AI-Based Android Application for Pothole Detection

Raghav Bhatia
Cse-AI &ML, Chandigarh University
Punjab, India

Ishal Walia
Cse- AI & ML, Chandigarh University
Himahal Pardesh, India

Hardik Sharma
Cse- AI & ML, Chandigarh
Himachal Pardesh, India

Abstract— Potholes are a type of defect that can damage the vehicle and pose a danger to the driver. Conflicts on the roads pose a danger not only for drivers but also for pedestrians. Floods cause many accidents and deaths every year around the world. In addition, these holes can cause serious damage to your vehicle, and this can be fixed by paying a large amount of money to the automobile company. These deficiencies are caused by the use of poor quality building materials and heavy rainfall. Nearly 9,500 people died due to potholes in 2017, according to a report submitted by the research wing of the Ministry of Road Transport and Highways in New Delhi. These incidents could have been prevented if the authorities had previously taken strict measures to monitor and repair these wells.

Around the world, correct road management is a very difficult task and an important issue. There is an urgent need for a powerful system that can detect irregularities in advance and warn the driver. This article presents a powerful application that can instantly control these conflicts without interruption and can be easily downloaded to the car ecosystem on Google Play Store. The app uses deep learning and machine learning to detect these anomalies and alert users in advance.

Keywords: Application Programming Interface, Deep Learning, Convolution Neural Network, TensorFlow, Tflite, Neural Networks, Pothole detection.

I. INTRODUCTION

A pothole is a distinct kind special organisms caused by severe weather conditions and can be detected by depressions in the road surface (on a flat surface at least 150 mm in size). According to a recent study by the World Health Organization, traffic accidents are thought to be responsible for 1.35 million deaths every year. Poor driving, bad weather conditions, etc.

Major accidents occur due to this. Ponds and other irregularities in the road are by far the most important in these situations. The second largest highway in the world is in India. This shows how important roads are to the Indian economy. According to reports, India's transportation sector grew by 10 percent, while the growth of the total market during the year was 6 percent.

Today the government spends a lot of money to build this road network, but the real problem arises due to the maintenance of these roads. If left unchecked, puddles resulting from poor maintenance can cause poor road conditions and lead to fatal accidents and serious injuries. According to government statistics, there were 37,000 injuries and 11,836 deaths due to flooding in India between 2013 and 2016. Although we cannot eliminate potholes, we can reduce the number of traffic accidents by examining potholes. It will notify the user if there is a problem.

The four main phases of the exploration process are: data collection, preliminary data, extraction and pit classification. We need enough data to detect potholes during data collection. In the data preprocessing step, we prepare for the third step by applying techniques such as data cleaning, data enrichment, data reduction, and data transformation by using previously created data. The feature extraction step is a technique used to find patterns that can distinguish anomalous pools. The pothole classification phase using pothole detection algorithms helps prove the existence of potholes. Determining the condition of a lake requires careful attention to data and processing. Below is an overview of the pothole detection process.. We can see this in **Figure 1**.

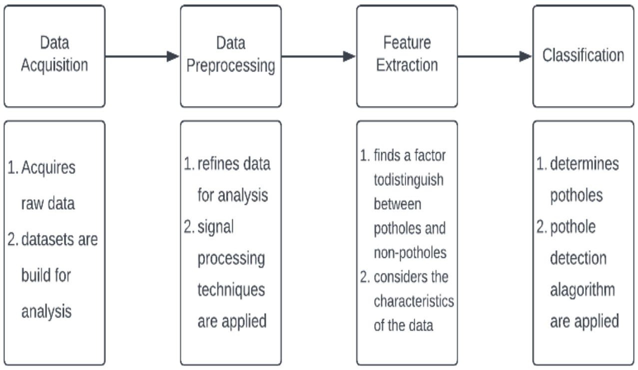


Fig 1: Phases of Pothole Detection

This research paper proposes an android application StreetSafe which is an end-to-end pothole detection application, which can be downloaded on any Android based device and which can help to combine all these processes and even save lives of innocent drivers and pedestrians. The methods proposed in this paper are proven to be 97% accurate in detecting these anomalies and alerting the user about these anomalies on the road.

II. LITERATURE SURVEY

In recent years, many researchers have been working on more accurate methods to detect gaps in time. Notifications using mobile applications, Open-CV based technology, sensor based technology, etc. to provide end-to-end solutions to users. Many ideas have been developed. These methods can be divided into three groups: 3D structure-based methods, vibration-based methods and optical methods. The visual method uses images or videos as input to determine the presence of potholes. The simplest and cheapest method is this. Because it primarily bases its analysis on the vibration of the acceleration sensors, the vibration-based technique has the drawback of not providing the precise shape and depth of potholes. The stereo-vision-based 3D reconstruction-based approach estimates the volume and determines the precise shape of the potholes. This is the most accurate model but is not cost efficient to implement. The summary of these methods is shown in **Figure 2 and Table 1**.

