IOT BASED SMART STICK FOR VISUALLY IMPAIRED PEOPLE

A MINI PROJECT REPORT

Submitted by

HARI VISHNU SK **(211420205303)**

TAMIL SELVAN S **(211420205164)**

SHARATH BABU R **(211420205142)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE, POONAMALLEE

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **" IOT BASED SMART STICK FOR VISUALLY IMPAIRED PEOPLE"** is the bonafide work of **"HARI VISHNU.S.K (211420205303),TAMIL SELVAN.S (211420205164), SHARATH BABU.R (211420205142)"** who carried out the project under my supervision.

SIGNATURE

Dr. M. HELDA MERCY M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Department of Information Technology

Panimalar Engineering College

Poonamallee, Chennai - 600 123

SIGNATURE

Dr.D.KARUNKUZHALI M.Tech., Ph.D

PROFESSOR

Department of Information Technology

Panimalar Engineering College

Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on _____

DECLARATION

I hereby declare that the project report entitled **" IOT BASED SMART STICK FOR VISUALLY IMPAIRED PEOPLE** " which is being submitted in partial fulfillment of the requirement of the course leading to the award of the 'Bachelor of Technology in Information Technology' in **Panimalar Engineering College, An Autonomous Institution Affiliated to Anna University- Chennai** is the result of the project carried out by me under the guidance and supervision of **Dr.D.KARUNKUZHALI, M.TECH., Ph.D. Professor in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

(**Hari Vishnu. S.K**)

(Tamil Selvan. S)

Date: **(Sharath babu. R)**

Place**:** Chennai

It is certified that this project has been prepared and submitted under my guidance.

Date**:** **Dr.D.KARUNKUZHALI, M.TECH.,Ph.D.**

Place: Chennai (Professor/ IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co- operation and support from many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Honorable Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs. C. VIJAYA RAJESHWARI and Dr.C.SAKTHI KUMAR, M.E.,Ph.D** and **Dr.SARANYA SREE SAKTHIKUMAR., B.E., M.B.A.,Ph.D** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our faculty in charge, **Dr.D.KARUNKUZHALI, M.Tech.,Ph.D** Associate Professor, Department of Information Technology for her guidance throughout the course of our project. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

ABSTRACT

Nowadays, the innovations are trying to make the visually challenged people can be able to travel anywhere without anyone's help. In our proposed system, it helps people to find and recognize the objects free paths in both indoor and outdoor places. The major advantage of this system is that it provides the detail information about images in the path. The information detected by the system is transmitted to the user and auditory communication methods. In this existing system, this device which can be acts as to detect the obstacles and steps by a white cane device. The proposed model will be able to detect the obstacles having a height below knee-level. It transmits the alert information to the user by two facilities such as vibration and audio alert. Vibration strength and audio track depends on distance and position of the obstacles that senses the person. It will be low cost and as a low power embedded device for blind people and identification.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Independence is an important methodology in achieving objectives, dreams, and goals in life. Visually impair/blind persons find themselves demanding the hazardous paths to go out separately. There are millions of visually impair or blind people in this world who are for all time needing help from others. For many years the normal walking stick becomes a well-known characteristic to blind person's navigation and later hard work has been made to recover the walking stick by adding remote sensors. Blind people has big troubles when they walk on the street or stairs using a typical walking stick, but they have sharp hepatic sensitivity.

The electronic walking stick will help the blind person by if a more efficient and convenient means of life. Moving during an unknown environment becomes a real challenge for the blind or impaired people.

Persons who go out of the house with the white stick frequently use well-known routes and difficulties with new ones. Moreover, many people simply afraid of being helpless in constant movement of people, vehicle and other road users. It is therefore advisable to offer new solutions for the problems with existing technologies. This paper proposes the design and develops a transportable stick for blind people/impair people for suitable use and navigation in open and private places. The most important drawbacks of these aids are the necessary skills and training period, range of motion and small information that they convey. There has been a creation of a lot of Electronic Travel Aids and they have been implemented to help the blind navigate separately and safely. Many high-level technological solutions have been introduced in recent times to help blind persons navigate independently.

