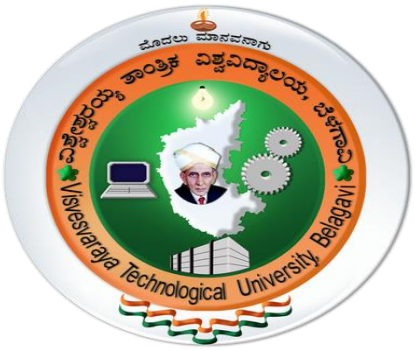
Visvesvaraya Technological University

Belgaum, Karnataka-590 018



A Project Report on

"Detection of Potholes Using Convolutional Neural Network"

Project Report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
In
Electrical & Electronics Engineering

Submitted by
APURVA SHIVAKUMAR (1CR16EE013)
ARJUN S.J (1CR16EE014)
KUMAR UTKARSH NIHAL(1CR16EE037)
MOHAMMED OMAR(1CR16EE046)

Under the Guidance of
Dr. S ZAHID NABI DAR, Assistant Professor
Department of Electrical & Electronics Engineering
CMR Institute of Technology

CMR Institute of Technology, Bengaluru-560 037

Department of Electrical & Electronics Engineering

2019-2020

CMR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

AECS Layout, Bengaluru-560 037

Certificate

Certified that the project work entitled **"Detection of Potholes Using Convolutional Neural Network**" carried out by APURVA SHIVAKUMAR (1CR16EE013), ARJUN S.J (1CR16EE014), KUMAR UTKARSH NIHAL(1CR16EE037), MOHAMMED OMAR(1CR16EE046)are bonafied students of CMR Institute of Technology, Bengaluru, in partial fulfillment for the award of Bachelor of Engineering in Electrical & Electronics Engineering of the Visvesvaraya Technological University, Belgaum, during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide	*Signature of the HOD*	*Signature of the Principal*
----------	----------	----------
Dr.S Zahid Nabi Dar,Ass. Professor Department CMRIT, Bengaluru	Dr. K. Chitra Professor & HOD EEE Department CMRIT, Bengaluru	Dr. Sanjay Jain Principal, CMRIT, Bengaluru

External Viva

Name of the Examiners Signature & Date

1.

2.

CMR INSTITUTE OF TECHNOLOY

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

AECS Layout, Bengaluru-560 037

DECLARATION

We, [APURVA SHIVAKUMAR (1CR16EE013), ARJUN S.J (1CR16EE014), KUMAR UTKARSH NIHAL(1CR16EE037), MOHAMMED OMAR(1CR16EE046)**],** hereby declare that the report entitled **"Detection of Potholes Using Convolutional Neural Network" has** been carried out by us under the guidance of Dr. S Zahid Nabi Dar, Ass. Professor, Department of Electrical & Electronics Engineering, CMR Institute of Technology, Bengaluru, in partial fulfillment of the requirement for the degree of **BACHELOR OF ENGINEERING in ELECTRICAL & ELECTRONICS ENGINEERING**, of Visveswaraya Technological University, Belgaum during the academic year 2019-20.The work done in this report is original and it has not been submitted for any other degree in any university.

Place: Bengaluru

Date: 26/05/2020

APURVA SHIVAKUMAR(1CR16EE013)

ARJUN S.J (1CR16EE014)

KUMAR UTKARSHNIHAL(1CR16EE037)

MOHAMMED OMAR(1CR16EE046)

Abstract

Poor road conditions like cracks and potholes can cause inconvenience to passengers, damage to vehicles, and accidents. Detecting those obstacles has become relevant due to the rise of the autonomous vehicle. Although previous studies used various sensors and applied different image processing techniques, performance is still significantly lacking, especially when compared to the tremendous leaps in performance with computer vision and deep learning.

Roads are considered to be the main mode of transportation. But due to this heavy use of roads and environmental factors, these roads need a scheduled maintenance. Often this maintenance is not performed since it is not possible to monitor each and every place or simply because of ignorance. This leads to the formation of potholes which causes unwanted traffics and the majority of accidents. This paper discusses about the detection of potholes using camera installed on light poles of roads. Image processing techniques have been used which informs the BMC officials in a timely manner using email system, thus keeping manual labor to the minimum. For testing its performance, the proposed system has been implemented under a Windows environment using OpenCV library. Simple image processing techniques like canny edge and contour detection with hough transform is used for effective pothole detection.

Research on damage detection of road surfaces using image processing techniques has been actively conducted, achieving considerably high detection accuracies. Many studies only focus on the detection of the presence or absence of damage. However, in a real-world scenario, when the road managers from a governing body need to repair such damage, they need to clearly understand the type of damage in order to take effective action. In addition, in many of these previous studies, the researchers acquire their own data using different methods.

This paper presents a pothole detection using image processing and Convolutional Neural Network algorithm for detecting multiple potholes on asphalt road surface.

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people, who are responsible for the completion of the project and who made it possible, because success is outcome of hard work and perseverance, but steadfast of all is encouraging guidance. So, with gratitude we acknowledge all those whose guidance and encouragement served us to motivate towards the success of the project work.

We take great pleasure in expressing our sincere thanks to ***Dr. Sanjay Jain****,* ***Principal, CMR Institute of Technology, Bengaluru*** *for providing an excellent academic environment in the college and for his continuous motivation towards a dynamic career.*

We would like to convey our sincere gratitude to ***Dr. K Chitra****,* ***Head of Electrical and Electronics Engineering Department, CMR Institute of Technology, Bengaluru*** *for her invaluable guidance and encouragement and for providing good facilities to carry out this project work.*

We would like to express our deep sense of gratitude to ***Dr. S Zahid Nabi Dar, Assistant Professor, Electrical and Electronics Engineering, CMR Institute of Technology, Bengaluru*** *for his/her exemplary guidance, valuable suggestions, expert advice and encouragement to pursue this project work.*

We are thankful to all the faculties and laboratory staffs of ***Electrical and Electronics Engineering Department, CMR Institute of Technology, Bengaluru*** *for helping us in all possible manners during the entire period.*

Finally, we acknowledge the people who mean a lot to us, our parents, for their inspiration, unconditional love, support, and faith for carrying out this work to the finishing line. We want to give special thanks to all our friends who went through hard times together, cheered us on, helped us a lot, and celebrated each accomplishment. Lastly, to the ***Almighty****, for showering His Blessings and to many more, whom we didn't mention here.*

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent years, the development of automotive industries has resulted in the mass deployment of automobiles on roads in both developing and developed countries. Although the quality of automobiles nowadays has been improved enormously, there are still low-cost models that operate on roads of developing countries.

