International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT228135

The Assay of Potholes and Road Damage Detection

Ankit Kumar*1, Neha kulshrestha2, Dr. Yusuf Perwej3, Ashish Kumar Srivastava4, Chandan Mani Tripathi5

1* B.Tech Scholar, Computer Science & Engineering, Ambalika Institute of Management & Technology, Lucknow, India

2Assistant Professor, Department of Computer Science & Engineering, Ambalika Institute of Management & Technology, Lucknow, India

3Professor, Department of Computer Science & Engineering, Ambalika Institute of Management and Technology, Lucknow, India

4Assistant Professor, Department of Computer Science & Engineering, Ambalika Institute of Management & Technology, Lucknow, India

5Assistant Professor, Department of Computer Science & Engineering, Ambalika Institute of Management & Technology, Lucknow, India

Article Info
Volume 8, Issue 1
Page Number : 202-211

Publication Issue :
January-February-2022

Article History
Accepted : 05 Feb 2022
Published :17 Feb 2022

ABSTRACT

Road conditions with holes are a common cause of accidents in a traffic environment. For motorcycle riders, car drivers, and other vehicle drivers, this can be fatal. For driving comfort, transportation safety, and infrastructure integrity, road surface monitoring and maintenance are critical. As a result, by identifying pot holes on the highway, this article seeks to develop a road contour damage information system. In this work, we suggest an Android-based application for executing data collecting points for government entities such as the NHAI and municipalities, among others. To create the app, we trained a model to recognize whether or not an image has a pothole. If a pothole is detected, the image is saved on a server, where it can subsequently be retrieved by appropriate authorities for maintenance and analysis. Instead of using the classic paper pen method, government personnel can utilise this mobile app to collect data. We plan to limit the quantity of photographs a person can upload because data can be very large. The findings of this study are expected to play a significant role in guaranteeing safe driving in the future by effectively detecting poor road conditions.

Keywords : Road Surface Damage, Firebase, Image Processing, Pothole Detection, Road Quality Monitoring, Road Damage Detection.

I. INTRODUCTION

Roads are one of the most important aspects of every growing or developed country's social and economic growth [1]. Road damage is a common occurrence in India. However, because we are well aware that maintaining the same by governmental [2] institutions such as municipalities is a major difficulty, many scholars are actively working on inventing efficient and effective strategies to assist municipalities. Most of the Indian people have often encountered damage to a road section both on alternate roads and main roadways. If frequent road inspections are not performed, the state of the roads will deteriorate over time owing to a variety of causes such as weather, traffic, aging, and poor material selection. As a result of the damage caused by potholes, as indicated in figure 1, expensive litigation and damage claims have been filed [3].

Some agencies use cars equipped with multiple pricey sensors and high-resolution cameras to conduct road surveys. Road supervisors and visual inspections are performed by some competent road management. However, these methods are both time consuming and costly. These agencies struggle to keep accurate and up-to-date databases of recorded structural damages even after the inspection is completed [4]. Poor management results in a disorganised and inefficient deployment of resources for road repair. As a result, we require a low-cost, quick, and well-organized system for detecting road damage. We are quite lucky that practically everyone now owns a smartphone with a camera. People have begun to undertake challenges and research in this arena since the introduction of Object Detection algorithms in AI [5].

Figure 1 The Potholes and Road Damage

After examining how government bodies such as the NHAI and Municipal Corporation operate, we discovered that the process of addressing road damage and repairing the road takes a long time because these departments must follow a number of protocols such as cost estimation, forming teams, and locating contractors, all of which take time.

Finally, both governmental agencies and ordinary citizens encounter issues that can be remedied more rapidly if these agencies are informed about the harm as soon as possible.

Later resolving these issues has major consequences, such as accidents resulting in serious injuries and, in some cases, death. The expense of correcting these damages continues to rise as more labour and money are required every day. Other issues that government agencies confront include not knowing which roads to repair first, as certain roads may be able to wait longer than those that require immediate attention [6]. It is now possible for the government to maintain track of road health, but the solutions are too expensive, time consuming, and ineffective. They may choose to appoint a few personnel to keep track of these circumstances, but because there are so many roads and so few employees, this will not always be correct. And fixing this problem with this kind of answer is impractical, or to put it another way, it's a highly sophisticated solution to a very simple problem that may not be particularly effective in the end. So we propose a product (software) capable of addressing these issues, as well as geolocation [7], to government bodies with the help of common users of roads (common people), using some authentication to ensure that the issues sent to them are as genuine as possible, as well as an analysis of road health severity. It will save them time and money while also assisting them in determining which concerns should be addressed first and which may be postponed until big ones are resolved.

