Pothole Detection Using Machine Learning Algorithms

BY

Saraban Tasnim Sharin

ID: 172-15-9660

AND

A.K.M. Jobayer Al Masud

ID: 172-15-9641

AND

Khandokar Farhan Tanvir Shawon

ID: 172-15-9731

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Ms. Zakia Zaman

Lecturer (Senior Scale)

Department of CSE

Daffodil International University

Co-Supervised By

Rubaiya Hafiz

Lecturer (Senior Scale)

Department of CSE

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

MAY 2021

APPROVAL

This Project titled "A Pothole Detection Using Machine Learning Algorithms", submitted by Saraban Tasnim Sharin, A.K.M. Jobayer Al Masud and Khandokar Farhan Tanvir Shawon to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 1st June 2021.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Abdus Sattar **Internal Examiner**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

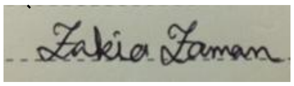
Md. Jueal Mia **Internal Examiner**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Dewan Md. Farid **External Examiner**
Associate Professor
Department of Computer Science and Engineering
United International University

DECLARATION

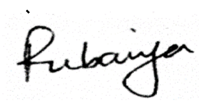
We hereby declare that this project has been done by us under the supervision of **Ms. Zakia Zaman, Lecturer (Senior Scale), Department of CSE,** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



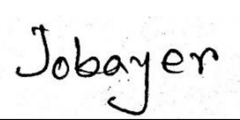
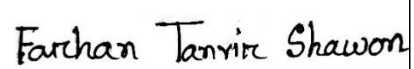
Ms. Zakia Zaman
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Co-Supervised by:



Rubaiya Hafiz
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Submitted by:

		
Saraban Tasnim Sharin ID: 172-15-9660 Department of CSE Daffodil International University	**A.K.M. Jobayer Al Masud** ID: 172-15-9641 Department of CSE Daffodil International University	**Khandokar Farhan Tanvir Shawon** ID: 172-15-9641 Department of CSE Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Ms. Zakia Zaman**, **Lecturer (Senior Scale)**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Machine Learning*" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Professor Dr. Touhid Bhuiyan, the Head**,** Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Potholes are holes on the surface of roads that are horizontally more than 75 mm and by depth more than 20 mm which are created naturally from weak construction, stuck water in rainy season, decay of rocks and also by overload vehicles and sometimes by all of them altogether. Statistics show that in our country in the last 20 years more than 57,000 people lost their lives on roads and a lot more were injured in 58,208 accidents. And a major percentage of these accidents are caused by potholes. The bikers nowadays are in grave danger because of this pothole. A good detection system is necessary to detect the potholes real time and alert the drivers to avoid any inconvenience or casualty. There is some work done on this specific problem, but we propose an approach to detect potholes real time by machine learning which will fulfil the requirements. We have applied some machine learning algorithms on our collected dataset's features and got promising results. And our model will help save a lot of lives by detecting potholes real time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter describes the whole study in brief. The title is Pothole Detection Using Machine Learning Algorithms. In this era Image Processing is a huge field of research under Artificial Intelligence. It is one of the most featured topics in the world as it plays an important role in the world of AI. It helps the world in thousands of ways in different fields such as agriculture, medical science, security, and thousands of other visual identification problems. This chapter consists of Motivation, Rationale of the study, Research questions, Expected output and Report layout. Where motivation discusses the issues on the roads of Bangladesh that are going on and it also describes the necessity of a solution. And the Rationale of the study gives the reason behind the study and about its necessity. Research questions show the list of objectives that this study is going to fulfil and the expected outcome shows the solution against those objectives. And finally, in the report layout chapter we describe the whole report.

1.2 Motivation

As ours is a densely populated country, with the number of people there are a huge number of vehicles and both the numbers are increasing day by day, but the roads are limited. When the limited road faces this many vehicles every day, roads are getting damaged and there creates potholes and other damages which creates numerous accidents. And accidents claim lives of general people and sometimes get people injured and make some of them cripple for life. According to The Daily Star, in the year of 2019, at least 5516 accidents happened, 7,855 people died and 13,330 were injured 13,330. Dhaka Tribune statistics shows that, in last year 2020, the number of road accidents were 5397 and claimed 7317 lives and injured 9021 people. And in the last 20 years a total of 58208 accidents happened and 56,987 people were killed.

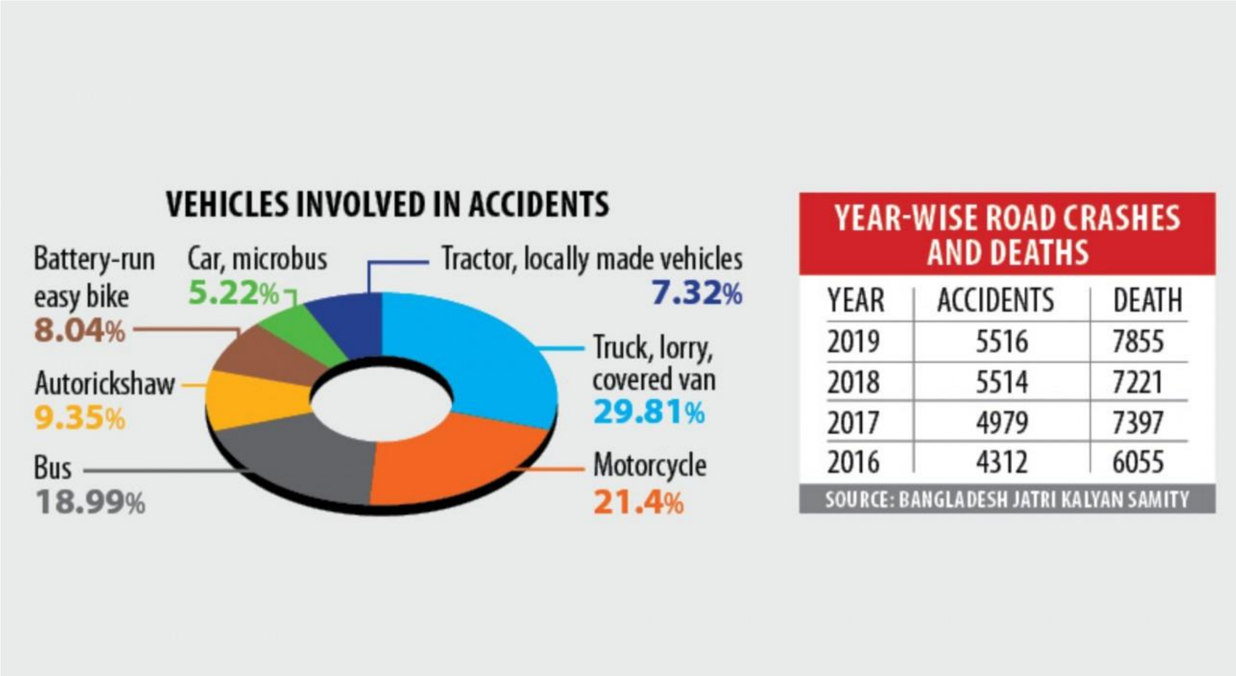


Figure 1.2.1: The statistics of road accidents (2016-2019)

