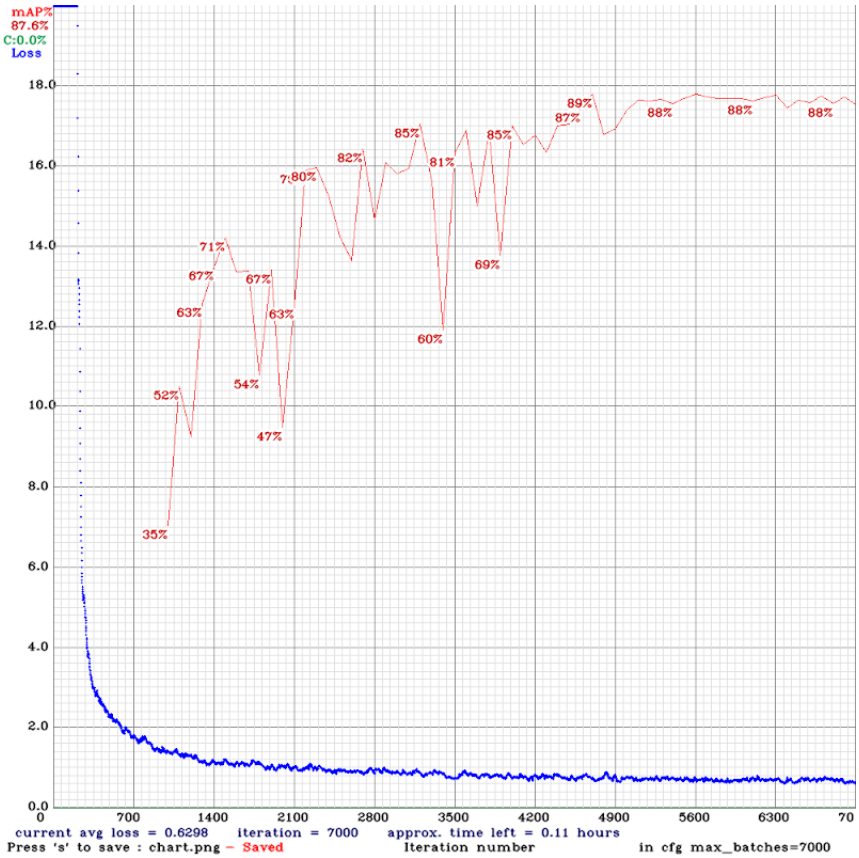



Figure 6.4: YOLOv4 accuracy graph

A comparison of two separate models was carried out so that we could determine whether model architecture is superior in terms of its ability to spot craters and other kinds of flaws. This was done so that we could make an informed decision. This step was taken since we give a method that is cutting-edge and, as a result, merits being acknowledged as being among the best available. The only images that are being used in the process of obtaining information are ones that were taken on Bangladesh's highways. This is because those are the only photographs available. As a direct consequence of this, a large amount of fruit has been produced as a result of its production. In order for us to achieve this goal, we plan to begin conducting tests as soon as we are able to start utilizing the newly calculated values for the model's parameters, and we will also investigate whether or not it will be possible to make adjustments to the model's fundamental components. Both of these steps will be taken in order for us to accomplish this objective. In order for us to accomplish what we set out to do, we are going to follow both of these processes.

| **Model name** | **Training accuracy** | **Validation accuracy** |
|---|---|---|
| Custom CNN | 97.58% | 97.35% |
| YOLOv4 | 87.60% | 87.21% |

Table 6.4: CNN vs YOLOv4

# Chapter 7

# Real-life applications

## 7.1 Why did we choose yoloV4 and CNN as object detection models?

The Yolo series is able to execute object detection in a single stage that is both effective and accurate because it makes use of its sophisticated image processing capabilities. YOLO v4 provides a solution to this problem by designing an object detector that can be trained without the use of several GPUs and that has a mini-batch size that is simpler to maintain. This is because training on a single GPU is impossible for the vast majority of cutting-edge models. The reason for this is due to the fact that GPUs have become increasingly powerful. Because these models can only be trained by distributing the training process across multiple GPUs in a large mini-batch size, training on a single GPU is not an option. As a consequence of this, it is feasible to train a rapid and accurate object detector on a single GPU that possesses either a 1080 Ti or a 2080 Ti. Research into artificial intelligence now has many more doors to explore as a result of this. In addition, the version of Yolo known as yolov5 can be obtained by going to the website known as GitHub, which is a repository for computer code. Yolo was developed by Yolo Labs, which gives the company its name. On the other hand, the novelist Alexey Bochkovskiy was at the pinnacle of his creative skills between the hours of midnight and four in the morning, where YOLO stands for midnight and yolov4 stands for four in the morning. Glenn Jocher was one of the authors who contributed to the development of YoloV5 along with the other authors. However, despite the fact that the essay has not yet been scrutinized by academics, it has been developed by other authors. There are some individuals who harbor reservations about whether or not this constitutes an improvement in contrast to yoloV4, and the degree to which they express those reservations varies. As a consequence of this, rather than settling on the idea of utilizing yoloV5, we have arrived at the conclusion that yoloV4 is the path that ought to be followed. This turned out to be the aspect that made the difference.