II. Related Work

Many factors contribute to road degradation. Cracked roads, rough, hollow roads are only a few examples [8]. A hole was one of the road defects that resulted in an accident for motorists and drivers. One of them is by detecting the presence of potholes in the road early on. T. N. R. Kumar [9] did research in 2015 on the application of image processing to the detection of holes in the highway. The identification of holes was carried out in this study utilising the K-Mean clustering approach, and it was concluded that the method had effectively discovered holes, however the efficacy of its success was not mentioned. Ochoa-Ruiz et al. [10] used deep learning [11] algorithms to detect road deterioration effectively. Accelerometers and gyroscopes, as well as the GPS, are the most often utilised sensors in vibration-based approaches because they are sensitive to shocks caused by road abnormalities (e.g., potholes and bumps). In 2015, S. Nienaber et al. [12] also carried out a research on the detection of holes on the highway. In this study, the convex hull algorithm method was used where it was concluded that the success of this system was 81.8%.

Many methods have been used, which can be divided into the following three categories first threshold-based methods, [13] second dynamic time warping (DTW) [14], and third machine learning together with feature engineering [15]. An important part of this research is the Google Maps Application Programming Interface (API) [16] special Google's server map [17] which includes a database of satellite imagery, road views, altitude profiles, directions, maps with a touch of style, demographics, analysis, and large databases and can be equipped with various specific attributes according to the desire of the programmer, to build a real time application to estimate the camera pose [18]. Pothole Patrol [19] collected data using mobile sensors [20] (accelerometer, GPS) installed on vehicles that travelled thousands of kilometres around the city of Boston, MA, USA. In their pothole-detection algorithm, a range of filters (speed, high-pass, z-peak, xz-ratio, speed vs. z-ratio) were applied to dismiss one or more none pothole event types.

III. Maintenance of the Roads

Various technologies in the realm of autonomous vehicles, such as sensors, have advanced significantly in recent years. Sensors can be used to detect the environment or recognize obstacles, which is important for safe personal mobility. Other cars and pedestrians may be met as impediments when driving. This technology is critical for avoiding vehicle collisions or collisional incidents [21]. Similarly, technology that can aid in the prevention of vehicular accidents is required. In intelligent transportation systems, subsystems that provide reliable information on current road conditions are critical. Road surface indicators, unexpected road obstructions, and road distress all play vital roles in creating a safe driving environment [22]. Among the aforementioned elements, government agencies are heavily involved in managing pavement conditions, thus there is a lot of interest from them in improving pavement quality through frequent road surface damage detection. Otherwise, water would permeate the damaged portions of the road surface, causing further road structure degradation [23] as well as raising the danger of traffic accidents in the near future. Because of ageing or deterioration, road distress manifests itself in a variety of ways [24]. Traditionally, road surface care was done by labor-intensive inspection and on-site repairs. When operating on highways and tunnels, this method entails a greater risk. Techniques involving inspection equipment, like as a mobile mapping system, have been developed to tackle this problem [25]. The road surface is always presumed to be in perfect condition in the autonomous driving field, and detection of the road area is deemed adequate to enable autonomous driving.

The road surface, on the other hand, may have multiple safety blind spots, such as surface deterioration caused by exposure to various environmental variables. Smart sensors in an integrated system are required for autonomous vehicles to identify road damage. Various maintenance technologies have been developed to this purpose in order to quickly fix road damage.