Fig 2: Characteristics of automated pothole-detection methods

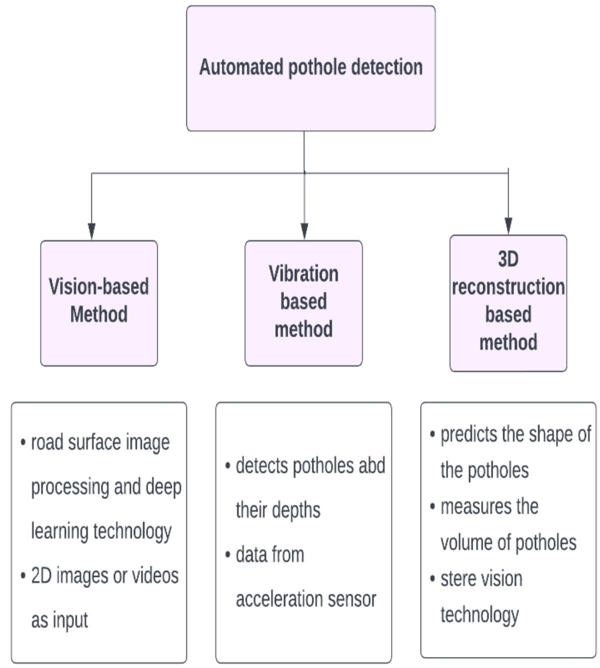


TABLE 1: The strengths and weakness of automated pothole-detection methods

Methods	Strengths	Weakness
Vision-based method	More affordable than 3D reconstruction based method Best for identifying potholes and estimating their size and shape	Cannot measure volume and depth of the potholes
Vibration-based method	Most cost-effective among the three methods Required small storage	Cannot provide the exact shape of pothole Real time detection cannot be done
3D reconstruction based method	Measures the shape of the potholes most accurately among the three methods	Most expensive among all the three methods

There has been a lot of research over the last decade trying to create an end-to-end system that could assist drivers and save innocent lives. In the following sections, we divide their work into three main areas: the authors, the research methods they used, the data they used to find the pits, and the activity and laboratory they used in the process. Pothole detection algorithm. After analyzing the dataset, we divided the classification data into two parts: all images used for training data and all images used to test the data. In the performance section, we use three metrics such as accuracy,
precision and recall.
This is clearly showcased in Table 2.

TABLE 2: A detailed comparison of the existing pothole detection techniques in terms if detection techniques, performance (accuracy, precision, recall), data set used and experimental circumstance.

S.NO.	Authors	Detection Techniques	Dataset	Performance	Experimental Circumstance
1.	**Lim et al [1]**	• Vision Based Pothole Detection • Training and testing is based on YOLO v2 • Uses a denser-grid model based on YOLO v2	Total: 1199 images Training : 996 images Testing:203 images	Accuracy: 67.7% Precision: 70.4% Recall: 74.9%	Used a GeForce GTX 1080
2.	**Baek et al [2]**	• Vision Based Pothole Detection • Feature Extraction is done using edge detection • Training and Testing is done using YOLO	Total: 13,376 images Training : 9364images Testing:2675 images	Accuracy: 77.8% Precision: 83.4% Recall: 72.9%	Used Windows 10, Intel Core i5-8440, 16GB RAM
3.	**Hanshen Chen et al [3]**	• Vision Based Pothole Detection • Training an testing is based on Location-aware CNN • Makes use of a component-primarily based type community(PCNN)	Total: 5676 images Training :4026 images Testing:1650 images	Accuracy: 95.0% Precision: 95.2% Recall: 92.0%	Used a GeForce GTX 1080
4.	**Nhat-Duc Hoang [4]**	• Vision Based Pothole Detection • Training an testing is based LS-SVM and ANN	Total: 200 images Training :160 images Testing: 40 images	Accuracy: 85.25% Precision: 92% Recall: 88%	N/A
5.	**Azza Allouch [5]**	• Vibration Based Pothole Detection • Data pre-processing done based on low pass filter • Feature Extraction is done based on Fourier transform and correlation • Classification is based on C4.05 decision tree, SVM and Naïve Bayes	Total: 2000 images Training : N/A Testing: N/A	Accuracy: 95.5% Precision: 98.5% Recall: 95.3%	Used cell phone mounted on the auto, android-based app operating on telephone
6.	**Chao Wu [6]**	• Vibration-based totally Pothole Detection • Statistics pre-processing is finished the usage of sliding window. • Feature Extraction is done using Fourier remodel and correlation • Education and trying out is done primarily based on Linear Regression, SVM and Random woodland	Total: 4088 images Training :3061 images Testing:474 images	Accuracy: 95.2% Precision: 85.1% Recall: 73.4%	Used experimental automobile, a telephone placed at the again seat and android primarily based app.
7.	**Amita Dhiman [7]**	• 3-D-reconstruction based Pothole Detection • Makes use of single-body stereo vision based totally method • Transfer learning with masks R-CNN and YOLOv2	N/A	Accuracy: 55.0% Precision: 45.8% Recall: 69.0%	Used GeForce GTX 1080 GPU and TESLA K80 GPU
8.	**Jinchao Guan [8]**	• 3D-reconstruction Based Pothole Detection • Training an testing is based on Principal Component Analysis • Uses Pixel-level pavement detection.	N/A	Accuracy: 95.9% Precision: 96.3% Recall: 95.5%	A vehicle equipped with GoPro HERO8

