Pothole Detection System

P. D. Lokhande*[1], Ashish Bagul[2], Rohit Gaikwad[2], Yadnyesh Maharshi[2]
[1]*Assistant Professor, Electronics and Telecommunication Department, GES's R. H. Sapat College of Engineering, Nashik*
[2]*Department of Electronics & Telecommunication, GES's R. H. Sapat College of Engineeering, Nashik, Maharashtra, India*

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. 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 passengers to travel safely without plunged in potholes.

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

I. INTRODUCTION

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 used mode of transport is by land. The two main forms of land transport can 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.

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. 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. 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.

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.

II. LITERATURE SUREY

In this paper, Chi-Wei Kuan, Wen-Hui Chen, and Yu-Chen Lin[1] 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. Dharneeshkar J, Soban D V, Aniruthan S A, Karthika R, and Latha Parameswaran[2] 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. Dhwani Desai, Abhishek Soni, Dhruv Panchal, and Sachin Gajjar[3] 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. Byeong-ho Kang and Su-il Choi[4] 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. Sudarshan S Rode, Shonil Vijay, Prakhar Goyal, Purushottam Kulkarni, Kavi Arya[5] 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. Roopak Rastogi, Uttam Kumar, Archit Kashyap, Shubham Jindal, Saurabh Pahwa[6] 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. Vigneshwar. K and Hema Kumar. B[7] 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. Kavitha R and Nivetha S[8] 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.

III. SCOPE OF PROJECT

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 PyCharm. 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.

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. If these 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.

IV. METHODOLOGY

1. Convolutional Neural Networks

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.

In a regular neural network, there are three types of layers as shown in figure 1:

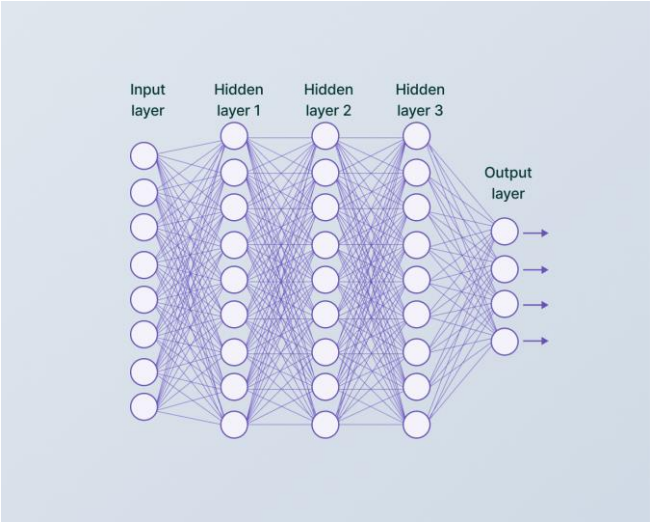

Fig 1: Convolutional Neural Networks

Input Layer: It is 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 layer 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.

2. 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. Table shows the overview of 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

3. Requirements

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. The project source code is needed to be run and executed on a system such as PC or laptop. The software requirements are python and IDE to run the code. 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.

Python is a popular general-purpose interpreted, interactive, object-oriented, and high level programming language. 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.

4. Libraries

The installation of two libraries in the IDE is required. These libraries are OpenCV and Geocoder. OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly for real-time computer vision. 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

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. Geocoder is installed in system using PyCharm Terminal by command: pip install geocoder

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. 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

5. Block Diagram

The above figure 2 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

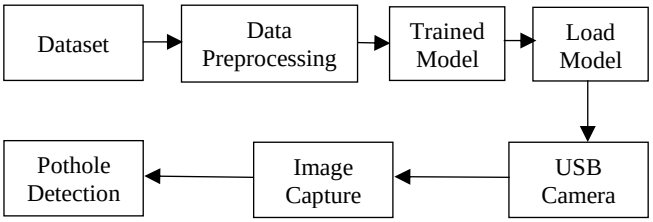

Fig. 2: Block Diagram

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.

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.

```
# 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)
```

Fig 3: DNN function in OpenCV

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.

V. IMPLEMENTATIONS

1. Dataset

Fig 3: 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. 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 3 shows sample images of pothole dataset. These images are collected from iStock.

2. Image Capturing in OpenCV

The OpenCV library has various methods and functions to capture the image. It is also useful in video capturing which can be further helpful to processing. Hence these functions of OpenCV are used for taking input as an image to the model from which the model can be trained.

```
import cv2

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

Fig 4: Image Capturing Function of OpenCV

VideoCapture(0) is used for external web camera and VideoCapture(1) is used for using default system's embedded camera. These functions are included in OpenCV.

3. 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.

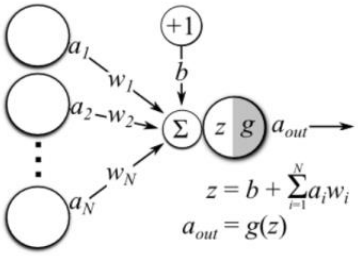

Fig 5: Operation in Single Neuron

The basic working of the neural network is as follows:

i. Let us 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.

ii. Each of these channels is assigned a numerical value known as weight.

iii. The inputs are multiplied by the corresponding weights and this weighted sum is then fed as input to the hidden layers.

iv. The output from the hidden layers is passed through an activation function which will determine whether the neuron will be activated or not.

v. 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.

vi. In the output layer, the neuron with the highest value predicts the output. These outputs are the probability values.

vii. 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.

viii. 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.

ix. 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 5, ai's is the set of inputs, wi's are the weights, z is the output and g is any activation function.

4. 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.