The reason is because expensive models are only attracted by the top earners in a region, while the midrange citizens can only afford low-cost automobiles as a mean of comfort transportation. In order to support automobiles in reducing accidents by hitting potholes on roads, this research aims to developed an improved real-time watershed-based pothole detection algorithm. The algorithm supports fully automated detection of various types of pothole shapes and sizes on different road surfaces such as smooth, aged, and degraded ones.

The quality of road infrastructure is crucial to people who drive. In some areas, drivers need to be cautious because potholes have been proven to cause catastrophes, especially during the rainy season.

Detecting potholes would allow vehicles to issue warnings so drivers can slow down and avoid them (or the vehicle itself can adjust settings to avoid them), minimize the impact, and make the ride smooth. Pothole detection is about sensing the road ahead of an autonomous vehicle. Nonetheless, studies and research on road-surface damage are still relatively few. Several of them (if not all) use traditional methods, with sensors and expensive equipment to label images in a classification task, but not to detect damage coordinates. Recently, object detection using end-to-end deep learning has been reported to outperform traditional methods. Costly sensors, battery life, computation power, and the complexity of data integration have been reduced by simply relying on imagery input to detect objects. In this study, we train and evaluate object detection with You Only Look Once version 2 (YOLOv2) that has a state-of-the-art convolutional neural network (CNN) at its core.

India is one of the most populous country, roads are the main mode of transportation in this developing country. But due to the heavy use of roads, there is a high amount of wear and tear carried out. Since these roads cannot sustain itself for a long time, a timely maintenance

is expected to be carried out in order to prevent the formation of potholes. The manner in which a pothole is formed depends on the type of bituminous pavement
surfacing. The heavy traffic on the road is the primary reasons for the fatiguing of the road surface, resulting in the formation of the crack. These depressions collect water and allow the water to mix with the asphalt. When vehicles drive through such holes the water is expelled along with some of the asphalt, and this slowly creates a cavity underneath the crack. If a regular road maintenance is neglected, the road surface will eventually collapse into the cavity, resulting in a visibly huge pothole over the surface. In order to repair these roads in a timely manner, it is necessary that the entity knows which area is affected by the pothole or decaying road section is located and an automated process could assist with this.

All these reasons demand that it is important to collect information of the road conditions and through a series of processing and analyzing the obtained information, appropriate conclusions are derived which in turn, warn the officials of the respective area. The simplest and highest accuracy approach to this is to click and upload pictures to the interface provided by the official, but this involves a strong participation from the users as well as manual image analysis. Thus, an automated detection sounds more promising in this case.

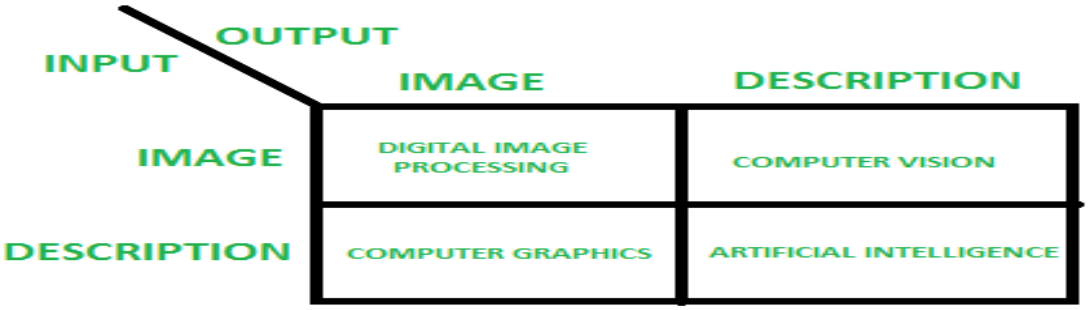


Figure 1.

Our approach includes the use of a computer vision-based system. In this system, a two-dimensional image of roads is used. The digital images are captured by the camera and are processed to capture the information related to road anomalies. In a 2D image-based approach, the system extracts the texture measure based on the histogram as the features of the image region, and neural network was built up to identify a potential region is a pothole or not.

CHAPTER 2

LITERATURE SURVEY

Dewiani Djamaluddin, Andani Achmad, Rivanto Parung, Prototype of vehicles potholes detection-based blob detection method, Journal of Theoretical and Applied Information Technology, 15th June 2017. Vol.95. No 11.

Potholes on road causes majority of traffic incidents. It happens with driver's unawareness to potholes on the road, while potholes location data mining is still performed manually by technician nowadays. This issue drives these following researches and prototyping to be performed. This paper presents a prototype of a pothole detection system using image processing. The system will detect the potholes by assuming potholes as a blob, using blob detection. Live feed from a camera module will be processed by the algorithm that is embedded to a microcontroller simultaneously to determine the potholes. Every detected pothole and its location will be recorded and displayed on a webpage. The prototype will be an embedded system Linux-based running on a Raspberry Pi, featured with camera and a computer program utilizing OpenCV. Several blob parameters such as shape, including circularity, convexity, and inertia value is utilized to optimize the pothole detection. The prototype is tested on actual road with different pothole shapes and size, and the result has 82.5% accuracy.

