**Abstract**

Road maintenance and safety are critical concerns in transportation infrastructure management. Potholes, often caused by wear and tear or adverse weather conditions, pose a significant threat to road safety and can lead to costly repairs and accidents. Traditional methods of pothole detection are time-consuming and often rely on visual inspections, which may not be efficient or accurate. This seminar project explores the use of LiDAR (Light Detection and Ranging) technology for automated pothole detection, aiming to improve road maintenance and reduce safety risks .

### Chapter 1

### 

### INTRODUCTION

**1.1Introduction**

The number of vehicles drastically increases every year, and the number of accidents proportionally does too. The condition of road surface affects directly on our safety .The American Automobile Association estimated in the five years prior to 2016 that16 million drivers in the United States had suffered damage from potholes to their vehicle with a cost of 3 billion USD a year. In India, 3,000 people per year are killed in accidents involving potholes. Britain has estimated that the cost of fixing all roads with potholes in the country would cost 12 billion EURO . According to the World Health Organization, road traffic injuries caused an estimated 1.25 million deaths worldwide in the year 2010. That is, one person is killed every 25 seconds. Only 28 countries, representing 449 million people (seven percent of the world's population), have adequate laws that address all five risk factors (speed ,drunk driving, helmets, seat-belts and child restraints).By the way, there is close relationship between the accident and road condition including a pothole. Road accidents occur as the result of one, or more than one of the following factors: human factors, vehicle factors, road and environment factors. Vogel and Bester introduced risk factors (human, vehicle and environment factors) for 14 accident types that can be used as a reference point to determine the likely cause of an accident of a specific type. A research had been done from a little bit different point of view, where the researchers proposed a cost-effective solution to identify the potholes and humps on roads and provide time alerts to drivers to avoid accidents or vehicle damages. This study aims to discover the potential and possible application of LiDAR systems in detecting road irregularities

**1.1.1 Problem statement**

Traditional methods of pothole detection involve manual inspections by road maintenance crews, which are both time-consuming and labor-intensive. Automating this process using advanced technologies like LiDAR can improve efficiency and accuracy, leading to safer roads and cost savings. Moreover it can also help reduce accidents caused by potholes and reduce error in spotting them , thus improving roadworks efficiency and road safety .

* 1. **Aim of the project**

The main objectives of this seminar project are as follows:

1. To understand the principles of LiDAR technology.
2. To explore the use of LiDAR for pothole detection.
3. To design and implement a pothole detection system using LiDAR.
   1. **Project domain**

The Pothole detection using LiDAR falls under the deep learning and sensors domains. implement ML techniques to recognize and alert on the presence of potholes.

* 1. **Scope of the project**

The project aims to create a system that is capable of detecting potholes with the help of 2D LiDAR cameras , the feed is then analysed and confirmed for errors .

* 1. **Methodology**

#### Data Collection

LiDAR data is collected using a LiDAR sensor mounted on a vehicle. The sensor emits laser pulses and records the time it takes for the pulses to return after hitting the road surface. This data is used to create a 3D point cloud representing the road surface.

#### Pothole Detection Algorithm

A pothole detection algorithm is developed to process the LiDAR point cloud data. The algorithm identifies depressions or irregularities in the road surface that meet specific criteria, classifying them as potential potholes.

#### System Implementation

The pothole detection system is implemented using suitable software and hardware components. The system is designed to work in real-time or near-real-time to ensure timely pothole detection.

### 

### Chapter 3

### PROJECT DESCRIPTION

**2.1 Existing system**

Currently the existing system is manual work , road department employees manually inspect roads for potholes or other damage. This system has low accuracy and leads to delays in repairs which makes the roads dangerous to drive on in poor conditions.

**2.2 Proposed system**

The proposed system offers a fully automated facility to detect potholes and alert authorities of their presence , it utilizes LiDAR cameras and an algorithm to analyse the feed from the cameras to spot potholes.

**2.2.1 Advantages**

The proposed system can significantly reduce time taken to identify and fix potholes. Further development can allow for pothole data to be shared to map services which then can warn civilians of the danger .Reduces human workload and potential errors . Relatively easy to implement in busy expressways where the required hardware is already present.