However, in the assessment of other technologies, many blind guidance systems use ultrasound because it is not prone to environmental noise. The major reason why ultrasonic is well-liked is that is knowledge is comparatively less classy, and also ultrasound emitters and detectors are light sufficient to be used without the need for complex circuitry. Apart from starting the old direction-finding aids, a blind aid system can add a new measurement of Real-space help and vision along with dedicated obstacle detection circuitry.Thus a system that can facilitate aid to the blind is the need of

the hour. The System will distinguish the obstacle and help the blind people to know the obstacle before them. Which are already planned to the system according to situation and condition to make blind people easy and efficient to move from source to desire destination?. The major goal of ASR research is to allow a computer to differentiate in real-time, with 100% exactness, all words that are reasonably spoken by any person, independent of terminology size, noise, speaker individuality or accent. Nowadays, if the system is educated to learn an individual speaker's voice, then much larger vocabularies are probable and accuracy cannot be lesser than 90%.

Smart walking stick is particularly planned to detect obstacles which may help the blind to navigate care-free. The audio communication will keep the user alert and considerably reduce accidents. without the help of the others. The main aim of the system is to provide the information for blind persons which gives a sense of vision by as long as the information about their environment and objects around them.

1.1 MOTIVATION

The blind people's life behaviour is greatly limited by the loss of eyesight.They can only walk-in set routes that are significant in their lives, with blind direction-finding tools and accumulate memories in their long-term examination. This situation has resulted in many difficulty encounter in a trip of the blind incorporated walking on the road and looking for usual life-arena. Several devices have been urbanized for mobility and direction-finding help of the blind and are characteristically known as travel aids or blind and are characteristically known as travel aids or blind mobility aids. So that the proposed system produced.

1.2 OBJECTIVE

- To simply the corporeal movement of visually impaired persons.
- To put back the conservative walking cane with a smart walking stick that identify the objects.
- To make the blind being safe while they are walking.
- To design a smart stick for a blind being using the ultrasonic antenna at distance range 2cm to 4cm using proteus simulation.

CHAPTER 2

LITERATURE SURVEY

2.1 “Safe local navigation for visually impaired users with time of flight and haptic feedback device”

R.K.Katzschmann,March 2018

This paper presents ALVU (Array of Lidars and Vibrotactile Units), a contactless, perceptive, hands-free, and prudent wearable device that allows visually impaired users to detect low- and high-hanging obstacles, as well as physical confines in their immediate environment. The solution allows for safe local navigation in both limited and open spaces by enabling the user to discriminate free space from obstacles. The device accessible is collected of two parts: a sensor belt and a haptic strap. The sensor belt is an array of time-of-flight distance sensors worn around the front of a user's waist, and the pulses of infrared light provide dependable and exact measurements of the distances among the user and surrounding obstacles or surfaces. The haptic strap communicates the considered distances through an array of vibratory motors worn around the user's upper abdomen, providing haptic feedback. The linear shuddering motors are united with a point-loaded pre-tensioned applicator to transmit isolated vibrations to the user. They validated the device's capacity in a general user study entailing 162 trials with 12 blind users. Users exhausting the device successfully walked through hallways, avoided obstacles, and predicted staircases.

2.2 “A haptic solution to assist visually impaired in mobility tasks”

B.Ando, October 2015

Electronic travel aids are used for detecting obstacles, identifying forces, and, usually, obtaining useful information from the environment, thus enabling safe and effective management of the environment. A drawback is perverted codification, which may lead to usability concerns. This paper introduces a haptic device expected to offer the user with information on the occurrence of obstacles inside the environment. The haptic boundary is intended to replicate the stimuli provided by a traditional white cane, without any contact with the environment. A prototype, implemented through a short cane with an embedded smart sensing approach an active handle, is obtainable. Twenty-five blindfolded normally sighted users participated to measure system performance in detecting obstacles

and accurately assigning their position by the haptic interface. With deference to detecting obstacles and their positions, the average values of the sensitivity in the case of left, center, and right positioned obstacles are 0.735, 0.803, and 0.830, while the specificity principles are 0.924, 0.835, and 0.827, respectively.