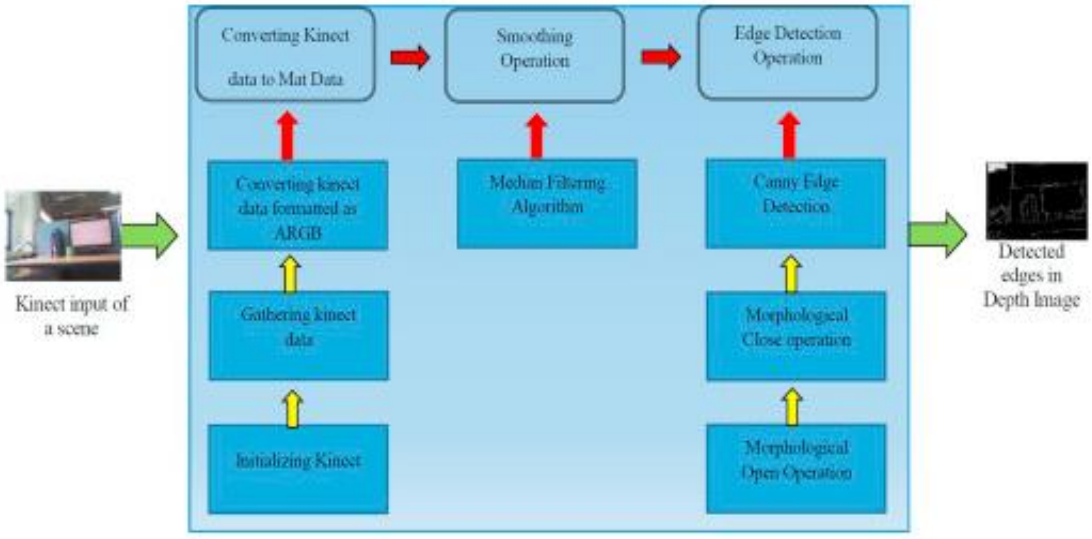


Figure 1.1

Road surface inspection is primarily based on visual observations by humans and quantitative analysis using expensive machines. Among these, the visual inspection approach not only requires experienced road managers, but also is time consuming and expensive. Furthermore, visual inspection tends to be inconsistent and unsustainable, which increases the risk associated with aging road infrastructure. Considering these issues, municipalities lacking the required resources do not conduct infrastructure inspections appropriately and frequently, increasing the risk posed by deteriorating structures.

In contrast, quantitative determination based on large-scale inspection, such as using a mobile measurement system (MMS) (KOKUSAI KOGYO CO., 2016) or laser-scanning method (Yu and Salari, 2011) is also widely conducted. An MMS obtains highly accurate geospatial information using a moving vehicle; this system comprises a global positioning system (GPS) unit, an internal measurement unit, digital measurable images, a digital camera, a laser scanner, and an omnidirectional video recorder. Though quantitative inspection is highly accurate, it is considerably expensive to conduct such comprehensive inspections especially for small municipalities that lack the required financial resources.

Therefore, considering the abovementioned issues, several attempts have been made to develop a method for analyzing road properties by using a combination of recordings by in-vehicle cameras and image processing technology to more efficiently inspect a road surface.

For example, a previous study proposed an automated asphalt pavement crack detection method using image processing techniques and a naive Bayes-based machine-learning approach (Chun et al., 2015). In addition, a pothole-detection system using a commercial black-box camera has been previously proposed (Jo and Ryu, 2015). In recent times, it has become possible to quite accurately analyze the damage to road surfaces using deep neural networks (Zhang et al., 2016.

Taehyeong Kim and Seung-Ki [1] Ryu proposed a detection system which starts with noise removal, followed by adjustment of brightness and simplification of video by binarization. Then, noise removal is applied to the binarized image. After noise removal, the process of extraction of the outlines of the segmented objects is carried out. Extraction is followed by

selection and square zoning for the objects. After all these processes, desired pothole area information is returned.

Sudarshan Rode [2] proposed a pothole detection system which is divided into three subsystems. First is sensing subsystem which senses the potholes encountered by it, by using accelerometer or by camera which scans the road. Both are mounted on the car. Then communication subsystem which transfers the information between Wi-Fi access point and mobile node. Access Point broadcasts the data about potholes in its area. Eriksson et al. [3] studied mobile sensing of roads to monitor and report any potholes. The system used accelerometer and GPS for detection and location respectively. Cars give detections which are fed to a central server.

[Eriksson et al. 2008] 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 (pp. 29-39). ACM. (2019, June).

This paper investigates an application of mobile sensing: detecting and reporting the surface conditions of roads. We describe a system and associated algorithms to monitor this important civil infrastructure using a collection of sensor-equipped vehicles. This system, which we call the Pothole Patrol (P2), uses the inherent mobility of the participating vehicles, opportunistically gathering data from vibration and GPS sensors, and processing the data to assess road surface conditions. We have deployed P2 on 7 taxis running in the Boston area. Using a simple machine-learning approach, we show that we are able to identify potholes and other severe road surface anomalies from accelerometer data. Via careful selection of training data and signal features, we have been able to build a detector that misidentifies good road segments as having potholes less than 0.2% of the time. We evaluate our system on data from thousands of kilometers of taxi drives, and show that it can successfully detect a number of real potholes in and around the Boston area. After clustering to further reduce spurious detections, manual inspection of reported potholes shows that over 90% contain road anomalies in need of repair.

Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals. In computer science AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

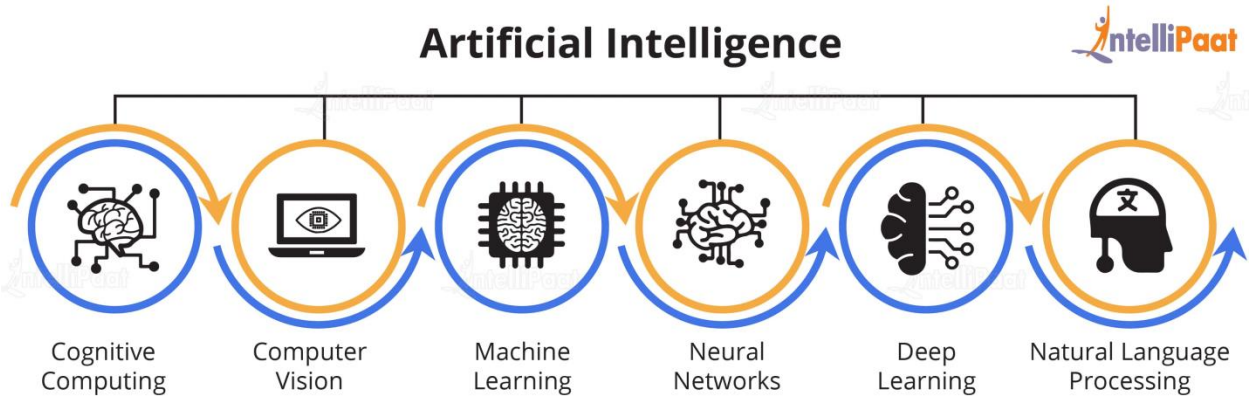


Figure 1.2

Several of them (if not all) use traditional methods, with sensors and expensive equipment to label images in a classification task, but not to detect damage coordinates. Recently, object detection using end-to-end deep learning has been reported to outperform traditional methods. Costly sensors, battery life, computation power, and the complexity of data integration have been reduced by simply relying on imagery input to detect objects. In this study, we train and evaluate object detection with You Only Look Once version 2 (YOLOv2) that has a state-of-the-art convolutional neural network (CNN) at its core.