### 

### 

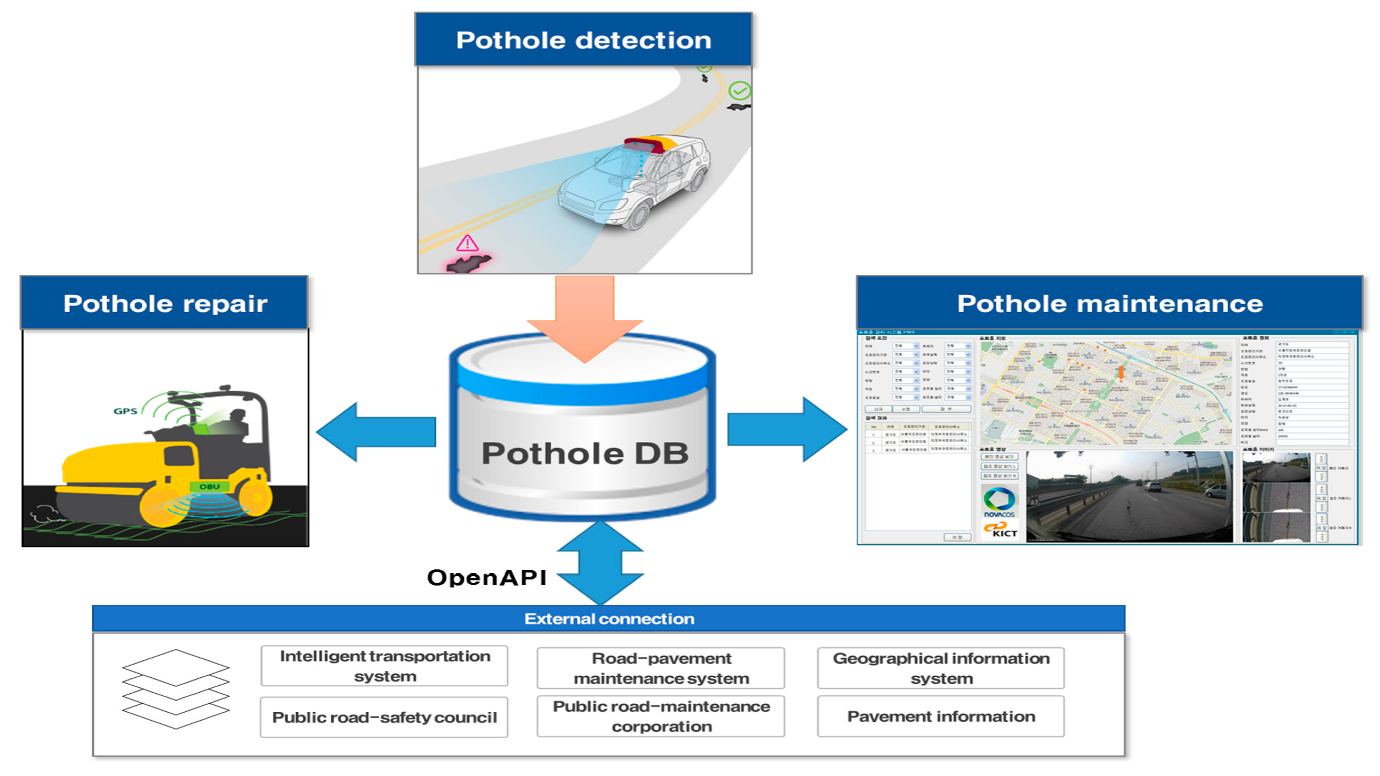
### Chapter 4

### 

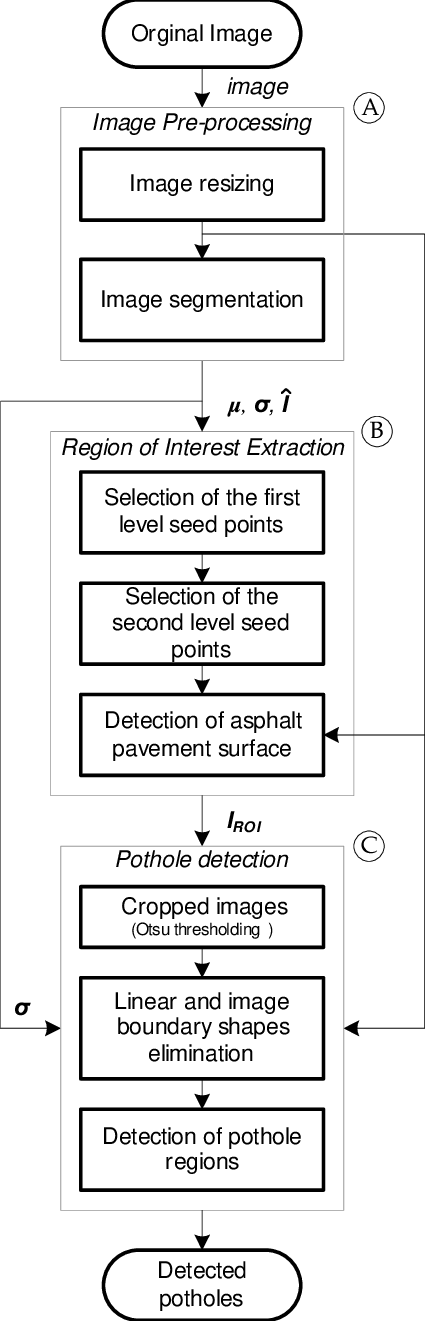
### MODULE DESCRIPTION

### 

***Fig3.1***

****

***Fig 3.2***

****

***Fig 3.3***

### Chapter 5

### IMPLEMENTATION AND TESTING

### 4.1 Input and output

### 

### *Fig4.1*

### 4.2 Testing

### Testing in the Pothole detection system involves evaluating various aspects of the software to ensure it functions correctly, efficiently, and securely. Key testing phases include :Unit Testing: Testing individual components or modules of the system to verify that work as intended. This includes testing functions like adding, editing, and deleting contacts. Integration Testing: Examining how different modules or components interact with each other to ensure they work seamlessly as a whole system. For example, testing the integration between the contact database and the user interface .Functional Testing: Evaluating the system against its functional requirements, such as searching for contacts, categorization, and data import /export .User Interface (UI) Testing: Assessing the user interface for usability, responsiveness ,and adherence to design specifications. It includes testing navigation, forms, and user interactions .Security Testing: Checking for vulnerabilities, such as unauthorized access or data breaches. This involves penetration testing and ensuring proper access control measures are in place . Performance Testing: Assessing the system’s performance, including response times, scalability , and resource utilization under varying workload s. Compatibility Testing: Verifying that the system works correctly on different devices ,browsers, and operating systems, ensuring a consistent user experience . Regression Testing: Ensuring that new updates or changes to the system do not introduce new bugs or negatively impact existing functionality .User Acceptance Testing (UAT): Involving end-users to validate that the system meets their requirements and expectations in real-world scenarios .Load and Stress Testing: Evaluating how the system performs under extreme conditions ,such as high user