```
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...")
```

Fig 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)

Benefits of 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 reduces the dimension. For example, In RGB images there are the three-color channels and three dimensions while grayscale images are single-dimensional.
- 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.

5. Image Resizing

Image resizing refers to the scaling of images. Scaling comes in handy in many images 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.

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.

```
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!")
```

Fig 7: Image Resizing in OpenCV

In our project code we have used the following function for image resizings:

img_ = cv2.resize(gray,(32,32))

VI. WORKING

1. *Image Capturing*

First, 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. Type Code imagecapturing2.py. Now video capturing will start. Capture the image using S from the keyboard.

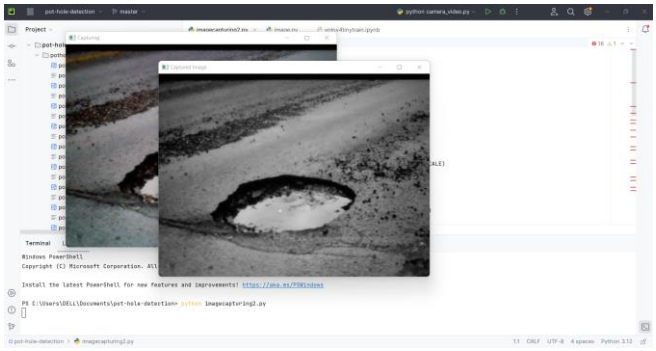

Fig 8: Captured Image

As shown in figure 8, the captured image will be shown in another python window.

2. *Image Saving*

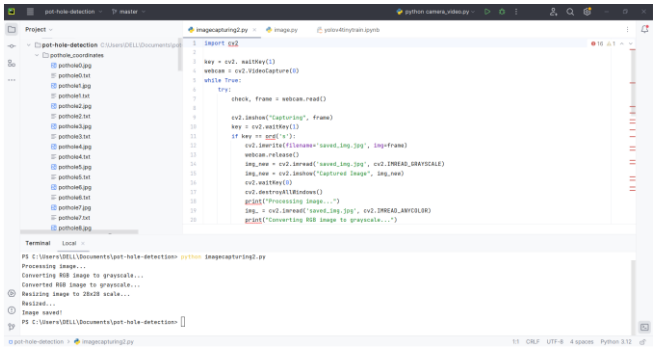

Fig 9: Image Saving

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.

3. *Displaying Result*

For displaying the result, we have to execute the second source code image.py. Type "python image.py" in the terminal.

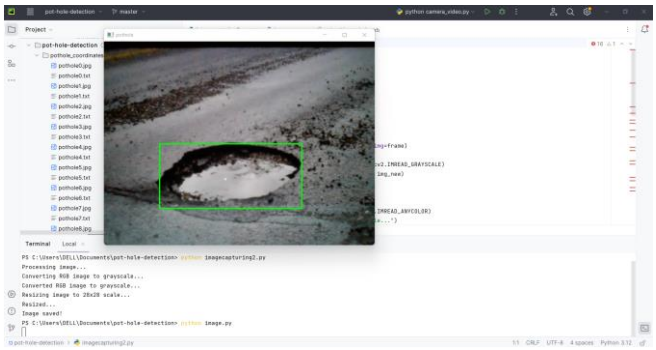

Fig 10: Result

Now the result will be displayed into another python window as shown in figure 10. 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.

4. *Flowchart*

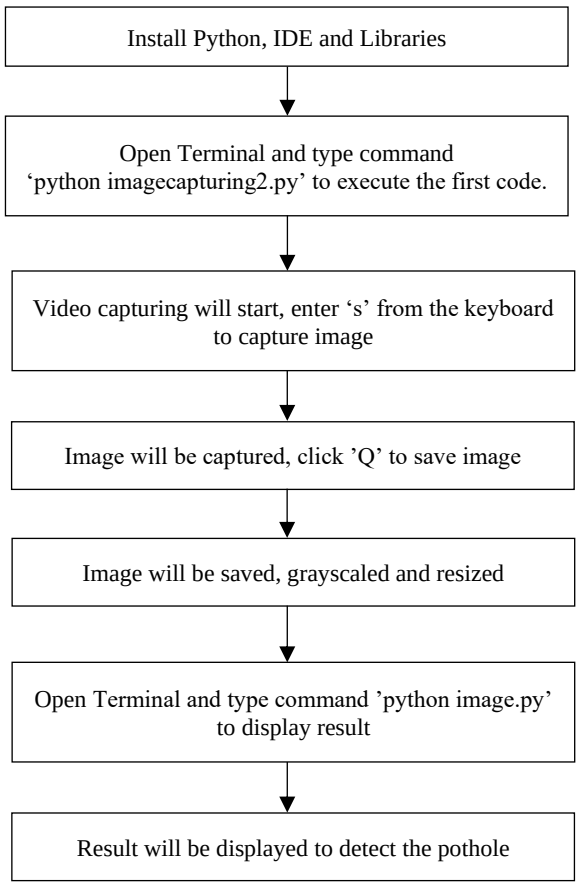

Fig 11: Complete Working Flowchart

Figure 11 shows the complete working flowchart of the Pothole Detection System. It starts with the installation of Python, Python IDE as PyCharm and installing libraries in the IDE. The libraries required are OpenCV and geocoder which are important for the image processing. After installation type commands to execute the code in the terminal and capture image through the S using the keyboard. Save the captured image using Q from the keyboard. Generate result using the second command using the terminal of the IDE. The result will be displayed in image format using a green frame around the pothole.

VII. 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 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 parameters can also be detected like the severity of potholes based on 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.

VIII. REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.