In addition, we develop a new architecture based on YOLOv2 but integrated with two different models. To the best of our knowledge, since there is no public dataset that Journal of Universal Computer Science, vol. 24, no. 9 (2018), 1244-1257submitted: 30/12/17, accepted: 30/6/18, appeared: 28/9/18 © J.UCS presents the real nature of potholes in roads, we created learning data of pothole images in a wide variety of weather conditions and illumination levels.

CHAPTER 3

PROBLEM STATEMENT

Our approach includes the use of a computer vision-based system. In this system, a two-dimensional image of roads is used. The digital images are captured by the camera and are processed to capture the information related to road anomalies. In a 2D image-based approach, the system extracts the texture measure based on the histogram as the features of the image region, and the Convolutional Neural Network was built up to identify a potential region is a pothole or not.

3.1. Existing System

Existing System uses the data mining concept in the pothole detection. Even though the detection accuracy with the approach can be high compared to the SVM approach, the accuracy of the system depends on the sample training images and this grows with the various types of potholes detectable by the system in several environments like wet, dry and shady. There are various application of pothole detection including protecting and improving safety for drivers and passengers, identifying and locating potholes for road repair and maintenance.

Disadvantages

1. Huge data set is required.
2. Support vector machine is to be trained to more extent to identify pothole.
3. Less in performance
4. More processing is required.
5. Less accuracy

3.2. Proposed System

Proposed system works with 2 phases one with image processing and convolutional neural network techniques. An open-source library of image processing called OpenCV is utilized as the framework for the image processing development. OpenCV is a library which is

designed for a computational efficiency for image processing and manipulation with many algorithms. for object detection, methods that apply an image classifier to an object detection task have become mainstream; these methods entail varying the size and position of the object in the test image, and then using the classifier to identify the object. deep learning has become best known for its ability to learn from experience, and is used in complex problems. Noticeably, deep convolutional neural networks (CNNs) have made tremendous progress in large-scale pothole recognition and in detection

Advantages

1. Less data set is required as neural network model learns the with different dataset values.
2. Using CNN is easy to identity pothole patterns.
3. More in performance
4. Faster in processing
5. More accuracy than data mining process.

CHAPTER 4

SYSTEM ANALYSIS

The systems development life cycle (SDLC) describes a set of steps that produces a new computer information system. The SDLC is a problem-solving process. Each step in the process delineates a number of activities. Performing these activities in the order prescribed by the SDLC will bring about a solution to the business situation. The SDLC process consists of the following phases:

1. Preliminary investigation—the problem is defined and investigated.
2. Requirements definition—the specifics of the current system as well as the requirements of the proposed new system are studied and defined.
3. Systems design—a general design is developed with the purpose of planning for the construction of the new system.
4. Systems development—the new system is created.
5. System installation—the current operation is converted to run on the new system.
6. Systems evaluation and monitoring—the newly operational system is evaluated and monitored for the purpose of enhancing its performance and adding value to its functions.
7. Looping back from a later phase to an earlier one may occur if the need arises.

4.1 Feasibility Study

The probability of the try is eviscerated in this stage of business proposal is progressive with an exceptionally wide strategy for the errand and some cost estimations. Amidst structure inspection the achievability examination of the proposed structure is done. Subsequently it guarantees that the proposed framework to any of the affiliation. For judgment skills study examination, a couple understands the veritable essentials for the structure is imperative. There are three sorts of feasibility study they are

4.1.1 Economical Feasibility

Monetary examination is done to survey the change cost weighed against a conclusive pay or focal points got from the made structure. The structure made and acquainted will be extraordinary point of preference with the affiliation. The made structure is worked in the existed hardware and programming establishment.

4.1.2 Technical Feasibility

Structure is simply critical just in case it can be changed into gen systems that will encounter the affiliation's particular essential. This test of believability is done to ask whether the structure will work or not when made and presented, any genuine impediments to use. Regarding these issues in particular examination there are a couple centers to focus on:

- Changes to procure the structure: Changes in the system are in positive heading, there should be better customer organization and extended level of profitability.
- Required capacities: Platforms and gadgets used as a piece of this errand are for the most part used.
- Acceptability: The model of the scheme is kept adequately achievable. So that there should not be any issue from the customer's viewpoint.

4.1.3 Operational Feasibility

- Question which emerge here are:
- If system is produced and actualized, will that can be utilized?
- Is there any of the adequate backing for this task by the stage on which it is created?
- Test situations where the system may fizzle and strategies to conquest?

CHAPTER 5

TOOLS USED

5.1. NetBeans IDE Frame Work for JAVA

In computer programming, a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user – written code, thus providing application specific software. A software framework is a universal, reusable software platform used to develop applications, products and solutions.

A software framework includes support programs, compilers, code libraries, tool sets and application programming interface that bring together all different components to enable development of a project or solutions.