loads or heavy data processing, to ensure it remains stable and responsive . Data Integrity Testing: Confirming that data remains accurate and consistent throughout various operations, including data import/export . Backup and Recovery Testing: Verifying that data can be reliably backed up and restored in case of system failures or data loss . Testing is an essential phase in system development to identify and address issues ,ensuring that the system functions as intended and meets user expectations in terms efficiency, security, and reliability.

### 

### 

### Chapter 6

### 

### RESULTS AND DISCUSSION

### 5.1 Efficiency of the Proposed System

### The proposed system offers a considerable boost in accuracy and speed in the detection of potholes , it also reduces instances where potholes are left unnoticed and would contribute to reducing accidents related to potholes considerably . It also reduces the workload of road department personnel and automates a rather taxing job.

### 5.2 Comparison of Existing and Proposed System

### Existing system

### Manual effort and inefficiency

### High error rate

### Long resolution time

### Cheap

### Proposed system

### High accuracy

### Reduced manual workload

### Little to no error rate

### Instant detection

### Moderately expensive

### 

### Chapter 7

### CONCLUSION

### 6.1 Conclusion

### This seminar project explores the use of LiDAR technology for automated pothole detection. The project aims to improve road safety and reduce maintenance costs by providing an efficient and accurate method for detecting potholes. The results of the evaluation suggest that LiDAR-based pothole detection has the potential to be a valuable tool for transportation infrastructure management. It addresses all the primary shortcomings of the existing primitive system in place and offers a step forward in the modernization of road infrastructure and the roadworks dept. in general the goal is to better the quality of life on highways and of road department staff and personal . Furthermore the new system boosts the urbanisation status of areas it is implemented in.