The increasing numbers of accidents every year can be seen from Figure 1.2.1. And a huge percentage of accidents occur from potholes and damaged roads. According to statistics for the last 20 years every day at least 8 people lost their lives on the road. And in 2019 everyday almost 21, and in 2020 almost 18 people died for the same reason. Nowadays the young generation have a huge addiction to motorbikes. The numbers of motorbike users are exploding day by day. Bikes are at great danger because of these potholes. Almost more than 50% of motorbike accidents are caused by potholes. Everyday a lot of motorbikes are falling in accidents and a lot of people are dying. It is almost not possible for a developing country like ours to rebuild or reform all the damages altogether and even if it happens, it won’t take much time to get damaged again because of the exceeded number of vehicles. One of the best solutions is to alert the drivers about the potholes so they can be cautious. It will reduce a huge number of accidents on the road and save a lot of lives.

1.3 Rational of The Study

In 1.2 the motivation part, we have discussed all the problems and limitations of our roads. As in our country there are a huge number of vehicles and a limited number of roads and the increasing numbers of vehicles cannot be controlled. Today's young people are the future of the country. And a huge percentage of these generations are addicted to speed. If we request them to stop speeding on motorbikes, we are pretty sure nobody would listen because we know the stubbornness of the youth. And the general people always need to go out. So, there is no way that we can decrease the number of vehicles on the road. So, another option is mandatory.

What could be the possible best option than to alert the drivers and riders about the upcoming danger. Our main goal is to detect the potholes and show it to the drivers to avoid any accidents. After turning this model into a software or an android application and if it collects all the possible locations of potholes and sends it to the cloud, the data will be saved forever and later these data can be sent to the authorities and they can try to recover the road. In this way a lot of accidents and inconveniences can be stopped using this model.

1.4 Research Questions

While working on this thesis we faced a lot of challenges and problems. Sometimes questions hit the mind and we were confused whether we can complete or not. Some common research questions are,

- How are we going to collect the data?
- How do we process the data and reduce the dimension into proper size?
- Will we be able to classify the dataset?
- Which of the algorithms are going to work well?
- Can our model detect potholes well?

1.5 Expected Outputs

- We worked as a team and worked step by step and did the work successfully.
- We collected the dataset and preprocessed the data.

- Our model classified the data well and all our applied algorithms worked very well and have shown great accuracy at detecting the potholes.

1.6 Report Layout

In chapter two, we discuss the background of our work. We discussed the analysis of potholes, related works, the summary of our research and the challenges we had to face during this whole work. In chapter three, we described the full workflow of our thesis and the instruments we have used. We also discussed how we collected the dataset and resized it. The feature extraction and completion of pre-processing along with classification and software development part is also discussed in this chapter. In chapter four we have shown the results of our experiment and discussed the descriptive analysis of these results. In the final chapter we discussed the whole system summary and also marked out the limitations of our system and the future of our work.

CHAPTER 2

BACKGROUND

2.1 Introduction

This chapter mainly describes the potholes and the background study. It also shows the similar kinds of works that have been done before. This chapter includes pothole analysis, related works, research summary and challenges. Where pothole analysis describes what can be called a pothole and what cannot! The difference between pothole image and normal road image. Related work shows the work that has been done for similar problems. Research summary demonstrates the whole work briefly and challenges shows the problem we had to face during this whole time we are working on this topic.

2.2 Pothole Analysis

Our dataset contains two parts of data. One is pothole image data and other is normal or clean road data. As we previously described about the negative sides of potholes. To describe the potholes, we must know what potholes are and what is not! Potholes are deep holes on the roads that are created naturally by the effect of stuck water on the road and by corrosion of rocks on the road. As in our country in rainy season a lot of rain falls, on the low-level area roads, often water gets stuck for a long time and damages the road surface and later when vehicles start moving around that place the rocks over the road starts to move away and sometimes by erosion, they disappear `and creates potholes. According to Northamptonshire County Council, on strategic roads or highways, greater than 40 mm dimensional holes are defined as potholes and on link roads or local roads more than 50 mm dimensional are defined as potholes. Clean or normal data contains plain and unharmed roads.

Figure 2.2.1: Pothole image sample.

Figure 2.2.2: Normal road image sample.

To make the terms simple, we have added two images where figure 2.2.1 shows the pothole image sample and figure 2.2.2 shows the normal or clean road image sample.

2.3 Related Works

As potholes are a major problem in any country, there is much study and research based on this topic.

2.3.1 ML and AI Algorithms

Four deep learning models are trained by P. Ping et al. to examine the best result. They have used Yolo V3 which gave them the best accuracy which was 82%. After applying SSD, they have acquired 80% accuracy which was also good and 74% accuracy by F-RCNN. But after getting the almost perfect accuracy from these three models, HOG gave them nothing after applying on 1500 images.

2.3.2 Deep Learning Models

A. Kumar et al. collected 1500 images and made a dataset and they have applied deep learning models on them such as Inception-V2, F-RCNN, and Transfer learning. While detecting images and video in real-time it gives the best accuracy.

2.3.3 Kirchhoff's Theory

By using Kirchhoff's theory G. B. R. et al. detected real-time potholes but the theory had some limitations, to overcome this they applied CNN-DL. They send the data to the control room by detecting the potholes location using GPS. CNN-DL gave 99.2%, KNN 95.4% and Kirchhoff's method gave 89.3% accuracy.

2.3.4 Depth Based Pothole Detection Technique

E. J. Reddy et al. calculated depth as a base with the ultrasonic sensor to detect the potholes and after detecting the location, a mail is sent. This technique doesn't require high computation. That's why it can be a great method to find the potholes. Node MCU, GPS module, Ultrasonic sensor, trilateration method which is used for distance calculation, server database for this technique. In this system for development of programming the microprocessor, software system design, program techniques, and system approaches are also used. The techniques which is used to detect the distance is,

Distance = time taken * (speed of ultrasonic wave /2)

And to calculate the depth,

Depth = Present distance – Ground clearance

They were able to find out if there is a pothole by accumulating the depth of the potholes because if there the depth of the potholes is zero then there is no pothole but if the value of the depth is bigger than zero then there must be a pothole and the location can be found by the GPS and sends to the microcontroller.