III. INFERENCE FROM RELATED WORK

As show in Table 2, the studies, Current research does not aim to establish a monitoring system that can accurately identify these wells. The above studies do not have much in terms of accuracy, precision and memorability. Small differences in these parameters can be dangerous when human life is at stake. There is a great need for a system that is robust, fast, and has higher accuracy, precision and repeatability. While existing studies generally provide accurate results based on images of test data, the method proposed in this research paper uses live images in videos. The accuracy given in this research paper is based on the image of the smartphone camera sensor or the vehicle's speed camera. Therefore, it is safe to conclude that there are other better ways to detect potholes.

IV. PROPOSED WORK

The pothole detection method detects potholes in real time. In this method, technologies such as "CNN", "Tflite", "Tfrecord" and "TensorBoard", "MobileNet" and "SSD" are used. The main advantages of combining these methods are that the training time is short and the training process is very simple and accurate. The data used to illustrate the model is taken from Kaggle.com. The data then goes through data preprocessing and feature extraction so that it can be used by the model. The transformed data is then sent to generate output through a region-based neural network (R-CNN), Tensorflow model, which is then output to the “StreetSafe” mobile app. The training model is then used to evaluate the smartphone's live camera and the car's camera sensors. These lakes are detected and confirmed using a red box connected around the lake, and the distance of the vehicle to the lake is also reported. The application also warns users with audio alerts, ensuring that drivers are aware of potholes in advance and can make decisions accordingly.

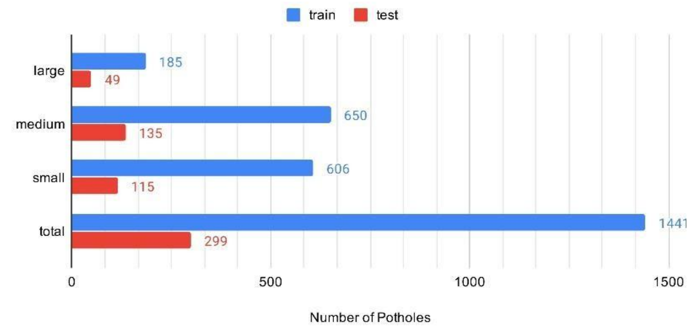
Data from Kaggle is divided into three groups: small, medium and large. The size of live pixels is used to create this group size. The number of pixels is determined by measuring the length of the image to 300 pixels. Pixels and groups of pits are shown in Table 3 below.

TABLE 3: Pixels Occupied v/s Categories of potholes

S.No.	Categories	Pixels Occupied
1.	Small	Area<= 32^2
2.	Medium	32^2< area<= 96^2
3.	Large	Area > 96^2

The distribution of the training and testing data over different categories is briefed under the Fig 3.

Fig 3: Training and Testing data brief



➔ Technologies Used

The application presented in this article uses languages such as Java and Python to build applications and create tests for mobile applications. This application is owned by Google Inc. It was developed using Android Studio (Dolphin), a mobile software development application developed by . It provides many useful functions to be efficient when developing applications such as integrated sites and flexible Gradle-based system.

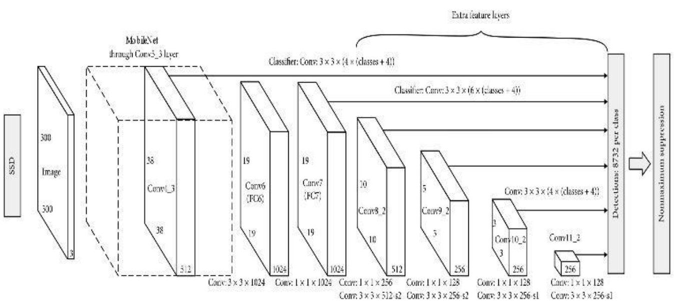
Use flexible learning methods to create goals. Describe the characteristics of the pit finding area. The fully trained MobileNet model is used to extract features. The program estimates the bounding box around pits in various images.

1. **CNN:** Deep learning algorithms such as convolutional neural networks are ideal for identifying and classifying images. Convolution layer, pooling layer and the whole process are connected with three main layers. The core of the CNN is the convolution layer, where multiple filters are used to remove features. The output is passed to a pooling layer where the spatial dimension is reduced to subsample objects. The output is then predicted from a fully connected cluster.

