**EFFICIENT MANHOLE VISUAL INSPECTION USING DEEP LEARNING IN UAV NAVIGATION**

**PROJECT REPORT**

***Submitted by***


| | |
|---|---|
| **SELVABHARATHI.P** | **(820620104051)** |
| **MOHAMED BASITH.B** | **(820620104308)** |
| **VENKATESH.S** | **(820620104315)** |
| **VISWAM.S** | **(820620104316)** |


***in partial fulfillment for the award of the degree***

***Of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**


**ARASU ENGINEERING COLLEGE, KUMBAKONAM**

**ANNA UNIVERSITY: CHENNAI 600 025**


**MAY 2024**

## DECLARATION

We hereby declare that the Project work entitled **"EFFICIENT MANHOLE VISUAL INSPECTION USING DEEP LEARNING IN UAV NAVIGATION"** is submitted in partial fulfillment of requirement for the award of the degree in B.E., Anna University Chennai, is a record of our own work carried out by us during the academic year 2023-2024 under the supervision and guidance of Mrs. R. SUDHA, M.E., Assistant Professor, Department of Computer Science and Engineering, Arasu Engineering College at Kumbakonam. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree, either in this or any other university.

| **Name** | **Register number** | **Signature** |
|---|---|---|
| SELVABHARATHI.P | 820620104051 | |
| MOHAMED BASITH.B | 820620104308 | |
| VENKATESH.S | 820620104316 | |
| VISWAM.S | 820620104316 | |

I certify that the declaration made above by the candidate is true.

MRS.R.SUDHA.,M.E.,

ASSISTANT PROFESSOR

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report **"EFFICIENT MANHOLE VISUAL INSPECTION USING DEEP LEARNING IN UAV NAVIGATION"** is the bonafide work of **"SELVABHARATHI.P (820620104051), MOHAMED BASITH.B (820620104308), VENKATESH.S (820620104315), VISWAM.S (820620104316)"**

| | |
|---|---|
| SIGNATURE | SIGNATURE |
| DR,KALAIMANI SHANMUGAM | MRS.R.SUDHA.,M.E., |
| M.Tech.,Ph.D.., | |
| HEAD OF DEPARTMENT | SUPERVISOR |
| PROFESSOR | ASSISTANT PROFESSOR |
| COMPUTER SCIENCE & ENGINEERING | COMPUTER SCIENCE & ENGINEERING |
| ARASU ENGINEERING COLLEGE | ARASU ENGINEERING COLLEGE |
| KUMBAKONAM-612501 | KUMBAKONAM-612501 |

**Submitted for Project Viva-Voce examination held on: ________________**

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

Apart from the efforts of us, the success of our project depends largely on the encouragement and guidelines of many others. It is our pleasure to give our special thanks to **ANNA UNVIERSITY**. Also we thank to **ARASU ENGINEERING COLLEGE** especially to our Chairman **Mr.R.THIRUNAVUKARASU M.A.,** and other management people for permitted us to carry out this project work.

We like to express our deep gratitude to our beloved Principal, **Dr. T. BALAMURUGAN M.E., Ph.D.,** for providing us all necessary facilities to carry out our project successfully.

Our profound sense of gratitude is due to our Head of Computer Science & Engineering department, **Dr.KALAIMANI SHANMUGAM M.Tech., Ph.D.,** for her constant encouragement and valuable guidance.

Our heartfelt gratitude to our project coordinator **R. KALAISELVI M.Tech.,** Assistant Professor of CSE department who continuously supported us in every possible way, from initial advice to encouragement till now.

We would like to show my greatest appreciation to our project guide **Mrs. R.SUDHA M.E.,** Assistant Professor of CSE department. We feel motivated and encouraged every time when we attended her meeting. Without her encouragement and guidance this project would not have materialized.

The guidance and support received from all the **Faculty members** who contributed to this project, was vital for the success of the project. We are grateful for their constant support and help. Last but not least my special thank to the all the supporting **staffs and lab technicians** at the institute who directly or indirectly helped us in completing our project.

# ABSTRACT

Manhole is a covered opening in a street or public area that provides access to a utility or maintenance vault underground. Manholes are typically constructed with a cover or lid that can be removed to allow entry for inspection, maintenance, or repair of utility infrastructure such as sewers, storm drains, telecommunications, electrical, or gas systems. Manholes are often located in public areas, and their covers need to be secure to prevent accidents. Broken or missing manhole covers can pose serious safety hazards to pedestrians, cyclists, and drivers. The increasing risk of traffic accidents due to the deterioration of manhole covers necessitates a more efficient and reliable inspection method. Manual observation, the traditional approach to monitoring manhole covers, faces challenges such as labour shortages and ethical concerns. Identifying open or broken manholes using image processing algorithms faces challenges related to variable image quality, complex backgrounds, scale changes, and dynamic environmental conditions. In response to this difficulty the aim of this project proposes an automated system architecture based on deep learning models to replace the manual examination process. The project involves the development of a deep learning model capable of analysing images of manhole covers. The model undergoes training using a diverse dataset to accurately classify covers into categories such as 'Close,' 'Open,' 'Broken,' and 'No Manhole.' Additionally, the system incorporates advanced techniques, including Convolutional Neural Networks (CNN) for image classification and You Only Look Once version 8 (YOLOv8) for accurate prediction and localization using UAV Images or CCTV Footages. The implementation of this deep learning-based architecture offers a promising avenue for enhancing urban safety and streamlining infrastructure maintenance processes.

**TABLE OF CONTENT**

# LIST OF FIGURE

# LIST OF ABBREAVTIONS

| | |
|---|---|
| YOLO | YOU ONLY LOOK ONCE |
| OPEN CV | OPEN SOURCE COMPUTER VISION |
| UAV | UNMANNED ARIEL VEHICLE |
| SVM | SUPPORT VECTOR MACHINES |

# CHAPTER 1

## 1.INTRODUCTION

A manhole or an inspection chamber is a unit constructed underground to provide access to the utilities like a sewer system, drainage system, etc. Hence, with the help of a manhole, underground utilities are inspected, modified, cleaned and maintained. Sewer systems are built underground with pipes that carry waste from homes and other buildings to a place of treatment or disposal. Part of maintaining a sewer system is providing frequent inspection, cleaning and repairs. Utility crews use manholes to gain closer access to pipes or other parts of the underground system to meet those needs.

### Purpose of a Manhole

Manholes are built primarily for trenchless restoration of the sewer system, drainage system inspection, cleaning of clogged lines, and maintenance purposes. Manholes are also used as a first step for accessing the inside of a sewer line to help diagnose any issues with it and facilitate the replacement of damaged pipes without the need for digging. Up until the end of the main sewer line or drainage point, manholes are positioned throughout the sewer line. There are usually manholes located at several intervals down the drainage system to allow for maximum access. If one area is clear yet another is blocked, the manhole closest to the issue can be lifted and inspected, and any necessary work such as high pressure water jetting can be carried out to clear the problem. If the water is flowing in along the pipe and then stops or backs up, the location of the problem can be confirmed by lifting the manholes and monitoring the water levels. If the levels are high, it suggested there is a problem nearby which requires attention. The manhole covers are composed of metal and composite material and come in a variety of sizes, materials, and designs, including rectangular, circular, and square. If the depth of the manhole chamber exceeds 2.5 m, a ladder must be installed inside; if the depth is little than 1 m, a step ladder is required.

There are three different types of manholes: shallow, normal and deep. “Normal” manholes are typically 4- to 5-feet deep and wide enough for the average person to fit in. "Shallow" manholes are 2- to 3-feet deep, often placed at the start of a sewer branch and in areas with low traffic. Manholes with a depth greater than 5-feet are considered "deep" and usually have an entry method like a ladder built-in, as well as a heavier cover. Manholes are designed with a cover or lid and comprised of grade adjusting rings, a top tapered section called the cone, a main cylinder section called the wall or barrel, and a bench and channel where the waste flows through.

## 1.1. AIM AND OBJECTIVES

The aim of the project is to develop an automated inspection system for manhole covers using deep learning models, with the primary objective being to enhance urban safety and streamline infrastructure maintenance processes.

- To develop deep learning models for analyzing images of manhole covers.
- To classify manhole covers into categories based on their condition using deep learning techniques.
- To accurately predict and localize the position of manhole covers within images.
- To curate a diverse dataset of manhole, cover images for training and validation purposes.
- To address challenges related to image quality, backgrounds, scale changes, and environmental conditions during processing.
- To integrate the system with UAV and CCTV systems for real-time image processing.
- To conduct thorough testing and validation of the automated inspection system.

- To evaluate the performance of the system in terms of speed, accuracy, and scalability.
- To deploy the system in urban areas for practical application.
- To document the development process and outcomes for dissemination and knowledge sharing.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1. Real-time Detection of Road Manhole Covers with a Deep Learning Model.

**Author:** Mehmet Akif Yaman, Frank Rattay, Abdulhamit Subasi.

Road manhole covers are crucial components of urban infrastructure; however, inadequate maintenance or poor marking can pose safety risks to vehicular traffic. This paper presents a method for detecting road manhole covers using a stereo depth camera and the MGB-YOLO model. We curated a robust image dataset and performed image enhancement and annotation. The MGB-YOLO model was developed by optimizing the YOLOv5s network with MobileNet-V3, Global Attention Mechanism (GAM), and BottleneckCSP, striking a balance between detection accuracy and model efficiency. Our method achieved an impressive accuracy of 96.6%, surpassing the performance of Faster RCNN, SSD, YOLOv5s, YOLOv7 and YOLOv8s models with an increased mean average precision (MAP) of 15.6%, 6.9%, 0.7%, 0.5% and 0.5%, respectively. Additionally, we have reduced the model's size and the number of parameters, making it highly suitable for deployment on in-vehicle embedded devices. These results underscore the effectiveness of our approach in detecting road manhole covers, offering valuable insights for vehicle-based manhole cover detection and contributing to the reduction of accidents and enhanced driving comfort.

**Algorithms / Techniques**: Feature Pyramid Networks (FPN)

**Drawbacks**: The model's performance with real-time detection standards.

### 2.2. Data-Augmented Deep Learning Models for Abnormal Road Manhole Cover Detection

**Author:** Apurva Kumari; P. Ashrani, Mudasar Basha, B. Sri Anjan Kumar

Anomalous road manhole covers pose a potential risk to road safety in cities. In the development of smart cities, computer vision techniques use deep learning to automatically detect anomalous manhole covers to avoid these risks. One important problem is that a large amount of data are required to train a road anomaly manhole cover detection model. The number of anomalous manhole covers is usually small, which makes it a challenge to create training datasets quickly. To expand the dataset and improve the generalization of the model, researchers usually copy and paste samples from the original data to other data in order to achieve data augmentation. In this paper, we propose a new data augmentation method, which uses data that do not exist in the original dataset as samples to automatically select the pasting position of manhole cover samples and predict the transformation parameters via visual prior experience and perspective transformations, making it more accurately capture the actual shape of manhole covers on a road. Without using other data enhancement processes, our method raises the mean average precision (mAP) by at least 6.8 compared with the baseline model.

**Algorithm / Techniques: Data Augmentation.**

**Drawbacks:** Data augmentation method based on deep learning is mentioned to be more time-consuming during training compared to traditional methods.

**2.3 Design and Implementation of Manhole Cover Safety Monitoring System based on Smart Light Pole.**

**Author:** Liang Yu ,Zhengkuan Zhang

Aiming at the current problems in the safety monitoring of urban manhole covers, this paper proposes a safety monitoring system for manhole covers based on smart light poles. The system uses STM32F103C8T6 as the microcontroller, and processes and controls the movement, loss, tilt, flooding, and positioning data of the manhole cover. Then, the data frame is sent to the LoRa gateway of the nearby smart light pole through the LoRa communication protocol, and the LoRa gateway will transmit the abnormal state of the manhole cover to the large screen display of the smart light pole through the bus. At the same time, a voice broadcast is carried out to remind localized road vehicles and pedestrians. Besides, it is sent to the IoT cloud platform KitLink, and the cloud platform will push the abnormal status data to the subscribed management user, and the user will quickly process it according to the positioning data. Among them, the terminal sensing control device of the manhole cover adopts an integrated package, and at the same time, it is installed at the center point under the manhole cover by using a waterproof material. It can not only carry out reliable and accurate installation without changing the original components but also can effectively monitor the safety. Through the design and implementation test of the system, it can be seen that the system can efficiently and accurately realize all-round and multi-modal safety linkage monitoring of urban manhole covers.

.**Algorithms / Techniques**: Transfer Learning with Pre-trained Models.

**Drawbacks**: Challenging related details about the algorithms

### 2.4: Detection and Localization of Manhole and Joint Covers in Radar Images by Support Vector Machine and Hough Transform

**Author:** Takahiro Yamaguchi, Tsukasa Mizutani

In this paper, a novel manhole and joint covers detection algorithm from radar images by Support Vector Machine (SVM) and Hough transform is proposed. Due to its dense and high-speed monitoring capabilities, Ground Penetrating Radar (GPR) is a promising tool. Furthermore, manhole and joint covers are apparent from surface reflections. An SVM model was developed utilizing Histogram of Oriented Gradient (HOG) feature and Laplacian filter. Classification accuracy of manhole, joint covers and pavement section was up to 98%. Hough transform was applied to the detected areas to visualize objects in a map. The algorithm detected manhole and joint covers accurately and fast by the combination of SVM and Hough transform.

**Algorithms / Techniques**: Support Vector Machines.

**Drawbacks:** The information provided does not include a reference link for further details and validation of the presented work.

### 2.5. Manhole Cover Detection on Rasterized Mobile Mapping Point Cloud Data Using Transfer Learned Fully Convolutional Neural Networks.

