SAVITRIBAI PHULE PUNE UNIVERSITY

A
Project Report
On

"POTHOLE DETECTION SYSTEM"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Submitted by
BAGUL ASHISH SURESH
GAIKWAD ROHIT DADAJI
MAHARSHI YADNYESH PRAMOD

Under the Guidance of
PROF. P. D. LOKHANDE

Gokhale Education Society's
R. H. Sapat College of Engineering,
Management Studies and Research, Nashik-5
DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

ACADEMIC YEAR: 2022-2023

Gokhale Education Society's

R. H. Sapat College of Engineering,

Management Studies and Research, Nashik-5

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Certificate

This is to certify that, this project report entitled **"POTHOLE DETECTION SYSTEM"** Submitted by **Mr. Bagul Ashish Suresh, Exam Seat No. B190693005; Mr. Gaikwad Rohit Dadaji, Exam Seat No. B190693017; Mr. Maharshi Yadnyesh Pramod, Exam Seat No. B190693030** for partial fulfillment of the requirement for the award of the degree of **Bachelor's of Engineering** in **ELECTRONICS AND TELECOMMUNICATION ENGINEERING** as laid down by **SAVITRIBAI PHULE PUNE UNIVERSITY**, Pune, is a record of their own work carried out by them under my supervision and guidance during year 2022 – 2023.

Place : Nashik

Date :

Prof. P. D. Lokhande
PROJECT GUIDE

Dr. S. P. Agnihotri
HEAD OF DEPARTMENT

Prof.
SIGNATURE OF EXAMINER

Dr. P. C. Kulkarni
PRINCIPAL

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to my guide, **Prof. P. D. Lokhande**, for his constant encouragement, wonderful technical guidance and support throughout the course. I sincerely thank **Dr. S. P. Agnihotri**, Head of Department of Electronics & Telecommunication Engineering for his advice and support during course of this work.

With deep sense of gratitude we thank to our Principal **Dr. P. C. Kulkarni** and Management of **Gokhale Education Society** for providing all necessary facilities and their constant encouragement and support.

We are very much thankful to our project coordinator **Mrs. H. H. Kulkarni** for giving us valuable guidelines for project work and its report writing. We are very much thankful to **Mr. P. D. Lokhande** for sponsoring our project and providing all guidance and necessary resources required to complete the task.

I express my thanks to all teaching & non- teaching staff for their kind co-operation and guidance for preparing and presenting this seminar. I take this opportunity to express my gratitude towards my parents without which it would have not been possible.

We are ending this acknowledgement with deep indebtedness to our friends who have helped us directly or indirectly.

BAGUL ASHISH SURESH
GAIKWAD ROHIT DADAJI
MAHARSHI YADNYESH PRAMOD

ABSTRACT

One of the most common modes of transportation in the world is road transport. In India, roads carry almost 85 percent of the country's passenger traffic[1]. While roads remain synonymous with development and growth in the country, they have also been a nemesis for users also carrying the dubious distinction of leading the global tally of annual deaths and injuries on account of road accidents. One of the major and primary causes of these road accidents is potholes which not only cause damage to vehicles but also put passengers into danger, hence identification and classification of roads using is very important. Manual pothole detection and classification is very human-intensive work. Hence the proposed system detects the potholes on the road surface using convolutional neural network (CNN) architecture using YOLOv4-tiny algorithm. This involves the pre-processing of input images and result demonstration by performing grayscaling. This detection proves advantageous for passangers to travel safely without plunged in potholes.

Keywords: *Convolutional Neural Network, YOLOv4-tiny, grayscaling.*

CONTENTS

ABBREVIATIONS

CARLA Car Learning to Act. 15

CMOS Complementary Metal Oxide Semiconductor. 20

CMYK Cyan, Magenta, Yellow, and Key (Black). 30

COVID COronaVIrus Disease. 12

CRREL Cold Regions Research and Engineering Laboratory. 12

CRRI Central Road Research Institute. 13

CUDA Compute Unified Device Architecture. 15

FPN Feature Pyramid Network. 25

GDP Gross Domestic Product. 11

Govt. Government. xi, 13

GPS Global Positioning System. 16

GPU Graphics Processing Unit. 15

GTX Giga Texel Shader eXtreme. 15

HSV Hue Saturation Value. 30

IDE Integrated Development Environment. 21

LiDAR Light Detection and Ranging. v, 16

MoRTH Ministry of Road Transport and Highways. xi, 13

MP Megapixel. 20

NCRB National Crime Records Bureau. 13

PC Personal Computer. 20

R-CNN Region-Based Convolutional Neural Network. 17

RAM Random Access Memory. 21

RGB Red, Green and Blue. 30

SSD Solid-State Drive. 17

USB Universal Serial Bus. 20

Wi-Fi Wireless Fidelity. 16

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Road Transportation and Issues

Transportation can be defined as the movement of persons and goods from one place to another using modes such as land, air, water, etc. The most commonly used mode of transport is by land. The two main forms of land transport can be considered to be rail transport and road transport. Road transport is one of the most cost effective and preferred mode of transport, both for freight and passengers. It is vital for economic development and social integration of the country.
According to the Ministry of Road Transport and Highways, Government of India, road Transport has emerged as the dominant segment in India's transportation sector with a share of 4.5% in India's GDP in 2005-06. The Road Transport Sector accounts for about 87% of passenger traffic and 60% of freight traffic movement in the country[2].

Figure 1.1: Mumbai–Pune Expressway[3]

Although roads are epitome of development and progress of the nation, there are many issues in the maintenance of the roads. The major road issues include poor road surfaces, insufficient capacity, poor sidewalks, unlimited crossings and a lack of road equipment and safety measures. Many road transportation problems arise due to rapid urbanization and over population.

The process of socio-economic development in the developing country is retarded by inadequate transportation facilities. India has the dubious distinction of accounting for 6% of the world's road deaths while having just 1% of the world's vehicles[4]. Hence road transportation issues needed to be solved. The most significant road issue is the pothole.

1.2 Pothole and Its Formation

A pothole is a depression in a road surface formed by gradual damage caused by traffic and weather, where traffic removes broken pieces of the pavement which is generally the result of water in the underlying soil structure and traffic passing over the affected area. According to the US Army Corps of Engineers' CRREL[5], pothole formation requires two factors which are water and traffic present at a same time. Potholes can grow to several feet in width, though they usually only develop to depths of a few inches. If not concerned properly, potholes may get large shapes.

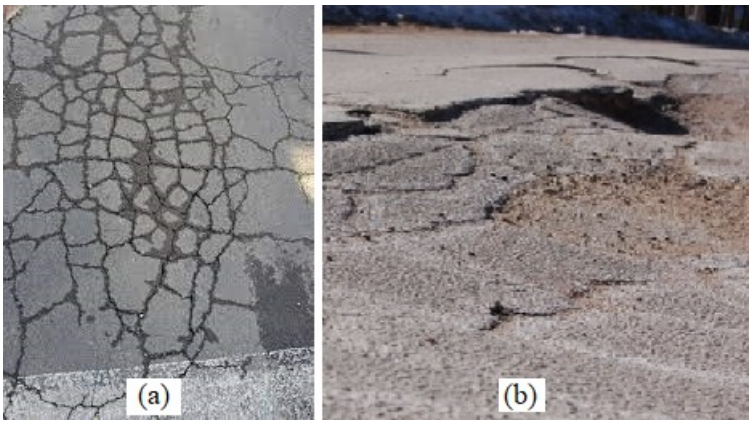


Figure 1.2: (a)Alligator Cracking and (b)Frost Heaves

Water seeps through the porous asphalt and road bed and pools on the top of frozen soil beneath. This water freezes and expands causing the road surface to rise called as frost heaves. The ice melts leaving a void beneath the asphalt. When the traffic pass over this, the tires collapse the surface into the void which results in the creation of pothole.

1.3 Problems Due to Potholes

Potholes are one of the major cause of accidents. If potholes become large enough, damage to the tires, wheels, and vehicle suspensions is liable to occur. Severe road accidents may occur if potholes are developed in large size.
The accidents cause by potholes may include traffic collision, vehicle trashing, vehicle collision with another objects. Traffic collisions often result in injury, disability, death, and property damage as well as financial costs to both society and the individuals involved.