The state of the art in machine learning places a significant amount of emphasis on highly functional feed-forward neural networks. These networks are an indispensable component of the field of study. They are sometimes referred to as "CNNs," which is an abbreviation for the phrase "convolutional neural networks," which is another common name for them. Additionally, they are sometimes referred to as

"deep learning networks," which is another prevalent moniker. When it comes to the identification and classification of photos, CNNs are of tremendous assistance due to the high degree of precision that they possess. Yann LeCun, a French computer scientist, is credited with being the first person to put forward the idea in the late 1990s. [32] He was the one who came up with the idea. The human ability to recognize patterns and shapes in the environment was the spark that ignited his creative fire. CNN makes use of a hierarchical model that constructs a network in the shape of a funnel before moving on to construct a fully connected layer in which all of the neurons are coupled to one another and the output is processed. This model begins by building a network in the shape of a funnel. The funnel-shaped network is followed by this tier in the hierarchy. After the formation of the network in the shape of a funnel, the subsequent step is the addition of this layer. CNN is able to generate the results that it does as a direct result of the work that it does because of the model that it uses. If it were possible for CNNs to develop the ability to "see" in two dimensions, it would be to their advantage to do so. When it comes to image processing, it is of the utmost importance for the model to have an understanding of the relative sizes of the many data structures that are in play at any one time. This is due to the fact that image processing requires the utilization of a wide variety of data structures. Because of the nature of the exercise, it is essential that you keep this degree of awareness throughout the entire thing.

## 7.2 Disadvantages of these models

There is no such thing as an error-free solution in the world of technology, and computer vision systems are not an exception to the rule that there is no such thing as an error-free solution. Computer vision systems are not an exception to the rule that there is no such thing as an error-free solution. This is due to the fact that there is no such thing as a solution that is completely devoid of errors. This is more or less the circumstance as a consequence of the fact that there is no such thing as a problem-solving method that is free of errors. The demand for continual monitoring is the most major disadvantage; faults or malfunctions in computer vision systems can result in catastrophic financial losses for enterprises. Due to the fact that there are several technologies that are currently available, businesses need to ensure that they have specialist staff in order to be able to monitor and evaluate these technologies. These people need to be able to accomplish it in a timely manner in order for it to work. Even if the two models detect things in very different ways and employ very distinct modes of operation, it is vital to keep in mind that both of these models have particular restrictions that must be taken into account. This is because the two models detect things in very different ways. This is still the case despite the fact that the two models have quite different perspectives on how things are presented to them. Remembering this is something that is essential at all times, so make sure that you do not allow even a moment for it to slip your mind for even a second. Always keeping this in mind is a necessary requirement, so make sure not to forget it.

| **Yolo v4** | **CNN** |
|---|---|
| i) When objects in the image have a typical ratio features, it is difficult to generalize. | i) CNN does not encode the position and orientation of objects. |
| ii) Struggles to detect close objects because each grid can propose only 2 bounding boxes. | ii) Lack of ability to be spatially invariant to the input data. |
| iii) Struggles to detect small objects Moreover One has to label the images manually, so working with a large dataset is near to impossible. | iii) Lots of training data is required and the training time is very long. So if there is any power cut one has to run the same training again from scratch |

Table 7.1: Disadvantages of two models

## 7.3 What can we do with these models?

Computer vision systems are not an exception to the norm that there is no such thing as a solution that is fully devoid of faults in the field of technology; it is impossible for there to be such a thing. Given that there is no answer that can completely eradicate the possibility of making mistakes, this is a legitimate assumption to make. The necessity of performing routine monitoring derives from the fact that even a single bug or malfunction in an organization's computer vision system could result in calamitous financial losses. One of the most significant drawbacks posed by the technology is this one. Businesses need to ensure that they have professionals on staff who are capable of monitoring, evaluating, and keeping up with all of the many technologies that are currently available because there are currently so many distinct technologies. It is necessary to bear in mind that each model has its own unique restrictions that need to be taken into account, even though the two models use different ways of detection and operation. This is because it is essential to remember that each model has its own special constraints. In spite of this, it is essential to bear in mind that the two models utilize different modes of detection and operation in order to function properly. The fact that this is the case persists despite the fact that the two models make use of entirely unique methods of detection. This is something that you definitely must have in mind at all times; you can't even let yourself forget about it for a second. You can't even give yourself permission to forget about it.