IV. Specifications for Requirements

We'll talk about Requirement Specifications in this part. First and foremost, we will address the most common issues faced by government agencies such as the National Highway Authority of India (NHAI) and the Municipal Corporation in terms of road maintenance following construction, as well as immediate actions to be taken to avoid road accidents and problems. To begin building our system, we must first design a model to detect road health, for which we will require a dataset of damaged roads. These data will be utilised to train our road health analysis model. Another problem is to make our model as accurate as feasible. This can be accomplished by training our model with a large amount of data. To register a user on our app and avoid spam, we need information such as his or her name, phone number, and email address. A user of the programme must first register with it, which can be done within the app. The data structures used by NoSQL [26] databases (for example, key–value pair, broad column, graph, or document) differ from those used by relational databases by default, allowing NoSQL to perform various tasks faster. The applicability of a NoSQL database is determined by the problem it must answer. NoSQL databases' data structures are sometimes seen to be more flexible than relational database tables. All of this information allows everyone to stay informed about the institution's progress. If a student or general user has an active internet connection and an Android mobile device, he or she can benefit from this service. Before the user may begin using the system, he should only be informed on the system's functions.

4.1 Requirements for Hardware and Software

The hardware and software requirements for the planned work are discussed in this section. Using a phone with Android OS [27] and SDK 5.0 (Lollipop) or later. A minimum of 512MB of RAM is required, as well as sufficient memory to install our app. Built on JetBrains' IntelliJ IDEA software and designed exclusively for Android development, Android Studio is the official integrated development environment (IDE) for Google's Android operating system. On May 16, 2013, at the Google I/O conference, Android Studio was revealed. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. In our Application we've used the latest Version 4.0.

4.1.1 Android Emulator

The Android Emulator simulates Android devices on our computer so that we can test our application on a variety of devices and Android API levels without needing to have each physical device. The emulator provides almost all of the capabilities of a real Android device [28]. We are simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play

store, and much more. Testing our app on the emulator is in some ways faster and easier than doing so on a physical device. For example, we can transfer data faster to the emulator than to a device connected over USB. The emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices. Each instance of the Android Emulator uses an Android virtual device (AVD) shown in figure 2 to specify the Android version and hardware characteristics of the simulated device. To effectively test our app, we are creating an AVD that models each device on which your app is designed to run. To create and manage AVDs, use the AVD Manager. Each AVD functions as an independent device, with its own private storage for user data, SD card, and so on. By default, the emulator stores the user data, SD card data, and cache in a directory specific to that AVD. When we launch the emulator, it loads the user data and SD card data from the AVD directory.

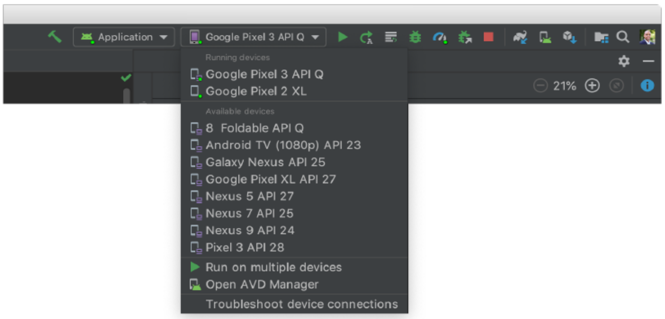

Figure 2 The Android Virtual Device (AVD)

4.1.2 Dependency

Project Dependencies are, in fact, Schedule Dependencies. It simply means that schedule of one task or activity is reliant on another one. It explains the project dependencies with the help of an example and differentiates between dependencies, assumptions, and constraints. These dependencies are legally or contractually required. They are, sometimes, inherent in the nature of the work.

```
implementation fileTree(dir: "libs", include: ["*.jar"])
implementation 'androidx.appcompat:appcompat:1.1.0'
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
implementation 'com.google.android.material:material:1.0.0'
implementation 'androidx.annotation:annotation:1.1.0'
implementation 'androidx.lifecycle:lifecycle-extensions:2.1.0'
implementation 'com.google.firebase:firebase-storage:16.0.4'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'androidx.test.ext:junit:1.1.1'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
```

4.1.3 UI Dependency

A user interface (UI) test that involves user interactions across multiple apps lets you verify that our app behaves correctly when the user flow crosses into other apps or into the system UI. An example of such a user flow is a messaging app that lets the user enter a text message, launches the Android contact picker so that the users can select recipients to send the message to, and then returns control to the original app for the user to submit the message.

```
implementation 'com.google.firebase:firebase-core:16.0.1'
implementation 'com.google.firebase:firebase-database:16.0.1'
implementation 'com.google.firebase:firebase-auth:16.0.1'
implementation 'com.firebaseui:firebase-ui-database:1.2.0'
implementation 'com.firebaseui:firebase-ui-auth:4.3.1'
```

Using UI Automator APIs we interact with visible elements on a device, regardless of which Activity is in focus. We are test can look up a UI component by using convenient descriptors such as the text displayed in that component or its content description. The UI Automator testing framework is an instrumentation-based API and works with the Android Junit Tester test runner. To optimize your UI Automator testing, you should first inspect the target app's UI components and ensure that they are accessible. These optimization tips are described in the next two sections.