2.3.5 Method to Avoid Potholes by Warnings

S. Hegde et al. proposed a method that gives earlier alerts to avoid the potholes to the driver. This idea can be used in real life by making a robot vehicle that detects the potholes and alerts the vehicles nearby. To detect the object from 0 to 254 inches Ultrasonic Sensor (LV-MaxSonar-EZ0) is used by them and from 6 inches out to 254 gives sonar range information. They have also used Zigbee (F-20) for sending data and microcontroller (MBED) for receiving data. For input decoding to the higher voltage Motor Driver - L293d was used. A threshold value was selected to use it in the detection and 3670 threshold value was set for the trail version. By using artificial potholes, the trials were made. For the artificial potholes, the technique was great but when they have used it for real potholes the threshold value could not be generated by them.

2.4 Research Summary

In this work we are going to detect potholes by images. So, we collected the images and resized them correctly. And used the pre-trained mobilenetV2 model to extract features and completed the training part and increased the visualization. We also did the dimensionality reduction of our features and made it ready for applying classifier algorithms. Preprocessing part is now complete and we applied machine learning algorithms and got amazing accuracy from them. And we also saved the whole model for further implementation of any software or mobile applications. And finally, our work process is completed successfully.

2.5 Challenges

Behind every successful work there are some challenges and problems. While doing all this work, we also faced some problems that we overcame successfully. While collecting data on the road we faced some problems. Mounting a camera in front of a vehicle in Dhaka city is a nightmare. In the traffic visualization is problematic cause every time most of the places there is jam and always a vehicle in front of our vehicle and the roads are too busy to go and collect them manually. Somehow, we collected the data and while we were extracting features, some images were problematic because of their format and we had to reformat them from the dataset manually and do that part. These are the main challenges we had to face; others are small and they didn't take much of our time to solve.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Introduction

This part is called the proposed methodology and we are going to discuss the methodology that we are proposing for this model. This chapter consists of a total of 6 parts which are Workflow, Pre-Processing, Feature Extraction, Learning and Classification, Pothole Detection, and Software Development. Where the workflow part briefly demonstrates about the process of our total work sequentially. And the pre-processing part describes the work we have done before applying an actual classifier on it and how we have made the dataset prepared for classification. Feature extraction part describes the process of feature extraction. Learning and classification part contains the process of how we have done the classification. The Pothole detection part demonstrates the detection part of our work and finally the last part describes if there will be any software development for our model or not.

3.2 Workflow

For the whole research we have worked in a sequence. After we have started the research, we collected the images and preprocessed them. After the preprocessing part is completed, we extracted the features successfully and trained the features. Then we reduced the dimension of features in a way that no information is lost from the features and also increased the visualization. We then applied the classification algorithms. Successfully, classification and detection were completed.

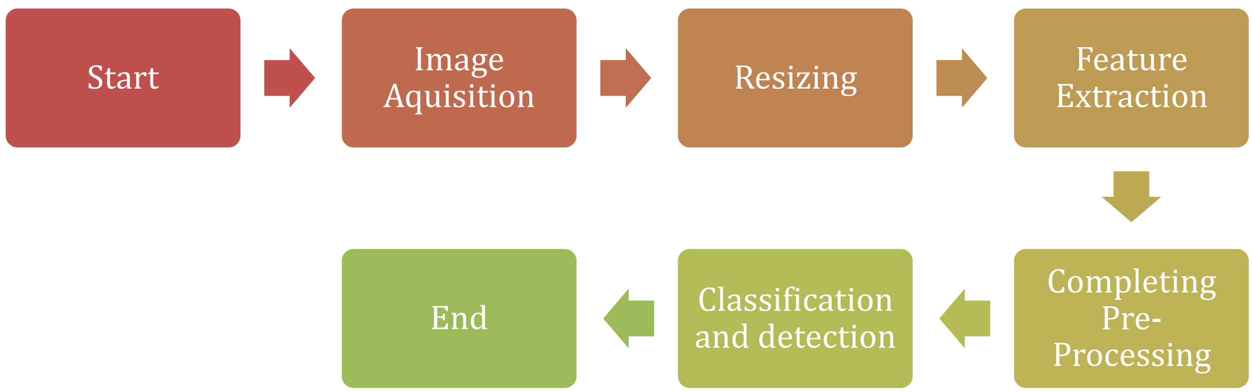


Figure 3.2.1: Workflow process of this research

3.3 Research Instrumentation:

We have used some tools, and also installed libraries to make this research complete. All the used materials are given below.

3.3.1: Tools we have used:

1. Google Colaboratory
2. Python Programming Language

3.3.2 Libraries we have installed:

We have imported a lot of libraries to make this process complete. All of them are given in the figure 3.3.1 and 3.3.2 which are given below,

```
import PIL.Image as Image
import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
import glob
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import NearestNeighbors
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.externals import joblib

import os
import librosa
import itertools
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import kurtosis
from scipy.stats import skew

import sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectKBest
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
```

Figure 3.3.1: Installed libraries

```
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

from sklearn.feature_selection import VarianceThreshold
from sklearn.feature_selection import SelectFromModel

import lightgbm as lgbm
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

Figure 3.3.2: Installed Libraries

3.4 Dataset Collection and Resizing

Our dataset contains two parts. One is normal road data which are normal roads and the other is pothole image data. The normal road image data is more than 350. And the pothole image data is more than 1600, from which we have collected about 400 data ourselves. But the traffic on the roads in Bangladesh and every intercity highway are too busy to collect data. At first, we thought about mounting a camera in a vehicle but the processing part seemed problematic. We had to collect images from videos. And in the heavy traffic, a lot of time would be lost and the visibility of the road would be compromised as well.

After the dataset collection was done, we went to resize all the images. And resized them in such a way that no information is lost and it would be easy for extracting features. The two figures 3.4.1 and 3.4.2 given below show the pre-resized image and post-resized image. First, we applied one image and checked if the system was ok or not, it was working well and then applied all of them.