### Appendix

### SOURCE CODE:

### import numpy as np

### import cv2

### import glob

### from keras.models import Sequential

### from keras.models import load\_model

### from sklearn.preprocessing import LabelBinarizer

### from sklearn.model\_selection import train\_test\_split

### from keras.utils import np\_utils

### global size

### size = 100

### model = Sequential()

### model = load\_model('C:/Users/anant/Desktop/Pothole Detection using Machine Learning/Pothole Detection using Machine Learning/sample.h5')

### # X\_test = np.load('./models/trainData/128x72x3x10000/X\_test.npy')

### # y\_test = np.load('./models/trainData/128x72x3x10000/y\_test.npy')

### ## load Testing data : non-pothole

### nonPotholeTestImages = glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/test/Plain/\*.jpg")

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.jpeg"))

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.png"))

### test2 = [cv2.imread(img,0) for img in nonPotholeTestImages]

### # train2[train2 != np.array(None)]

### for i in range(0,len(test2)):

### test2[i] = cv2.resize(test2[i],(size,size))

### temp4 = np.asarray(test2)

### ## load Testing data : potholes

### potholeTestImages = glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/test/Pothole/\*.jpg")

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.jpeg"))

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.png"))

### test1 = [cv2.imread(img,0) for img in potholeTestImages]

### # train2[train2 != np.array(None)]

### for i in range(0,len(test1)):

### test1[i] = cv2.resize(test1[i],(size,size))

### temp3 = np.asarray(test1)

### X\_test = []

### X\_test.extend(temp3)

### X\_test.extend(temp4)

### X\_test = np.asarray(X\_test)

### X\_test = X\_test.reshape(X\_test.shape[0], size, size, 1)

### y\_test1 = np.ones([temp3.shape[0]],dtype = int)

### y\_test2 = np.zeros([temp4.shape[0]],dtype = int)

### y\_test = []

### y\_test.extend(y\_test1)

### y\_test.extend(y\_test2)

### y\_test = np.asarray(y\_test)

### y\_test = np\_utils.to\_categorical(y\_test)

### tests = model.predict\_classes(X\_test)

### for i in range(len(X\_test)):

### print(">>> Predicted=%s" % (tests[i]))

### # metrics = model.evaluate(X\_test, y\_test)

### # for metric\_i in range(len(model.metrics\_names)):

### # metric\_name = model.metrics\_names[metric\_i]

### # metric\_value = metrics[metric\_i]

### # print('{}: {}'.format(metric\_name, metric\_value))

### import pandas as pd

### import numpy as np

### import matplotlib.pyplot as plt

### import matplotlib.mlab as mlab

### import tensorflow as tf

### from tensorflow.contrib.layers import flatten

### from keras.layers.pooling import MaxPooling2D

### from keras.models import Sequential, Model

### from keras.callbacks import EarlyStopping, Callback

### from keras.layers import Dense, Dropout, Activation, Flatten, Lambda, ELU,GlobalAveragePooling2D, regularizers

### from keras.layers.convolutional import Convolution2D, Cropping2D, Conv2D

### from keras.layers.pooling import MaxPooling2D

### from keras.optimizers import adam

### from sklearn.utils import shuffle

### from keras.utils import np\_utils

### import time, cv2, glob

### global inputShape,size

### def kerasModel4():

### model = Sequential()

### model.add(Conv2D(16, (8, 8), strides=(4, 4), padding='valid', input\_shape=(size,size,1)))

### model.add(Activation('relu'))

### model.add(Conv2D(32, (5, 5), padding="same"))

### model.add(Activation('relu'))

### model.add(GlobalAveragePooling2D())

### # model.add(Dropout(.2))

### # model.add(Activation('relu'))

### # model.add(Dense(1024))

### # model.add(Dropout(.5))

### model.add(Dense(512))

### model.add(Dropout(.1))

### model.add(Activation('relu'))

### # model.add(Dense(256))

### # model.add(Dropout(.5))

### # model.add(Activation('relu'))

### model.add(Dense(2))

### model.add(Activation('softmax'))

### return model

### size=100

### ## load Training data : pothole

### potholeTrainImages = glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Pothole/\*.jpg")

### potholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Pothole/\*.jpeg"))

### potholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Pothole/\*.png"))

### train1 = [cv2.imread(img,0) for img in potholeTrainImages]

### for i in range(0,len(train1)):

### train1[i] = cv2.resize(train1[i],(size,size))

### temp1 = np.asarray(train1)

### # ## load Training data : non-pothole

### nonPotholeTrainImages = glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.jpg")

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.jpeg"))