Gradle is a build system (open source) which is used to automate building, testing, deployment etc. "Build. gradle" are scripts where one can automate the tasks. For example, the simple task to copy some files from one directory to another can be performed by Gradle build script before the actual build process happens. Every Android project needs a gradle for generating an apk from the .java and .xml files in the project. Simply put, a gradle takes all the source files (java and XML) and apply appropriate tools, e.g., converts the java files into dex files and compresses all of them into a single file known as apk that is actually used.

V. Proposed System and Services

For Analytics phase we are using Google analytics is a cost-free app measurement solution that provides insights on app usage and user engagement. For messages using Firebase Cloud Messaging Formerly known as Google Cloud [30] Messaging (GCM), Firebase Cloud Messaging (FCM) shown in figure 3 is a cross-platform solution for messages and notifications for Android, iOS, and web applications. In this project authentication process we are using Firebase Authentication is a service that can authenticate users using only client-side code. It supports social login providers Facebook, GitHub, Twitter and Google as well as other service providers like Google Play Games, Apple, Yahoo, and Microsoft.

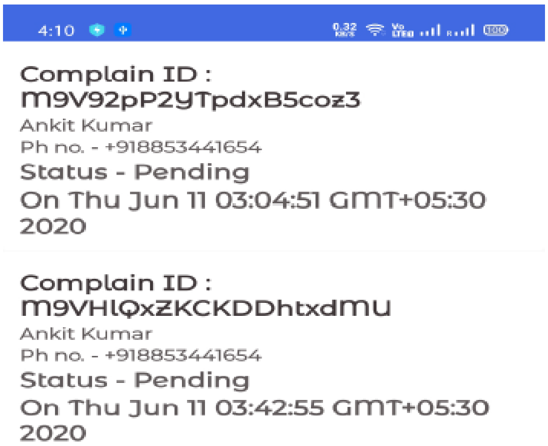

Figure 3 The Complain Massage

Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase shown in figure 4. Again for database using Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data [29] to be synchronized across clients and stored on Firebase's cloud. It's also provides client libraries that enable integration with Android, iOS, JavaScript, Java, ObjectiveC, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Using the real-time database can secure their data by using the company's server-side-enforced security rules. Next Firebase storage provides secure file uploads and downloads for Firebase apps, regardless of network quality, to be used for storing images, audio, video, or other user-generated content. It is backed by Google Cloud Storage.

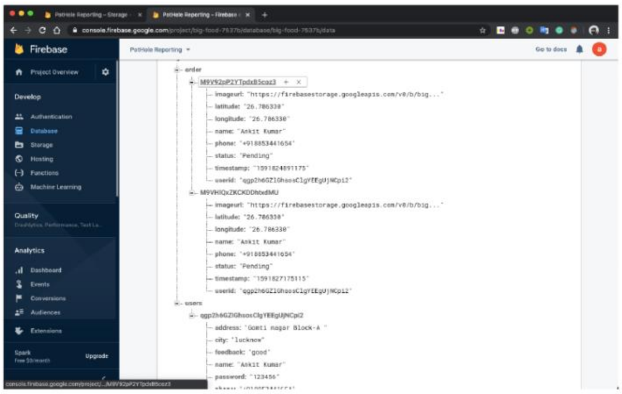

Figure 4 The Firebase Storage

In the Firebase Hosting is a static and dynamic web hosting Service. It supports hosting static files such as CSS, HTML, JavaScript and other files, as well as support through Cloud Functions. The service delivers files over a content delivery network (CDN) through HTTP Secure (HTTPS) and Secure Sockets Layer encryption (SSL). Firebase partners with Fastly, a CDN, to provide the CDN backing Firebase Hosting. The company states that Firebase Hosting grew out of customer requests; developers were using Firebase for its real-time database but needed a place to host their content. We are using ML Kit is a mobile machine learning system. ML Kit APIs [31] feature a variety of features including optical character recognition, detecting faces, scanning barcodes, labelling images and recognising landmarks. It is currently available for iOS or Android developers. We are also import our own TensorFlow Lite models, if the given APIs are not enough.