1.4 Potholes in India

In India, pothole is very serious issue in road transport systems. It is a major cause of road accidents in the country in which several deaths also happen every year. According to government data, potholes claimed, on average, 2,300 lives annually across the country during the five years from 2016 to 2020[6]. The situation is more worsen because of the less. working on road transport during COVID pandemic. Increasing vehicular traffic and rains deteriorate the condition of roads.

Figure 1.3: Potholes on Ring Road near Maharani Bagh in South Delhi[7]

According to the NCRB, about 5% of the deaths on roads are due to potholes or open manholes[8]. According to the CRRI data, metro cities like Delhi and Mumbai record highest number of potholes[9].

Year	Road Accidents	Deaths
2019	4775	2140
2020	3564	1471
2021	3625	1481

Table 1.1: Total Number of Road Accidents Occurred and Persons Killed Due To Pothole On All Category of Roads in India, Source: MoRTH, Union Govt.[10]

1.5 Detection of Potholes

Due to these serious problems of potholes, they are needed to be detected by automatic system using software tools and machine learning algorithms. Manual pothole detection is very time consuming task hence the potholes should be detected using object detection technique using deep learning. Object Detection is a software and fundamental task for an autonomous driving system that provides remarkable change.
Detection and counting of potholes using different image processing techniques helps in classification of different types of road profile. Pothole detection involves different processing methods such as image filtering, image segmentation and clustering techniques like edge detection, thresholding, etc. The result where evaluated based on the traditional performance measures - accuracy, sensitivity, specificity, and computational time.
Vibration based method uses accelerometers in order to detect potholes. However, these methods could provide wrong results that the hinges and joints of road can be detected as potholes. 3D reconstruction using laser scanning systems can detect potholes in real time. However, the cost of laser scanning equipment is still significant. Although the vision-based methods are cost effective than 3D laser scanner methods, it may be difficult

Figure 1.4: Manual Detection of Pothole and Repairing of Road in Bengaluru[11]

to recognize pothole due to noise in a form like pothole such as shadow, road surface patch.

1.6 Overview of Pothole Detection System

The proposed system “Pothole Detection System” makes use of deep learning to detect the potholes in the road modules. In deep learning, an algorithm is used to detect the object which is placed in front of the system.
It is a completely software based system which can work on any operating system and consists of a source code. The user needs to runs the source code which is in Python programming language on any open source code editing and compiling software such as Microsoft Visual Studio Code.
In the software of the system involves a binary classifier which classifies the received input into an output from two choices for pothole and no pothole. This classifier requires a dataset of images which is needed to be uploaded and train this classifier accordingly to work properly for the generation of output.
The output of the system will be visible on the screen in the generated result window.This result image will have the captured image with detected output highlighted using frame structure from which potholes can be easily identified.

Chapter 2

LITERATURE SURVEY

2.1 Paper 1: Pothole Detection and Avoidance via Deep Learning on Edge Devices

In this paper, Chi-Wei Kuan, Wen-Hui Chen, and Yu-Chen Lin[12] proposed a pothole avoidance system deployed on an edge platform is proposed, which can avoid potholes automatically. They used Deep Q-Network (DQN) as an agent, and train it on CARLA driving simulator with cross-task unsupervised transfer learning. According to the type of input data, these methods can be concluded into three categories: vibration-based, depth-based, and image-based. This consists of two steps: bounding box calibration and perspective bounding box transformation.
This research also deals with pothole avoidance which is based on deep reinforcement learning that can effectively avoid the potholes. In addition, both of the systems are deployed on an energy-efficient edge platform, and can be executed in real time.

2.2 Paper 2: Deep Learning based Detection of potholes in Indian roads using YOLO

Dharneeshkar J, Soban D V, Aniruthan S A, Karthika R, and Latha Parameswaran[13] developed a pothole detection system based on deep learning using YOLO & advanced image processing. In this paper, a new 1500 image dataset has been created on Indian roads. The dataset is annotated and trained using YOLO (You Only Look Once). The new dataset is trained on YOLOv3, YOLOv2, YOLOv3-tiny, and the results are compared. The results are evaluated based on the mAP, precision and recall.
The experiment was carried out on a GeForce GTX 1060 with MAX-Q design and the operating system was Ubuntu 18.04. To increase the computation speed and efficient use of graphics the system is installed with CUDA. CUDA is used when there is a need for GPU computation. It ensures a superfast neural network but it supports only Nvidia GPU. By using CUDA the training time is drastically reduced.

2.3 Paper 3: Design, Development and Testing of Automatic Pothole Detection and Alert System

Dhwani Desai, Abhishek Soni, Dhruv Panchal, and Sachin Gajjar[14] introduced a system which automatically generates an alarm on handset after detection of pothole making use of stereo-visual camera and GPS.It also specifies working of ultrasonic sensor. GPS u-blox neo 6M is used to identify the location (latitudes and longitudes) of pothole or speed breaker.

2.4 Paper 4: Pothole Detection System using 2D LiDAR and Camera

Byeong-ho Kang and Su-il Choi[15] provides a way to detect pothole using LiDAR and Camera using edge extraction, object extraction and adaptive breakpoint detection.The pothole detection algorithm includes noise reduction pre-processing, clustering, theline segment extraction, and gradient of pothole data function. Next, image-based pothole detection method is used to improve the accuracy of pothole detection and to obtain pothole shape. Image-based algorithm includes noise filtering, brightness control, edge extraction, binarization, addictive noise filtering, and object extraction.

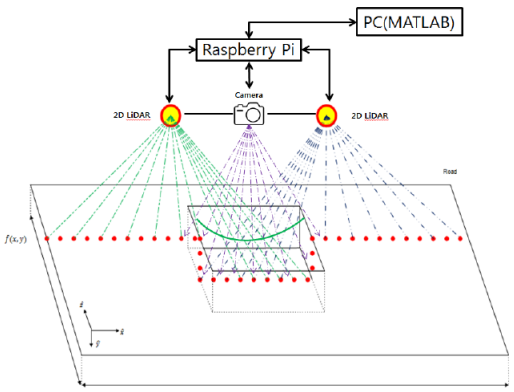


Figure 2.1: Pothole Detection Model Developed By Byeong-ho Kang and Su-il Choi

2.5 Paper 5: Pothole Detection and Warning System: Infrastructure Support and System Design

Sudarshan S Rode, Shonil Vijay, Prakhar Goyal, Purushottam Kulkarni, Kavi Arya[16] detected potholes using Wi-Fi and access point tracking. During the setup phase of the access points, each of them would be given correct location coordinates using a GPS device. The Wi-Fi standard leaves connection criteria and roaming totally open to the client. This is the strength of Wi-Fi, but it also means that one wireless adapter may perform substantially better than another. Since Wi-Fi transmits in the air, it has the same properties as a non-switched wired Ethernet network, and therefore collisions may occur.

2.6 Paper 6: A Comparative Evaluation of Deep Learning Algorithms for the Pothole Detection

Roopak Rastogi, Uttam Kumar, Archit Kashyap, Shubham Jindal, Saurabh Pahwa[17] studied the evolution in the technologies used in pothole detection using SSD algorithm. This model can be deployed in autonomous vehicles for real-time geotagged pothole detection from photographs or video streams. The pothole detection application can also suggest potential alternate eco-friendly routes and guide the commuters in low light navigation. A modified YOLOv2 was proposed to unravel the class imbalance problem of "pothole" and "normal road" classes, and it outperformed the YOLOv2 base model and Faster R-CNN architectures.

2.7 Paper 7: Detection and Counting of Pothole using the Image Processing Techniques

Vigneshwar. K and Hema Kumar. B[18] developed algorithm based on image processing which detects and counts the potholes using DoG filter. The main goal of this paper is to identify a better method which is highly efficient and accurate compared to the conventional methods. Different image pre-processing and segmentation methods for pothole detection where reviewed using performance measures.