```
[ ] # Load image
    img = Image.open('/content/drive/MyDrive/train/pot.131.jpg')
    img_batch = np.array(img, dtype=np.float32)[np.newaxis, :, :, :]/255
    print('Batch size:', img_batch.shape) # (1, 256, 256, 3)
    img

    Batch size: (1, 294, 320, 3)
```

Figure 3.4.1: Pre-Resized image sample

```
[ ] img_resized = img.resize([224, 224], resample=Image.BILINEAR)
    img_batch_resized = np.array(img_resized, dtype=np.float32)[np.newaxis, :, :, :]/255
    print('Batch size:', img_batch_resized.shape) # (1, 224, 224, 3)
    img_resized

    Batch size: (1, 224, 224, 3)
```

Figure 3.4.2: Post-Resized image sample

3.5 Feature Extraction

After the resizing process we removed the output layer from the images and extracted features for one image and then made a graph that shows the high-level feature distribution for that image which is given in figure 3.5.1. And when it is done, we extract features for the full dataset altogether and put them into a csv file. To extract the features, we used the pre-trained MobilenetV2 model which is a great model for feature extraction.

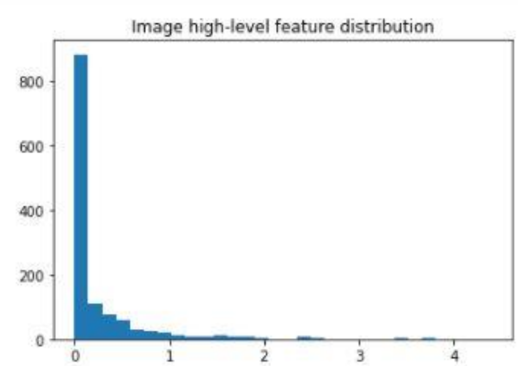


Figure 3.5.1: High-level feature distribution

3.6 Completing the Pre-Processing

After we have extracted the features, we had to apply classifiers. But to get better performance we applied PCA, LDA and t-SNE. Where,

Principal Component Analysis (PCA) is a technique that is utilized to decrease the dimensions of datasets. Howley et.al. in their paper described PCA in the case of classification. It is often hazardous to work with enormous datasets in case of classification. PCA primarily lessens the factors of a dataset bargaining as little data as could be expected. It works in different steps. The first one is standardization or normalization. Normalization is utilized to ensure that all factors with both high and low-scale go to the examination similarly. The next step is covariance Matrix computation. It is a table that sums up every one of the potential associations between variables. Now comes the computation of eigenvalues and eigenvectors and then sorting them and developing the principle matrix. In our case, we applied PCA on our features, and after embedding it on a figure we got the result given below.

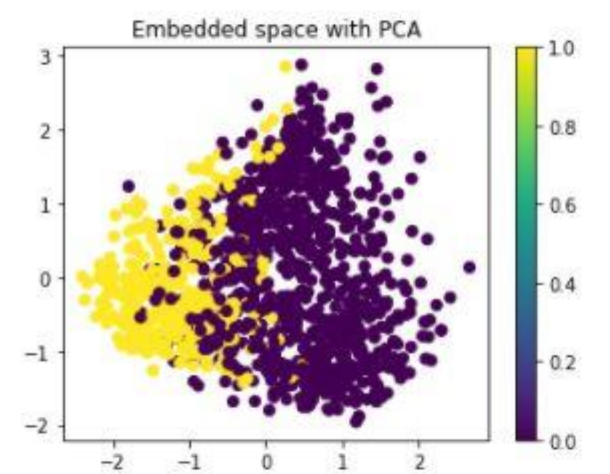


Figure 3.6.1: PCA analysis result

Linear Discriminant Analysis (LDA) is a novel strategy to diminish dimension. It is mostly utilized for solving the issues of classifications. It helps really great at showing the dissimilarities in various groups of datasets. To transform a big dimensional feature into a low dimensional feature without losing much information, it does its job. After applying LDA on the features of our dataset and embedding the result on a figure, we got-

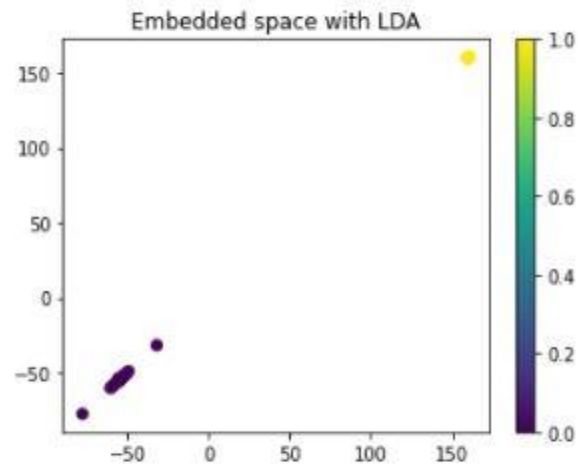


Figure 3.6.2: LDA analysis result

As high-dimensional data is troublesome to envision, to solve the problem t-Distributed Stochastic Neighbor Embedding(t-SNE) is utilized. It actually reduces the dimension of data and makes it convenient to visualize. This function generator for space of high-dimensional original data. So, we can visualize the main theme of the data. Applying it to our feature we got,

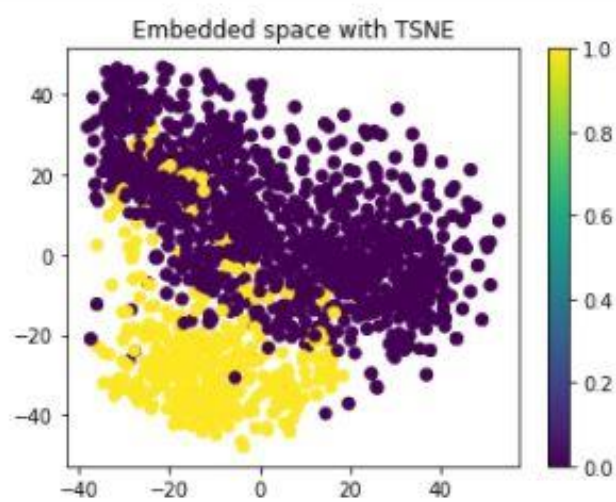


Figure 3.6.3: t-SNE analysis result

3.7 Classification and Detection

We completed all the preprocessing parts and the features are now more than ready to go to the next phase which is classification algorithms. We have applied five different algorithms of machine learning and all of them did a very good job at classifying and detecting the potholes. We are going to elaborate each of them down below.

3.7.1 Logistic Regression

A supervised machine learning algorithm which is an arrangement calculation algorithm used to appoint perceptions to a discrete set of classes. It shows great results at classifying our features. It's true-positive prediction accuracy is 99% and true negative prediction accuracy is 93% which is great. And this much accuracy means our pre-processing and all the work is done well. And it also shows the test set accuracy 97.14% and the validation accuracy 97.03% which is quite a satisfying result. Figure 3.7.1 shows the test set and validation accuracy.