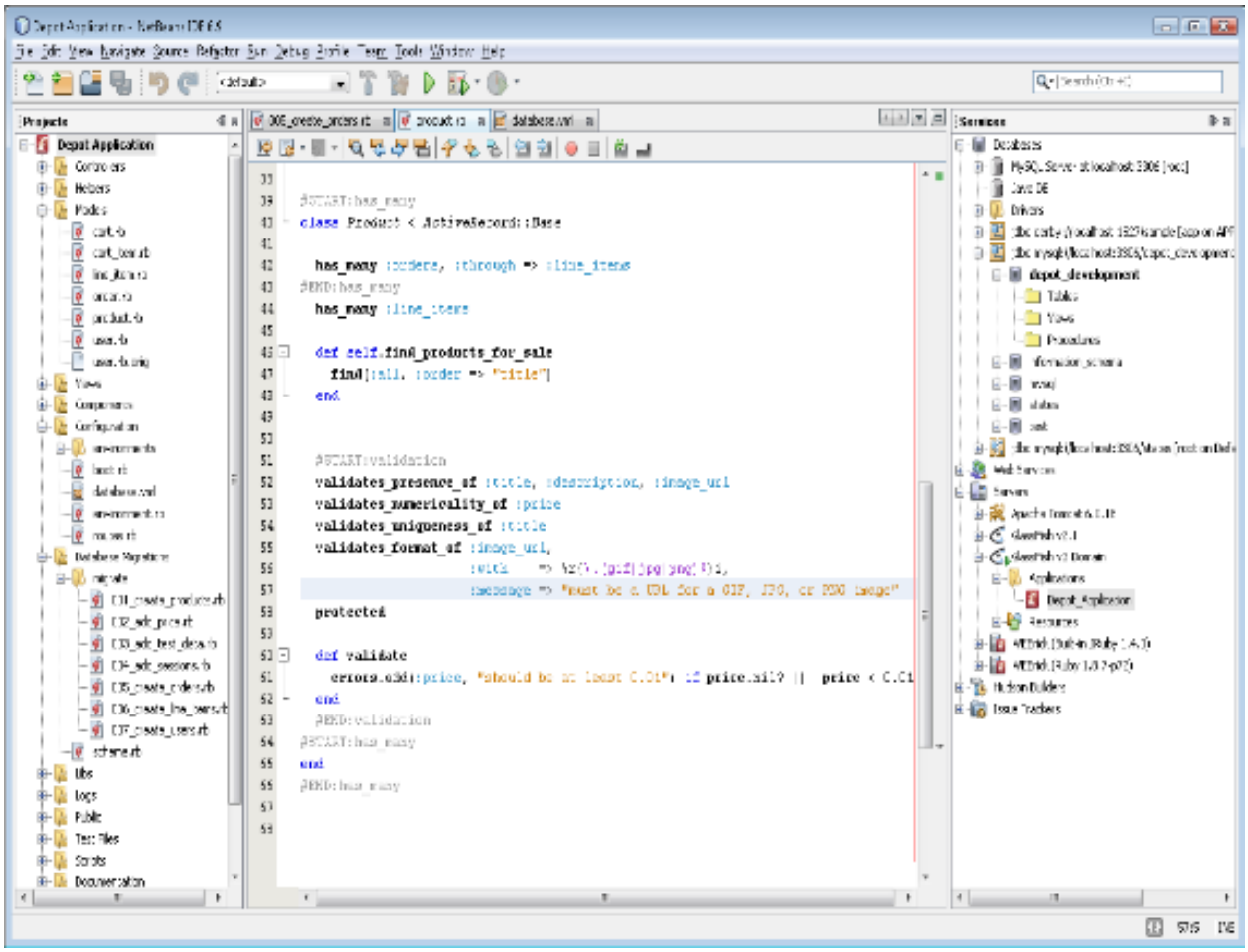


Figure 1.3

5.2. Selection of Language

Usage stage ought to superbly outline plan report in a reasonable programming dialect keeping in mind the end goal to accomplish the fundamental last and right item. The JAVA programming dialect is utilized for usage reason. The propelled java helps more in making web applications. The Servlets and JSPs (java server pages) are utilized to build up our venture.

5.2.1 JAVA

Java technology is high level programming language, can be characterized by following

- **Simple** based on OOP standard and designed in such way that people can learn easily if they recognize the basic concept of C++.
- **Object Oriented** In everything is treated as object and it can be easily protracted since it is founded on the object model.
- **Distributed** process of making a facility or a product that is available for use of the customer and business users. By direct means of work or indirect means of work with the help of mediators. The three parts of marketing mixes are product, pricing.
- **Multithreaded** this feature is conceivable to write plans that can-do multiple tasks concurrently. This plan allows programmers to hypothesis smoothly running cooperative system.
- **Dynamic** is intended to adjust to creating environment. Java programs convey more measure of runtime confirmation that can be utilized to check and resolve gets to objects on run time.
- **Architecture Neutral** compiler generates construction impartial object code which makes the byte code to be executable on many platforms, with attendance of java run time system.
- **Portable** operation dependent aspects of the requirement make java portable.
- **High Performance** use of Just in Compilers increases high performance.

- **Robust:** Java removes error disposed to situations by highlighting mainly on compile time error and run time checking.
- **Secure:** Secure feature allows to develop virus free systems.

5.2.2 Java Platform Discussion

A Java stage is things or environment in which a framework runs. Without a doubt the most surely understood stages are Windows 2000, Linux, Solaris and Mac OS. The Java stages are showed as a mixture at work framework and hardware. It is not the same as other stage is that it is simply programming organize that continues running on top of other gear-based stages.

- **The Java has modules:**
  - The Java-Virtual-Machine (JVM)
  - The Java-Application-Programming-Interface (JAPI)

5.3. Coding Standards

5.3.1. Naming Convention

Enlightening names are one of the finest implements for making effectively justifiable code. Dodge theoretical names that are prone to be reused by different parts of the framework.

5.3.2. Function Naming

Functions names must recognize the activity performed or the data gave by the capacity. Steps are taken to see that the names will for the most part start with a verb.

5.3.3. Variable Naming

The sort and motivation behind every variable ought to be obvious the code is utilized. E.g. per user will anticipate that username will be no less than 6 characters. Names are shaped from delimited by lower case letters, composite words, with underscores permitted.

Variable names are given long, and important. Variables are named with the substance of the variable. All variables contain lower case letters for e.g., which is the cluster comprising of the landing values.

The sort and purpose behind each variable is evident inside the code in which it is used.

Per user will foresee that username will be no under 6 characters.

- Variable names are given long and imperative.
- Variables are termed with the substance of the variable.
- It contains lower case letters, which is the group involving the passage values.

5.4 System Maintenances

The bolster time of the item series is the time programming thing performs profitable work. Framework is successful completed it must be kept up in a suitable way.

The necessity for framework upkeep is to take off flexible to the changes in the framework environment. Upkeep can be depicted by taking after doings after reenactment writing computer programs is released for use.

- In a framework if any botches happen it can be represented to the framework manager or customer.
- Quick change in framework is an irrefutable desiring in the planned framework so the diversion programming needs to give programming interface to the customer.
- The item is used, references for new limits, alteration to existing limits, general overhauls are gotten from the customers. To gratify requests from this federation, perfective backing is achieved.

CHAPTER 6

REQUIREMENT SPECIFICATION

6.1. General Explanation

A System Requirement Specification-(SRS) is done delineation of the performance programming to deliver and arranged motivation to add to that item. To achieve decided essentials the originators, require less time and effort and the change cost is minimized. A fair SRS describes how the application will associate with system hardware and customers in real circumstances.

The normal understanding in the middle of clients and engineers is caught in provisions report. The SRS examines about the item not about the undertaking used to build up that. SRS gives base in improvement of the created item.

We have identified five potential classifications of users of our system:

- Software Developers: These are the general population that take the model created by the originators and apparatus it in code. They may likewise utilize the framework to recognize the outline of a current framework keeping in mind the end goal to look after it.
- Quality Engineers: These clients are typically liable for guaranteeing that an outline is achievable and/or dependable. They will thusly likewise should see at the yield from our framework.
- Software Designers: These are the clearest clients. They will utilize the framework as a method for setting out the outline of a so far unimplemented framework.
- The Client: More frequently than not, programming is intended for a customer. The customer may wish to see the outline as laid out by the framework and have the capacity to see what precisely they are purchasing.
- System Administrator: Due to the customer/server/simultaneous nature of the framework, somebody should be liable for security and protection of the framework.