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.png"))

### train2 = [cv2.imread(img,0) for img in nonPotholeTrainImages]

### # train2[train2 != np.array(None)]

### for i in range(0,len(train2)):

### train2[i] = cv2.resize(train2[i],(size,size))

### temp2 = np.asarray(train2)

### ## load Testing data : non-pothole

### nonPotholeTestImages = glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/test/Plain/\*.jpg")

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.jpeg"))

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.png"))

### test2 = [cv2.imread(img,0) for img in nonPotholeTestImages]

### # train2[train2 != np.array(None)]

### for i in range(0,len(test2)):

### test2[i] = cv2.resize(test2[i],(size,size))

### temp4 = np.asarray(test2)

### ## load Testing data : potholes

### potholeTestImages = glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/test/Pothole/\*.jpg")

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.jpeg"))

### # nonPotholeTrainImages.extend(glob.glob("C:/Users/anant/Desktop/pothole-and-plain-rode-images/My Dataset/train/Plain/\*.png"))

### test1 = [cv2.imread(img,0) for img in potholeTestImages]

### # train2[train2 != np.array(None)]

### for i in range(0,len(test1)):

### test1[i] = cv2.resize(test1[i],(size,size))

### temp3 = np.asarray(test1)

### X\_train = []

### X\_train.extend(temp1)

### X\_train.extend(temp2)

### X\_train = np.asarray(X\_train)

### X\_test = []

### X\_test.extend(temp3)

### X\_test.extend(temp4)

### X\_test = np.asarray(X\_test)

### y\_train1 = np.ones([temp1.shape[0]],dtype = int)

### y\_train2 = np.zeros([temp2.shape[0]],dtype = int)

### y\_test1 = np.ones([temp3.shape[0]],dtype = int)

### y\_test2 = np.zeros([temp4.shape[0]],dtype = int)

### print(y\_train1[0])

### print(y\_train2[0])

### print(y\_test1[0])

### print(y\_test2[0])

### y\_train = []

### y\_train.extend(y\_train1)

### y\_train.extend(y\_train2)

### y\_train = np.asarray(y\_train)

### y\_test = []

### y\_test.extend(y\_test1)

### y\_test.extend(y\_test2)

### y\_test = np.asarray(y\_test)

### X\_train,y\_train = shuffle(X\_train,y\_train)

### X\_test,y\_test = shuffle(X\_test,y\_test)

### # X\_train.reshape([-1,50,50,1])

### # X\_test.reshape([-1,50,50,1])/

### X\_train = X\_train.reshape(X\_train.shape[0], size, size, 1)

### X\_test = X\_test.reshape(X\_test.shape[0], size, size, 1)

### y\_train = np\_utils.to\_categorical(y\_train)

### y\_test = np\_utils.to\_categorical(y\_test)

### print("train shape X", X\_train.shape)

### print("train shape y", y\_train.shape)

### inputShape = (size, size, 1)

### model = kerasModel4()

### model.compile('adam', 'categorical\_crossentropy', ['accuracy'])

### history = model.fit(X\_train, y\_train, epochs=500,validation\_split=0.1)

### metrics = model.evaluate(X\_test, y\_test)

### for metric\_i in range(len(model.metrics\_names)):

### metric\_name = model.metrics\_names[metric\_i]

### metric\_value = metrics[metric\_i]

### print('{}: {}'.format(metric\_name, metric\_value))

### print("Saving model weights and configuration file")

### model.save('sample.h5')

### model\_json = model.to\_json()

### with open("truesample.json", "w") as json\_file:

### json\_file.write(model\_json)

### model.save\_weights("truesample.h5")

### print("Saved model to disk")

### References

### [1]Khaled R. Ahmed , 2021 , Smart Pothole Detection using Deep Learning Based on Dilated Convolution , MDPI Open access journals.

### [2]Hyunwoo Song , Kihoon Baek ,2018 , Pothole Detection using Machine Learning , Research gate journals.

### [3]Varsha Kumari , 2021 , Pothole Detection using LiDAR , Advances in Automobile Engineering.

### [4]X. Yu , E. Salari , 2011 , Pavement pothole detection and severity measurement using laser imaging , IEEE Xplore.

### [5]Roopak Tamboli , 2015 , Laser based detection and depth estimation of dry and water filled potholes: A geometric approach , Research Gate.