2.7.1 Difference of Gaussian-Filtering(DoG)

The DoG- Filter was applied as a pre-processing filter to original pothole image for better edge detection with reduced noise. It finds the difference between two sigma values of two Gaussian profiles and find the edges in the grayscale image,

$$G(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{1}{2\pi 0.5\sigma^2}e^{-\frac{x^2+y^2}{2\ 0.5\sigma^2}} \tag{2.1}$$

2.8 Paper 8: Pothole and Object Detection for an Autonomous Vehicle Using YOLO

Kavitha R and Nivetha S[19] proposed a system for pothole detection based on object detection technologies using model training. Detecting potholes in Indian road help the autonomous vehicle to move smoothly without getting struck in the potholes.The computer vision and machine learning algorithm is the important tool for detecting objects in and around the environment.In this paper, which consists of two parts The first part is implemented on object detection in the surrounding with YOLO Algorithm provides exact classification and position which is configured on newly created datasets for classes of object. In part two of the proposed method is implemented on Raspberry pi4 a popular embedded computer board explores suitability for the running objects.

Chapter 3

PROBLEM STATEMENT

In India, potholes in the road are the major cause for serious to fatal accidents. Potholes not only leads to injuries to human body by harming spinal cord but also damages the vehicles by making it less efficient, damages wheels and reems, harms exhaust system of vehicles which pollutes environment. Hence potholes are required to be detected. It is very tedious and time consuming to manually detect all potholes on the road surface, hence we should make use of deep learning algorithm for pothole detection.

Chapter 4

OBJECTIVE

1. To develop a software-based image processing system which detects potholes using source code on machine learning algorithm.
2. To develop a system based on tiny-YOLOv4 algorithm which generate result of detected potholes by framing the pothole structure from the captured input images given by the user of the system.

Chapter 5

REQUIREMENTS SPECIFICATION

5.1 Hardware Requirements

5.1.1 Web Camera

Figure 5.1: HIKVISION DS-U02 1080p Webcam

In this project, a web camera is required for capturing the images of road. Here we are using HIKVISION DS-U02 web camera for taking the pictures of road surface. This camera is connected to the system USB cable. Camera specifications are given in the following table 5.1[20]:

Manufacturer	HIKVISION
Model	DS-U02
Resolution	1080p
Connectivity Technology	USB
Photo Sensor Technology	2MP CMOS
Focal Length	3.6

Table 5.1: Webcam Specifications

5.2 System Requirements

The project source code is needed to be run and executed on a system such as PC or laptop. This minimum requirements are given in the table 5.2:

Operating System	Windows / Linux / Mac OS
Programming Language	Python
IDE	VS Code / Jupyter / PyCharm
Processor	i3 or AMD
Minimum RAM	8GB

Table 5.2: System Requirements

5.3 Software Requirements

5.3.1 Python

Figure 5.2: Python

Python is a popular general-purpose interpreted, interactive, object-oriented, and high level programming language[21]. We are developing our system in Python programming language because it is easy to interpret the code in it and python has very large collection of libraries which are easy to install. Also Python is easy to debug.
We have installed 64-bit Python 3.9.0 version which was released on Oct. 5, 2020 from official Python website. This version supports flexible function and variable annotations. This is the first version of Python to default to the 64-bit installer on Windows[22].

5.3.2 PyCharm

Figure 5.3: PyCharm

There are various of IDEs available to run the Python code. For example, VS Code, Jupyter, Google Colab. In our system we are going to use PyCharm IDE. PyCharm is developed by the Czech company JetBrains in 2010. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django.It is cross-platform, working on Microsoft Windows, macOS and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License. We used Community Edition which is free of cost. You can directly download and install this software from JetBrains official website[23].

Chapter 6

METHODOLOGY

6.1 Block Diagram

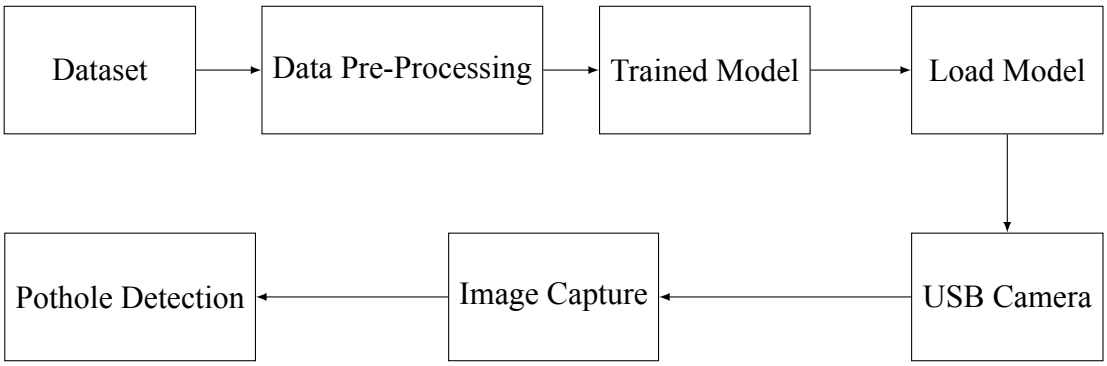


Figure 6.1: Block Diagram

The above figure 6.1 shows the functional block diagram of Pothole Detection System. It consist of 7 steps to detect the pothole which are:
Dataset: Dataset of pothole images is required to train the model. We have used 45 pothole images for model training.
Data Pre-Processing: It is a data mining technique which is used to transform the raw data in a useful and efficient format. This step involves processing of pothole images.
Trained Model: It is generated after data preprocessing stage using machine learning algorithm. We are using tiny YOLO v4 algorithm for model training.
Load Model: In this step, load the weights and re-evaluate the saved model
USB Camera: Connect USB camera to system port and add port number.
Image Capture: Capture image of road surface using system keyboard.
Pothole Detection: This is the final step where result will be generated automatically. Potholes are detected by framing them.

6.2 Flowchart

Figure 6.2 shows flowchart of the system. This flowchart is user's action for working of the project.

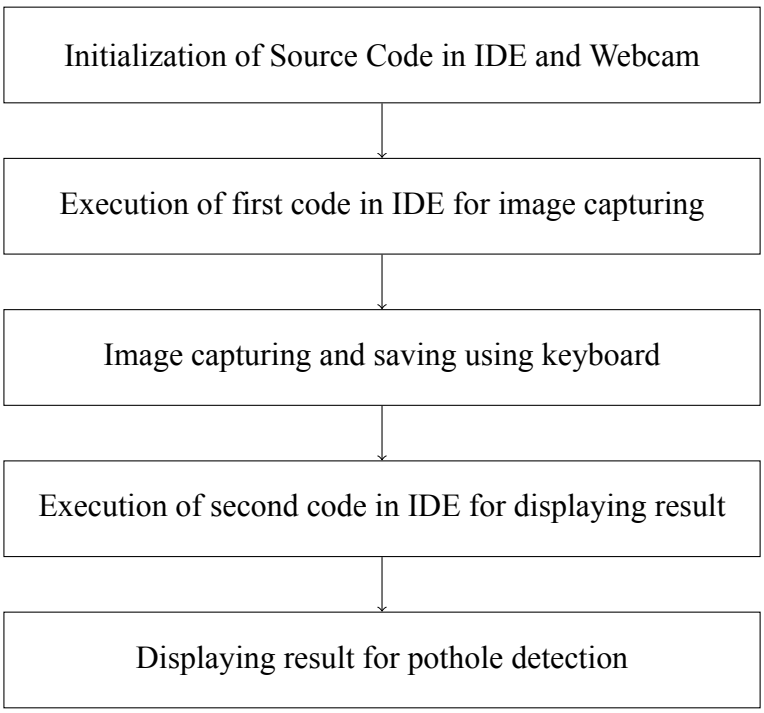


Figure 6.2: Flowchart

6.3 Convolutional Neural Network

Convolutional neural network is a kind of network architecture for the deep learning algorithms and is specifically used for image recognition and tasks involves the processing of pixel data. It is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.