**Author:** Lukas Mattheuwsen, Maarten Vergauwen

Large-scale spatial databases contain information of different objects in the public domain and are of great importance for many stakeholders. These data are not only used to inventory the different assets of the public domain but also for project planning, construction design, and to create prediction models for disaster management or transportation. The use of mobile mapping systems instead of traditional surveying techniques for the data acquisition of these datasets is growing. However, while some objects can be (semi)automatically extracted, the mapping of manhole covers is still primarily done manually. In this work, we present a fully automatic manhole cover detection method to extract and accurately determine the position of manhole covers from mobile mapping point cloud data. Our method rasterizes the point cloud data into ground images with three channels: intensity value, minimum height and height variance. These images are processed by a transfer learned fully convolutional neural network to generate the spatial classification map. This map is then fed to a simplified class activation mapping (CAM) location algorithm to predict the center position of each manhole cover. The work assesses the influence of different backbone architectures (AlexNet, VGG-16, Inception-v3 and ResNet-101) and that of the geometric information channels in the ground image when commonly only the intensity channel is used. Our experiments show that the most consistent architecture is VGG-16, achieving a recall, precision and $F_2$-score of 0.973, 0.973 and 0.973, respectively, in terms of detection performance. In terms of location performance, our approach achieves a horizontal 95% confidence interval of 16.5 cm using the VGG-16 architecture.

**Algorithms**: Fully Convolutional Neural Networks(F-CNN).

**Drawbacks:** While achieving a 95% confidence interval of 16.5 cm is commendable, there may be scenarios or environments where the method's location performance could be further challenged.

### 2.6. Customized Mobile LiDAR System for Manhole Cover Detection and Identification.

**Author:** Zhanying Wei

Manhole covers, which are a key element of urban infrastructure management, have a direct impact on travel safety. At present, there is no automatic, safe, and efficient system specially used for the intelligent detection, identification, and assessment of manhole covers. In this work, we developed an automatic detection, identification, and assessment system for manhole covers. First, we developed a sequential exposure system via the addition of multiple cameras in a symmetrical arrangement to realize the joint acquisition of high-precision laser data and ultra-high-resolution ground images. Second, we proposed an improved histogram of an oriented gradient with symmetry features and a support vector machine method to detect manhole covers effectively and accurately, by using the intensity images and ground orthophotos that are derived from the laser points and images, respectively, and apply the graph segmentation and statistical analysis to achieve the detection, identification, and assessment of manhole covers. Qualitative and quantitative analyses are performed using large experimental datasets that were acquired with the modified manhole-cover detection system. The detected results yield an average accuracy of 96.18%, completeness of 94.27%, and F-measure value of 95.22% in manhole cover detection. Defective manhole-cover monitoring and manhole-cover ownership information are achieved from these detection results. The results not only provide strong support for road administration works, such as data acquisition, manhole cover inquiry and inspection, and statistical analysis of resources, but also demonstrate the feasibility and effectiveness of the proposed method, which reduces the risk involved in performing manual inspections, improves the manhole-cover detection accuracy, and serves as a powerful tool in intelligent road administration.

**Algorithms / Techniques**: Support Vector Machines.

**Drawbacks:** It would be important to consider factors such as scalability, robustness in diverse environments, and potential challenges in real-world implementation

# CHAPTER 3
# PROBLEM DEFINITION

## 3.1 PROBLEM STATEMENT

Manholes present several significant challenges. Firstly, they pose safety hazards to pedestrians, cyclists, and motorists if left uncovered or improperly maintained, leading to accidents and injuries. Additionally, uncapped or poorly maintained manholes can become breeding grounds for pests and bacteria, raising public health concerns. Infrastructure damage is another issue, as damaged or corroded manholes can result in collapsed road surfaces or sewer lines, necessitating costly repairs. Moreover, clogged or blocked manholes can cause flooding and drainage problems, resulting in property damage and environmental pollution. Accessibility challenges arise for individuals with disabilities when manholes lack proper accommodations, hindering their ability to navigate public spaces safely. Furthermore, open or unsecured manholes present security risks by providing unauthorized access to underground infrastructure, leading to vandalism and theft. Lastly, the maintenance burden of manholes is substantial, requiring regular inspection, cleaning, and upkeep, which can strain municipal resources. Addressing these challenges requires proactive measures such as regular maintenance, infrastructure upgrades, public awareness campaigns, and improved design and security measures. The problem statement of the project revolves around addressing the multitude of challenges associated with manholes, particularly in urban environments. Manholes, while crucial for utility access and maintenance, present significant safety hazards, public health concerns, infrastructure damage, and accessibility issues if left unattended or improperly maintained. The lack of proper management and maintenance can lead to accidents, injuries, property damage, environmental pollution, and hindered mobility for individuals with disabilities. Additionally, the security risks posed by open or unsecured manholes further exacerbate the problem. Therefore, the project aims to develop innovative solutions and technologies to enhance the detection, management, and maintenance of manholes, ensuring public safety, health, infrastructure resilience, and accessibility in urban areas.

## 3.2 EXISTING SYSTEM

- **Manual Visual Inspection**

The existing manual system for manhole defect prediction involves trained inspectors visually inspecting manholes for signs of damage or wear and tear. Inspectors typically use checklists or forms to document their findings, which are then manually entered into a database or spreadsheet for further analysis. This manual process is time-consuming and can be prone to errors or inconsistencies due to human error or subjective judgments. Inspectors may also miss certain defects or fail to identify trends or patterns in the data, leading to incomplete or inaccurate information. Furthermore, the manual system is often reactive rather than proactive, meaning that repairs or maintenance are only initiated after a defect has been identified, rather than being detected and addressed before the issue becomes critical.

- **Image Processing Based System**

There are some existing image processing systems for manhole defect prediction. These systems use various image processing techniques such as edge detection, morphological operations, and thresholding to identify defects in manhole images. Some of the commonly used techniques are:

**Sobel Edge Detection**: This technique is used to detect edges in an image by calculating the gradient in the x and y directions. The edges in the image can be used to identify defects in manholes.

**Morphological Operations**: This technique is used to enhance or reduce certain features in an image. For example, erosion can be used to reduce the thickness of lines or edges in an image, while dilation can be used to increase the thickness of lines or edges. These operations can be used to remove noise or enhance defects in manhole images.

**Thresholding**: This technique is used to convert a grayscale image to a binary image by setting a threshold value. Pixels with intensity values above the threshold are set to white, while pixels with intensity values below the threshold are set to black. This technique can be used to identify defects in manhole images by setting the threshold value to highlight areas of damage or wear and tear.

❖ **Machine Learning Based System**

There are some existing machine learning systems for manhole defect prediction. These systems use various machine learning algorithms such as decision trees, support vector machines (SVM), and random forests to identify defects in manhole images. These algorithms can be trained on a dataset of labelled manhole images to predict the presence of defects.

Some of the commonly used techniques are:

**Decision Trees:** This technique is used to create a model that predicts the value of a target variable based on several input variables. The decision tree consists of nodes that represent the input variables and branches that represent the possible values of these variables. This technique can be used to predict defects in manholes by training the decision tree model on a dataset of manhole images and their associated defect labels.

**Random Forests:** This technique is an extension of decision trees that uses multiple decision trees to improve prediction accuracy. The random forest model is trained on a dataset of manhole images and their associated defect labels. During training, the model creates multiple decision trees using different subsets of the input variables and data. The final prediction is made by averaging the predictions of all decision trees.

**Support Vector Machines:** This technique is used to create a model that predicts the value of a target variable based on several input variables. The SVM model separates the input data into different classes by creating a hyperplane in the input space. This technique can be used to predict defects in manholes by training the SVM model on a dataset of manhole images and their associated defect labels.

**DRAWBACKS**

- Labor-intensive manual inspection is prone to human error.
- Limited coverage and frequency of inspections may miss emerging issues.
- Safety risks for workers during manual maintenance tasks.
- Inefficient response to emergencies due to delayed identification of issues.
- Difficulty in tracking maintenance records leads to inefficiencies.
- High maintenance costs due to labour-intensive processes and potential inefficiencies.
- Traditional machine learning methods can be limited in their ability to learn complex features and patterns in images.
- Existing machine learning systems may not be able to generalize well to different types of manholes, conditions, and defects.
- Require significant computational resources and training data to achieve high accuracy.
- Some existing methods may be prone to false positives and false negatives, leading to inaccurate predictions and inefficient maintenance operations.

## 3.3 PROPOSED SYSTEM

The proposed system of the Manhole Predictor project is a web-based efficient manhole maintenance system that uses deep learning algorithms to classify manhole images into six categories: close, open, broken, overflow, manhole with pedestrian and no manhole.

**ADVANTAGES**

- Simplified data collection and evaluation.
- Suitable for drone implementation
- Adequate accuracy while maintain a low computational load

# CHAPTER 4
# SYSTEM STUDY

## 4.1 FEASIBILITY STUDY

This project is feasible provided given unlimited resources and infinite time. Unfortunately the development of a computer-based system is more likely to be plagued by resource scarcity and stringent schedules. It is both necessary and prudent to evaluate the feasibility of a project at earliest possible time. Wastage of manpower and financial resources and untold professional embarrassment can be avoided if an ill-conceived system is recognized early in the development phase. So a detailed study was carried out to check the workability of the proposed system. Feasibility study is a test of system proposal regarding its workability, impact on the organization, ability to meet user needs and effective use of resources. Thus, when an application is proposed, it is normally goes through a feasibility study before it is approved for development.Feasibility and risk analysis is related in many ways. If project risk is great, the feasibility of producing quality is reduced. Thus during feasibility analysis for this project, following three primary areas for interest was considered very carefully. There are several types of feasibility. Three key consideration involved in the feasibility analysis are,

1) Technical Feasibility

2) Operational Feasibility

3) Economic Feasibility

### 4.1.1 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility; this is the technical requirements of the system. Any system developed must not have a importance to consider the monetary factors also it might happen that developing a particular system may be technically possible. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

In this technical feasibility the following issues are taken into considerations. Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investment and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

### 4.1.2 OPERATIONAL FEASIBILITY

The proposed system normally solves the problem and takes advantages of the opportunities identified during scope definitions; it satisfies the requirements identified in the requirement analysis phase of system development. Since the statistical figures are stored in a certain format in the computer, reduce the manual work and enhance the standard of presentation also.

Nonoperational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This measures how well your company will be able to solve problems and take advantage of opportunities that are presented during the course of the project.

### 4.1.3 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.In economic feasibility, cost benefits analysis is done in which expected cost and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system. The developed system is economical when compared to the existing system job done manual system. So the proposed system is so fast that planning can be made easily.

# CHAPTER 5

# SYSTEM REQUIREMENTS AND SPECIFICATION

## 5.1 HARDWARE REQUIREMENTS

| | | |
|---|---|---|
| **Processor** | : | Intel Core I5. |
| **RAM** | : | 8GB or more. |
| **Storage** | : | SSD for read/write operations. |

Any modern computer , laptop.

## 5.2 SOFTWARE REQUIREMENTS

| | | |
|---|---|---|
| **Operating System** | **:** | Windows( 10 Or 11). |
| **Back End Development** | **:** | Python , Flask |
| **Front-End** | : | HTML,CSS. |
| **Library** | : | Pandas , Numpy. |
| **Language** | : | Python. |
| **Server** | : | XAMPP. |

## 5.3 SOFTWARE DESCRIPTION

### 5.3.1 PYTHON

- Python is a high-level, interpreted, interactive and object-oriented scripting Language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner- level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.
- Variables in Python do not require explicit declaration of data types. For instance, integers, floating-point numbers, strings, and booleans are assigned dynamically.
- Control structures in Python, such as if statements, for loops, and while loops, provide powerful mechanisms for decision-making and repetition in code execution. If statements allow for conditional branching, while loops iterate based on a condition, and for loops iterate over a sequence of items.
- Functions in Python are defined using the `def` keyword followed by the function name and parameters. This allows for modularizing code and

promoting reusability. Python also supports lambda functions for creating small anonymous functions.

- Python offers a wide range of built-in data structures. Lists, for example, are mutable sequences that can contain a variety of data types. Tuples are similar to lists but are immutable. Dictionaries are unordered collections of key-value pairs, and sets are unordered collections of unique elements.
- Python's standard library is extensive and versatile. It includes modules for mathematics, random number generation, file handling, and more. For instance, the `math` module provides functions for mathematical operations, `random` module for random number generation, and `os` module for interacting with the operating system.
- Exception handling in Python enables graceful handling of errors and exceptions that may occur during program execution. This is achieved using `try`, `except`, and optional `finally` blocks.
- Python supports object-oriented programming (OOP) features such as classes, objects, inheritance, polymorphism, and encapsulation. Classes encapsulate data and behavior into objects, allowing for clean and organized code design.

- In summary, Python's simplicity, versatility, and extensive standard library make it a popular choice for a wide range of applications, including web development, data analysis, scientific computing, machine learning, and more. Its ease of use and readability make it an excellent language for both beginners and experienced developers alike.

**Assigning Values to Variables**

- **Locating Modules**
- When you import a module, the Python interpreter searches for the module in the following sequences −
- The current directory.
- If the module isn't found, Python then searches each directory in the shell variable PYTHONPATH.
- If all else fails, Python checks the default path. On UNIX, this default path is normally /usr/local/lib/python/.
- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.
- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

### 5.3.2 MYSQL

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

Database Queries

A query is a question or a request. We can query a database for specific information and have a recordset returned.
Look at the following query (using standard SQL):
SELECT LastName FROM Employees

The query above selects all the data in the "LastName" column from the "Employees" table.

### 5.3.3 PHP OVERVIEW

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language .PHP scripts are executed on the server .PHP is free to download and use.

**Note**

It is powerful enough to be at the core of the biggest blogging system on the web WordPress. It is deep enough to run the largest social network Facebook. It is also easy enough to be a beginner's first server side language!

**PHP FILE**

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"
- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies

- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

**NEED OF PHP:**

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

A PHP script is executed on the server, and the plain HTML result is sent back to the browser.

**BASIC PHP SYNTAX:**

A PHP script can be placed anywhere in the document.