2.3 "An Electronically Guided Walking Stick for the Blind University Tenaga Nasional, Malaysia"

Mobility for the blind is always a great problem. Just like a sighted, blind also needs to travel around inside a closed premises like house, factory, office, school etc. They may also like to go for shopping, visiting friends and other places of their interest. Presently available electronic travelling aids like sonic path finder, sonic torch etc. are not suitable for using inside a closed premises such as school, factory, office etc. In this paper an electronically guided walking stick that can be used conveniently inside a closed premises has been discussed.

2.4 "Smart Cane: Assistive Cane for Visually-Impaired People"

A.Noraziah, Roslina M. Sidek,

This paper reports on a study that helps visually-impaired people to walk more confidently. The study hypothesizes that a smart cane that alerts visually-impaired people over obstacles in front could help them in walking with less accident. The aim of this paper is to address the development work of a cane that could communicate with the users through voice alert and vibration, which is named Smart Cane. Their development work involves coding and physical installation. A series of tests have been carried out on the smart cane and the results are discussed. This study found that the Smart Cane functions well as intended, in alerting users about the obstacles in front.

CHAPTER 3

PROBLEM DEFINITION

3.1 PROBLEM STATEMENT

Physical movement is a challenge for a visually impaired person. The conservative walking stick used by them very imperfectly in its range of discovery and it is only used to detect the object which is near to the user. The drawback of the conservative cane, however, is its stoppage to detect obstacles outside of its arrival. That is the user has to tap the position of the object to notice the obstacle. The visually challenged people can avoid the thing better if the walking stick can create. Vibration and sound warming when there is an object in the exact range of distance.

3.2 EXISTING SYSTEM

Blind Cane:

1) Recognition of obstacles up to knee level.

2) Does not protect from obstacles at torso and face level.

3) Prone to injuries.

Trained Guide Dogs:

1) 1% usage.

2) Expensive to Train Dogs.

3) Training period on an average 6 months.

4) Difficulty in dog up-keeping costs and lifestyle changes.

Human Guide:

1) Dependency.

2) Feeling of being a burden.

DISADVANTAGES

- Only detect the obstacles and steps.
- There is no alternate path.
- GPS can’t attach in.
- Little sensor support in these fields.
- Limited and fixed route to follow daily routine.

3.3. PROPOSED SYSTEM

In the proposed system, it is able to detect the obstacles having a height below knee-level. It transmitting the alert information to the user by two facilities:

1) Vibration
2) Audio alert

Vibration strength and audio track depends on distance and position of the obstacles that senses the person.

ADVANTAGES

- It can be detect more distant obstacles.
- It is easily to recognize the destination.
- Having feature to left and right turn alarm signal.
- Simple to use and low cost.
- It can detect the digs and water present in the ground.
- Device can be converted easily.

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.

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

4.1.1 Technical Feasibility:

A study of resource availability that may affect the ability to achieve an acceptable system. Technical feasibility is the most difficult area to ensure at initial stage. Since the objectives, functions, performance cannot be predicted to its fullest, everything seems possible, provided the right assumptions are made.

4.1.2 Operational Feasibility:

It is essential that the process of analysis and definition can be conducted in parallel with an assessment of technical feasibility. The consideration that is normally associated with technical feasibility includes resource availability at the organization where the project is to be developed and implemented.

4.1.3 Economic Feasibility:

An evaluation of development cost weighted against the ultimate income or benefit derived from the proposed system. The proposed system will not cause any expenditure since all the requirements are available in the firm.