2. **MobileNet:** Object detection, feature classification, face etc. It is a high-performance, simple and effective convolutional neural network that can be used for many real-world applications such as. It was created by Google researchers in 2017. MobileNets uses a total of 4.2 million parameters, which is a significant reduction compared to other CNN architectures. It also provides the ability to adjust the size of the model.Object detection, feature classification, face etc. It is a high-performance, simple and effective convolutional neural network that can be used for many real-world applications such as. It was created by Google researchers in 2017. MobileNets uses a total of 4.2 million parameters, which is a significant reduction compared to other CNN architectures. It also provides the ability to adjust the size of the model.

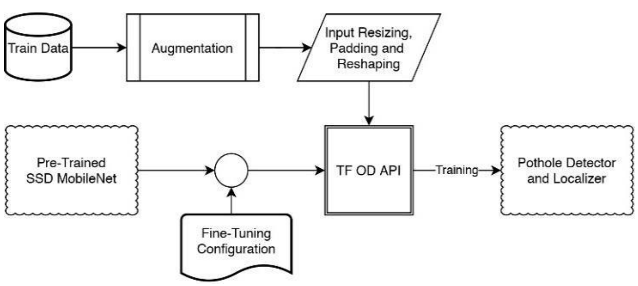
3. **SSD (Single Shot Multibox Detection):** Deep learning is used to find mobile video content. It is a stone-material process that has so far proven to be very effective in solving difficult problems. The backbone version and SSD head consist of two parts. As a function extractor, the kernel model is an inexpensive version of image distribution. This skeleton has a layer called SSD Head, whose output is defined as containers. Fig. 4 depicts the complete structure of the SSD.

Fig 4: Architecture of Single Shot MultiBox Detection



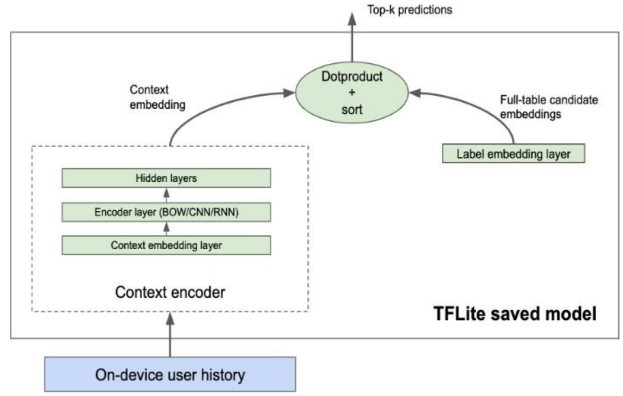
4. **Tensorflow API:** The first open source library developed by researchers and experts in the Google Brain team at the Google Research Machine Group. Do research on machine learning and deep neural networks. Tensorflow has APIs in various languages. The Python API is far from easy to use. In this project, a Tensorflow API model is trained using prepared data and some quality metrics. The structure of this API and how it is used in the project is shown below.

Fig 5: WorkFlow of TF OD API



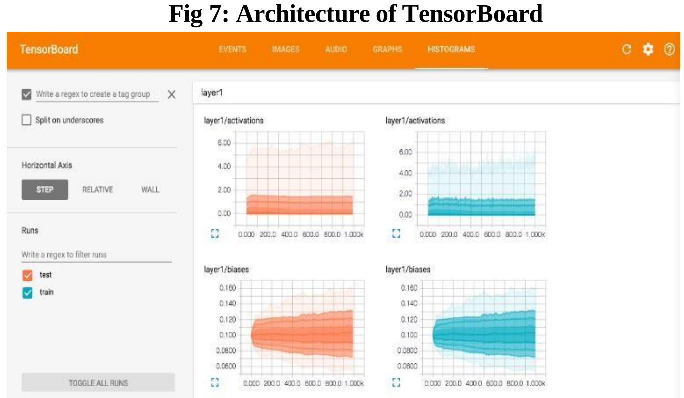
5. **Tflite:** Tensorflow lite is a hard and fast of equipment, which assist in on-device gadget mastering by helping the developers to run complex machine studying and deep gaining knowledge of strategies over their cellular, embedded and side gadgets. the important thing functions of Tflite is that is optimized for on-tool machine studying using five key constraints: latency, connectivity, and privacy, length and energy consumption. it is supported by different platforms along with Android, IOS, Embedded Linus and microcontrollers. It has a completely various language assist together with JAVA, fast, C++ and Python. The architecture of Tflite is showcased in the below **Fig 6**.

Fig 6: Architecture of Tflite



6. **TensorBoard:** TensorBoard is a visualization toolkit, that's used for machine mastering experimentation. It enables in tracking and visualizing metrics which includes loss and accuracy. It visualizes histograms of weights, biases as they exchange over the time. It shows text, audio and video statistics and lots greater. it is used in preparing the visualizations of accuracy, precision and don't forget of the version.