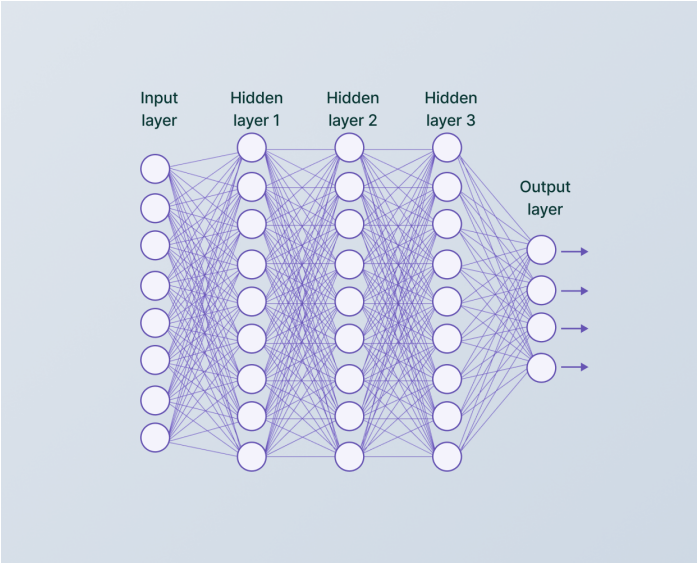


Figure 6.3: Layers in Convolutional Neural Network

In a regular neural network there are three types of layers as shown in figure 6.3[24]:
Input Layer: It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of pixels in the of an image.
Hidden Layers: There can be more than one hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.
Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

6.4 YOLOv4-tiny

YOLO stands for You Only Look Once. YOLO trains full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. YOLOv4 is a one-stage object detection model that improves on YOLOv3 with several bags of tricks and modules introduced in the literature. We used YOLOv4-tiny for our project which is a compressed version of YOLOv4 designed to train on machines that have less computing power. Its model weights are around 16 megabytes large, allowing it to train on 350 images in 1 hour when using a Tesla P100 GPU.
In program folder of the project, configuration and weight files of YOLOv4-tiny are required. Also python file is required in training files folder. In this program it is clone into Darknet and mounted into drive. Darknet is an open source neural network framework written in C and CUDA. Then dataset is set and unzipped from Darknet.

```
#getting yolo darknet repo
!git clone https://github.com/AlexeyAB/darknet
```

```
Cloning into 'darknet'...
remote: Enumerating objects: 15395, done.K
remote: Total 15395 (delta 0), reused 0 (delta 0), pack-reused 15395K
Receiving objects: 100% (15395/15395), 13.99 MiB | 18.30 MiB/s, done.
Resolving deltas: 100% (10354/10354), done.
```

```
#mounting drive
%cd ..
from google.colab import drive
drive.mount('/content/gdrive')
```

```
/
Mounted at /content/gdrive
```

Figure 6.4: Cloning into Darknet and Mounting into Drive

6.4.1 Overview of YOLOv4-tiny

The table 6.1 shows details about YOLOv4-tiny algorithm.

Date of Release	Nov 09, 2020
Model Type	Object Detection
Architecture	ResNet-D, YOLO
Framework Used	Darknet
Annotation Format	YOLO Darknet TXT

Table 6.1: Overview of YOLOv4-tiny

6.4.2 YOLOv4-tiny Architecture

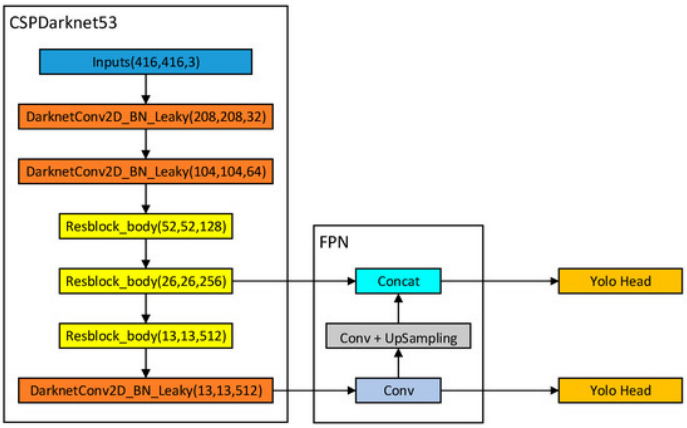


Figure 6.5: YOLOv4-tiny Model Structure[25]

Figure 6.5 demonstrates the structure of the YOLOv4-tiny algorithm, in which the CSP darknet53 is used as the backbone network for feature extraction. The FPN part is a bottom-up pyramid structure that uses a single-layer convolution structure to up sample features. This algorithm only uses the last two Yolo Heads for feature regression to obtain classification results. Different Yolo Heads can detect targets of different sizes.

6.5 Deep Neural Network

Deep Neural Network (DNN), also called Deep Nets, is a neural network with a certain level of complexity. It can be considered as stacked neural networks, or networks that composed of several layers, usually two or more, that include input, output, and at least one hidden layer in between.

The NVIDIA CUDA® Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. cuDNN provides tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers[26].

Deep learning researchers and framework developers worldwide rely on cuDNN for high performance GPU acceleration. It allows them to focus on training neural networks and developing software applications rather than spending time on low GPU performance tuning. cuDNN accelerates widely used deep learning frameworks, including Caffe2, Chainer, Keras, MATLAB, MxNet, PaddlePaddle, PyTorch, and TensorFlow.

6.5.1 DNN Function in OpenCV

```
# importing model weights and config file
net = cv2.dnn.readNet('project_files/yolov4_tiny.weights',
                      'project_files/yolov4_tiny.cfg')
model = cv2.dnn_DetectionModel(net)
model.setInputParams(scale=1 / 255, size=(416, 416), swapRB=True)
classIds, scores, boxes = model.detect(
    img, confThreshold=0.6, nmsThreshold=0.4)
```

Figure 6.6: Creating Model by Importing Weight and Configuration File

The cv2.dnn.readNet() function is used for importing model weights and configuration file from folders as shown in figure. The cv2.dnn_DetectionModel(Net) is used to create model from deep learning network. These DNN functions are used in image.py program

Chapter 7

IMPLEMENTATION

7.1 Dataset

A machine learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions.

Figure 7.1: Pothole Dataset

In this system, we required a dataset of potholes on the road surface. We are using 45 pothole images and their coordinates as a raw data. Figure 7.1 shows sample images of pothole dataset. This images are collected from iStock.

7.2 Installation of Libraries

A Machine Learning library, or a Machine Learning framework, is a set of routines and functions that are written in a given programming language.

7.2.1 Installing OpenCV library

Figure 7.2: OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly for real-time computer vision[27]. The library is cross-platform and licensed as free and open-source software. OpenCV features GPU acceleration for the real time operations.
OpenCV is installed in system using PyCharm Terminal by command:
pip install opencv - python

7.2.2 Installing Geocoder library

Geocoder is a simple and consistent geocoding library written in Python. Dealing with multiple different geocoding provider such as Google, Bing, OSM & many more has never been easier[28].
Geocoder is installed in system using PyCharm Terminal by command:
pip install geocoder

7.3 Image Capturing in OpenCV

The OpenCV library include various methods and functions to capture an image and video also. OpenCV library is compatible with the Linux and windows both operating system. To capture images from Webcam, it should be connected to the user's laptop through USB cable.

```
import cv2

key = cv2. waitKey(1)
webcam = cv2.VideoCapture(0)
while True:
    try:
        check, frame = webcam.read()
```

Figure 7.3: Image Capturing Function of OpenCV in PyCharm

The function used for image capturing in OpenCV is cv2.VideoCapture(Port). By using this function, camera will start working to capture the image. First, you have to import cv2 and declare a variable for image capturing.
VideoCapture(0): For using the external web camera.
VideoCapture(1): For using default system's embedded camera

7.4 Image Processing using CNN

Image processing is the process of transforming image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying predetermined signal processing methods. It is a computer technology applied to images that helps us process, analyze and extract useful information from them[29].