A PHP script starts with **<?php** and ends with **?>**:

```
<? php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php".A PHP file normally contains HTML tags, and some PHP scripting code.

**COMMENTS IN PHP:**

A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is looking at the code.

Comments can be used to:

- Let others understand what you are doing

- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code.

**PHP VARIABLES:**

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _ )
- Variable names are case-sensitive ($age and $AGE are two different variables).

**PHP VARIABLES SCOPE:**

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/ used.

PHP has three different variable scopes:

- Local
- Global
- Static

Global and Local Scope

A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function. A variable declared within a function has a LOCAL SCOPE and can only be accessed within that function

PHP the global Keyword

The global keyword is used to access a global variable from within a function. To do this, use the global keyword before the variables (inside the function).PHP also stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.

# CHAPTER 6

# WORKING OF SYSTEM

## 6.1. SYSTEM ARCHITECTURE

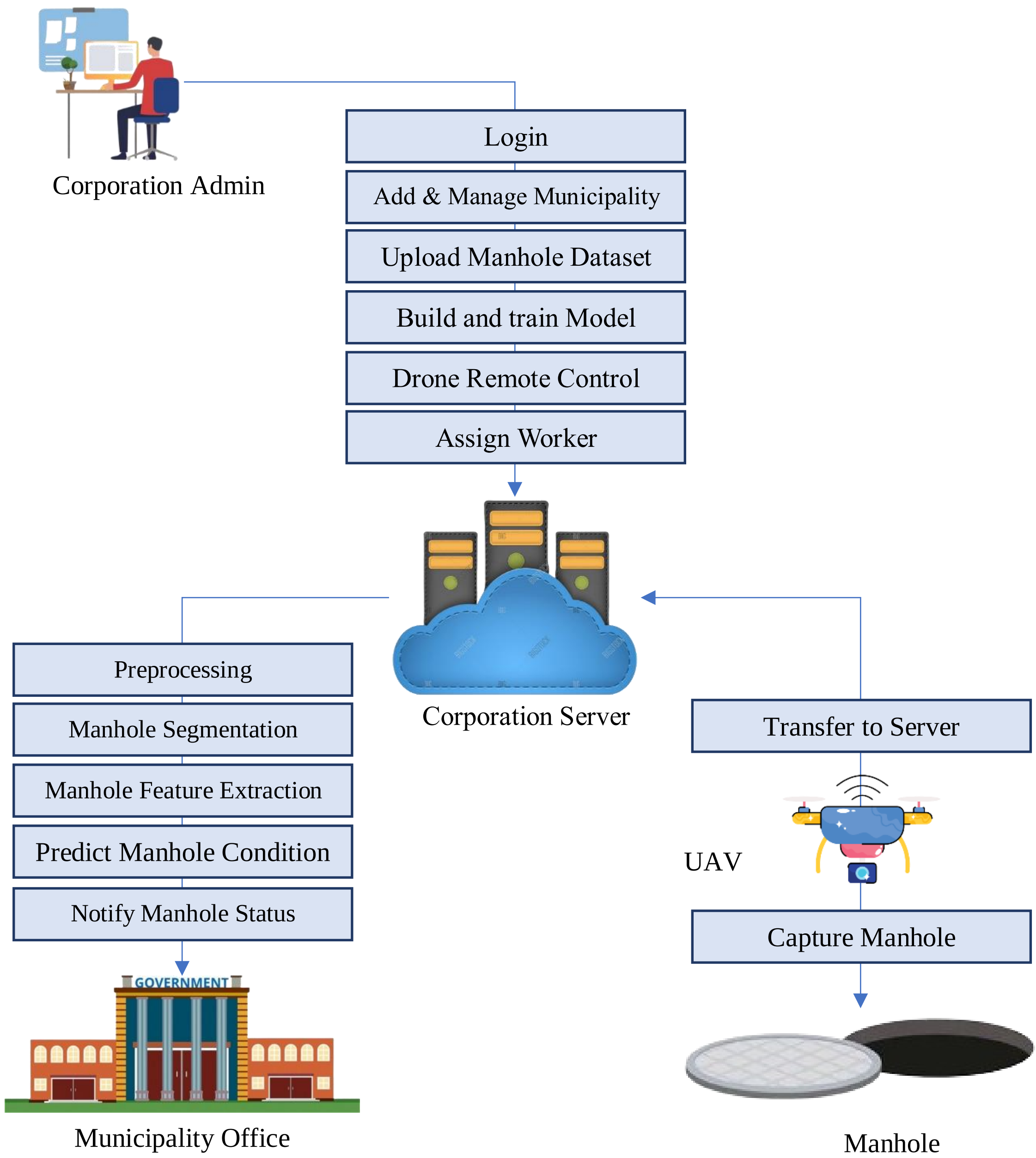



**Fig 6.1**

## 6.2 DATAFLOW DIAGRAM

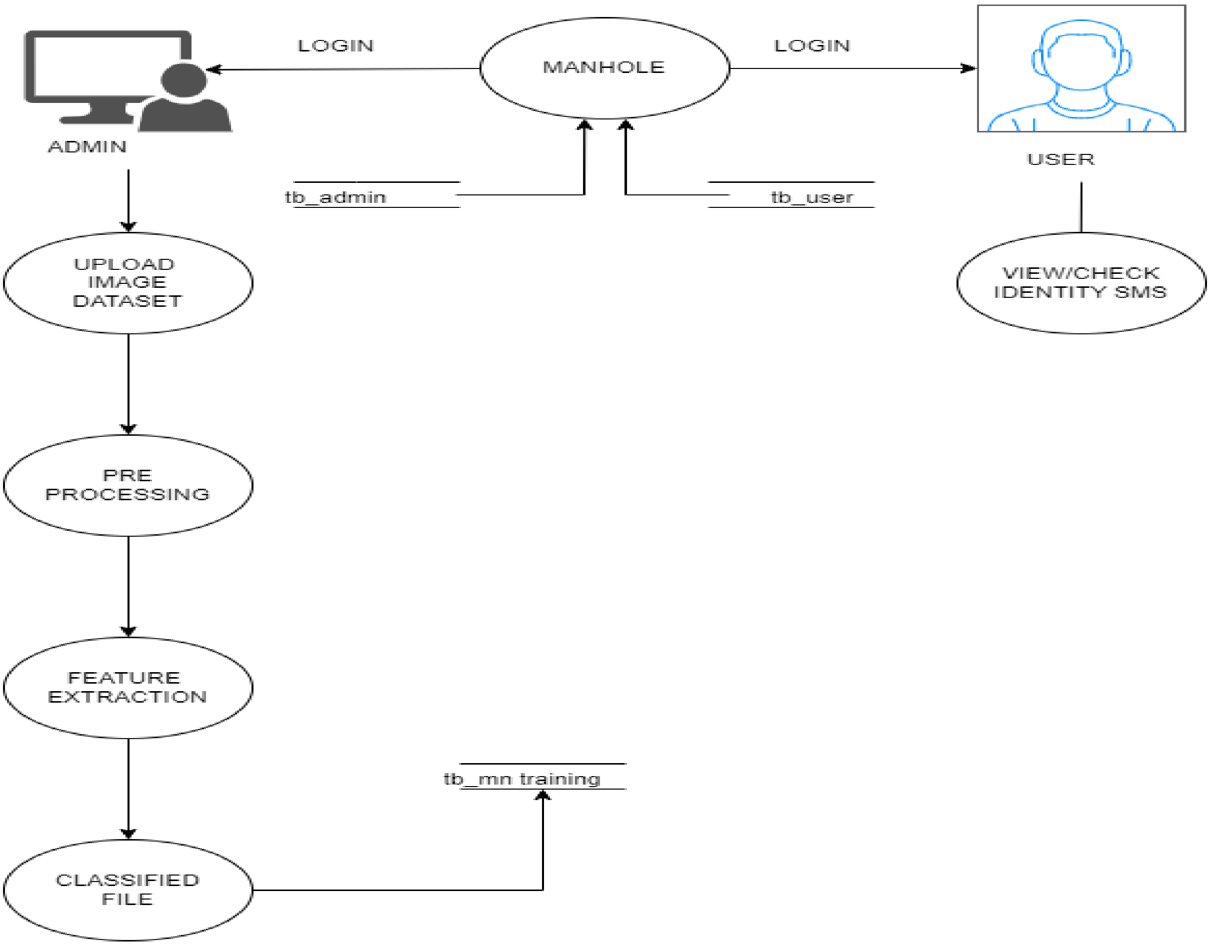



**Fig 6.2**

## 6.3 UML DIAGRAM

### 6.3.1 USECASE DIAGRAM

Use case diagrams are referred to as behaviors diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

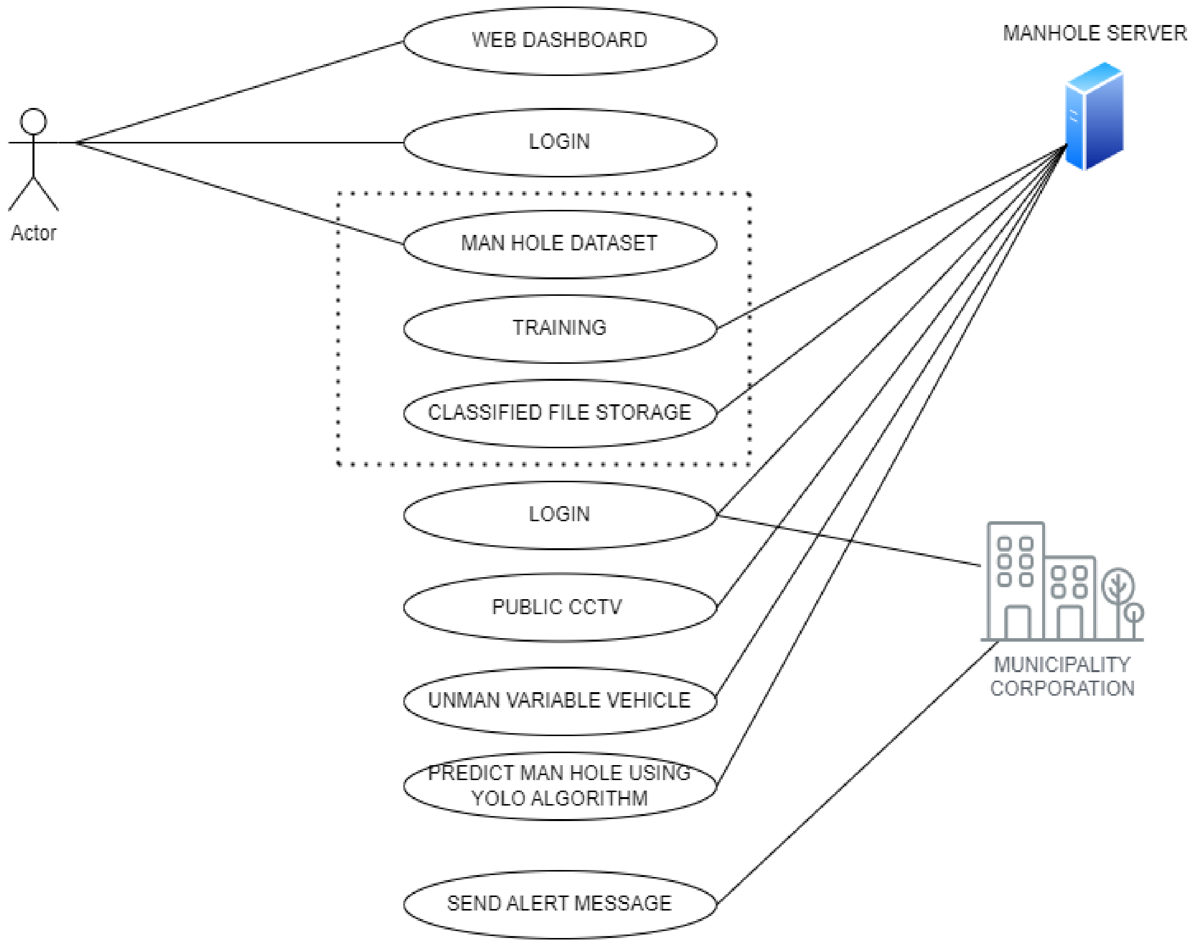



**Fig 6.3.1**

### 6.3.2 CLASS DIAGRAM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing a system's classes, their attributes operations (or methods), and the relationships among objects. The class diagram is the main building block of object oriented modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent boy the main elements, interactions in the applications, and the classes to be programmed.

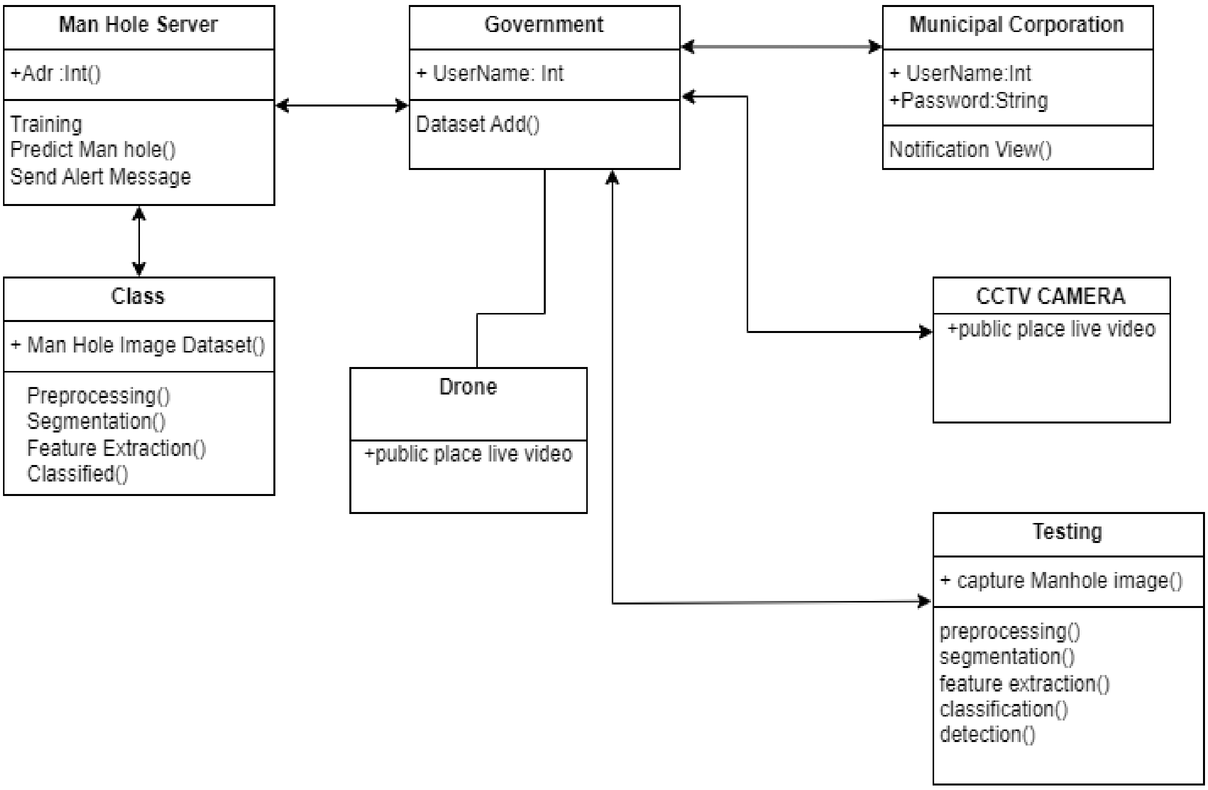



**Fig 6.3.2**

### 6.3.3 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

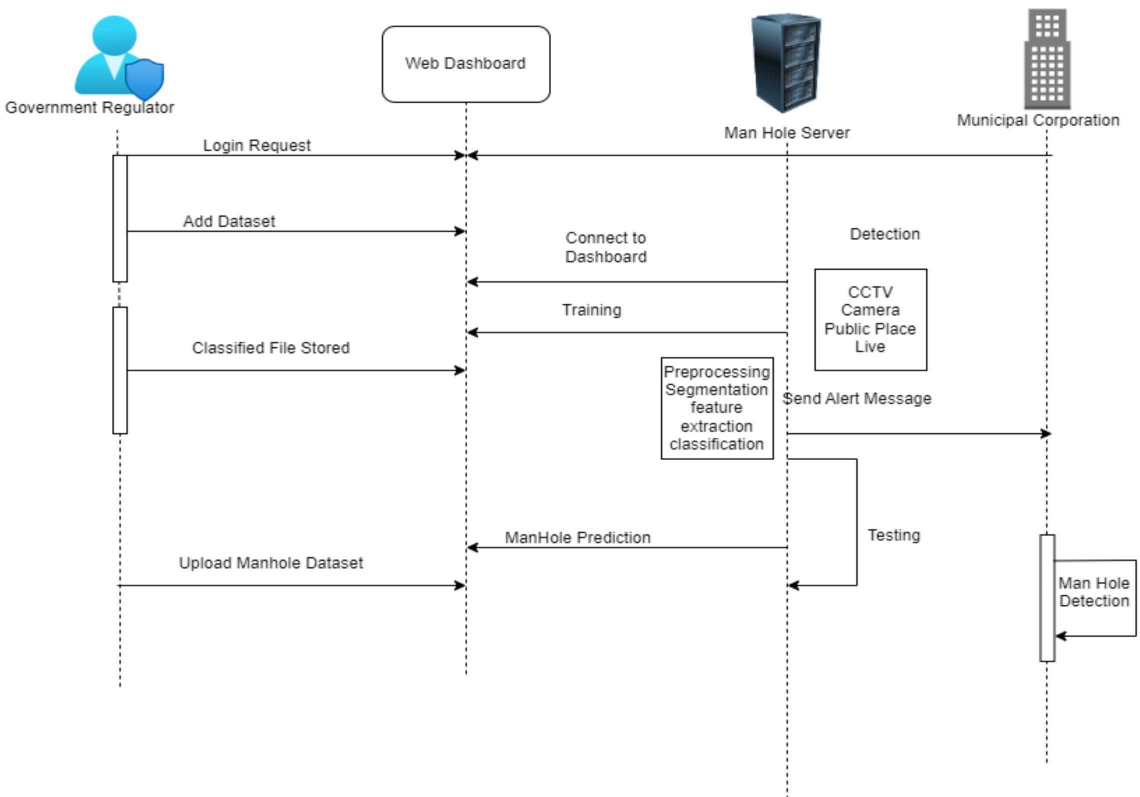



**Fig 6.3.3**

### 6.3.4COLLABORATION DIAGRAM

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

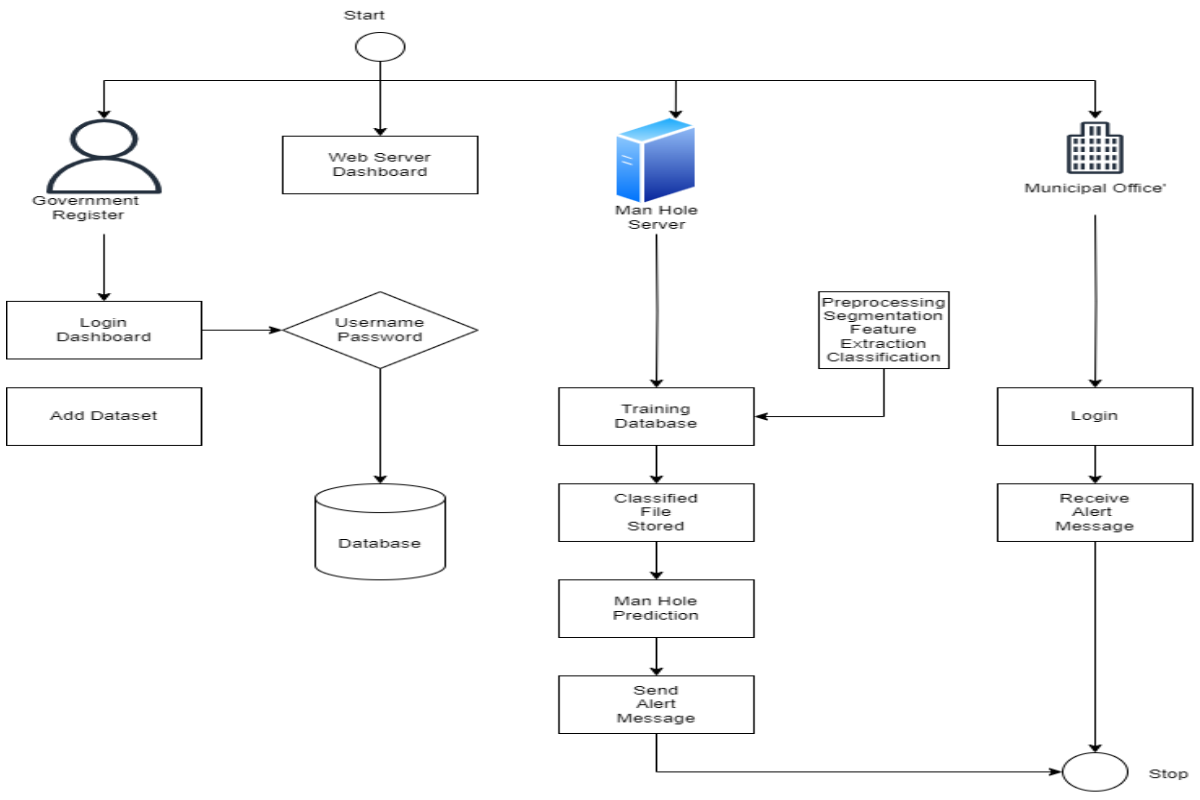



Fig 6.3.4

# CHAPTER 7

# SYSTEM IMPLEMENTATION

## 7.1 SYSTEM DESCRIPTION

System implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successfully system and in giving confidence on the new system for the user that it will work efficiently and effectively.

## 7.2 MODULES

### 7.2.1 User Interface

- **Web Admin Interface**

The Web Admin interface is accessible only to the authorized admin users. The admin can log in to the system with their credentials and perform various tasks like training the Manhole model, adding and deleting users, and monitoring the system logs.

- **Citizen or User Interface**

The citizen or user interface is accessible to all registered users of the system. The user can upload the manhole images and get the prediction results of manhole defects.

- **Municipality Officer Interface**

The Municipality Officer Interface is accessible to authorized officers who can view the defected manhole images, its location, and the severity of the damage. Based on this information, they can plan to schedule repairs for identified defects.

### 7.2.2 Manhole Classifier: Build and Train

- **Dataset Collection**

This component involves collecting a large and diverse set of manhole images to build a comprehensive dataset.

- **Import and Visualize Manhole Image Dataset**

The import and visualize manhole image dataset module of the manhole predictor web app is responsible for importing the manhole images dataset from the user's computer or cloud storage and visualizing them in a user-friendly manner.

- **Pre-processing**

The pre-processing module of the Manhole Predictor Web App involves several steps to prepare the manhole images for feature extraction and classification.

- **Segmentation**

The Segmentation module of the Manhole Predictor Web App is responsible for identifying the region of interest (ROI) in the manhole image.

- **Feature Extraction**

The Feature Extraction module of the Manhole Predictor Web App is responsible for extracting important features from the segmented manhole images.

- **Classification**

The classification module of the Manhole Predictor web app is responsible for predicting the condition of the manhole image

### 7.2.3 Build and train Model

The Build and Train module of the Manhole Predictor Web App is responsible for building and training the deep learning models used for manhole classification, prediction, and localization.

- **Input Image:** The input image of the manhole is received from the user interface.
- **Pre-processing:** The input image is pre-processed, which includes converting the RGB image to grayscale, resizing the image to a standard size, de-noising the image, and binarizing it.
- **Segmentation:** The pre-processed image is then passed through a region proposal network (RPN) to segment the manhole cover from the background.
- **Feature Extraction:** After segmentation, grey-level co-occurrence matrix (GLCM) features are extracted from the segmented manhole cover image.
- **Classification:** The extracted features are then used as input to a convolutional neural network (CNN) for classification of the manhole cover's condition.

- **Prediction:** The output of the CNN is the classified the manhole cover's condition (1-Close, 0-Open, 2-Broken, 4-No Manhole).
- **Output:** The prediction class is returned as output and displayed to the user through the web app interface.

### 7.2.4 Notification

The notification module of the Manhole Predictor Web App is responsible for sending notifications to the Municipality Officer regarding the identified manhole defects.

### 7.2.5 Combine CNN and YOLO Algorithm

**Data Collection and Preprocessing**:

- Collect a dataset of images of manhole covers, including normal and defective covers.
- Preprocess the images to enhance quality, remove noise, and standardize dimensions.

**CNN Feature Extraction**:

- Utilize a pre-trained CNN model (e.g., VGG16, ResNet) to extract features from the preprocessed images.
- Fine-tune the CNN on the manhole cover dataset to adapt it to the specific task of cover detection.

**YOLO Object Detection**:

- Implement the YOLO algorithm to perform real-time object detection on the feature maps generated by the CNN.
- Train the YOLO model to detect manhole covers and classify them as open or closed.

**Integration**:

- Integrate the CNN feature extraction and YOLO object detection stages into a unified pipeline.
- Fine-tune the integrated model on the manhole cover dataset to optimize performance.

# CHAPTER 8
# TESTING

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

## PSYCHOLOGY OF TESTING

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

## TESTING OBJECTIVES

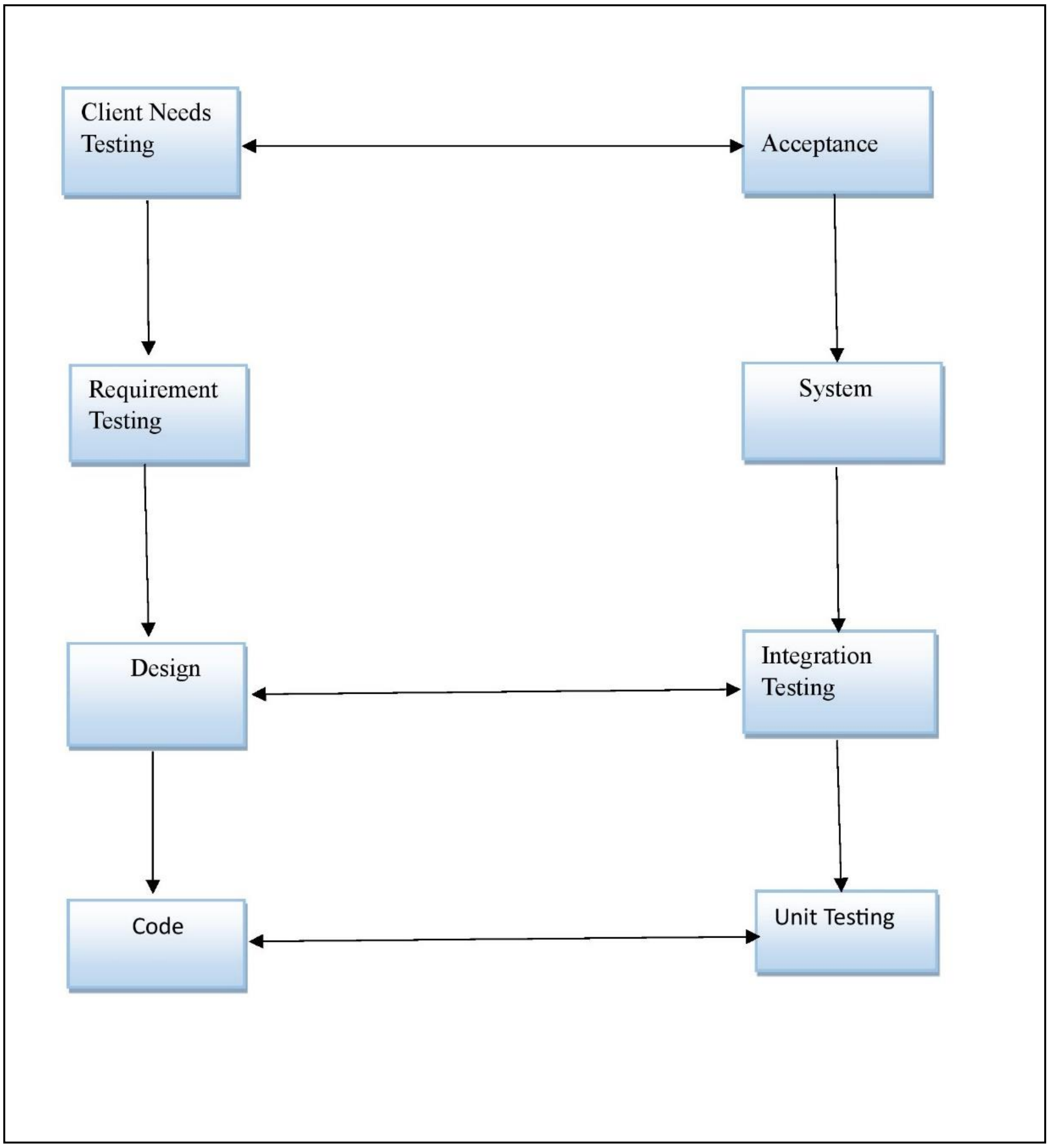
The main objective of the testing is to uncover the host of errors, systematically and with minimum effort and time. Stating formally we can say,

Testing is the process of executing a program with the intent of finding errors.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding errors, if exists.
- The tests are adequate to detect possibly present error.
- The software more or less confirms to the quality and reliable standards.

## 8.1 LEVELS OF TESTING

In order to uncover the error present in different phases we have the concept of levels of testing. The basic levels of testing are,
of levels of testing.



### 8.1.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover error within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project "Driver assistant for the detection of drowsiness and emergency alert " , each service through of a module. When developing the modules as well as finishing the development so that each module works without any error. The inputs are validated when accepting from user.

### 8.1.2 INTEGRATION TESTING

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, emphasis being on testing interface between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project, "Driver assistant for the detection of drowsiness and emergency alert ", the main system performed by integrating all the modules.

When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of input with which the four services run perfectly before integration.

### 8.1.3 SYSTEM TESTING

Here the entire software system is tested. The reference document for this process is the requirements documents, and the goal so to see if software meets its requirements documents. Here entire "Driver assistant for the detection of drowsiness and emergency alert", has been tested against requirements of projects and it is checked whether all requirements of project have been satisfied or not.

### 8.1.4 ACCEPTANCE TESTING

Acceptance test is performed with realistic data of the client to documents that the software is working satisfactorily. Testing here is focused on external behavior of the system: the internal logic of program is not emphasized. In this project for "Driver assistant for the detection of drowsiness and emergency alert". I have collected some data and tested whether project is working correctly or not.

Testing phase is an important part of development. It is the process of finding errors and missing operations and also a complete verification to determine whether objectives are met and the user requirements are satisfied.

### 8.1.5 WHITE BOX TESTING

This is a unit testing method where a unit will be at a time and tested thoroughly at a statement level2 find the maximum possible errors. I tested step wise every piece of code, taking care that every statement in the code is executed at least once.

The white box testing is also called glass box testing. I have generated the list of test cases, sample data, which is used to check all possible combinations of execution paths through the code at every module level.

### 8.1.6 BLACK BOX TESTING

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details as statement level.

Here the module will be treated as black box that will take some input and generate output. Output for a given set of inputs combinations are forwarded to other modules.

# CHAPTER 9
# SAMPLE CODE

**Packages**