Fig 7: Architecture of TensorBoard



7. **Tfrecord:** all of the data is stored within the shape of tfrecords. It internally uses Protocol Buffers to serialize/deserialize the statistics and the shop them in bytes, as it will consume much less memory area to keep an adequate quantity of records. Protocol Buffers makes use of technology such as JSON & XML.

8. **Mediaplayer:** We have employed MediaPlayer, a component of the Android multimedia framework, which generates an audio warning if a pothole is detected.

V. PROJECTILE ANALYSIS

The work showcased in the research paper is highlighting the abilities of artificial neural networks and machine learning in the field of pothole detection. The mobile application developed has certain steps in order to provide accurate results. Here the input is the dataset, which is discussed earlier, and the output is the detection of the bumps and potholes.

Steps used to create Street Safe:

1. Training model: Training version:

First of all, training materials are developed in a special way. The table below describes the improvements used in the training materials.

a) Horizontal Flip: Flips the pixel line or precision line horizontally.

b) Random Cropping: This is the method we designed to create a random subset of the original image.

This allows further generalization of the version.

c) Resizing aspect ratio: Usually used to have a simple image grid.

d) Zero padding: Used to increase the original entry length. This is particularly compatible with the basis of convolutional layers.

2. Prepare the model before training: The model is created by modifying learning. Transfer learning techniques help data scientists leverage knowledge gained from previous models used for similar projects. This technique can be applied to many techniques such as deep learning models. The purpose of this model is to reinforce the model's knowledge before training while performing different tasks.

In this mobile application, we use single-shot multi-box detection and MobileNet v2 pre-learning model. With every pre-trained model in the Tensorflow Model Well, we have a "pipeline.config" file that is used for configuration information during training and evaluation. Some changes have been made to this document as shown in the table below.

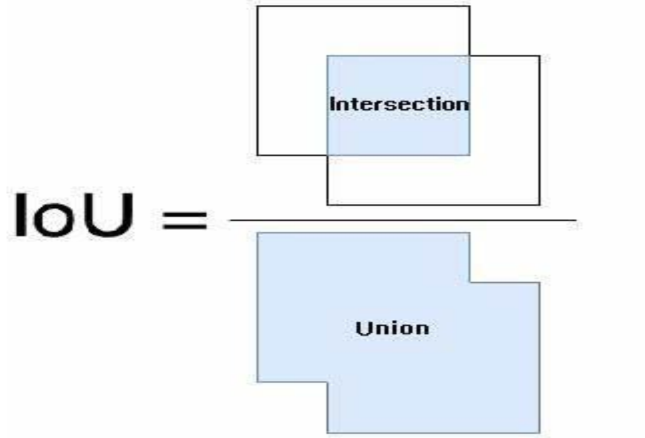
Configuration Options	**Changes Made**
Input-Shape	300,300,3
Num-Classes	1
Batch-Size	24
Min/Max Dimension	300
Initial-Learning-rates	0.005
Decay-Steps	6000
Decay-factor	0.85

1. Run the training process: The Tensorflow model forms the basis of the training process and interprets the important python code. It also provides the ability to save the checkbox in case the mode is affected in any way. Scalar and image values are saved in TensorBoard.

2. Run the evaluation process: After completing the training process, we will continue the verification and evaluation process. Here we use the tensorflow model repository, which always makes the training tool a version checkpoint. This function can display the overall performanceof the model at a certain

3. Evaluation Methods: On this step, we evaluated the overall performance of the model. To take detected bounding box as true positive, different IoU (Intersection over Union) thresholds were taken into consideration. IoU is used to explain the extent of overlap of two packing containers. It's far immediately proportional to the location of overlap. it's far utilized in applications associated with object detection, in which a model is trained to model an output of a subject that suits perfectly around an object. The purpose of that is to improve the prediction of the devices.

$$\text{IOU} = \frac{Area\ of\ Intersection\ of\ two\ boxes}{Area\ of\ Union\ of\ two\ boxes}$$



10 IoU thresholds had been considered from 50% upto 95% with intervals of five% and these values have been then averaged to get the price for IoU. Taking the above IoU thresholds we have:

TP= Number of True Positive Predictions
FP= Number of False Positive Predictions
TN= Number of True Negative Predictions
FN= Number of False Negative Predictions

a) **Precision Calculation:** Precision is defined as the ratio of correctly classified positive samples divided by total number of positive samples(either correct or incorrect)

$$Precision = \frac{TP}{TP + FP}$$

Average precision is the area under the precision-recall curve.

$$AP = \int_0^1 p(r)dr$$

b) **Recall Calculation**: It is determined as the proportion of positively classed positive samples to all positively classified positive samples.

$$Recall = \frac{TP}{TP + FN}$$

Average Recall is the mean of the recall values of all individual classes.

$$AR = 1/N \sum_{i=1}^{N} ri$$

c) **Accuracy:** It is the ratio of correct prediction divided by correct prediction + incorrect prediction. If a model has high accuracy, we can infer that the model makes correct prediction most of the time.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

4. **Creating Application Layouts:** In this process we are creating the Application layouts using the .xml files. In addition, creating the classes, which are to be utilized while creating methods for the execution of the TensorFlow.