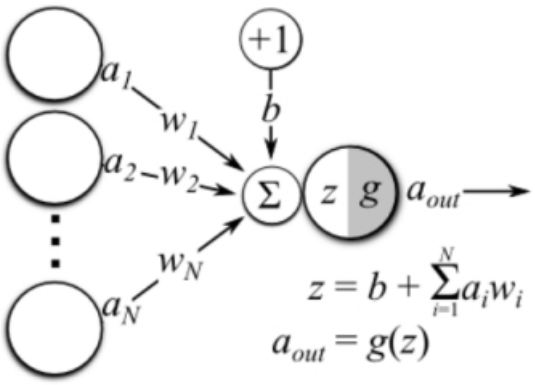


Figure 7.4: Operation in Single Neuron

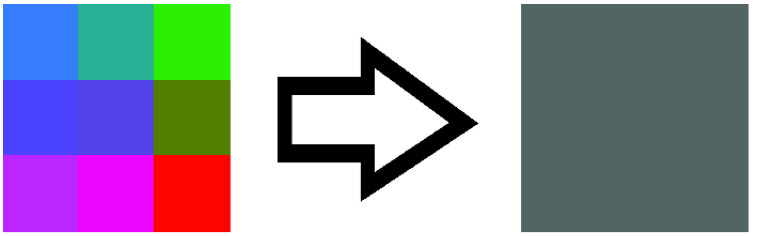
The basic working of the neural network is as follows:

1. Let's consider an image, each pixel is fed as input to each neuron of the first layer, neurons of one layer are connected to neurons of the next layer through channels.
2. Each of these channels is assigned a numerical value known as weight.
3. The inputs are multiplied by the corresponding weights and this weighted sum is then fed as input to the hidden layers.
4. The output from the hidden layers is passed through an activation function which will determine whether the particular neuron will be activated or not.
5. The activated neurons transmits data to the next hidden layers. In this manner, data is propagated through the network, this is known as Forward Propagation.
6. In the output layer, the neuron with the highest value predicts the output. These outputs are the probability values.
7. The predicted output is compared with the actual output to obtain the error. This information is then transferred back through the network, the process is known as Backpropagation.
8. Based on this information, the weights are adjusted. This cycle of forward and backward propagation is done several times on multiple inputs until the network predicts the output correctly in most of the cases.
9. This ends the training process of the neural network. The time taken to train the neural network may get high in some cases.

In the Figure 7.4, ai's is the set of inputs, wi's are the weights, z is the output and g is any activation function.

7.5 Image Grayscaling

In digital photography, computer-generated imagery, and colorimetry, a grayscale image is one in which the value of each pixel is a single sample representing only an amount of light; that is, it carries only intensity information. Grayscale images, a kind of black and white or gray monochrome, are composed exclusively of shades of gray.
Grayscaling is the process of converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white[30].



(a) RGB pixels. (b) Grayscale pixels.

Figure 7.5: Conversion of RGB to Grayscale Image[31]

The average pixel values(ranging from 0-255) of the primary colours which are red green and blue (popularly referred to as RGB) are combined. The luminous intensity of each colour band(which is 24 bits) is combined into a reasonable approximated grayscale value(8 bits)

7.5.1 OpenCV Function for Grayscaling

```
img_ = cv2.imread('saved_img.jpg', cv2.IMREAD_ANYCOLOR)
print("Converting RGB image to grayscale...")
gray = cv2.cvtColor(img_, cv2.COLOR_BGR2GRAY)
print("Converted RGB image to grayscale...")
```

Figure 7.6: Image Grayscaling in OpenCV

In our project code we have used the following function to convert the RGB image to grayscale:
gray = cv2.cvtColor(img_, cv2.COLOR_BGR2GRAY)

7.5.2 Benefits of Image Grayscaling

- It helps in simplifying algorithms and as well eliminates the complexities related to computational requirements.
- It reduces model complexity. Consider training neural articles on RGB images of 10x10x3 pixels. The input layer will have 300 input nodes. On the other hand, the same neural network will need only 100 input nodes for grayscale images.

- It enhances easy visualisation. It differentiates between the shadow details and the highlights of an image because it is mainly in 2 spatial dimensions (2D) rather than 3D.
- It reduces the dimension. For example, In RGB images there are the three color channels and three dimensions while grayscale images are single-dimensional.

7.6 Image Resizing

Image resizing refers to the scaling of images. Scaling comes in handy in many image processing as well as machine learning applications. It is the process of increasing or decreasing the size of an image. It also helps in zooming in on images. Many times we need to resize the image i.e. either shrink it or scale it up to meet the size requirements.

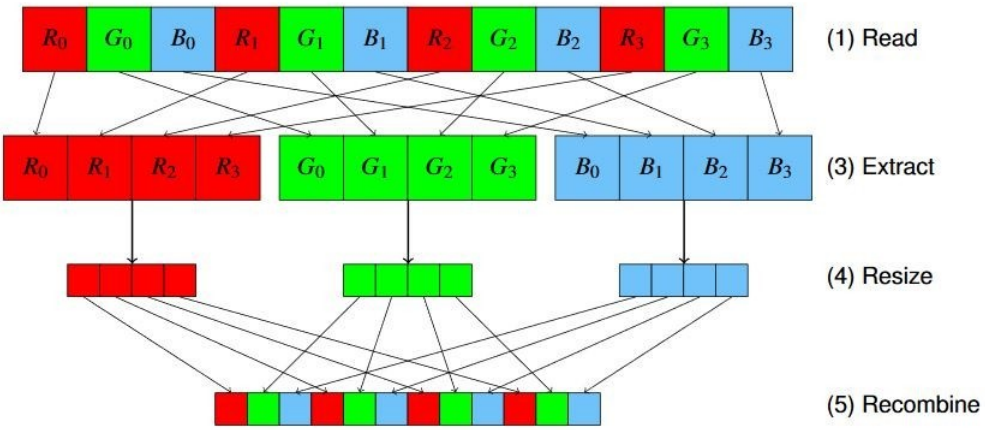


Figure 7.7: Image Resizing Algorithm[32]

Image resizing is necessary when you need to increase or decrease the total number of pixels, whereas remapping can occur when you are correcting for lens distortion or rotating an image. Zooming refers to increase the quantity of pixels, so that when you zoom an image, you will see more detail.

7.6.1 OpenCV Function for Image Resizing

```
print("Resizing image to 28x28 scale...")
img_ = cv2.resize(gray,(32,32))
print("Resized...")
img_resized = cv2.imwrite(filename='saved_img-final.jpg', img=img_)
print("Image saved!")
```

Figure 7.8: Image Resizing in OpenCV

In our project code we have used the following function for image resizings:
img_ = cv2.resize(gray,(32,32))

Chapter 8

WORKING OF THE SYSTEM

8.1 Image Capturing

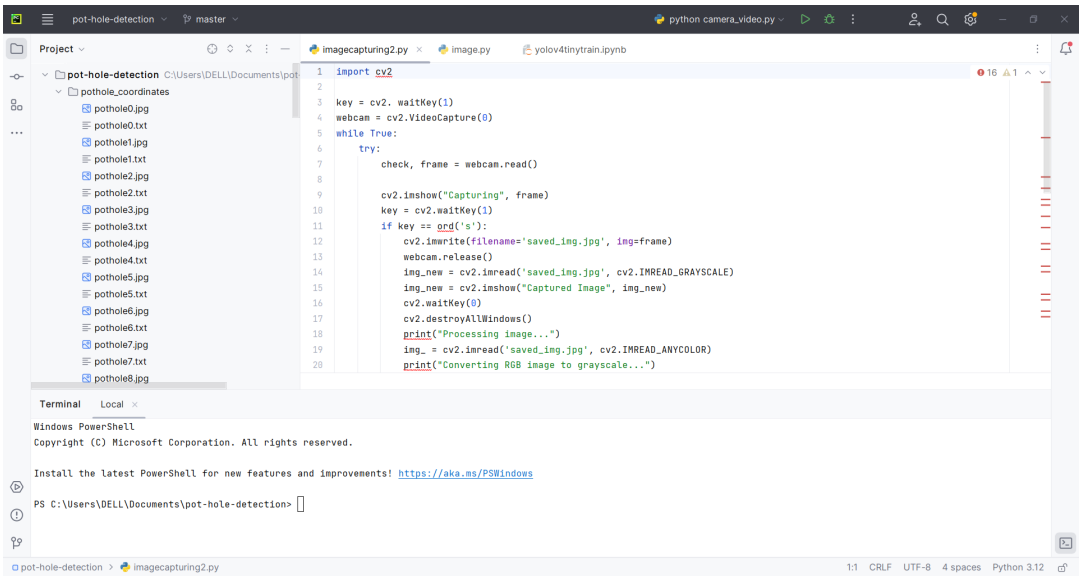


Figure 8.1: Opening of Source Code and Terminal in PyCharm IDE

First of all, you need to install Python, PyCharm IDE and required OpenCV and Geocoder libraries as mentioned earlier. Connect the webcamera to system. Now, open the PyCharm IDE and open the project's source code folder in it. It has two Python source codes, i. e. imagecapturing2.py and image.py. Now open both codes in two windows and open the Terminal for code execution as shown in figure 8.1