This is the System Administrators part. The Administrator of the framework, venture or model need not be a partner of the any of alternate parts distinguished here.

User

can be assumed to have the following characteristics:

- Ability to recite and comprehend English.
- Familiarity with basic Graphical User Interface (GUI) workings of the Windows operating system.
- Further facility with computer knowledge can be assumed.

- **Software Designer**

can be assumed to have the following characteristics:

- A thorough knowledge of object-oriented plan notation.
- Familiarity with shared design surroundings
- Software Developers User Structures.

Quality Engineer

can be assumed to have the following characteristics:

- A detailed knowledge of object-oriented plan representation.

Administrator

will not be mandatory to have any specific features.

Client

Understanding of modelling techniques. (Won't essentially have to use the scheme, look at its output).

6.2 Specific Requirements

❖ Hardware Requirement Specification:

- Hardware Dual Core
- Hard Disk 50 GB
- Speed 1.4 GHz
- RAM 1GB
- Key Board Standard Keyboard
- Touch Pad Button Mouse
- Monitor LED

❖ Software Requirement Specification:

- Operating System : Windows
- IDE : Net Beans 7.3.1
- Technology : Java
- Java Version : JDK1.7

6.3 Functional Requirements

Utilitarian Requirement portrays a segment of an item scheme and structure must portable when given detailed inputs and conditions.

6.4 Non-Functional Requirements

- **Usability:** The connection is accommodated every edifice. The client has the volume view and makes sections in the structures. Acceptances are given in every document to stay away from conflicting or invalid passage in the application/system.
- **Availability:** System will be accessible all the time with the exclusion of the time needed for the strengthening of information.
- **Portability:** The application is produced in Java. It is compact to other working system gave JDK is available to the working system.
- **Integrity:** The task work is fundamentally outlined in an incorporated development environment, where every class, individuals, traits is planned under java bundle. Building and examining the principle class will incorporate every one of the classes in like manner for the best possible assemblage of the undertaking work.
- **Extensibility:** The venture work is similarly open for any future alteration.

CHAPTER 7

System Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

7.1. INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.
-

7.2. OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user

will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow

7.3. OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.

- Trigger an action. Confirm an action.