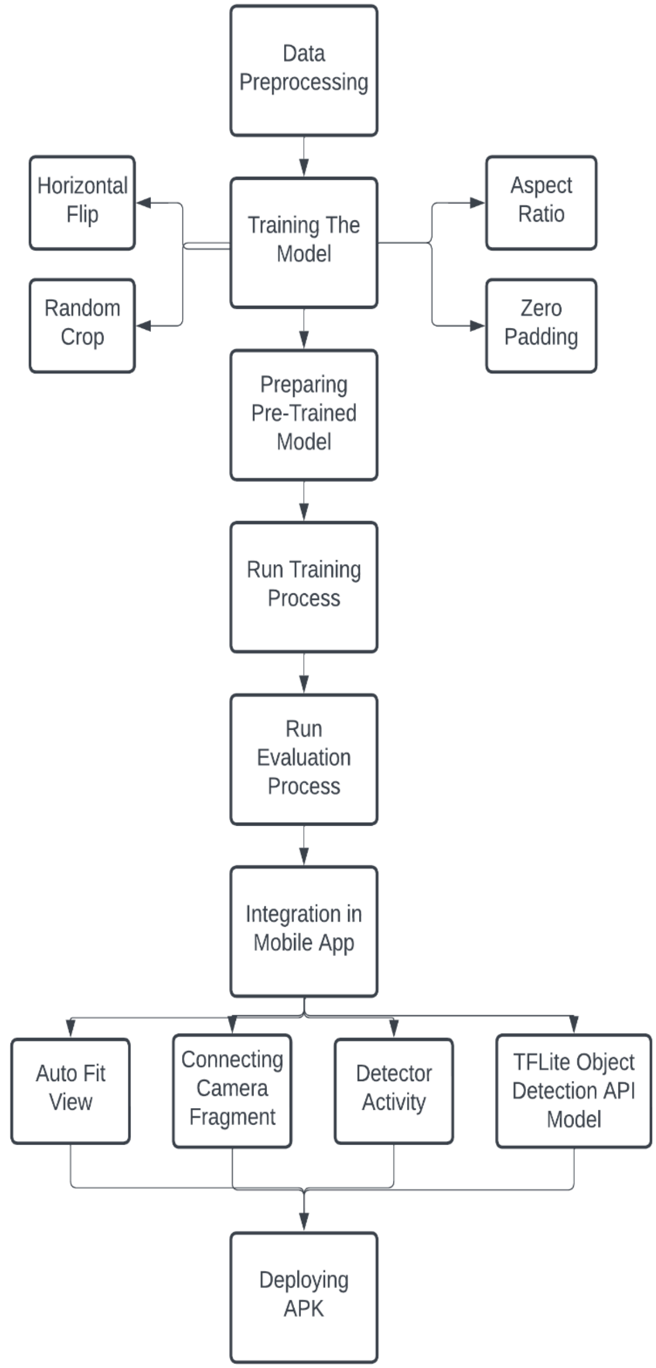
5. **Creating Camera Connection Fragments and Detector Activity:** In this step, we are connecting the xml files with the fragments of Camera. Here we are creating the optimal size of the red boxes that is to be showcased over the detected potholes. Detector Activity is used to import and detect the potholes by using the model, which was created in the previous steps.

6. **Creating TFLite Object Detection API Model in Android Studio:** The API model of the TFLite is made in this step by utilizing the training dataset prepared earlier. The file handles each type of errors using the try and except clauses and recognizes the images and applies the model over it to provide accurate results and sends information to the detector file to mark the region as the potholes by using the red bounding-boxes and alert the user.

7. **Building the Gradle and the Application:** The software is built using Gradle, an automation tool. The code is compiled, linked, and packaged during the construction process. In this mobile Application, we have preferred a stable SDK 29 for which stable TensorFlow API are present and can be utilized easily. These sdk provides the android studio about the total number of features to be used by the app.

8. **Running the App:** The final step is to check that the application is running without any error and is able to detect the potholes without any delay. If any error occur then we are required to debug it and find solution for the same. After successful run over a demo smart-phone, we can use its Apk from anywhere.