Figure 8.2: Command for Execution of First Code

Now in the terminal you can see the path of the folder. Next to it you have to type commands for execution. We have to execute the imagecapturing2.py source code first. This source code is for image capturing and saving the input image.
Now, type command python “imagecapturing2.py” as shown in figure 8.2

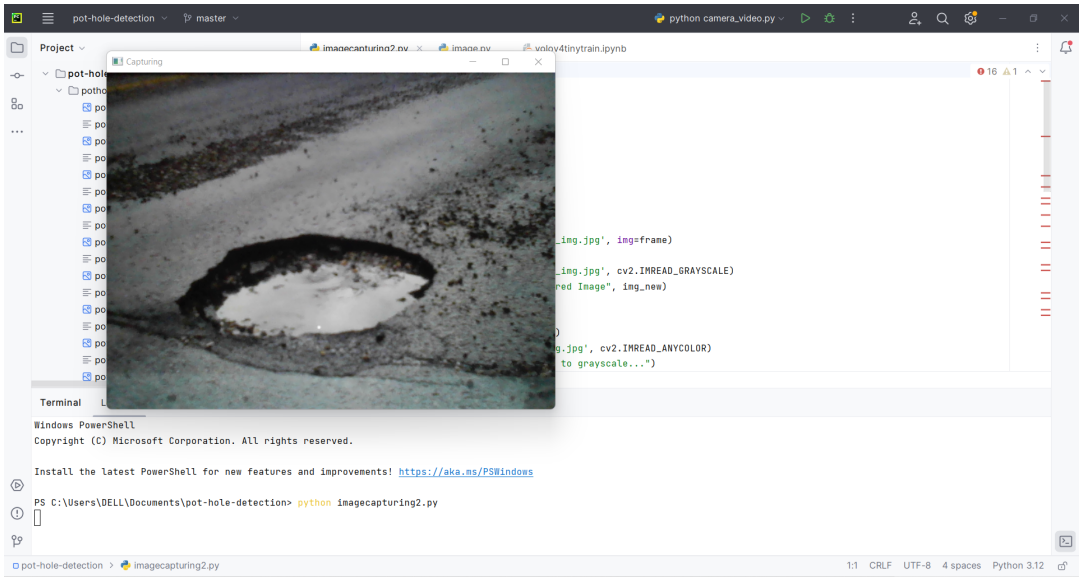


Figure 8.3: Video Capturing through Web Camera

This command will execute first code which results into start of video capturing through the connected web camera in python window as shown in figure 8.3. Now you can capture the image of road surface from the web camera connected to your system or laptop. This camera should be connected to the USB port of laptop through th USB cable.

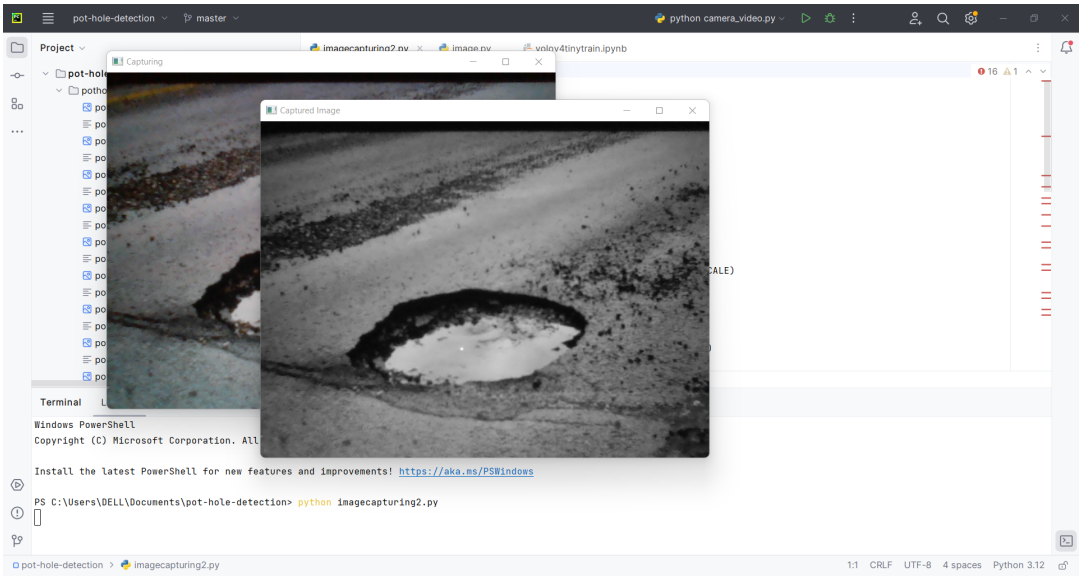


Figure 8.4: Captured Image

After the start of video capturing, user can go for capturing the pothole input image. To capture the image, click ‘S’ on the keyboard. As shown in figure 8.4, the captured image will be shown in another python window.

8.2 Image Saving

After capturing the image it needed to be saved. This is done using the ‘Q’ key on your keyboard. This key will start the image processing. First step will start with the converting of captured RGB image into grayscale image. Now this grayscaled image is resized into 32X32 pixels. After resizing image will be saved and message will be displayed to user through the Terminal.

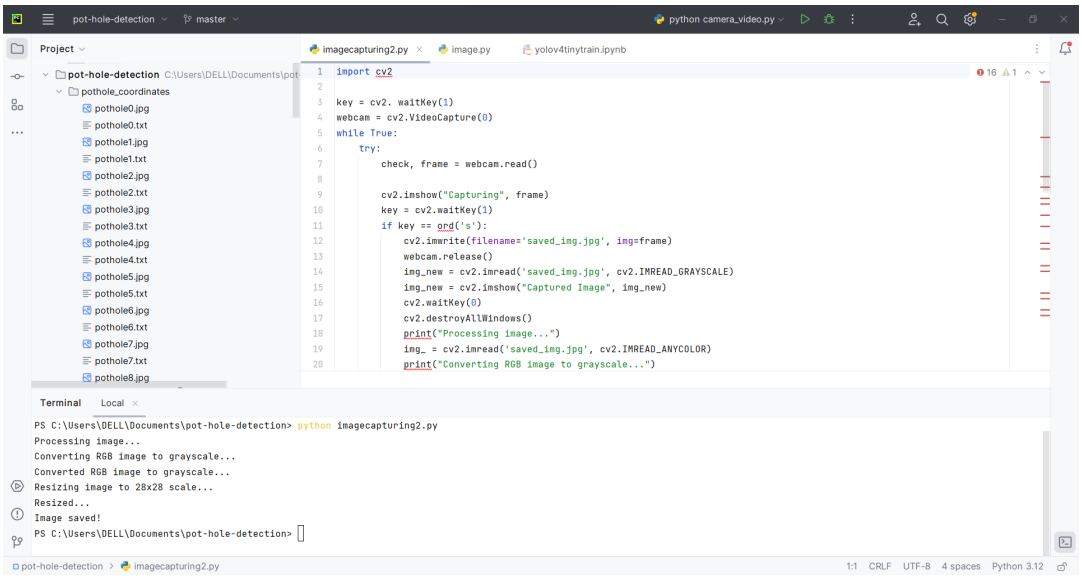


Figure 8.5: Image Processing and Saving

8.3 Displaying Result

After the successful image saving, it’s time to display the result which is the given image contains a pothole or not. For displaying the result, we have to execute the second source code image.py. This source code contains DNN and tiny-YOLOv4 configuration files. Once again we have to type command in the terminal for its execution. Type “python image.py” as shown in the figure 8.6

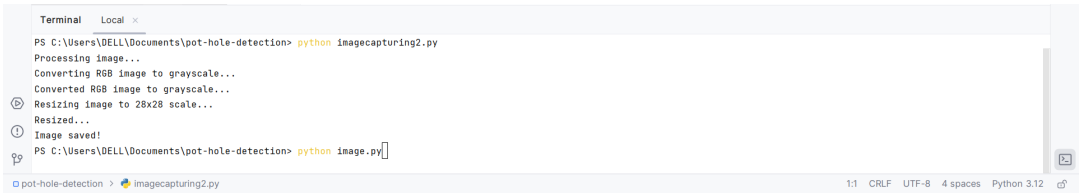


Figure 8.6: Command for Execution of Second Code

Now the result will be displayed into another python window as shown in figure 8.7. We know that, the given input image in this demonstration was containing a pothole. You can see this pothole is highlighted using a green frame around it in the result image.