Figure 5 The Pothole Reporting

The APIs can be used on-device or on-cloud. For reporting we are using Crashlytics. Crash Reporting creates detailed reports of the errors in the app shown in figure 5. Errors are grouped into clusters of similar stack traces and triaged by the severity of impact on app users. In addition to automatic reports, the developer can log custom events to help capture the steps leading up to a crash. Before acquiring Crashlytics, Firebase was using its own Firebase Crash Reporting.

Again, Firebase Performance provides insights into an app's performance and the latencies the app's user's experience. Using testing Firebase Test Lab provides cloud-based infrastructure for testing Android and iOS apps in one operation. We can test their apps across a wide variety of devices and device configurations. Test results including logs, videos, and screenshots are made available in the Firebase console. If hasn't written any test code for their app, Test Lab can exercise the app automatically, looking for crashes. Test Lab for iOS is currently in beta. Admob is a Google product that integrates with Firebase audience. Firebase Dynamic Links Dynamic [32] Firebase links are smart URLs that dynamically change their behaviour to provide "the best available experience" across multiple platforms, including desktop web browsers, iOS, and Android, and in-depth links to mobile apps. We are opens dynamic link on iOS or Android and the application is not installed, the user will be prompted to install the app first. Once installed, the application will start running and can access the link shown in figure 6.

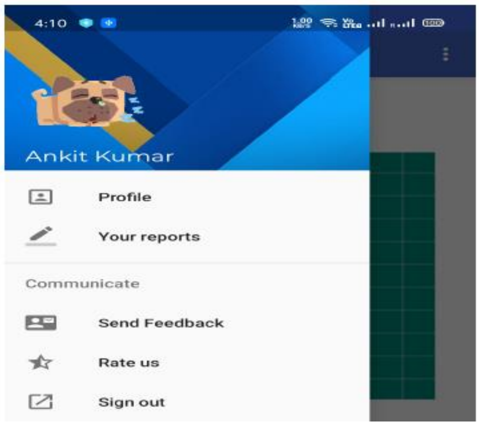

Figure 6 The App view

VI. Analysis and Findings

Our model is able to identify potholes with an accuracy of 84 percent in this research. So far, the outcomes have been good and far better than we had anticipated. The health of the system is also beneficial throughout the life cycle. The main goal of this module is to forecast the presence of potholes in a set of photos. The model and photos from the machine are loaded into the module. The photos are pre-processed in the same way that they were in the training module. The photos are fed into the model, and the results are displayed on the console as predictions and accuracies. Everything went according to plan. The system that was constructed is compared to the initial specifications that were developed at the start of the project. To improve model accuracy even more, we're trying to train it on a variety of different and much larger datasets. We also attempted to compress the image in order to reduce server load. Noises, such as other items that we are not interested in, can be found in an image. We attempted to eliminate these noises in order to improve accuracy. Multithreading is a good way to improve the speed of our programme. The only security problem we encountered when developing an Android app was reverse engineering, which exposes our code to hackers. Hackers will be able to bypass authentication as a result of this. Some people may want to crack the app in order to figure out how it works and what particular features it has, either to develop a better app than ours or to entirely copy it. It's likely that different users will post the same report many times, increasing redundancy. In the future, we're considering identifying geo-locations based on blocks and limiting the number of photographs a user can upload at a time.