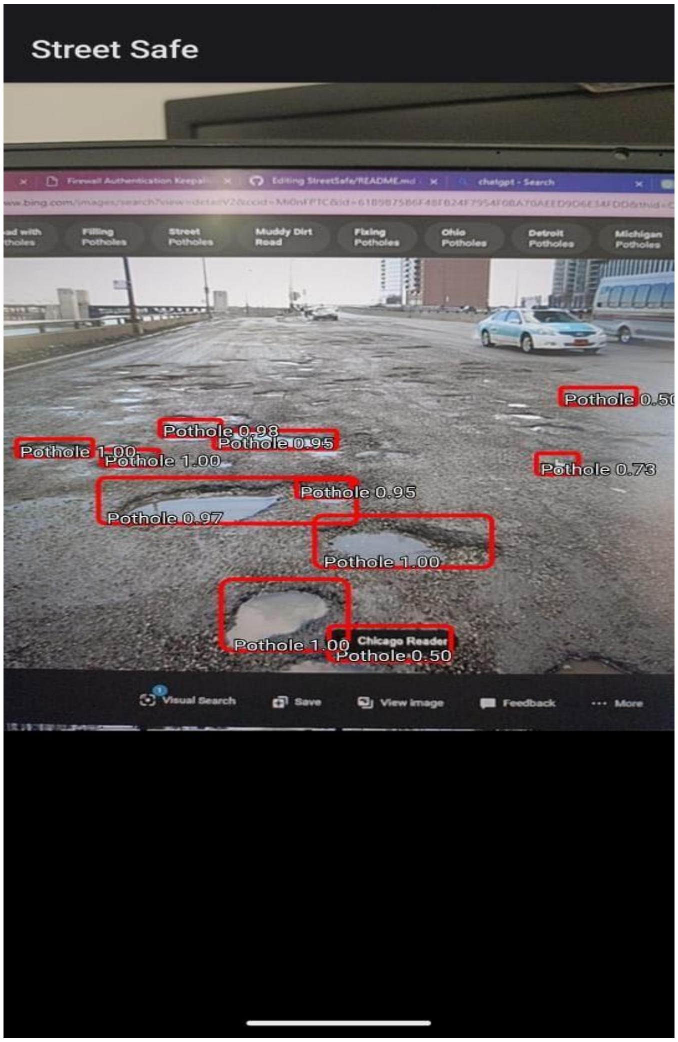
Fig 8: Data Flow Diagram



VI. RESULTS & DISCUSSION

The proposed mobile application and model, which detects the potholes, is shown below in Fig 9.

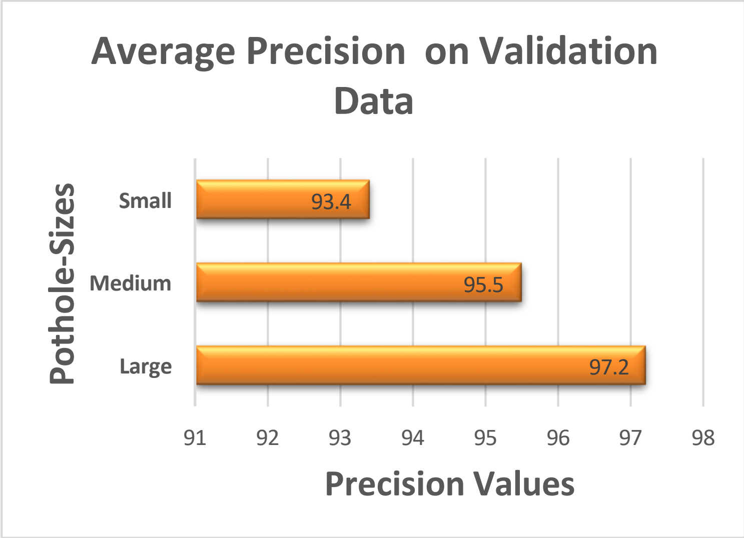
Fig 9: Screen Shot of Mobile Application StreetSafe detecting Potholes



The app screenshot above shows the app's ability to detect potholes and tell the user how close the car is to them. The training and evaluation process was carried out over 10,000 steps. The evaluation process works with the help of TensorFlow repository. The performance of the model is considered as the true average and the IoU threshold as the Good Quality threshold. The average accuracy is 95.7%, higher than other models to date, as shown in Figure

1. The accuracy of the model is approx. 97.3% is the highest rate yet considering previous deep learning models.

Chart 1: Precision V/s Pothole Sizes



It is clearly depicted that the model performs better on large potholes when compared to small potholes. The precision and accuracy values are good, which is unusual for a lightweight model that is based on MobileNet and SSD and designed to work with both high-end and low-end smartphone ecosystems as well as automotive ecosystems.

VII. CONCLUSION

One Found stone plays one of the most important roles in construction and renovation. Thousands of lives can be saved by early detection and prevention of accidents caused by lakes. Thanks to this technology, the effort required to manage potholes is reduced, which is beneficial for road maintenance. Given the current state of technology, it is logical to assume that research in this area will continue.

The end-to-end Android application we describe in this study can be used to instantly detect potholes and send alerts to help drivers avoid potholes. The app can identify potholes more accurately than computer vision models. A convolutional neural network called MobileNet and the Single Shot MultiBox Detector are used in an alternative way to detect potholes. The pothole detection model proposed in this article can be used in many areas such as intelligent traffic management. The data collected by the application can be shared and used to create protection monitoring policies. It can help improve the car's suspension and also improve automatic driving performance.