```
[ ]  preds = grid_lr.predict(X_test)
     print("best score on validation set (accuracy) = {:.4f}".format(grid_lr.best_score_))
     print("best score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))

     best score on validation set (accuracy) = 0.9703
     best score on test set (accuracy) = 0.9714
```

Figure 3.7.1: Logistic regression Accuracy

3.7.2 Elastic Net

This machine learning algorithm did also very well. This gives us the accuracy of true-positive prediction is also 99%. But it was a bit more confusing for the true negative prediction. Though it predicted 89% correctly, which is not bad but a bit less than the logistic regression algorithm. It did very well at the detection where it gave test set accuracy 96.51% and validation accuracy 97.03%, which is not bad at all. Figure 3.7.2 shows the accuracy of this algorithm from our features,

```
[ ] preds = grid_en.predict(X_test)
    print("best score on validation set (accuracy) = {:.4f}".format(grid_en.best_score_))
    print("best score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))

    best score on validation set (accuracy) = 0.9703
    best score on test set (accuracy) = 0.9651
```

Figure 3.7.2: Elastic Net algorithm result

3.7.3 Decision Tree

Decision Trees are directed machine learning where the information is persistently parted by a specific boundary. The tree can be clarified by two substances, in particular decision nodes, and leaves. This is one of the most used algorithms in machine learning. Although this algorithm gave us the least accuracy. Somehow it didn't work well for our system. It gave us the true positive accuracy of 94% and true negative accuracy of 76% which is not acceptable, where others were doing much better. It gave us the test set accuracy of 89.52% and validation accuracy of 91.72%. It is a bit disappointing for us. Figure 3.7.3 shows the test and validation accuracy.

```
[ ] preds = grid_cart.predict(X_test)
    print("best score on validation set (accuracy) = {:.4f}".format(grid_cart.best_score_))
    print("best score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))

    best score on validation set (accuracy) = 0.9172
    best score on test set (accuracy) = 0.8952
```

Figure 3.7.3: Decision Tree algorithm accuracy

3.7.4 Random Forest

This algorithm is a bootstrapped collection of many decision trees. It is a commonly used algorithm. It gave us a true positive prediction accuracy of 100% which is an amazing result and true negative of 85%, it is a bit less expected result. In detecting potholes this algorithm gives us 95.87% accuracy for the test set and 95.96% for validation accuracy. Figure 3.7.4 shows the test and validation accuracy of our features from our model.

```
[ ]  preds = grid_rf.predict(X_test)
     print("best score on validation set (accuracy) = {:.4f}".format(grid_rf.best_score_))
     print("best score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))

     best score on validation set (accuracy) = 0.9596
     best score on test set (accuracy) = 0.9587
```

Figure 3.7.4: Random Forest algorithm accuracy

3.7.5 SVM

Support vector machine, a supervised algorithm of machine learning is one of the most popular. It does a great job at classifying and detecting. In our features it's prediction accuracy of the true positive is 99% and true negative is 94%, which is better than all the previous ones. It gave us 97.46% accuracy for test sets and 97.45% for validation accuracy. Which also is better than all the previous ones. The logistic regression is just a little behind. Figure 3.7.5 shows the test and validation accuracy from our model.

```
[ ]  preds = grid_svm.predict(X_test)
     print("best score on validation set (accuracy) = {:.4f}".format(grid_svm.best_score_))
     print("best score on test set (accuracy) = {:.4f}".format(accuracy_score(y_test, preds)))