CHAPTER 5

SYSTEM REQUIREMENTS AND SPECIFICATION

5.1 HARDWARE REQUIREMENTS:

The hardware used for the development and installation of the project is:

Processor	:	Intel Dual Core
Hard Disk	:	160 GB
RAM	:	2 GB
IO Devices	:	Basic Keyboard and Mouse
Acquisition Device	:	Camera
Sound Output	:	Speakers

5.2 SOFTWARE REQUIREMENTS:

The software used for the development of the project is:

Operating System : WINDOWS XP / 7 / 8

Package : VISUAL STUDIO 2008 / 2010

Language : C#

Framework : .NET 3.5 / 4

Front End : C#.Net

Back End : Training Dataset

5.3 SOFTWARE ENVIRONMENT

5.3.1 C#.NET

Introduction to the C# Language and the .NET Framework

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

Note:

The Visual C# documentation assumes that you have an understanding of basic programming concepts. If you are a complete beginner, you might want to explore Visual C# Express, which is available on the Web. You can also take advantage of books and Web resources about C# to learn practical programming skills.

5.3.2 C# Language:

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it

may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

Encapsulated method signatures called delegates, which enable type-safe event notifications. Properties, which serve as accessors for private member variables. Attributes, which provide declarative metadata about types at run time. Inline XML documentation comments. Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop." Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical.

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events.

5.3.3 .NET Framework Architecture:

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.

5.3.4 C# source code to machine execution

Language interoperability is a key feature of the .NET Framework. Because the IL code produced by the C# compiler conforms to the Common Type Specification (CTS), IL code generated from C# can interact with code that was generated from the .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single assembly may contain multiple modules written in different .NET languages, and the types can reference each other just as if they were written in the same language.

CHAPTER 6

SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

The architecture diagram describes the overall function of our project.