System Architecture

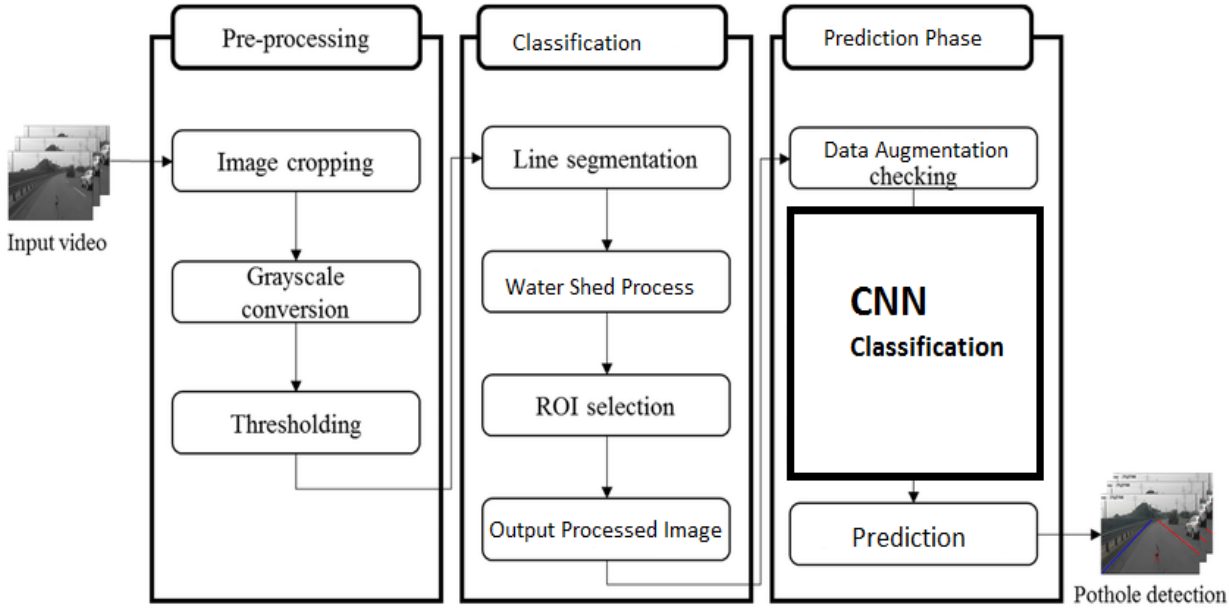


Figure 1.4

Data Flow

DFD Image Processing

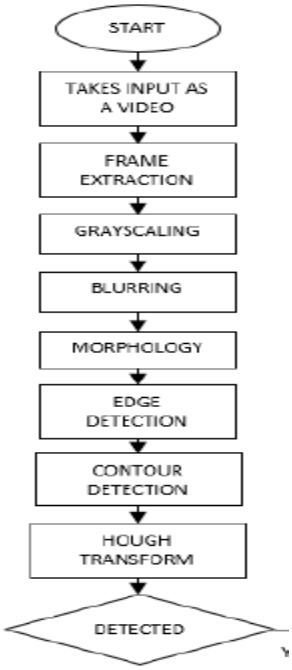


Figure 1.5

Data Flow diagram of CNN

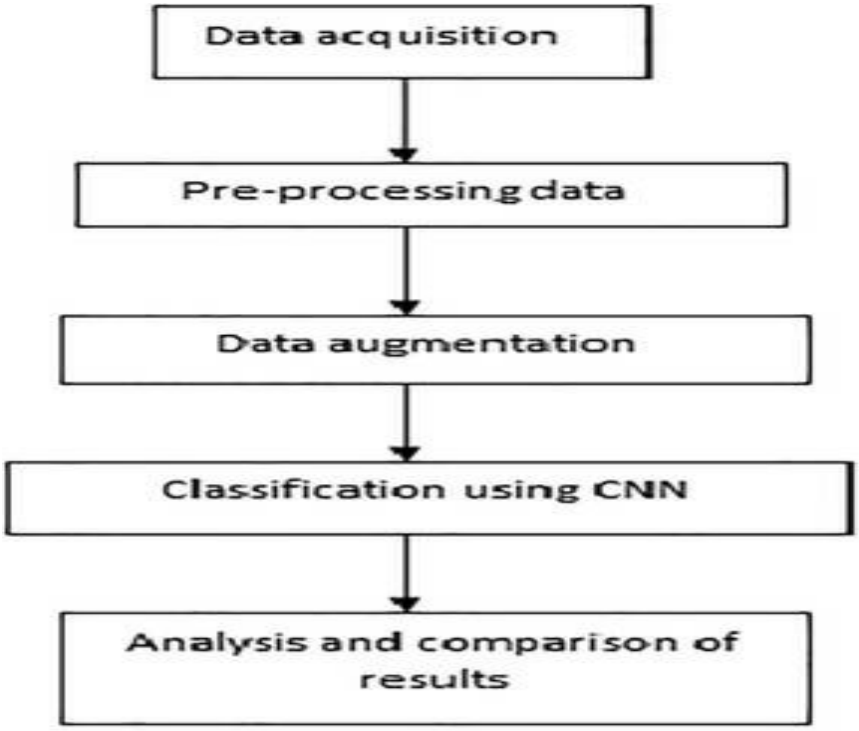


Figure 1.6

7.4. Implementation

Implementation consist of two stages image processing and Neural Network process that is preprocess, classification stage and Prediction phase. Input is provided as image and output is predicted with pothole detected or not detected results.

Image input phase

Once an input image is given to the application, it checks and read the the image in order to process next stage.

7.5. Bilateral Filtering phase

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

The bilateral filter is defined as[1][2]

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|),$$

and normalization term, W_p, is defined as

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

where

I^{filtered} is the filtered image;

I is the original input image to be filtered;

x are the coordinates of the current pixel to be filtered;

Ω is the window centered in x, so $x_i \in \Omega$ is another pixel;

f_r is the range kernel for smoothing differences in intensities (this function can be a Gaussian function);

g_s is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function).

The weight W_p is assigned using the spatial closeness (using the spatial kernel g_s) and the intensity difference (using the range kernel f_r).[2] Consider a pixel located at (i, j) that needs to be denoised image using its neighbouring pixels and one of its neighbouring pixels is located at (k, l). Then, assuming the range and spatial kernels to be Gaussian kernels, the weight assigned for pixel (k, l) to denoise the pixel (i, j) is given by

$$w(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|I(i, j) - I(k, l)\|^2}{2\sigma_r^2}\right),$$

where σ_d and σ_r are smoothing parameters, and $I(i, j)$ and $I(k, l)$ are the intensity of pixels (i, j) and (k, l) respectively.

After calculating the weights, normalize them:

$$I_D(i, j) = \frac{\sum_{k,l} I(k, l) w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)},$$

where I_D is the denoised intensity of pixel (i, j).

Figure 1.7

7.6. Canny Edge Detection

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges

5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

1. K = 1, set the iteration n and the coefficient of the amplitude of the edge h.
2. Calculate the gradient value $G_x(x,y)$ and $G_y(x,y)$
3. Calculate the weight according to the formula below:

$$d(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}$$

$$w(x,y) = \exp\left(-\frac{\sqrt{d(x,y)}}{2h^2}\right)$$

4. The definition of the adaptive filter is:

$$f(x,y) = \frac{1}{N}\sum_{i=-1}^{1}\sum_{j=-1}^{1} f(x+i,y+j)w(x+i,y+j)$$

to smooth the image, which

$$N = \sum_{i=-1}^{1}\sum_{j=-1}^{1} w(x+i,y+j)$$

5. When K = n, stop the iterative, otherwise, k = k+1, keep do the second step

Figure 1.8

7.7. Dilation Image Phase

In binary morphology, dilation is a shift-invariant (translation invariant) operator, equivalent to Minkowski addition.

A binary image is viewed in mathematical morphology as a subset of a Euclidean space $\mathbf{R}^d$ or the integer grid $\mathbf{Z}^d$, for some dimension *d*. Let *E* be a Euclidean space or an integer grid, *A* a binary image in *E*, and *B* a structuring element regarded as a subset of $\mathbf{R}^d$.

The dilation of *A* by *B* is defined by

$$A \oplus B = \bigcup_{b \in B} A_b,$$

where A_b is the translation of *A* by *b*.

Dilation is commutative, also given by.

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a.$$

If *B* has a center on the origin, then the dilation of *A* by *B* can be understood as the locus of the points covered by *B* when the center of *B* moves inside *A*. The dilation of a square of size 10, centered at the origin, by a disk of radius 2, also centered at the origin, is a square of side 14, with rounded corners, centered at the origin. The radius of the rounded corners is 2.

7.8. Water Shed Segmentation Algorithm

The watershed is a classical algorithm used for **segmentation**, that is, for separating different objects in an image.

Starting from user-defined markers, the watershed algorithm treats pixels values as a local topography (elevation). The algorithm floods basins from the markers until basins attributed to different markers meet on watershed lines. In many cases, markers are chosen as local minima of the image, from which basins are flooded.

In the example below, two overlapping circles are to be separated. To do so, one computes an image that is the distance to the background. The maxima of this distance (i.e., the minima of the opposite of the distance) are chosen as markers and the flooding of basins from such markers separates the two circles along a watershed line.

To simulate this process, the algorithms proposed by Beucher and Meyer are used the most often, and the first of them consists of the following steps [35]:

A1 We find the local minima in the image. Each of them is assigned with a unique marker.

A2 We simulate a flooding process that uses a priority queue that consists of H queues, where H is the number of possible image element values (for example, H = 256 if the gray level varies from 0–255). When it is filled, the elements of the image with the value of h are pushed into the corresponding queue with the number h.

(a) We fill the priority queue with the elements of image local minima.

(b) We scan the priority queue in sequence from smaller values of h to larger ones. Then, we select the element from the first non-empty queue. If all queues from the priority queue are empty, the algorithm terminates.

(c) We remove the selected element from the queue, and its marker propagates on all unmarked neighbors.

(d) We place all the neighbors marked in the previous step into the priority queue. Then, we proceed to Step A2.

The result will not contain the watershed lines in this case. If they are required, the second approach is applied [39]:

B1 Similar to Step A1. B2 A similar priority queue is used, as described in Step A2. Furthermore, each element of the image needs to be marked additionally, depending on whether it was already placed into this queue or not (an additional marker).