```
from flask import Flask, render_template, Response, redirect, request, session,
abort, url_for
import mysql.connector
import hashlib
import datetime
from urllib.request import urlopen
import webbrowser
import cv2
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import shutil
import imagehash
from werkzeug.utils import secure_filename
from PIL import Image
import argparse
import urllib.request
import urllib.parse
import seaborn as sns
from PIL import Image, ImageOps
import scipy.ndimage as ndi
from skimage import transform
from keras.preprocessing.image import ImageDataGenerator , load_img ,
img_to_array
```

from keras.models import Sequential

from keras.layers import Conv2D, Flatten, MaxPool2D, Dense

**Training**

#Preprocessing

img = cv2.imread('static/dataset/'+fname)

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imwrite("static/trained/g_"+fname, gray)

#noice

img = cv2.imread('static/trained/g_'+fname)

dst = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 15)

fname2='ns_'+fname

cv2.imwrite("static/trained/"+fname2, dst)

def kmeans_color_quantization(image, clusters=8, rounds=1):

h, w = image.shape[:2]

samples = np.zeros([h*w,3], dtype=np.float32)

count = 0

for x in range(h):

for y in range(w):

samples[count] = image[x][y]

count += 1

compactness, labels, centers = cv2.kmeans(samples,

clusters,

None,

(cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10000,

0.0001),

rounds,

cv2.KMEANS_RANDOM_CENTERS)

centers = np.uint8(centers)

res = centers[labels.flatten()]