     best score on validation set (accuracy) = 0.9745
     best score on test set (accuracy) = 0.9746
```

Figure 3.7.5: SVM algorithm accuracy

3.8 Software Development

After all the work is completed, we save this model for the future. We are going to use this model and develop software and mobile applications (android and iOS). This software can be implemented on hardware that can be mounted on a vehicle and work the same as the software just the drivers won't be needing a mobile device or anything like that.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

The name of this chapter is experimental results and discussion. In this part, we will be describing the results of our research and describing it. This chapter consists of three main parts: Experimental Results, Descriptive Analysis. The experimental results describe all the models and algorithms and their results and accuracy. Descriptive analysis describes each model and algorithms that we have worked with and detailed review on them. Finally, the summary part summarizes this whole chapter in brief and gives a final decision.

4.2 Experimental Results

We have used some algorithms of machine learning on the features and acquired results from them. The main machine learning algorithms are Logistic Regression, Elastic Net, Decision Tree, SVM and Random Forest. In the table below all the results and accuracies are given sequentially,

Algorithm	Validation Accuracy	Test set Accuracy
Logistic Regression	97.03%	97.14%
Elastic Net	97.03%	96.51%
Decision Tree	91.72%	89.52%
Random Forest	95.96%	95.87%
SVM	97.45%	97.46%

Table 4.2.1: The result of applied algorithms

Among all the applied algorithms we got the best result from Logistic Regression, Elastic Net, and SVM, which is more than 97%. And the others haven’t disappointed us at all. Though the decision tree algorithm gave us 91% of accuracy, others are promising. Now, we are going to check the values of precision, recall and f1 score for all the algorithms.

Algorithm	**Data**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	Clean	0.97	0.99	0.98
	Pothole	0.96	0.93	0.94
Elastic Net	Clean	0.96	0.99	0.98
	Pothole	0.97	0.89	0.93
Decision Tree	Clean	0.92	0.94	0.93
	Pothole	0.81	0.76	0.79
Random Forest	Clean	0.95	1.00	0.97
	Pothole	0.99	0.85	0.91
SVM	Clean	0.98	0.99	0.98
	Pothole	0.96	0.94	0.95

Table 4.2.2: Precision, Recall and F1 Score results for all algorithms

4.3 Descriptive Analysis

Among all the algorithms that we have applied, we can make a decision about the one that is best for our features, after doing a deep analysis. We must do the analysis in terms of something that we have applied on our system. As we have extracted confusion matrix, precision, recall and f1 score for each of the algorithms, we will choose the ones that work best for us by comparing each of their confusion matrix, precision, recall and f1 score. Now, we are going to describe them and compare them accordingly.

4.3.1 Confusion Matrix

Confusion matrix is a matrix in the form of a table that shows us the true positive or a true negative value. It is mainly used to demonstrate the accuracy of a classifier to know the true value when the classifier is run on a test dataset. Let's give an example of a confusion matrix. Suppose an experiment results in a confusion matrix where the system is detecting male or female.

	Male	Female
Male	0.70(TP)	0.30(FP)
Female	0.20(FN)	0.80(TN)

Table 4.3.1: Example of Confusion Matrix

In table 4.3.1 we can see in the male row, at first, there is a True-Positive value of 70%. Which means the algorithm has predicted 70% males who are truly male and predicted 30% of true male as female in the False-Positive section. So, the system is confusing for about 30% of male. And again, for the female row in the True-Negative section, the system predicted 80% of females who are truly female and in the False-Negative section it falsely detected 20% of females as male. In this case, the system confused about 20% of females. So, we can call a system best for something where the confusion will be less than others. We hope this example describes the confusion matrix well.

4.3.2 Precision, Recall, F1-Score

To check the quality of an algorithm on a dataset, precision measurement is used. If precision value for an algorithm is good, that means the algorithm returns more applicable results than inapplicable. And recall measures the quantity of the returned applicable results. High recall value means whether the irrelevant results are returned or not the most relevant results are returned. The F1 score measures the classification ability of an algorithm. Good f1 score value means a very low amount of false-positive and false-negative value. For all of these, 1 is considered the perfect result for an algorithm and 0 is a failure. We are going to check all of these for each algorithm and compare them accordingly.

4.3.2 Logistic Regression

Logistic regression is a well-known algorithm in machine learning. It is an arrangement calculation algorithm used to appoint perceptions to a discrete set of classes. This algorithm works well on our system where it gives 97% of accuracy. That means it predicted very well. But we have to test it on the terms of the confusion matrix. The figure 4.3.2.1 shows the confusion matrix of this algorithm where the algorithm predicts the real normal data 99% correctly. And it is confused for only 1% of the normal data and wrongly predicts them as potholes. And for the pothole data the algorithm detects 88% of real pothole data as potholes and it gets confused about 12% of the pothole data and wrongly predicts them as normal data which is average so far. And we must admit that this algorithm predicts well overall.

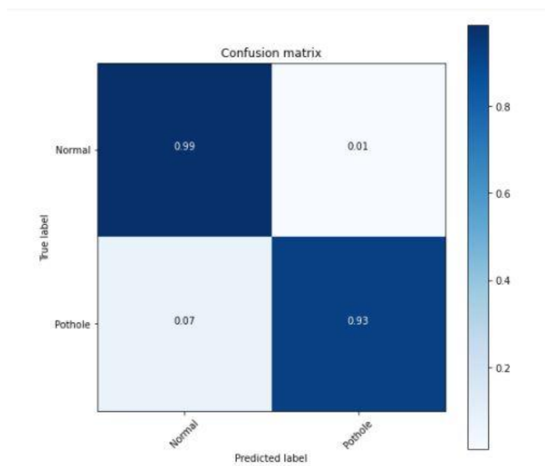


Figure 4.3.2.1: Confusion matrix of Logistic Regression algorithm

For this algorithm precision result is 0.97 and 0.96 for clean and pothole roads. So, it is a great score for the algorithm. The result quality is amazing. And we can see the recall score of 0.98 and 0.94 in the figure 4.3.2.2 as well. This score shows the quantity of relevant results that the algorithm returned but somewhere it lagged a bit for the pothole data but we have to admit its accuracy. And the f1 score 0.98 and 0.94 is also very promising. Which shows the perfectness of classification of this algorithm.

	precision	recall	f1-score	support
0	0.97	0.99	0.98	235
1	0.96	0.93	0.94	80
accuracy			0.97	315
macro avg	0.97	0.96	0.96	315
weighted avg	0.97	0.97	0.97	315

Figure 4.3.2.2: Precision, Recall and F1 Score result for Logistic Regression

4.3.3 Elastic Net