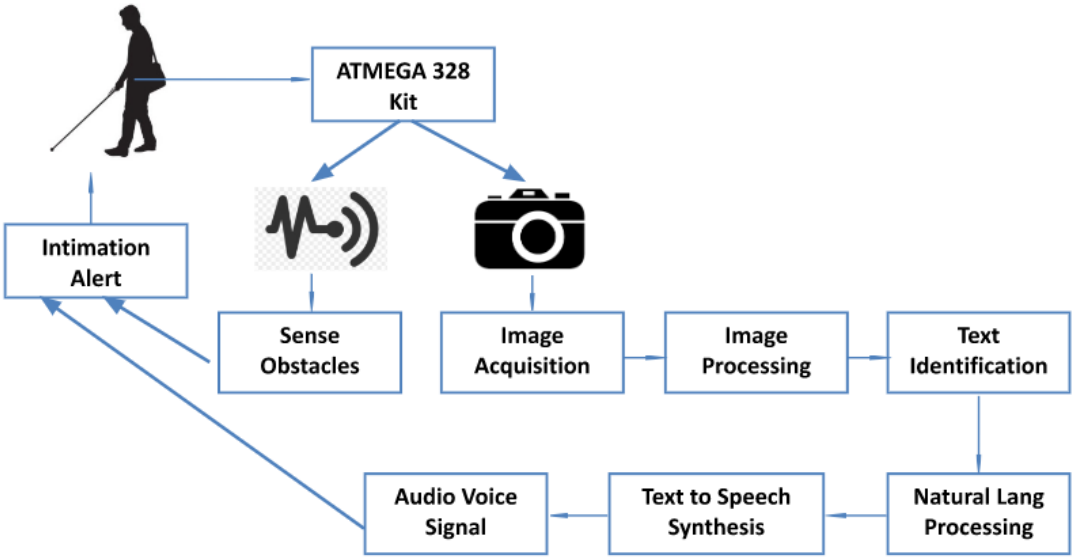


Fig 6.1 System Architecture

6.2 DATA FLOW DIAGRAMS

6.2.1 Data Flow Diagram Level 0

Fig 6.2.1 Data Flow Diagram level 0

6.2.2 Data Flow Diagram Level 1

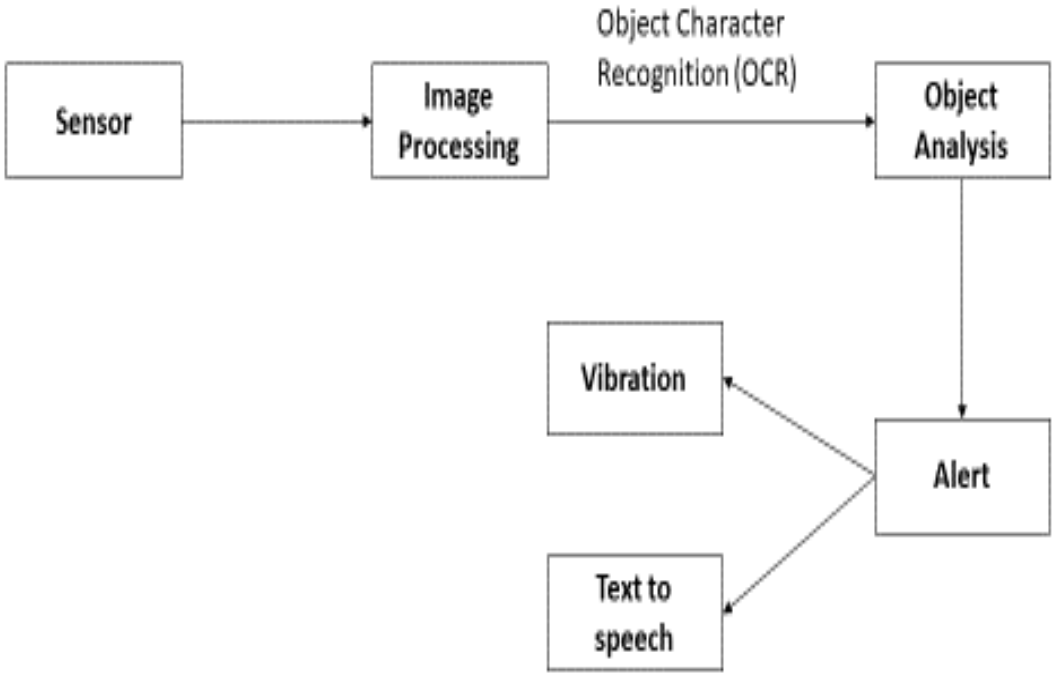


Fig 6.2.2 Data Flow Diagram Level 1

6.2.3 Data Flow Diagram Level 2

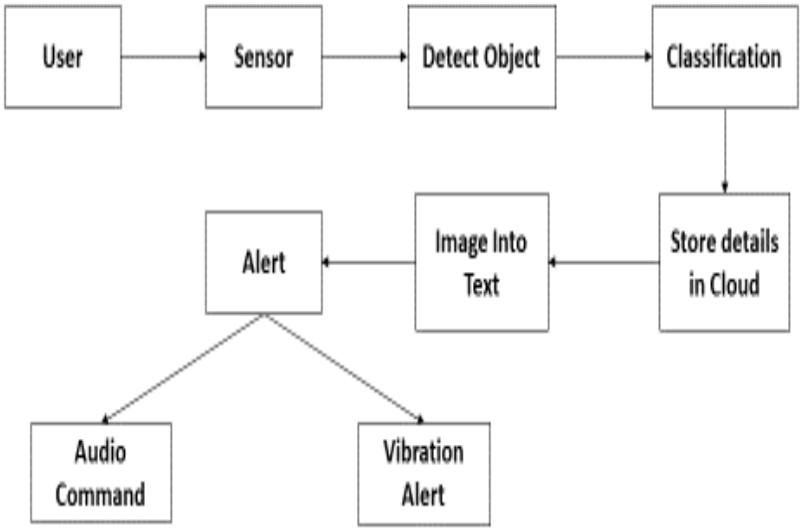


Fig 6.2.3 Data Flow Diagram Level 2

6.3 UML DIAGRAMS

6.3.1 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.