```
return res.reshape((image.shape))
#bin
'''image = cv2.imread('static/dataset/'+fname)
original = image.copy()
kmeans = kmeans_color_quantization(image, clusters=4)
# Convert to grayscale, Gaussian blur, adaptive threshold
gray = cv2.cvtColor(kmeans, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (3,3), 0)
thresh =
cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY_INV,21,2)
# Draw largest enclosing circle onto a mask
mask = np.zeros(original.shape[:2], dtype=np.uint8)
cnts = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
for c in cnts:
((x, y), r) = cv2.minEnclosingCircle(c)
cv2.circle(image, (int(x), int(y)), int(r), (36, 255, 12), 2)
cv2.circle(mask, (int(x), int(y)), int(r), 255, -1)
break
# Bitwise-and for result
result = cv2.bitwise_and(original, original, mask=mask)
result[mask==0] = (0,0,0)
cv2.imshow('thresh', thresh)
cv2.imshow('result', result)
cv2.imshow('mask', mask)
cv2.imshow('kmeans', kmeans)
```

```
cv2.imshow('image', image)
cv2.imwrite("static/trained/bb/bin_"+fname, thresh)
##RPN
img = cv2.imread('static/trained/g_'+fname)
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ret, thresh =
cv2.threshold(gray,0,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
kernel = np.ones((3,3),np.uint8)
opening = cv2.morphologyEx(thresh,cv2.MORPH_OPEN,kernel, iterations = 2)
# sure background area
sure_bg = cv2.dilate(opening,kernel,iterations=3)
# Finding sure foreground area
dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
ret, sure_fg = cv2.threshold(dist_transform,1.5*dist_transform.max(),255,0)
# Finding unknown region
sure_fg = np.uint8(sure_fg)
segment = cv2.subtract(sure_bg,sure_fg)
img = Image.fromarray(img)
segment = Image.fromarray(segment)
path3="static/trained/sg/sg_"+fname
#segment.save(path3)
###
g1=fname.split(".")
g2=g1[0]
ffname=g2+".png"
g3=g2.split('-')
if g3[1]==ex[3]:
a=1
segment.save("static/trained/sg/"+fname)
```

```
else:
image = cv2.imread('static/dataset/'+fname)
mask = cv2.imread('static/trained/cc/'+ffname, 0)
mask = cv2.threshold(mask, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)[1]
image[mask==255] = (36,255,12)
cv2.imwrite("static/trained/classify/"+ffname, image)

###Feature extraction & Classification
def CNN():
#Lets start by loading the Cifar10 data
(X, y), (X_test, y_test) = cifar10.load_data()
#Keep in mind the images are in RGB
#So we can normalise the data by diving by 255
#The data is in integers therefore we need to convert them to float first
X, X_test = X.astype('float32')/255.0, X_test.astype('float32')/255.0
#Then we convert the y values into one-hot vectors
#The cifar10 has only 10 classes, thats is why we specify a one-hot
#vector of width/class 10
y, y_test = u.to_categorical(y, 10), u.to_categorical(y_test, 10)
model = Sequential()
#We want to output 32 features maps. The kernel size is going to be
#3x3 and we specify our input shape to be 32x32 with 3 channels
#Padding=same means we want the same dimensional output as input
#activation specifies the activation function
model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), padding='same',
activation='relu'))
#20% of the nodes are set to 0
model.add(Dropout(0.2))
```

```
#now we add another convolution layer, again with a 3x3 kernel
#This time our padding=valid this means that the output dimension can
model.add(Conv2D(32, (3, 3), activation='relu', padding='valid'))
#maxpool with a kernet of 2x2
model.add(MaxPooling2D(pool_size=(2, 2)))
#In a convolution NN, we neet to flatten our data before we can
#input it into the ouput/dense layer
model.add(Flatten())
#Dense layer with 512 hidden units
model.add(Dense(512, activation='relu'))
#this time we set 30% of the nodes to 0 to minimize overfitting
model.add(Dropout(0.3))
#Finally the output dense layer with 10 hidden units corresponding to
#our 10 classe
model.add(Dense(10, activation='softmax'))
#Few simple configurations
model.compile(loss='categorical_crossentropy',
optimizer=SGD(momentum=0.5, decay=0.0004), metrics=['accuracy'])
#Run the algorithm!
model.fit(X, y, validation_data=(X_test, y_test), epochs=25,
batch_size=512)
#Save the weights to use for later
model.save_weights("cifar10.hdf5")
#Finally print the accuracy of our model!
print("Accuracy: &2.f%%" %(model.evaluate(X_test, y_test)[1]*100))
i=0
vd=[]
data4=[]
while i<4:
```

```
vt=[]
vi=i+1
vv[i]
vt.append(cname[i])
vt.append(str(vv[i]))
vd.append(str(vi))
data4.append(vt)
i+=1
dd2=vv
doc = cname #list(data.keys())
values = dd2 #list(data.values())
fig = plt.figure(figsize = (10, 8))
# creating the bar plot
cc=['green','yellow','orange','red']
plt.bar(doc, values, color =cc,
width = 0.4)
plt.ylim((1,20))
plt.xlabel("Class")
plt.ylabel("Count")
plt.title("")
fn="tclass.png"
#plt.xticks(rotation=20)
plt.savefig('static/trained/'+fn)
plt.close()
#Test
ff=open("static/trained/class.txt",'r')
ext=ff.read()
ff.close()
cname=ext.split(',')
```

```
if request.method=='POST':
file = request.files['file']
if file.filename == '':
flash('No selected file')
return redirect(request.url)
if file:
fname = file.filename
filename = secure_filename(fname)
f1=open('static/test/file.txt','w')
f1.write(filename)
f1.close()
file.save(os.path.join("static/test", filename))
cutoff=1
path_main = 'static/dataset'
for fname1 in os.listdir(path_main):
hash0 = imagehash.average_hash(Image.open("static/dataset/"+fname1))
hash1 = imagehash.average_hash(Image.open("static/test/"+filename))
cc1=hash0 - hash1
print("cc="+str(cc1))
if cc1<=cutoff:
ss="ok"
fn=fname1
print("ff="+fn)
break
else:
ss="no"
gs=get_data.split('|')
fn=gs[1]
ts=gs[2]
```

```
nn=gs[3]
ff=open("static/trained/class.txt",'r')
ext=ff.read()
ff.close()
cname=ext.split(',')
n=int(nn)
i=0
for cc in cname:
if i==n:
res=cc
break
i+=1
r1=res.split('-')
res1=r1[1]
```

**MYSQL**

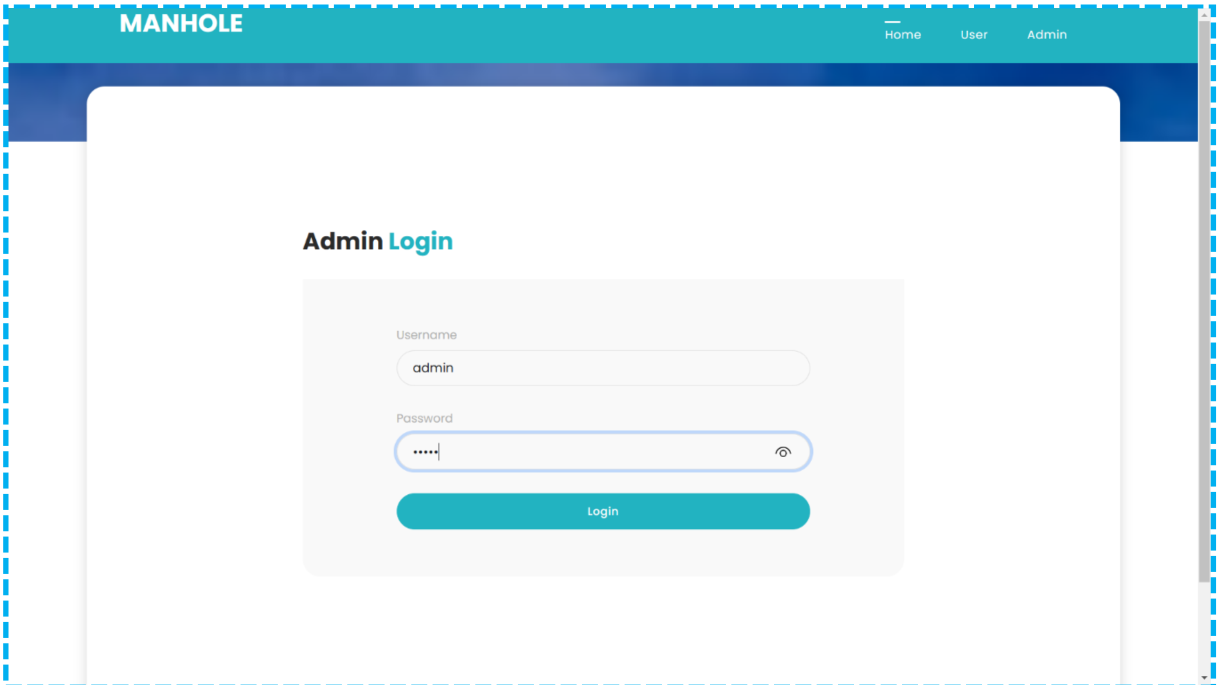
```
-- phpMyAdmin SQL Dump
-- version 2.11.6
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Apr 14, 2023 at 10:34 AM
-- Server version: 5.0.51
-- PHP Version: 5.2.6

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";


/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
```

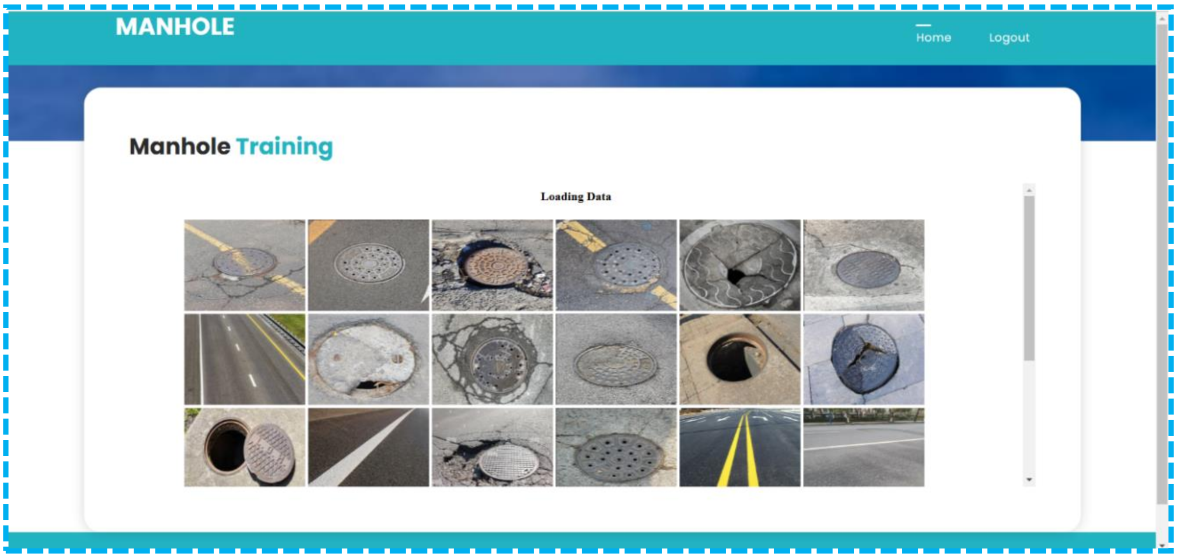
```
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8 */;