This is also a great algorithm and it has shown us a great accuracy of 97% of accuracy which is a very promising result. But we cannot judge only by accuracy. Let's check it through the confusion matrix. As in the figure 4.3.3.1 which is given below shows us the true positive value of 99% which means the algorithm predicts the real normal road data 99% correctly. And it gets confused and wrongly predicts 1% of normal road data as potholes. Which is the same as logistic regression. But now let’s check its true negative values. It predicts 89% of real potholes as potholes correctly and gets confused for 11% of data and wrongly predicted as normal road data. Which is a little bit disappointing and this result is not as accurate as logistic regression but it is also promising.

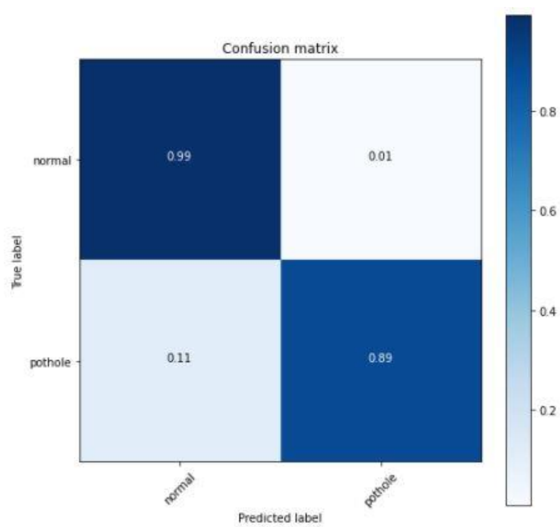


Figure 4.3.3.1: Confusion matrix of Elastic Net algorithm

In the figure 4.3.3.2 we can see the precision, recall and f1 score results sequentially for clean road data and pothole data. Precision is 0.96 and 0.97, showing the high quality of the results which is quite a great result. And recall value is 0.99 and 0.89 which is a great quantity of relevant results for clean roads and not that good for pothole data. The f1 score shows great results of 0.98 and 0.93 which shows the perfect classification accuracy.

	precision	recall	f1-score	support
0	0.96	0.99	0.98	235
1	0.97	0.89	0.93	80
accuracy			0.97	315
macro avg	0.97	0.94	0.95	315
weighted avg	0.97	0.97	0.96	315

Figure 4.3.3.2: Precision, Recall and F1 Score result for Elastic Net algorithm

4.3.4 Decision Tree

This is a very well-known algorithm in machine learning, and it is used enormously by researchers all over the world. But this one gives us the least accuracy in detecting which is 91.72%. For the confusion matrix which is in the figure 4.3.4.1 given below, it shows 94% of true positive results which means it predicts 94% of normal data which are actually normal and becomes confused about 6% data which are actually normal but it predicts them as potholes. And again, it predicts 76% of true negative data which are potholes and it predicts them as potholes. But it wrongly predicts 24% of the pothole data as normal data. Which is a bit disappointing.

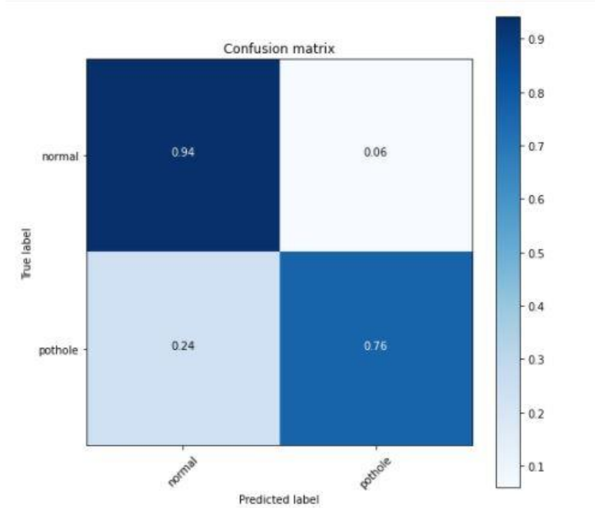


Figure 4.3.4.1: Confusion matrix of Decision Tree algorithm

As for the precision recall and f1 score, the figure 4.3.4.2 shows the results. Precision value is 0.92 and 0.81 which is less than the previous ones by quality of relevant results. For recall the value is 0.94 and 0.76 which shows the quantity of relevant results which can be called average. And the f1 score result is also average where it shows 0.93 and 0.79. The classification accuracy of this algorithm is not perfect like the others. This algorithm is not perfect by any means for our dataset.

	precision	recall	f1-score	support
0	0.92	0.94	0.93	235
1	0.81	0.76	0.79	80
accuracy			0.90	315
macro avg	0.87	0.85	0.86	315
weighted avg	0.89	0.90	0.89	315

Figure 4.3.4.2: Precision, Recall and F1 Score result for Decision Tree algorithm

4.3.5 Random Forest

This algorithm is a bootstrapped collection of many decision trees. It is a commonly used algorithm. In detecting potholes, this algorithm gives us almost 96% of accuracy. In the figure 4.3.5.1 which is given below, we see that it predicts 100% of true positive data which is normal data. That means it detects all the normal road data correctly. But for pothole data it predicts 85%

correctly and gets confused about 15% of pothole data and predicts them wrong. We can see, it did good at predicting.

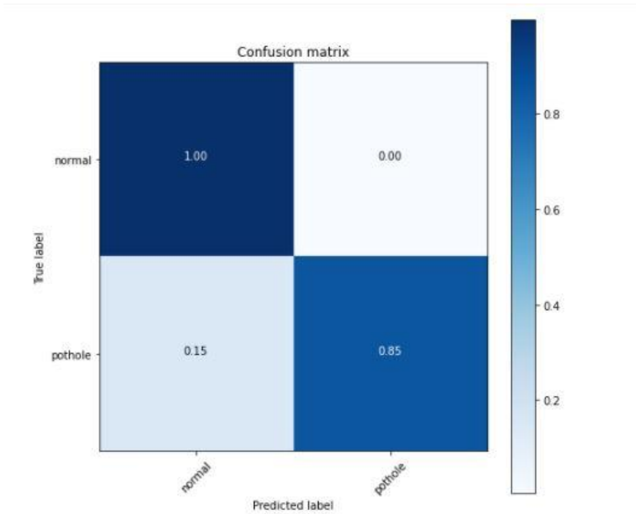


Figure 4.3.5.1: Confusion matrix of Random Forest algorithm.

In the given figure 4.3.5.2, the precision value for this algorithm is 0.95 and 0.99 which is great and this shows the good quality of the relevant results that the algorithm returned. The recall value shows 1.00 and 0.85 which cannot be any better for clean roads and for potholes it is a bit disappointing quantity of relevant results. The f1 score shows 0.97 and 0.91 and it shows the perfect classification of clean road data but for the pothole data it is not that perfect compared to others.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	235
1	0.99	0.85	0.91	80
accuracy			0.96	315
macro avg	0.97	0.92	0.94	315
weighted avg	0.96	0.96	0.96	315

Figure 4.3.5.2: Precision, Recall and F1 Score result for Random Forest algorithm

4.3.6 SVM (Support Vector Machine)

It is a supervised learning algorithm which is one of the most popular algorithms among all in machine learning. It is mostly used for classification in machine learning. It gave us the highest accuracy rate which is 97.14% for the test set and 97.24% for validation. If we look at the figure 4.3.5 which is given below, it shows the prediction rate of 99% of normal data which are truly normal, and that is great. For the pothole data it predicts 91% correctly about the true potholes. And gets confused for 9% of the pothole image and predicts them as normal data. Which is a huge achievement. This is the best result so far.

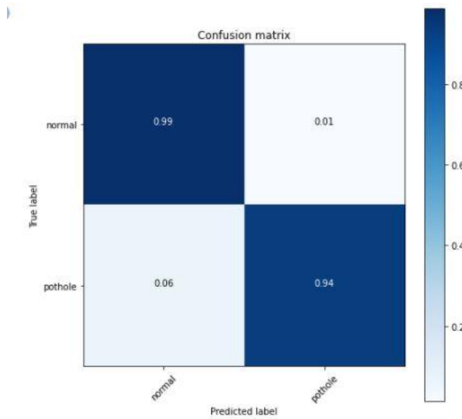


Figure 4.3.6.1: Confusion matrix of SVM algorithm

In figure 4.3.6.2 we can see the result of precision, recall and f1 score for the SVM algorithm where it shows 0.98 and 0.96 value for precision. It is the best quality of the result so far. And the recall is 0.99 and 0.94 which shows the quantity of relevant results. F1 score values are 0.98 and 0.95 which is also an amazing result. All three of the results are best among all the algorithms we have applied so far. So, SVM is the algorithm that is best for our dataset.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	235
1	0.96	0.94	0.95	80
accuracy			0.97	315
macro avg	0.97	0.96	0.97	315
weighted avg	0.97	0.97	0.97	315

Figure 4.3.6.2: Precision, Recall and F1 Score result for SVM algorithm

CHAPTER 5

CONCLUSION

5.1 Introduction

Our system works and fulfilled its goal successfully which was to detect the potholes accordingly. The images we have chosen for this system, there were two classes. One is pothole image data, and another is normal or clean road data. Our extracted features worked very well for the models of machine learning and the potholes were detected successfully by those models. Before extracting features of the dataset, we resize the image altogether. Then we started training the features and after training we applied different algorithms to reduce the dimension of the dataset, reduced the features in such a manner that no information was lost from the features, and we also increased the visualization. By doing so, the features were ready and it would be easy for the machine learning algorithms to work properly. We applied a lot of algorithms which are Logistic Regression, Elastic Net, Decision Tree, SVM and Random Forest. Almost all of them gave amazing results. The best result was given by SVM and Logistic regression, which are more than 97%. And all the others gave more than 95% except for Decision tree which gave the lowest that is also more than 91%. This result proves that our classification and feature extraction was perfect. This accuracy can be better in future by adding more images in the dataset. And after the implementation will be completed, the system will be easy for overall use.