## 7.4 How can our country benefit from this work?

It is feasible that the government may come up with a gadget that is known as a pothole detector and model it after our prototype in order to have something to use as a starting point. The results of the count would be compared to those obtained from other regions, and then, depending on the conclusion, the device would decide which roads need maintenance first and in what order. It will be a lot simpler for people who use wheelchairs or who ride motorcycles to get to their destinations if

certain models provide the capability for drivers to avoid taking particular roadways altogether. This is because people who use wheelchairs or who travel on motorcycles are more likely to experience road hazards. When there are fewer potholes on the roads, there will be a lower likelihood for individuals to get injuries while driving on routes that have such hazards as a result of their exposure to those hazards. This is due to the fact that fewer people will drive on roads that have been damaged by potholes. As an immediate consequence of this, there will be a discernible reduction in the overall number of persons who are fatally injured or killed.

## 7.5 How are we portraying the solutions in terms of accuracy?

Mean Average Precision is the metric that yoloV4 uses to evaluate how successful an object identification model is in order to determine whether or not it should be utilized. The goal of this evaluation is to determine whether or not the model should be used. The purpose of this analysis is to assess whether or not the model ought to be utilized in the process. This metric was designed with the goal of evaluating the degree to which an identification model functions in an accurate manner (mAP). The degree to which the bounding box that was generated in advance is equivalent to the box that was identified is one of the factors that is used by the mAP to determine how high of a score to assign. This is done by comparing it to the box that was identified. When there is a greater total score, the model has more information to work with, which results in more accurate detections than when there is a lower overall score. This is in contrast to when there is a lower overall score. The rectangular box that we have already designated as either normal or pothole serves as a representation of the ground-truth bounding box. It is quite uncommon for the actual box that is produced to be an identical duplicate of the box that was visualized. It is more likely that the real box will be comparable to the one that was pictured in your head.

We must determine AP (Average Precision) for each class in order to calculate mAP. The mAP is the average of all class Aps. The equation of mAP is given below. [31]

# Chapter 8

# Conclusion and Future work

This article shows cutting-edge technology for detecting defects in roadways and demonstrates how it can be done. The technology can be found in the accompanying video. Our CNN model has performed incredibly well, and we are pleased with the way that things have transpired as a result. Despite this, we are working to secure our position in the years to come. Detection through the utilization of machine vision and deep learning technologies The process kicks off with the collection of raw data in the form of images of potholes, which is the first step. After that, these photographs go through a process called pre-processing, during which annotations and other information that places the images in their proper context are added. Additionally, we created separate subfolders for pictures and annotations. While all of our training images were saved in a single folder that we simply referred to as "images," the annotations themselves were stored in an entirely different location. The XML file format was applied throughout the process of constructing the annotated files. After that, we put our model through its paces by instructing it in deep learning for a while to see how well it can perform. The process of doing object recognition using deep learning was made possible for us by downloading a pre-trained yolov4 weight file that had been learned on a coco dataset. This particular file had previously undergone training using the dataset. After that, we modified the setting for the train-from-pre-trained model to use the value pre-trained-yolov4, increased the number of tests to 200, and made sure that the version of TensorFlow for the graphics processing unit (GPU) we used was at least 2.8.0. (the weight file we downloaded). After that, we carried out an analysis, and after that, we saved the trained model. We were successful in locating various photos of potholes while using the OpenCV package to do real-time pothole recognition on a streaming video. The footage contained a number of different potholes. A helpful next step could consist of expanding the scope of this project by developing an android application that makes use of Google Map to record the locations of potholes. This would be a step that would be advantageous to take. The software would indicate on the road exactly where potholes are located, making it much simpler for drivers to steer clear of them.

# References


1) 4,200km of roads in bad condition (2019) The Daily Star. Available at: https://www.thedailystar.net/frontpage/news/4200km-roads-bad-or-very-bad-1764895?fbclid=IwAR3YvuXCBYmv15Lo35sb_ytLeqohZcyOCQgDVBpVgOZpLI-6mqYo91zFD8k (Accessed: January 18, 2022).

2) Potholes rule roads in Kalapara town (2021) The Daily Star. Available at: https://www.thedailystar.net/news/bangladesh/news/potholes-rule-roads-kalapara-town-2177631 (Accessed: January 18, 2022).