--
-- Database: `manhole`
--

-- --------------------------------------------------------

--
-- Table structure for table `admin`
--

CREATE TABLE `admin` (
  `username` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`username`, `password`) VALUES
('admin', 'admin');

--
-- Table structure for table `register`
--

CREATE TABLE `register` (
  `id` int(11) NOT NULL,
  `name` varchar(20) NOT NULL,
  `email` varchar(30) NOT NULL,
  `mobile` bigint(20) NOT NULL,
  `uname` varchar(20) NOT NULL,
  `pass` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `register`
--
```

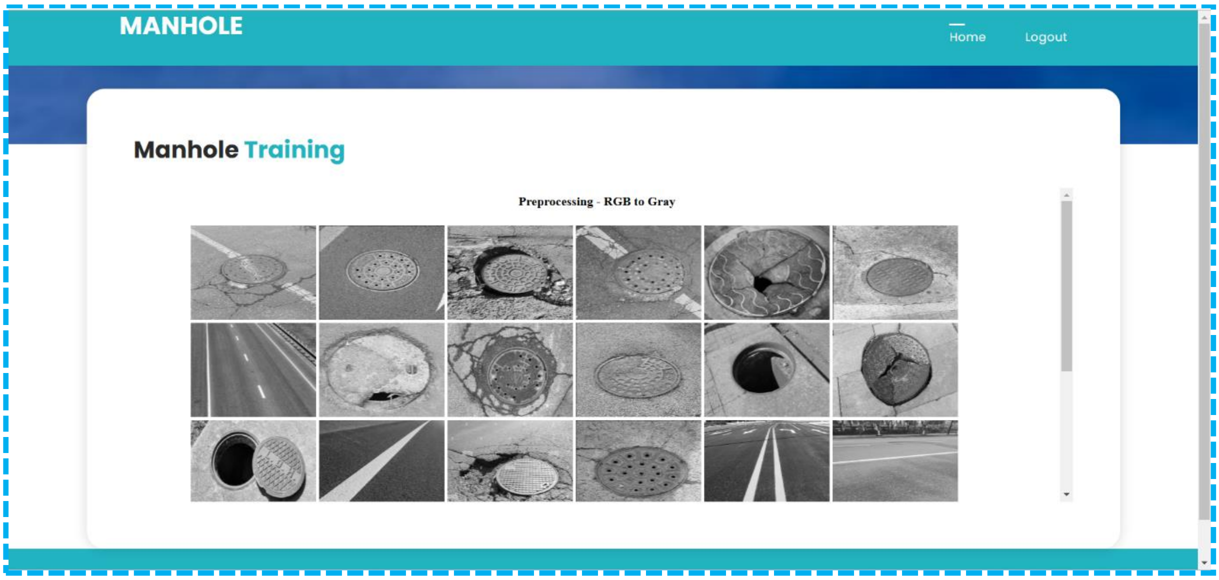
```sql
INSERT INTO `register` (`id`, `name`, `email`, `mobile`, `uname`, `pass`)
VALUES
(1, 'Thiru', 'thiru@gmail.com', 9582659563, 'thiru', '123456'),
(2, 'Raja', 'raja@gmail.com', 9662582255, 'raja', '123456');

-- --------------------------------------------------------

--
-- Table structure for table `result`
--

CREATE TABLE `result` (
  `id` int(11) NOT NULL,
  `uname` varchar(20) NOT NULL,
  `status` varchar(20) NOT NULL,
  `latitude` varchar(20) NOT NULL,
  `longitude` varchar(20) NOT NULL,
  `date_time` timestamp NOT NULL default CURRENT_TIMESTAMP on
update CURRENT_TIMESTAMP,
  `img_name` varchar(20) NOT NULL,
  `process_st` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `result`
--