Finally, our model succeeded in detecting the potholes and this system could be a perfect prototype for the road authority of Bangladesh and by giving order to every vehicle to use this, and this will reduce the accident rate and can save a lot of lives and other harmful occurrences.

5.2 System Limitations

Our system is very effective and helpful for general people and we have no doubt that it will help people of all sectors who put their lives in risk by travelling around by roads. It successfully reaches our main goal, which is to detect potholes accurately. Though the system has some limitations.

To make our system usable, every vehicle will have to mount a camera in front of their vehicle. This system is not implemented on any software or android applications yet. No GPS system has been added to specify the exact location of the potholes. We have not added any dataset that was collected at night. So, we are really worried about this system's accuracy during the night-time. The system cannot send data to the cloud with the location of potholes, to give the information to the authority. Even our system is not connected to the map so other drivers who will go through a road can understand how many potholes are there in the road and if they will choose the road for traveling or not.

5.3 Future Work

Our main goal of this thesis was to detect the potholes accurately. And our model has done its part. But we will work hard on this topic further and take this work to a new level. As we have discussed in the limitation part about the things that are limited in our system. We are going to improve and develop them. We will improve the accuracy by applying more different algorithms. As our model lacks the ability to detect potholes at night, we are going to add a GPS system with a model that will store and send the location with the image of the pothole for uploading it to the server. The app that we are going to develop will receive the data and further that data will be sent to the app of the person who is currently using the app on that specific road and also it will help at night and alert the drivers about upcoming danger.

We are going to develop both android and iOS applications, which will add a whole new chapter to this system. As the vehicles will have cameras mounted on the vehicle front, the application will be connected to the camera. But after the server system with map is active not all vehicles will need to have a camera cause the app will let them know about the exact locations. And the bikers and old vehicles will never have to worry in days as well as at night-time. There is also a possibility that we are going to implement this system in a hardware that will be implemented on the vehicle with the camera and will beep while the camera detects potholes on the road. And it will also have GPS and it will do the same work as that app. And the dash monitor of the car will be added with it for giving an alert on those specific pothole locations.

REFERENCE

1. "21 died on roads every day," *The Daily Star*. January 12, 2020. Accessed on: April 26th, 2021. [Online]. Available: https://www.thedailystar.net/backpage/road-accident-in-bangladeh-21-died-every-day-1852867

2. "18 people killed every day on average in road accidents in 2020," *Dhaka Tribune*. January 9th, 2021. Accessed on: April 26th 2021. [Online]. Available: https://www.dhakatribune.com/bangladesh/2021/01/09/report-18-people-killed-every-day-on-average-in-road-accidents-in-2020#:~:text=Op%2DEd-,Report%3A%2018%20people%20killed%20every%20day%20on,in%20road%20accidents%20in%202020&text=As%20many%20as%206%2C686%20people,accidents%20in%202020%20in%20Bangladesh.&text=Cumulatively%2C%20a%20total%20of%205%2C397,and%20left%209%2C021%20others%20injured

3. *"CATEGORIZATION OF POTHOLES," Northamtonshire County Council*. Accessed on: April 27, 2021. [Online]. Available: https://www-uat.northamptonshire.gov.uk/councilservices/northamptonshire-highways/roads-and-streets/Documents/Categorisation%20of%20Potholes.pdf

4. T. Howley, M.G. Madden, ML O'Connell and A.G. Ryder. "The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data," *Macintosh A., Ellis R., Allen T. (eds) Applications and Innovations in Intelligent Systems XIII*. SGAI 2005. Springer, London.

5. A. Kumar, Chakrapani, D. J. Kalita and V. P. Singh, "A Modern Pothole Detection technique using Deep Learning," *International Conference on Data, Engineering and Applications (IDEA)*, Bhopal, India, 2020, pp. 1-5.

6. Jetashri R. Gandhi, U. K. Jaliya, D. G. Thakore, "A Review Paper on Pothole Detection Methods," *International Journal of Computer Sciences and Engineering*, Vol.7, Issue.2, pp.379-383, 2019.

7. G. B. R., C. C., S. B. Rao M., S. M., K. E. and S. J., "Deep Learning Based Pothole Detection and Reporting System," *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, 2020, pp. 1-6.

8. J. Lekshmipathy, S. Velayudhan and S. Mathew, “Effect of combining algorithms in smartphone-based pothole detection," *International Journal of Pavement Research and Technology*. 14, 63–72 (2021).

9. A. Maredia, A. Mishra, R. Mishra, M. P. Janrao.” Machine Learning for Automated Pothole Detection,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887Volume 7IssueIV, Apr 2019

10. P. Ping, X. Yang and Z. Gao, "A Deep Learning Approach for Street Pothole Detection," *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2020, pp. 198-204.

11. J. D. Crisman and C. E. Thorpe, "UNSCARF-a color vision system for the detection of unstructured roads," In Proceedings of. *1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2496-2501 vol.3.

12. H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama and H. Omata, “Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images: Road damage detection and classification,” *Computer-Aided Civil and Infrastructure Engineering 33*. June 2018.

13. S. Hegde, H. V. Mekali and G. Varaprasad, "Pothole detection and inter vehicular communication," *IEEE International Conference on Vehicular Electronics and Safety*, 2014, pp. 84-87.

14. K. Pawar, S. Jagtap and S. Bhoir. “Efficient pothole detection using smartphone sensors,” *ITM Web of Conferences*, January 2020, 32:03013

15. E. J. Reddy, P. N. Reddy, G. Maithreyi, M. B. C. Balaji, S. K. Dash and K. A. Kumari, "Development and Analysis of Pothole detection and Alert based on NodeMCU," *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1-5.

16. N. Van Khang and É. Renault, "Cooperative Sensing and Analysis for a Smart Pothole Detection," *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 1785-1790.

Turnitin Originality Report

Processed on: 13-Jun-2021 15:55 +06
ID: 1605517868
Word Count: 8536
Submitted: 1

172-15-9660 By Saraban Tasnim Sharin

Similarity Index	Similarity by Source	
12%	Internet Sources:	9%
	Publications:	5%
	Student Papers:	9%