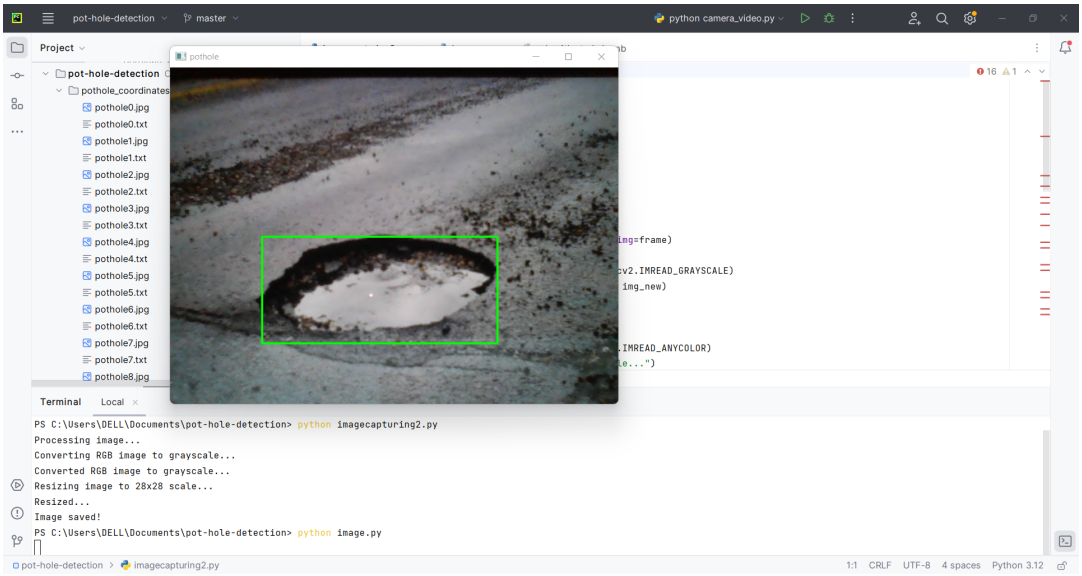


Figure 8.7: Result Image

8.4 Working Flowchart

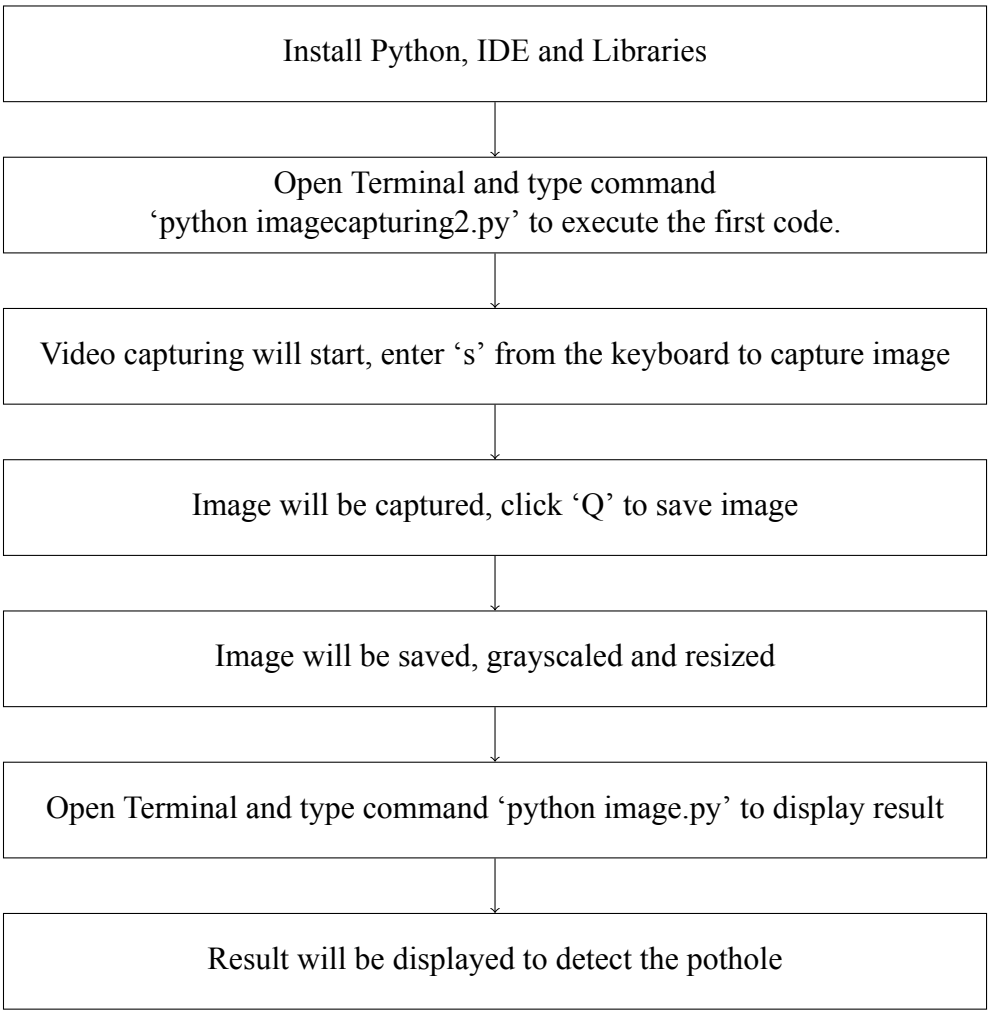


Figure 8.8: Complete Working Flowchart

Chapter 9

ANALYSIS

9.1 Scope of the Project

The main goal of the 'Pothole Detection System' is to developed image processing based software system which detects the potholes. This project is used to spot the potholes on the road surface using machine learning. This project deals with the study of image processing algorithms. The work is mainly based on the convolutional neural networks. CNN works through the YOLOv4 tiny algorithm. The input image is first processed, grayscaled and resized and the oputput image is displayed.
DNN also called as DeepNets is a neural network which has a certain level of complexity composed of several layers. Multiple hidden layers in this neural network helps to extract the desired features from the input image. We used CUDA DNN functions in the source code in project for importing the weight and configuration file and developed model from that file.
This project also utilizes the OpenCV library in python programming language. This libary is very helpful in order to use its functions for image capturing and processing. We have used various functions as mentioned in the methodology and implementation chapters.
This project's idea is based on the safety of travelling of people through the road transport by avoiding the accidents. Majority of accidents happen due to the potholes on the road surfaces[33]. If this potholes avoided, many accidents will not happen and many lives can be saved. Hence detection of the pothole is done automatically by this system.
This system can also be set up in the vehicle. It requires only an operating system which is easily available now a days in the car's dashboard or control panel. It can warn the driver about the upcoming pothole on the road which should be avoided.
In case of self driving cars, it is a vehicle capable of sensing its environment and operating without human involvement[34]. This car runs fine on a smooth and pothole-less road. But on damaged road surface, it may be stuck into the pothole or can passed through it which result into the discomfort of the passenger and damage of car assembly. Hence pothole detection in advancement of travelling of road is required in these cars.
Manual detection of potholes is tedious and challenging task which includes high labour efforts and exorbitant price. It also very difficult to store and categorized the data received from this inspections. Hence these should be spotted autonomously and it should collect the data which can be further moved to road regulatory authority. The proposed system reduces the human efforts and detects the potholes ensuring the safety of traveller.