VII. CONCLUSION

Potholes arise as a result of road wear and tear and weathering. They not only create inconvenience to individuals, but they also result in deaths as a result of car accidents. The goal of this research was to come up with a mechanism for detecting potholes in road surfaces that would save time and money. The user can utilize the application at any time and from any location. The project has an in-built machine learning kit, allowing users to easily prepare for interviews or compile their scripts whenever they want. The codes are readily modifiable and may be saved and shared across multiple platforms. The programme will function as the team has suggested, and any future upgrades will be automatically notified. The project's major purpose was to shorten the time it took from receiving a report of damaged roads to completing it. As a result, there are numerous accidents and significant vehicle damage. This job will be completed in less time and with fewer man hours as a result of our initiative. The tracking of these issues will become much more simple and straightforward. We believe that this project will expand in the future to make government officials' jobs easier, faster, and more effective. Because mobile phones are getting cheaper and more process power heavy than ever before, a strong android app like this can be quickly put in a phone. This will make using the app as simple as using any other app that a regular person uses on a daily basis. Unlike others, our solution is less expensive and easier to maintain. Only a few engineers are needed to keep the system running.

VIII. FUTURE WORK

We will enhance the amount of training data and design the structure of a new neural network in future research to improve the detection accuracy of categories that are difficult to identify. We're also trying to figure out how many potholes there are in the image. Which of these will improve the usability and effectiveness of our system?

IX. REFERENCES

[1]. Rajeshwari Madli, Santosh Hebbar, Praveenraj Pattar, Varaprasad Golla, "Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers", Sensors Journal IEEE, vol. 15, no. 8, pp. 4313-4318, 2015

[2]. Al-Mushayt O, Haq K and Yusuf P., “Electronic-Government in Saudi Arabia; a Positive Revolution in the Peninsula”, International Transactions in Applied Sciences, India, 1(1), 87-98, 2009

[3]. Transport Committee, Local roads funding and maintenance: Filling the gap. 2019

[4]. Tran Duc Chung, M. K. A. Ahamed Khan, "Watershed-based Real-time Image Processing for Multi-Potholes Detection on Asphalt Road", System Engineering and Technology (ICSET) 2019 IEEE 9th International Conference on, pp. 268-272, 2019

[5]. Yusuf Perwej, “An Evaluation of Deep Learning Miniature Concerning in Soft Computing”, the International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 4, Issue 2, Pages 10 - 16, 2015, DOI: 10.17148/IJARCCE.2015.4203

[6]. J. Javier Yebes, David Montero, Ignacio Arriola, "Learning to Automatically Catch Potholes in Worldwide Road Scene Images", Intelligent Transportation Systems Magazine IEEE, vol. 13, no. 3, pp. 192-205, 2021

[7]. C. Guo, Y. Liu, W. Shen, H. J. Wang and Y. Zhang, "Mining the Web and the Internet for Accurate IP Address Geolocations", Infocom. IEEE, 2009

[8]. Hall, Kathleen T. dkk, Rehabilitation Strategies for Highway Pavement, National Cooperative Highway Research Program, Amerika Serikat, 2001.

[9]. Kumar, T.N.R., A Real Time Approach for Indian Road Analysis using Image Processing and Computer Vision, Department of Computer Science M.S.Ramaiah Institute of Technology Bangalore, India, 2015.

[10]. Ochoa-Ruiz, G., Angulo-Murillo, A. A., Ochoa-Zezzatti, A.; Aguilar-Lobo, L.M.; Vega, J.A.; Natraj, S.,“An asphalt damage dataset and detection system based on retinanet for road conditions assessment”. Appl. Sci., 10, 3974. 2020

[11]. Yusuf Perwej, “An Evaluation of Deep Learning Miniature Concerning in Soft Computing”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), ISSN (Online): 2278-1021, ISSN (Print): 2319-5940, Volume 4, Issue 2, Pages 10 - 16, 2015, DOI: 10.17148/IJARCCE.2015.4203

[12]. Nienaber, S. dkk, Detecting Potholes using Simple Image Processing Techniques and Real-World Footage, Department of E&E Engineering Stellenbosch University, Afrika Selatan, 2015

[13]. Yusuf Perwej, “The Bidirectional Long-Short-Term Memory Neural Network based Word Retrieval for Arabic Documents”, Transactions on Machine Learning and Artificial Intelligence (TMLAI), Society for Science and Education, United Kingdom (UK), ISSN 2054-7390, Volume 3, Issue 1, Pages 16 - 27, 2015, DOI: 10.14738/tmlai.31.863

[14]. Yusuf Perwej , Firoj Parwej, Asif Perwej, “Copyright Protection of Digital Images Using Robust Watermarking Based on Joint DLT and DWT ”, International Journal of Scientific & Engineering Research (IJSER), France, ISSN 2229-5518, Volume 3, Issue 6, Pages 1- 9, 2012

[15]. Sattar, S.; Li, S.; Chapman, M. Road surface monitoring using smartphone sensors: A review. Sensors 2018, 18, 3845

[16]. Punjabi, Himanshu dkk, Intelligent Pothole Detection System, Department Thadomal Shahani Engineering College, India, 2014

[17]. Bray, M., “Application Programming Interface”, The Software Engineering Institute, 1997.