(a) We mark elements, that already have a unique marker, with an additional marker.

(b) Into the priority queue, we add elements that have marked neighbors. We also mark them with an additional marker.

(c) Similar to Step A2.

(d) We remove the selected element from the queue. If all its marked neighbors were marked with the same marker, we mark the element itself by this marker. If the selected element was a neighbor of elements with different markers, we mark it with a special marker, meaning that it is an element of the watershed.

(e) Neighbors of this element that have not been marked with an additional marker yet are placed into the priority queue. Then, we proceed to Step B2c

7.9. Convolutional Neural Network

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.[1][2] They have applications in image and video recognition, recommender systems,[3] image classification, medical image analysis, natural language processing,[4] and financial time series.

CNNs are regularized versions of multilayer perceptron's. Multilayer perceptron's usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

Convolutional

When programming a CNN, the input is a tensor with shape (number of images) x (image width) x (image height) x (image depth). Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map width) x (feature map height) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- Convolutional kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameter).
- The depth of the Convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

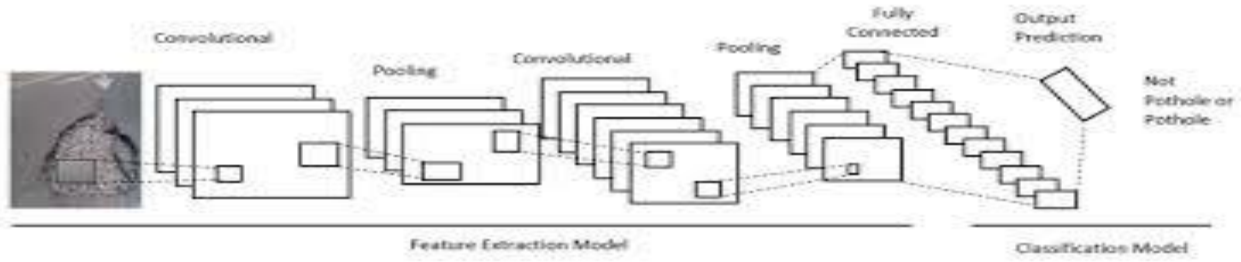


Figure 1.9

7.9.1. Convolutional layers

convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for *each* neuron in the second layer. The convolution operation

brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using backpropagation.

7.9.2. Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer.In addition, pooling may compute a max or an average. *Max pooling* uses the maximum value from each of a cluster of neurons at the prior layer. *Average pooling* uses the average value from each of a cluster of neurons at the prior layer.

7.9.3. Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

7.9.4. Receptive field

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from *every* element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically, the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its *receptive field*. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

7.9.5. Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights.

The vector of weights and the bias are called *filters* and represent particular features of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces memory footprint because a single bias and a single vector of weights are used across all receptive fields sharing that filter, as opposed to each receptive field having its own bias and vector weighting.

CHAPTER 8
TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.1. TYPES OF TESTS

8.1.1. Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.
This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2. Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of

components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing.

8.1.4. System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.1.5. White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.1.6. Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of

tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot ―see‖ into it. The test provides inputs and responds to outputs without considering how the software works.

8.2. TEST CASES

Name of the test	Image Processing for Binary Filtering
Test Description	Binary Filter Processing for removing noise in the image
Sample Input	Grayscale Image
Expected Output	Filtered Image
Actual result/Remarks	Same as expected
Passed (?)	Pass

Name of the test	Canny Edge Detection
Test Description	Gaussian Filter to remove weak and unwanted edges
Sample Input	Filtered Image
Expected Output	Filtered Image with strong edges
Actual result/Remarks	Same as expected
Passed (?)	Pass

Name of the test	Dilation Image Phase
Test Description	Expansion of Image to to extract the region shape
Sample Input	Filtered Image with strong edges

Expected Output	Image with expanded pixels
Actual result/Remarks	Same as expected
Passed (?)	Pass

Name of the test	Segmentation
Test Description	Watershed Algorithm for segmentation of the image
Sample Input	Image with expanded pixels
Expected Output	Segmented Image (9*9 pixels)
Actual result/Remarks	Same as expected
Passed (?)	Pass

Name of the test	CNN Classification
Test Description	Predicting whether the segmented image is pothole or not
Sample Input	Segmented Image
Expected Output	Pothole (Yes/No)?
Actual result/Remarks	Same as expected
Passed (?)	Pass

8.3. Integration Testing of System

Integration testing is needed to test urged programming sections to grasp whether they truly continue running as one structure. Exchange off particularly away to uncover the matters that climb up the mix of pieces. Event driven and it more worried with the critical aftereffect of screens or fields. This test show that paying little personality that the pieces were solely satisfaction, as displayed by tastefully unit testing.

The mix of sections is correct and solid. Programming testing is the testing of two or more made programming parts on a single stage to go on disillusionments brought on by crossing point misshapenness. Attempt of testing is to watch that parts or programming application e.g. pieces in a thing framework or programming application at the association level to interface without lurch.

Mix testing is deliberate to test joined programming portions to make sagacity of whether they continue running as one framework. Combination testing is especially away to uncover the issues that rise up out of the mix of sections.

Testing is more stressed with the basic consequence of screens fields. Combination tests display that in spite of the way that the sections were solely gratification, as showed up by viably unit testing. Programming blend testing is the coordination testing of two or more joined programming sections on a lone stage to make frustrations realized by interface disfigurements.

The task that compromise test is to watch that parts or programming application, e.g. fragments in an item framework or programming application at the association level to interface without bungle.

CHAPTER 9
CONCLUSION AND FUTURE SCOPE

Potholes have nothing but negative effects and hence it must be eradicated as soon as possible. The current system includes the use of manual detection by people who are willing to contribute for the betterment of the road. Thus, it is important that manual labor approach is kept to a minimum and switched to an automatic approach instead.

The system will be installed in a fixed position on the light poles which ensures less handling. Also, this system keeps a track of the negligence and delay. The system makes use of Raspberry Pi, which has a low cost and high compatibility with other interfaces, we also make use of 2D vision-based approach, this makes our system more affordable.

The system also detects potholes in time without damaging the cars for potholes detection. Thus,
making the system more feasible and favorable.

We presented a deep-learning architecture designed to detect potholes. The model shrinks the number of parameters by a large margin, with a notable increase in performance. We showed that a deep CNN can indeed be applied to detecting potholes with promising results in accuracy and speed, even considering the much smaller size of real-world, labeled pothole data. We believe that with more training data, the results will substantially improve beyond those reported here.