9.2 Advantages of the System

The 'Pothole Detection System' has numerous advantages which can be described as follow:

- **Less hardware requirement:** It only requires a web camera and a system to which it is connected and code executed. Hence the hardware requirement is very less. Also it does not require any electronic circuitry and connection which reduces the complexity of the system.
- **Cost-effective:** Due to less hardware requirement, cost of the system is very low. It only requires a webcam which ranges from Rs. 1000 to 5000 depending on the resolution[35]. We have used Hikvision DS-U02 which is priced Rs. 3000.
- **Less complex algorithm:** It makes use of YOLOv4-tiny algorithm which is very less complex as compared to other YOLO versions. It is a compressed version of YOLOv4 designed to train on machines that have less computing power[36].
- **Simple to operate:** the system is very simple to operate. You just need to execute the code using simple commands to detect the pothole.

Chapter 10

CONCLUSION

The pothole detection using artificial intelligence methods can help in better maintenance of the road conditions especially in developing countries where resources are limited. For this purpose, the proposed system based on convolutional neural networks using image processing does have the potential to compete with the existing techniques of pothole detection. The proposed 'Pothole Detection System' using CNN based on YOLOv4-tiny model has several advantages over the other techniques such as more accurate, low cost, less complex and also does not involve risk of passing through potholes. Further, this work can be extended to detect the region of pot- holes after classifying an image as pothole and furthermore param- eters can also be detected like the severity of potholes on the basis of which it can be figured out which area requires urgent repair work. This system will surely minimize the risk of road accidents caused by potholes and will help for the society.

Chapter 11

CERTIFICATES OF PROJECT COMPETETION

Gokhale Education Society's
R. H. Sapat College of Engineering, Management Studies and Research Nashik-422005
Department of Electronics and Telecommunication Engineering

CERTIFICATE
OF APPRECIATION

THIS CERTIFICATE IS AWARDED TO

Bagul Ashish Suresh

for participation in project competition of
G-ESTRONICA Organized by department of Electronics &
Telecommunication Engineering in association with Electronics &
Telecommunication Engineering Student's Technical Alliance (ESTA) during
20 May 2022-23

(Dr. S. P. Agnihotri)
Head, E&TC

(Dr. P. C. Kulkarni)
Principal

(Prof. P. M. Deshpande)
Director (P)

Figure 11.1: Certificate awarded to Bagul Ashish Suresh for participating in project competition G-ESTRONICA


Gokhale Education Society's
R. H. Sapat College of Engineering, Management Studies and Research Nashik-422005
Department of Electronics and Telecommunication Engineering

CERTIFICATE
OF APPRECIATION

THIS CERTIFICATE IS AWARDED TO

Gaikwad Rohit Dadaji

for participation in project competition of
G-ESTRONICA Organized by department of Electronics &
Telecommunication Engineering in association with Electronics &
Telecommunication Engineering Student's Technical Alliance (ESTA) during
20 May 2022-23

(Dr. S. P. Agnihotri)
Head, E&TC

(Dr. P. C. Kulkarni)
Principal

(Prof. P. M. Deshpande)
Director (P)

Figure 11.2: Certificate awarded to Gaikwad Rohit Dadaji for participating in project competition G-ESTRONICA

Figure 11.3: Certificate awarded to Maharshi Yadnyesh Pramod for participating in project competition G-ESTRONICA

REFERENCES

[1] *https://www.worldbank.org/en/news/feature/2011/09/23/india-transportation*

[2] *https://morth.nic.in/road-transport*

[3] *https://www.trawell.in/blog/tourist-spots-close-to-mumbai-pune-expressway*

[4] *https://trid.trb.org/view/451887*

[5] Robert A. Eaton, Robert H. Joubert and Edmund A. Wright, *"Pothole primer: A public administrator's guide to understanding and managing the pothole problem*, Special Report 81-21, US Army Corps of Engineers Cold Regions Research & Engineering Laboratory, September 1981.

[6] *https://www.deccanherald.com/opinion/first-edit/india-must-rise-out-of-the-potholes-1139374.html*

[7] *https://www.hindustantimes.com/cities/delhi-news/maharani-bagh-battles-pothole-riddled-roads-heavy-vehicles-rush-101665600113892.html*

[8] *chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ncrb.gov.in/sites/default/files/ADSI-2021/adsi2021_Chapter-1A-Traffic-Accidents.pdf*

[9] *https://www.constructionworld.in/latest-construction-technology/Say-NO-To-Potholes/11831*

[10] *chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://morth.nic.in/sites/default/files/RA_2021_Compressed.pdf*

[11] *https://www.deccanherald.com/opinion/first-edit/money-down-the-pothole-1193618.html*

[12] Chi-Wei Kuan, Wen-Hui Chen, Yu-Chen Lin, *"Pothole Detection and Avoidance via Deep Learning on Edge Devices"*, International Automatic Control Conference (CACS), IEEE, 2020.

[13] Dharneeshkar J, Soban Dhakshana V, Aniruthan S A, Karthika R, Latha Parameswaran, *"Deep Learning based Detection of potholes in Indian roads using YOLO"*, International Conference on Inventive Computation Technologies (ICICT), IEEE, 2020.

[14] Dhwani Desai, Abhishek Soni, Dhruv Panchal, Sachin Gajjar, *"Design, Development and Testing of Automatic Pothole Detection and Alert System"*, 16th India Council International Conference (INDICON), IEEE,2019.

[15] Byeong-ho Kang, Su-il Choi, *"Pothole Detection System using 2D LiDAR and Camera"*, Ninth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2017.

[16] Sudarshan S Rode, Shonil Vijay, Prakhar Goyal, Purushottam Kulkarni, Kavi Arya, *"Pothole Detection and Warning System: Infrastructure Support and System Design"*, International Conference on Electronic Computer Technology, 2009.

[17] Roopak Rastogi, Uttam Kumar, Archit Kashyap, Shubham Jindal, Saurabh Pahwa, *"A Comparative Evaluation of the Deep Learning Algorithms for Pothole Detection"*, 17th India Council International Conference (INDICON), IEEE, 2020.

[18] Vigneshwar. K, Hema Kumar. B, *"Detection and Counting of Pothole using Image Processing Techniques"*, IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE, 2016.

[19] Kavitha R, Nivetha S, *"Pothole and Object Detection for an Autonomous Vehicle Using YOLO"*, 5th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2021.

[20] *https://www.hikvision.com/en/products/Turbo-HD-Products/Turbo-HD-Cameras/webcam-series/ds-u02/*

[21] *https://www.python.org/*

[22] *https://www.python.org/downloads/release/python-390/*

[23] *https://www.jetbrains.com/pycharm/*

[24] *https://www.v7labs.com/blog/convolutional-neural-networks-guide*

[25] *https://www.mdpi.com/2072-666X/13/11/1983*

[26] *https://developer.nvidia.com/cuda-toolkit*

[27] *https://opencv.org/*

[28] *https://geopy.readthedocs.io/en/stable/*

[29] *https://neptune.ai/blog/image-processing-python*

[30] *https://towardsdatascience.com/transfer-learning-on-greyscale-images-how-to-fine-tune-pretrained-models-on-black-and-white-9a5150755c7a*

[31] *https://www.researchgate.net/figure/Conversion-of-RGB-to-grayscale-pixels-This-image-is-best-viewed-in-color_fig3_353507293*

[32] *https://xilinx.github.io/Vitis-Tutorials/2021-1/build/html/docs/Hardware_Acceleration/Introduction/07-image-resizing-with-vitis-vision.html*

[33] *https://www.legalserviceindia.com/legal/article-3376-the-pothole-crisis-an-inconspicious-issue.html*

[34] *https://www.synopsys.com/automotive/what-is-autonomous-car.html*

[35] *https://www.pricedekho.com/web-cams/web-cams-price-list.html*

[36] Upesh Nepal, Hossein Eslamiat, *"Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs"*, MDPI, January 2022.