[18]. M. Pérez, I. Strictest, C. Confidence, P. One, I. Strictest, and C. Confidence, Mastering OpenCV, no. January. 2003

[19]. Eriksson, J.; Girod, L.; Hull, B.; Newton, R.; Madden, S.; Balakrishnan, H. 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, Breckenridge, USA, 17–20, pp. 29–39, 2008

[20]. Yusuf Perwej, Majzoob K. Omer, Osama E. Sheta, Hani Ali M. Harb, Mohmed S. Adrees, “The Future of Internet of Things (IoT) and Its Empowering Technology”, International Journal of Engineering Science and Computing, ISSN: 2321- 3361, Volume 9, Issue No.3, Pages 20192– 20203, 2019

[21]. C. Ilas, Electronic sensing technologies for autonomous ground vehicles: a review, in: 8th International Symposium on Advanced Topics in Electrical Engineering, pp. 1–6, 2013

[22]. Y. Yu, J.Z. H. Guan, Automated detection of urban road manhole covers using mobile laser scanning data, IEEE Trans. Intell. Transport. Syst. 16 (6), 3258–3269, 2015

[23]. KICT, Final Report of the National Highway Pavement Management System 2016, Korea Institute of Civil Engineering and Building Technology, Gyeonggi-do, South Korea, 2017

[24]. Y. Pan, X. Zhang, G. Cervone, L. Yang, Detection of asphalt pavement potholes and cracks based on the unmanned aerial vehicle multispectral imagery, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 11 (10), 3701–3712, 2018

[25]. M. Brogan, S. McLoughlin, C. Deegan, Assessment of stereo camera calibration techniques for a portable mobile mapping system, IET Comput. Vis. 7 (3), 209–217, 2013

[26]. Nathan Hurst, "Visual Guide to NoSQL Systems.", http://blog.nahurst.comlvisual-guide-to-NoSQL-systems/

[27]. Yusuf Perwej, Shaikh Abdul Hannan, Firoj Parwej, Nikhat Akhtar , "A Posteriori Perusal of Mobile Computing", International Journal of Computer Applications Technology and Research (IJCATR), , Volume 3, Issue 9, Pages 569 - 578, September 2014, DOI: 10.7753/IJCATR0309.1008

[28]. K. Yaghmour, "Embedded Android: Porting Extending and Customizing", O'Reilly Media, 2013

[29]. Y. S. Yilmaz, B. I. Aydin and M. Demirbas, "Google cloud messaging (GCM): An evaluation", 2014 IEEE Global Communications Conference, pp. 2807-2812, 2014

[30]. Nikhat Akhtar, Bedine Kerim, Yusuf Perwej, Anurag Tiwari, Sheeba Praveen, "A Comprehensive Overview of Privacy and Data Security for Cloud Storage", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 08, Issue 5, Pages 113-152, 2021 DOI: 10.32628/IJSRSET21852

[31]. N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi and F. Kawsar, "Squeezing Deep Learning into Mobile and Embedded Devices", IEEE Pervasive Computing, vol. 16, no. 3, pp. 82-88, 2017

[32]. R. Slavin, X. Wang, M. B. Hosserni, J. Hester, R. Krishnan, J. Bhatia, et al., "Toward a framework for detecting privacy policy violation in Android application code", ICSE, pp. 25-36, 2016

Cite this article as :

Ankit Kumar, Neha kulshrestha, Dr. Yusuf Perwej, Ashish Kumar Srivastava, Chandan Mani Tripathi, "The Assay of Potholes and Road Damage Detection", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 1, pp. 202-211, January-February 2022. Available at doi : https://doi.org/10.32628/CSEIT228135
Journal URL : https://ijsrcseit.com/CSEIT228135