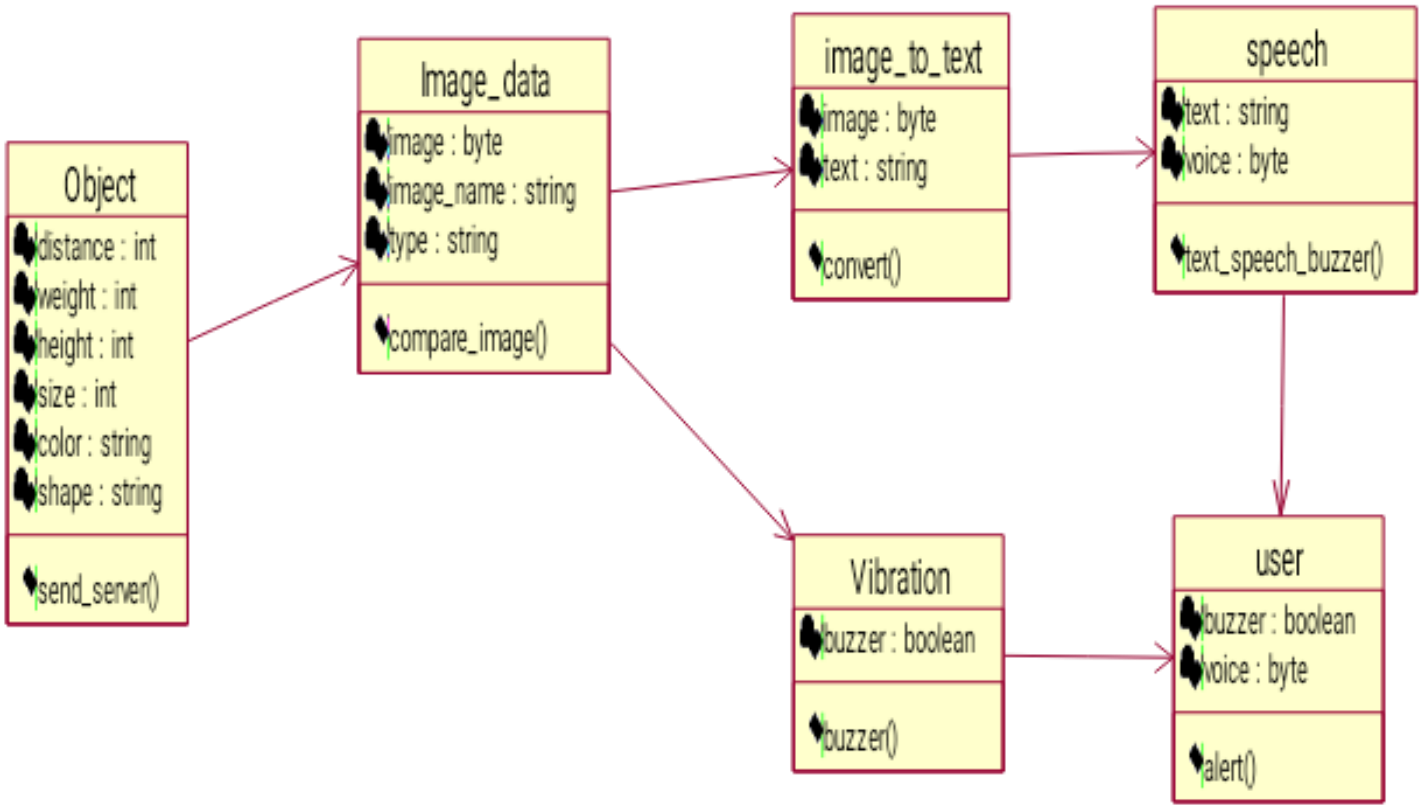


Fig 6.3.2 Class Diagram

6.3.2 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes.