3)The New Nation (no date) Potholes on roads cause damage to vehicles Commuters face immense suffering, The New Nation. Available at: https://thedailynewnation.com/news/303727/Potholes-on-roads-cause-damage-to-vehicles-Commuters-face-immense-suffering (Accessed: January 18, 2022)

4. Koch, C. et al. (2015) "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," Advanced engineering informatics, 29(2), pp. 196–210. doi: 10.1016/j.aei.2015.01.008.

5) Dhiman, A., Chien, H.-J. and Klette, R. (2017) "Road surface distress detection in disparity space," in 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE.

6) Dhiman, A., Chien, H.-J. and Klette, R. (2018) "A multi-frame stereo vision based road profiling technique for distress analysis," in Proc. ISPAN, pp. 7–14.

7) Koch, C. and Brilakis, I. (2011) "Pothole detection in asphalt pavement images," Advanced engineering informatics, 25(3), pp. 507–515. doi: 10.1016/j.aei.2011.01.002.

8) Buza, E., Omanovic, S. and Huseinovic, A. (2013) "Stereovision techniques in the road pavement evaluation," in Proceedings of the 2nd International Conference On Information Technology and ComputerNetworks, pp. 48–53.

9) Koch, C., Jog, G. M. and Brilakis, I. (2013) "Pothole detection with image processing and spectral clustering," Journal Computing in Civil Engineering, 27(4), pp. 370–378.

10) Pereira, V. et al. (2018) "A deep learning-based approach for road pothole detection in Timor Leste," in 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). IEEE.

11) Mednis, A. et al. (2011) "Real time pothole detection using Android smartphones with accelerometers," in 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS). IEEE.

12) Zhang, A. et al. (2017) "Automated Pixel - Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network," Computer-Aided Civil and Infrastructure Engineering, 00, pp. 1–15.

13) Cha, Y.-J., Choi, W. and Büyüköztürk, O. (2017) "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks: Deep learning-based crack damage detection using CNNs," Computer-aided civil and infrastructure engineering, 32(5), pp. 361–378. doi: 10.1111/mice.12263.

14) Maeda, H. et al. (2018) "Road damage detection using deep neural networks with images captured through a smartphone," arXiv [cs.CV]. Available at: http://arxiv.org/abs/1801.09454.

15) Koch, C. and Brilakis, I. (2011) "Pothole detection in asphalt pavement images," Advanced engineering informatics, 25(3), pp. 507–515. doi: 10.1016/j.aei.2011.01.002.

16) Shaghouri, A. A., Alkhatib, R. and Berjaoui, S. (2021) "Real-time pothole detection using deep learning," arXiv [cs.CV]. Available at: http://arxiv.org/abs/2107.06356.

17) Y. Tsai, A. Chatterjee, "Pothole Detection and Classification Using 3D Technology and Watershed Method", Jouranl of Computing in Civil. Engineering, 2018, 32(2), 04017078, 2018.

18) Maeda H, Sekimoto Y, Seto T, et al. Road damage detection using deep neural networks with images captured through a smartphone[J]. arXiv preprint arXiv:1801.09454, 2018.

19) Chablani, M. (2017) YOLO — You only look once, real time object detection explained, Towards Data Science. Available at: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006 (Accessed: January 18, 2022).

20) Hollemans, M. (no date) Real-time object detection with YOLO, Machinethink.net. Available at: https://machinethink.net/blog/object-detection-with-yolo/ (Accessed: January 18, 2022).

21) Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET). 2017.

22) https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network

23) https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d

24) https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d

25) Tedeschi A, Benedetto F. A real-time automatic pavement crack and pothole recognition system for mobile Android-based devices[J]. Advanced Engineering Informatics, 2017, 32: 11-25.

26) Bochkovskiy, A., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. [online] ResearchGate. Available at: https://www.researchgate.net/publication/340883401_YOLOv4_Optimal_Speed_and_Accuracy_of_Object_Detection¿ [Accessed 19 August 2022].

27) Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. CSPNet: A new backbone that can enhance learning capability of cnn. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop), 2020.

28) Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768, 2018.

29) Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018

30) https://github.com/pjreddie/darknet

31) Gad, A., 2022. Mean Average Precision (mAP) Explained — Paperspace Blog. [online] Paperspace Blog. Available at: ¡https://blog.paperspace.com/mean-average-precision/#: :text=To%20evaluate%20object%20detection%20models,model%20is%20in%20its%20detections. [Accessed 12 August 2022].

32) LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539