VIII. FUTURE WORK

We need a 100% accurate system when dealing with human life. In the future, we will work on a stronger, smarter model that will predict more accurately. In the future, we will introduce new features to the app, such as a map that will find and show users where the lakes are along the way. Order through which we can send information about traffic rules. Consumer-focused user interface that is interactive and fun to use. We will create a system where we can collect data from our app users and share it with government officials so they can fix it and provide a better way to the public.

IX. REFERENCES

[1] Lim, K.S.; Kwon, J.W. Detection of Potholes Using a Deep Convolutional Neural Network. J. Univers. Comput. Sci. 2018, 24,

[2] Baek, J.-W.; Chung, K. Pothole Classification Model Using Edge Detection in Road Image. Appl. Sci. 2020, 10, 6662.

[3] Chen, H.; Yao, M.; Gu, Q. Pothole detection using location-aware convolutional neural networks. Int. J. Mach. Learn. Cybern. 2020,11, 599–911.

[4] Hoang, N.-D. An Artificial Intelligence Method for Asphalt Pavement Pothole Detection Using Least Squares Support Vector Machine and Neural Network with Steerable Filter-Based Feature Extraction. Adv. Civ. Eng. 2018, 2018, 7419058.

[5] Allouch, A.; Koubâa, A.; Abbes, T.; Ammar, A. RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope. IEEE Sens. J. 2017, 17, 4231–4238.

[6] Wu, C.; Wang, Z.; Hu, S.; Lepine, J.; Na, X.; Ainalis, D.; Stettler, M. An Automated Machine-Learning Approach for Road Pothole Detection Using Smartphone Sensor Data. Sensors 2020, 20, 5564.

[7] Dhiman, A.; Klette, R. Pothole Detection Using Computer Vision and Learning. IEEE Trans. Intell. Transp. Syst. 2020, 21,3536–3550.

[8] Guan, J.; Yang, X.; Ding, L.; Cheng, X.; Lee, C.S.; Jin, C. Automated pixel-level pavement distress detection based on stereo vision and deep learning. Autom. Constr. 2021, 129, 103788.

[9] Maxwell, A.E.; Warner, T.A.; Guillén, L.A. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review. Remote Sens. 2021, 13, 2450.

[10] B. X. Yu, and X. Yu, Vibration-based system for pavement condition evaluation, In Proceedings of the 9th International Conference on Applications of Advanced Technology in Transportation (2006), 183-189.

[11] K. De Zoysa, C. Keppitiyagama, G. P. Seneviratne, and W.W.A.T. Shihan, A public transport system based sensor network for road surface condition monitoring, In Proceedings of Workshop on Networked Systems for Developing Regions (2007), 1-6.

[12] J. Erikson, L. Girod, and B. Hull, The pothole patrol: using a mobile sensor network for road surface monitoring, In Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (2008), 29-39.

[13] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, Real time pothole detection using Android smartphones with accelerometers, In Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (2001), 1-6.

[14] K. C. P. Wang, Challenges and feasibility for comprehensive automated survey of pavement conditions, In Proceedings of 8th International Conference on Applications of Advanced Technologies in Transportation Engineering (2004), 531-536.

[15] K.T. Chang, J. R. Chang, and J. K. Liu, Detection of pavement distress using 3D laser scanning technology, In Proceedings of the ASCE International Conference on Computing in Civil Engineering (2005), 1-11.

[16] Z. Hou, K. C. P. Wang, and W. Gong, Experimentation of 3D pavement imaging through stereovision, In Proceedings of the International Conference on Transportation Engineering (2007), 376-381.

[17] Ai D, Jiang G, Kei LS, Li C (2018) Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods. IEEE Access 6:24,452–24,463

[18] Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: 2017 International conference on engineering and technology (ICET). IEEE, pp 1–6

[19] Ali A, Zhu Y, Zakarya M (2021) A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. Multimed Tools Appl:1–33

[20] All about Android Platforms. [Online] Available: https://developer.android.com/about

[21] Anaissi A, Khoa NLD, Rakotoarivelo T, Alamdari MM, Wang Y (2019) Smart pothole detection system using vehicle-mounted sensors and machine learning. J Civ Struct Health Monitor 9(1):91–102

[22] Anand S, Gupta S, Darbari V, Kohli S (2018) Crack-pot: Autonomous road crack and pothole detection. In: 2018 Digital image computing: Techniques and applications (DICTA). IEEE, pp 1–6

[23] Arya D, Maeda H, Ghosh SK, Toshniwal D, Mraz A, Kashiyama T, Sekimoto Y (2020) Transfer learning-based road damage detection for multiple countries. arXiv:2008.13101

[24] Bhatia Y, Rai R, Gupta V, Aggarwal N, Akula A et al (2019) Convolutional neural networks based potholes detection using thermal imaging Journal of King Saud University-Computer and Information Sciences