INSERT INTO `result` (`id`, `uname`, `status`, `latitude`, `longitude`,
`date_time`, `img_name`, `process_st`) VALUES
(1, 'raja', 'Open', '13.013', '80.1697', '2023-03-28 20:14:27', 'M1.png', 0),
(2, 'raja', 'Open', '10.836258', '78.689229', '2023-03-28 20:08:45', 'M2.png', 0),
(3, 'raja', 'Broken', '10.836258', '78.689229', '2023-03-28 20:24:52', 'M3.png', 1);
```

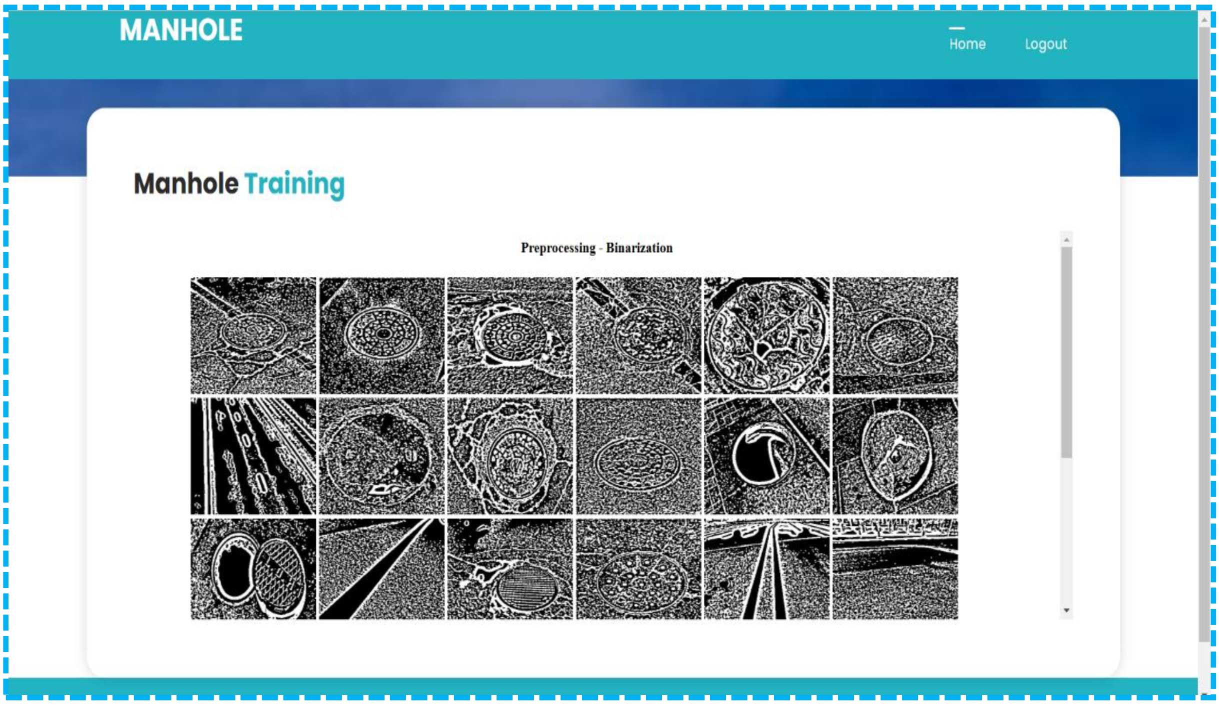
# CHAPTER 10

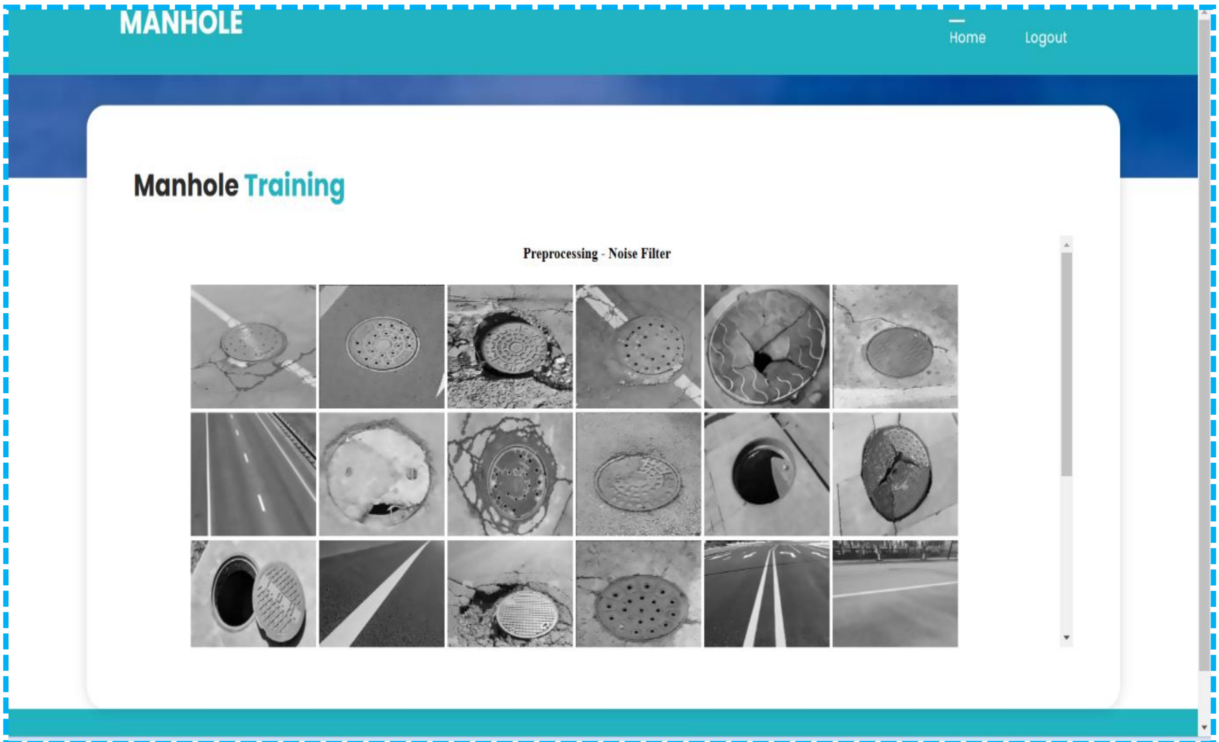
# SCREENSHOTS

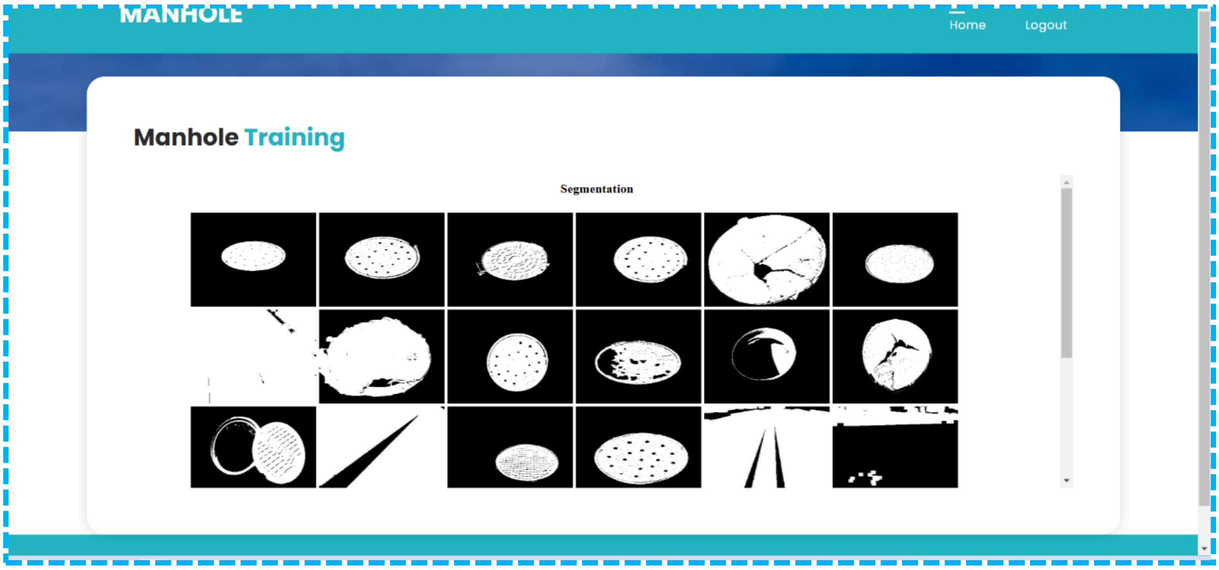



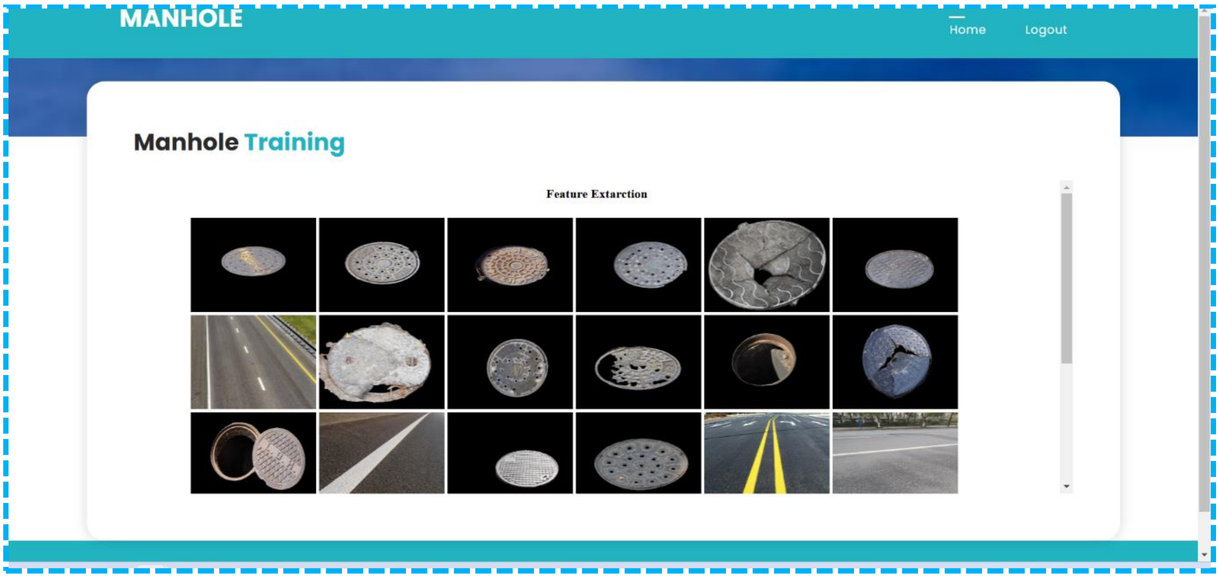

MANHOLE
Home
Logout
Manhole Training
Loading Data

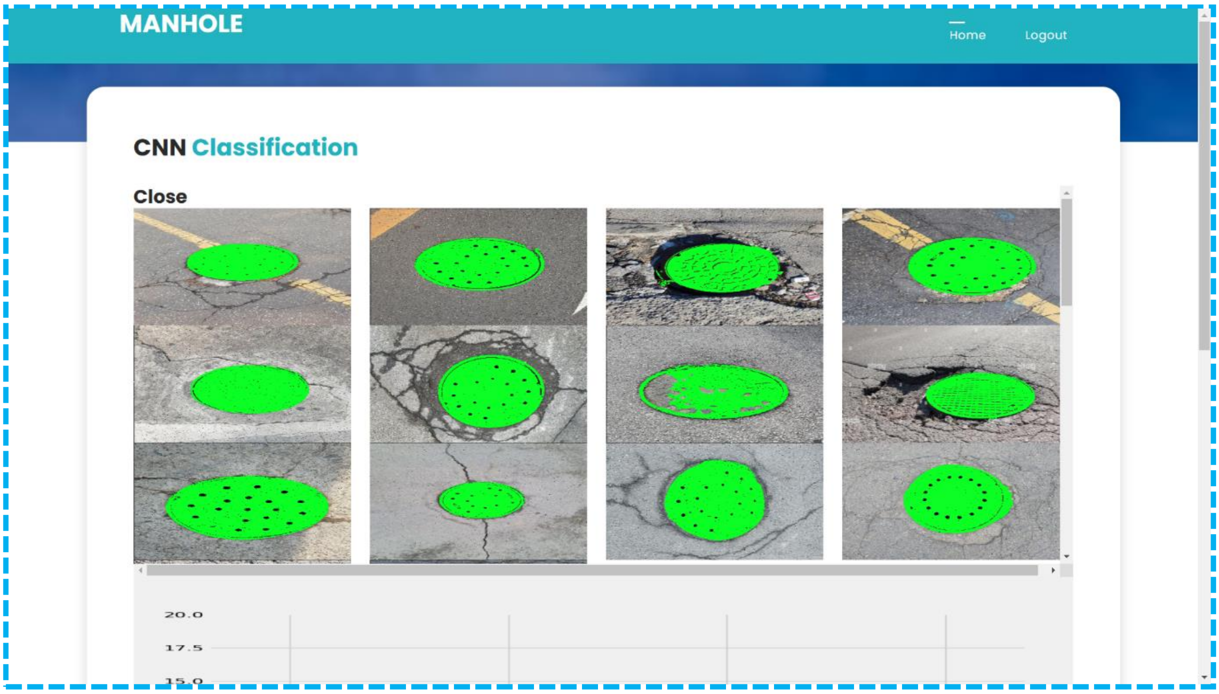

MANHOLE
Home
Logout
Manhole Training
Preprocessing - RGB to Gray

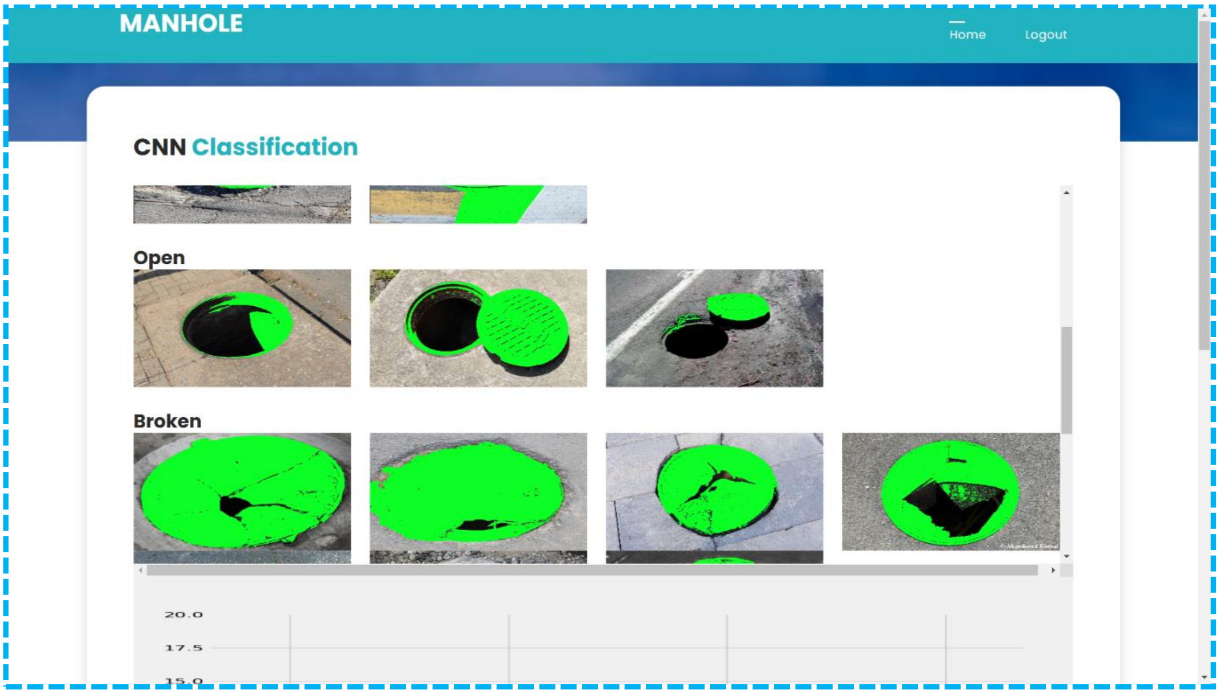

MANHOLE
Home
Logout
Manhole Training
Preprocessing - Binarization

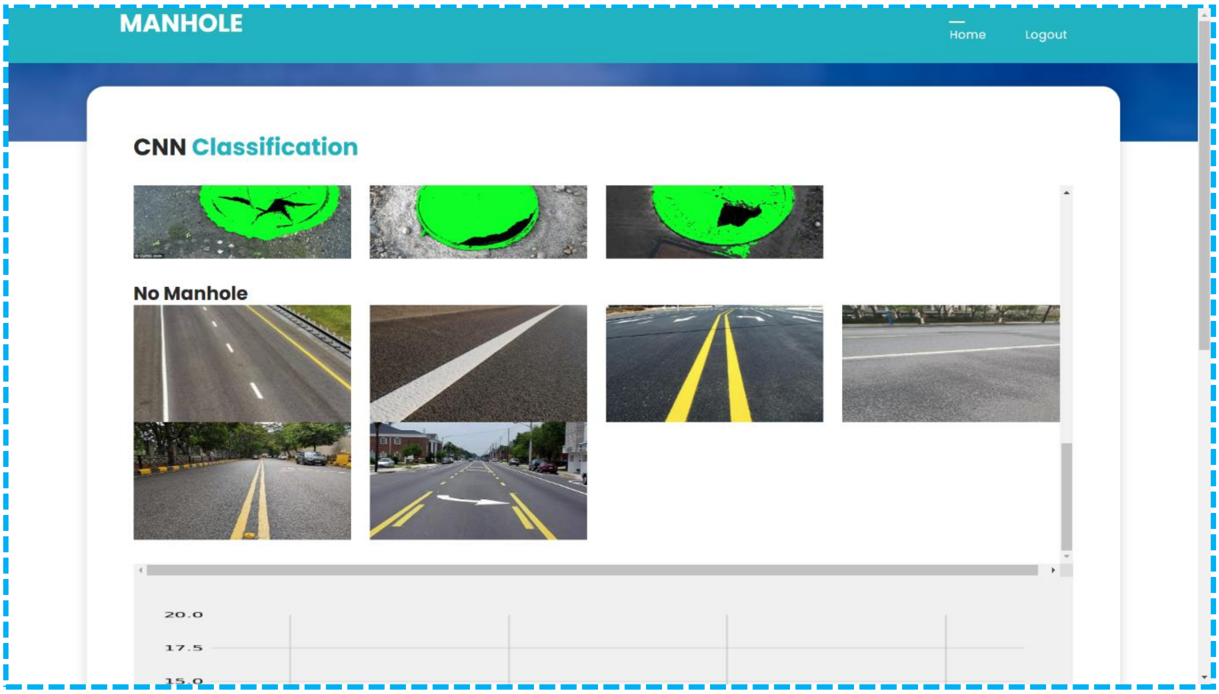

MANHOLE
Home
Logout
Manhole Training
Preprocessing - Noise Filter


MANHOLE
Home
Logout
Manhole Training
Segmentation


MANHOLE
Home
Logout
Manhole Training
Feature Extarction

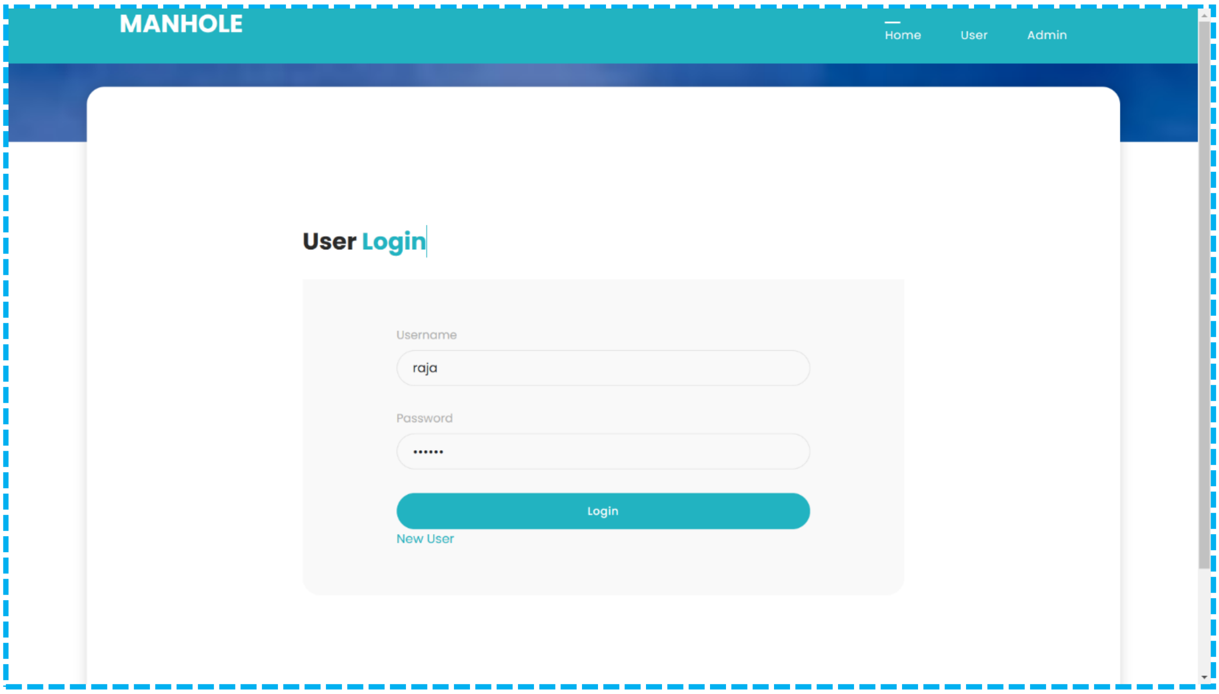

MANHOLE
Home
Logout
CNN Classification
Close
20.0
17.5
15.0

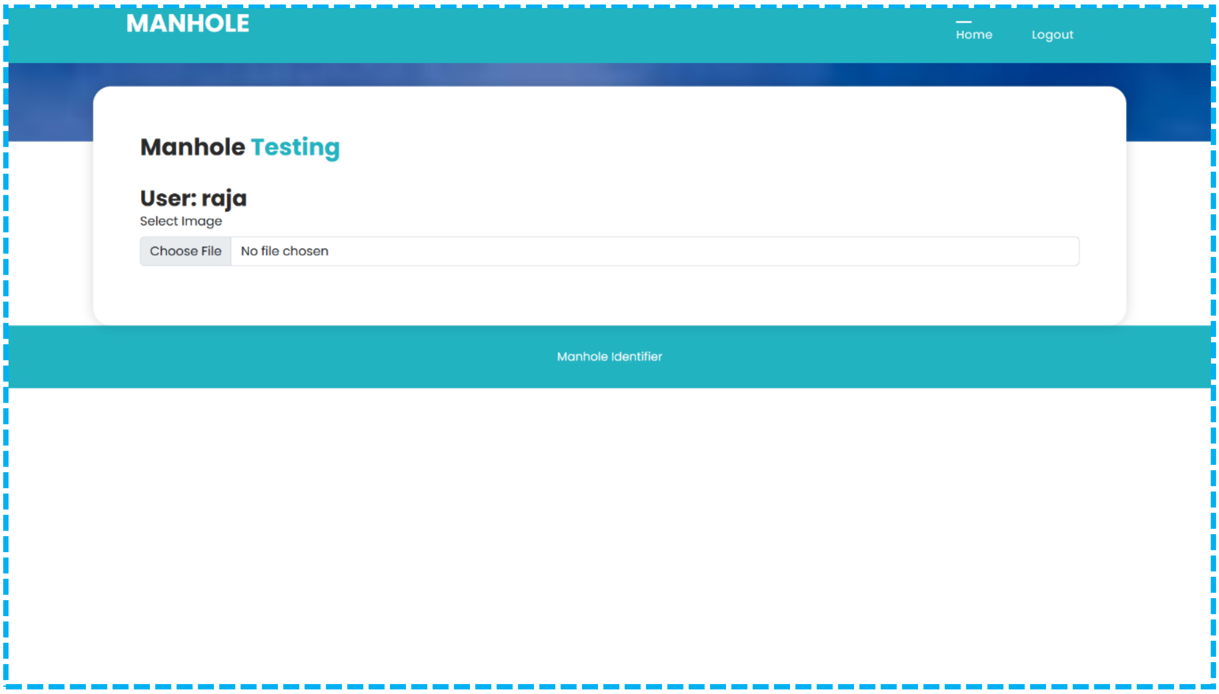

MANHOLE
Home
Logout
CNN Classification
Open
Broken
20.0
17.5
15.0


MANHOLE
Home
Logout
CNN Classification
No Manhole
20.0
17.5

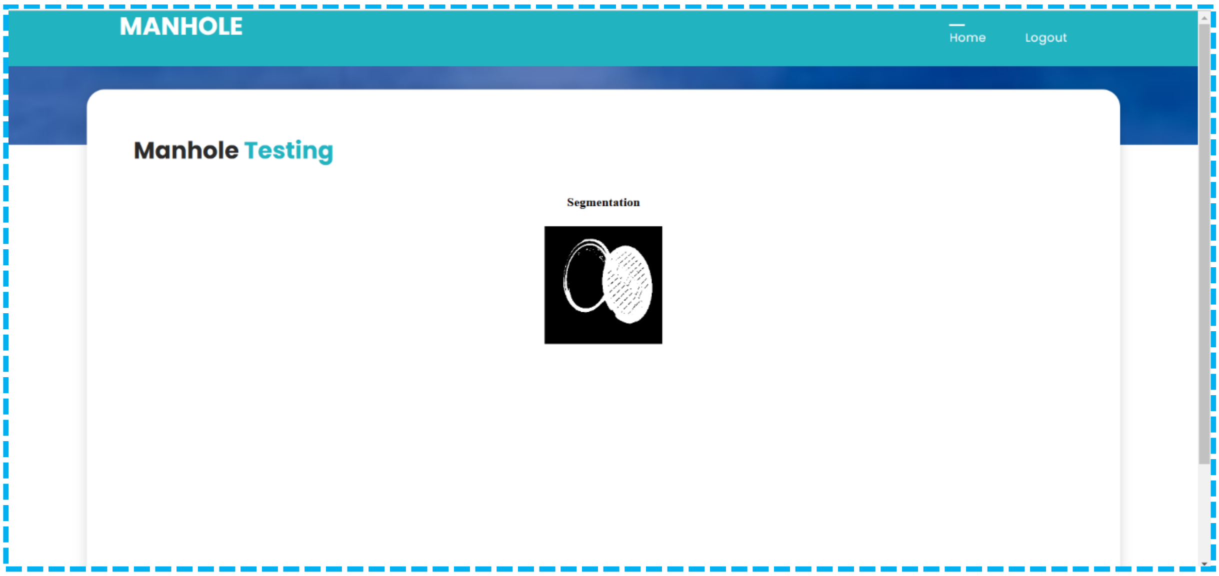

MANHOLE
Home
Logout
Open
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
Count
Close
Open
Broken
No Manhole
Class
Performance Analysis
Model Accuracy
Model Loss

15.0
12.5
10.0
7.5
5.0
2.5
Count
Close
Open
Broken
No Manhole
Class
Performance Analysis
Model Accuracy
Model Loss
Accuracy %
Epochs
train
val
Model loss
Manhole Identifier


MANHOLE
Home
User
Admin
User Login
Username
raja
Password
Login
New User

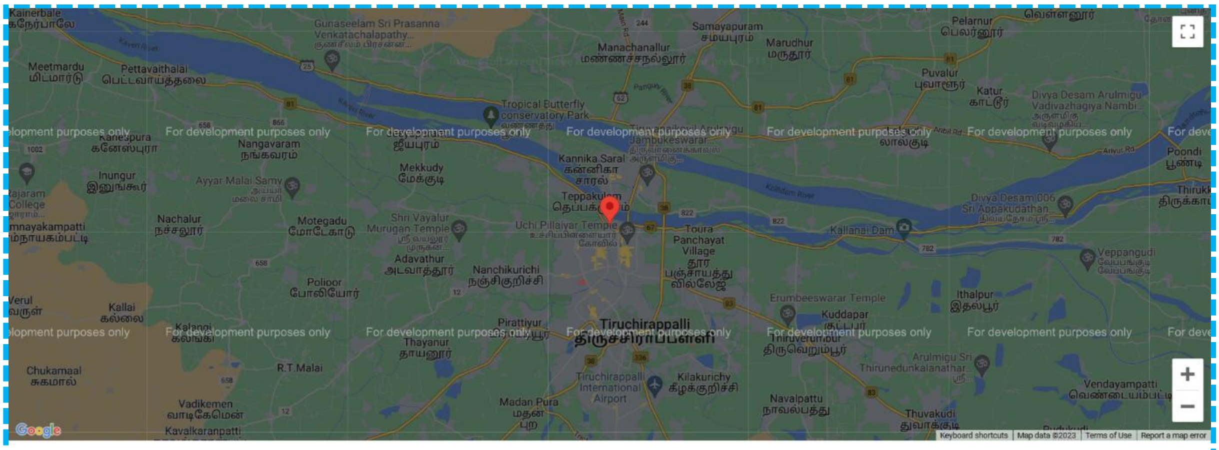

MANHOLE
Home
Logout
Manhole Testing
User: raja
Select Image
Choose File
No file chosen
Manhole Identifier


MANHOLE
Home
Logout
Manhole Testing
Test Image


MANHOLE
Home
Logout
Manhole Testing
Segmentation


MANHOLE
Home
Logout
Manhole Testing
Predicted Result
Manhole Opened
Complaint and Location has sent to Municipality
Manhole Identifier


MANHOLE
Home
Complaints
Logout
Manhole Detected
User: raja on 2023-03-28 20:24:52
Manhole: Broken
Location: View Map
Status: Solved
User: raja on 2023-03-28 20:08:45
Manhole: Open
Location: View Map
Click to Solved
User: raja on 2023-03-28 20:14:27
Manhole: Open
Location: View Map



Kainerbale
Meetmardu
Pettavaithalai
Gunaseelam Sri Prasanna Venkatachalapathy
Manachanallur
Samayapuram
Marudhur
Pelamur
Puvalur
Katur
Divya Desam Arulmigu Vadivazhagiya Nambi
Tropical Butterfly conservatory Park
For development purposes only
Nangavaram
Jambukeswarar
Mekkudy
Kannika Sarai
Ayyar Malai Samy
Inungur
Teppakulam
Uchi Pillaiyar Temple
Toura Panchayat Village
Kallanai Dam
Divya Desam 006 Sri Appakudathan
Nachalur
Motegadu
Shri Vayalur Murugan Temple
Adavathur
Nanchikurichi
Poliyor
Veppangudi
Erumbeeswarar Temple
Kuddapar
Ithalpur
Kallai
Pirattiyur
Thayanur
Tiruchirappalli
திருச்சிராப்பள்ளி
Thiruverumbur
Arulmigu Sri Thirunedunkalanathar
R.T.Malai
Chukamaal
Vadikemen
Kavalkaranpatti
Tiruchirappalli International Airport
Kilakurichy
Madan Pura
Navalpattu
Thuvakudi
Vendayampatti
Keyboard shortcuts
Map data ©2023
Terms of Use
Report a map error

# CHAPTER 11

## CONCLUSION

In conclusion, the development and implementation of the Manhole Predictor Web App represent a significant advancement in infrastructure maintenance practices. The proposed system of the project utilizing advanced deep learning algorithms such as Convolutional Neural Networks (CNN) for manhole classification, Region Proposal Network (RPN) for segmentation, and You Only Look Once version 8 (YOLOv8) for accurate prediction and localization, marks a significant advancement in infrastructure maintenance practices. The project has demonstrated the potential to revolutionize the inspection and maintenance of manhole covers in urban environments. The accuracy and efficiency of the prediction model showcased in the project's results underscore its effectiveness in accurately classifying manhole covers and predicting their conditions. By providing municipalities and maintenance authorities with timely and precise information about the state of manhole covers, the web app enables proactive maintenance measures, ultimately enhancing urban safety and minimizing the risk of accidents. Furthermore, the user-friendly interface and seamless communication features of the web app facilitate efficient collaboration between users and municipal officers, streamlining the maintenance process and optimizing resource allocation. Thus this project represents a valuable tool for enhancing infrastructure management practices, contributing to safer urban environments, and laying the foundation for future advancements in smart city initiatives. With ongoing refinement and widespread adoption, the project holds the potential to significantly improve the maintenance and safety standards of urban infrastructure systems globally.

## FUTURE ENHANCEMENT

There are several areas for future improvement and expansion for the Manhole Predictor web app:

- **Automatic manhole repair scheduling:** The app currently provides information about the location and type of manhole defects to the municipality officer. In the future, this information could be used to automatically schedule repairs based on factors like severity of the defect and priority of the location.
- **Mobile app version:** A mobile app version of the Manhole Predictor could be developed to make it easier for citizens to report manhole defects. The app could use the device camera to capture images of manholes and automatically send them to the municipality officer for processing.
- **Integration with GIS data:** The app could be integrated with geographic information system (GIS) data to provide more detailed information about manholes in a particular area. This could include information about the age of the manhole, its size, and any previous repairs that have been done.

# CHAPTER 12

# REFERENCES


1. Yin, X., Li, X., Gao, Q., & Li, H. (2019). Manhole defect detection algorithm based on convolutional neural network. Journal of Intelligent & Fuzzy Systems, 36(6), 6071-6079.
2. Naini, H. K., Ahmadi, M., & Moini, A. (2017). Manhole detection and tracking for urban road mapping using an unmanned aerial vehicle. Measurement, 104, 50-58.
3. Bhat, S. S., & Sundaram, S. (2018). Manhole cover detection and classification for intelligent vehicles using deep learning. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1814-1819). IEEE.
4. Kataria, K., & Patel, M. (2021). Manhole Defect Detection Using Computer Vision and Image Processing. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 516-519). IEEE.
5. Kim, J. H., Lee, Y. H., Kim, J. H., & Oh, H. (2019). Automatic detection of manhole covers using a deep convolutional neural network. Journal of Computing in Civil Engineering, 33(1), 04018048.
6. Ding, J., Liu, X., & Wang, J. (2020). Urban Manhole Cover Detection Based on Improved YOLOv3. Journal of Physics: Conference Series, 1693(1), 012012.
7. Muralidharan, V., & Balasubramanian, R. (2020). Smart manhole covers monitoring system using IoT. In 2020 IEEE International Conference on Innovative Research and Development (ICIRD) (pp. 104-107). IEEE.
8. Zhang, S., & Jiang, H. (2021). Urban Manhole Cover Detection Algorithm Based on Deep Learning. In Proceedings of the 2021 International Conference on Computer Science and Big Data Applications (pp. 67-71).

9. Sahu, S., & Sahu, N. (2021). Detection of Manhole Defects Using Deep Learning. In International Conference on Intelligent Communication and Computational Techniques (pp. 609-616). Springer, Singapore.
10. Alzahrani, A., Alhajri, M., & Alshammari, M. (2020). A smart manhole monitoring system based on IoT and deep learning. In 2020 IEEE International Conference on Electro Information Technology (EIT) (pp. 564-568). IEEE.
11. Chandran, S., Gopakumar, R., & Paul, V. (2019). Detection of manhole cover using machine learning. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS) (pp. 1195-1200). IEEE.
12. Zhou, B., Gao, X., Wang, J., & Yan, Y. (2019). Detection of manhole cover based on YOLOv3 network. In 2019 3rd International Conference on Control Science and Systems Engineering (ICCSSE) (pp. 68-71). IEEE.
13. Kim, J., Jung, S., Kang, S., & Park, J. (2019). An Automatic Detection Method of Manhole Covers from Image Data using Deep Learning. Journal of Institute of Control, Robotics and Systems, 25(11), 1024-1029.
14. G. J. Jiji, B. R. Santhi, and T. G. Palanivelu, "Automatic Manhole Detection and Defect Identification Using Machine Learning Algorithms," Journal of Testing and Evaluation, vol. 47, no. 1, pp. 1–12, 2018.
15. M. B. J. R. Kumar, S. S. S. Kumar, M. Rajesh, and N. R. Raajan, "Real Time Manhole Defect Detection System using Raspberry Pi," in 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2018, pp. 1–5.
16. A. Farid and A. T. Khalifa, "Defect Detection and Classification for Manhole Covers and Rings," Journal of Engineering, vol. 2015, pp. 1–10, 2015.

17. Classification using Image Processing Techniques," in 2020 IEEE International Conference on Power, Electronics and IoT (PEIOT), 2020, pp. 1–6.
18. A. M. Al-Qershi, T. M. E. Othman, and A. M. A. Bakar, "Defect detection of manhole covers using image processing," in 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI), 2017, pp. 1–6.
19. S. S. Senthilkumar, V. P. Arun, K. R. K. Lalithaa, and P. D. Ananthi Jebaseeli, "Automated Manhole Defect Detection using Image Processing and SVM," in 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp. 514–519.
20. S. S. Senthilkumar, V. P. Arun, and K. R. K. Lalithaa, "Automated Manhole Detection and Defect Classification using Image Processing and Machine Learning Techniques," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1–7.
21. S. Sharma and S. Bhattacharya, "Automated Manhole Defect Detection and Localization using Deep Learning," in 2019 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, pp. 726–731.
22. P. Mittal and V. Jain, "Defect Detection and Localization in Manhole Covers using Image Processing Techniques," in 2021 IEEE 8th International Conference on Signal Processing and Integrated Networks (SPIN), 2021, pp. 317–321.
23. J. J. Pacheco, D. Salazar, and S. S. Fonseca, "Detection of Defective Manhole Covers using Convolutional Neural Networks," in 2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2018, pp. 1–6.