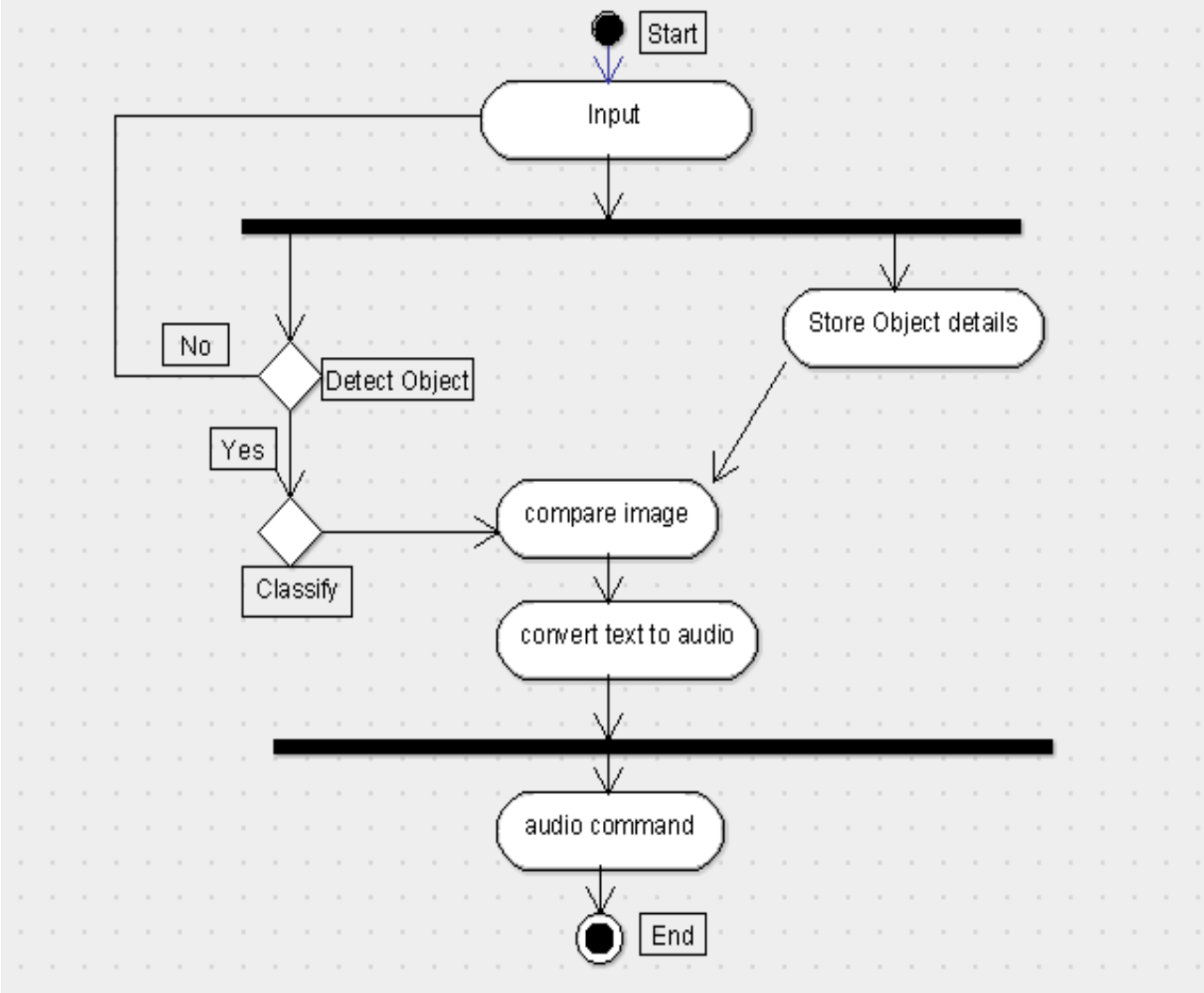


Fig 6.3.2 Activity Diagram

6.3.3 COLLABORATION 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).

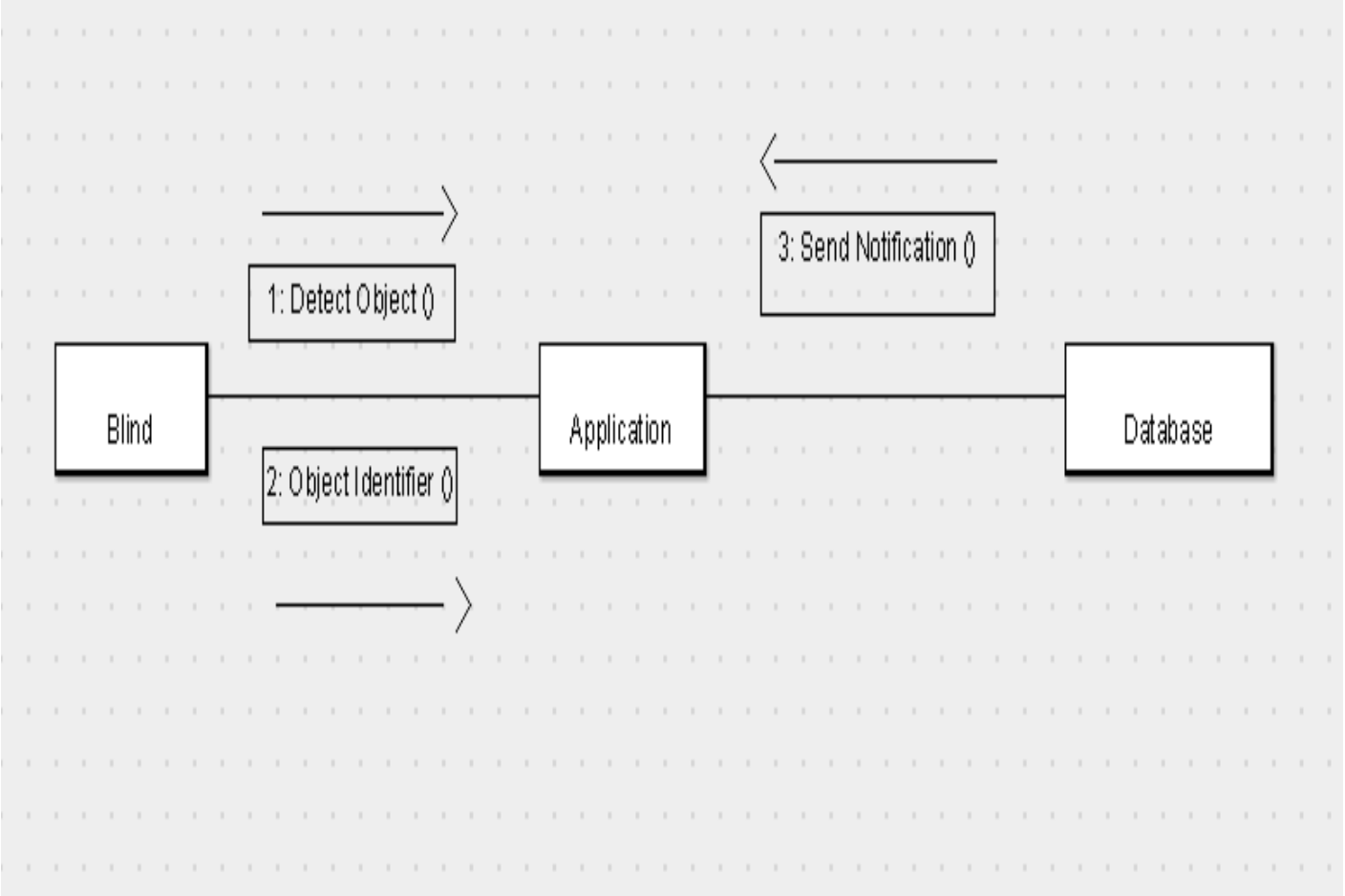


Fig 6.3.3 Collaboration Diagram

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 SYSTEM DESCRIPTION

System implementation is the stage in the project where the theoretical design 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 LIST OF MODULES

- Device Initialization Module
- Image Acquisition and Pre-Processing
- Image Enhancement and Feature Extraction

7.3 MODULES DESCRIPTION

7.3.1 Device Initialization Module

In this module, we initialize the board is equipped with sets of digital and analog input/output (I/O) pins. And using sensors to analyze the signals. A buzzer or beeper is a signaling device, used to alert the user. Sensors measure the distance to the target by measuring the time between the emission and reception.

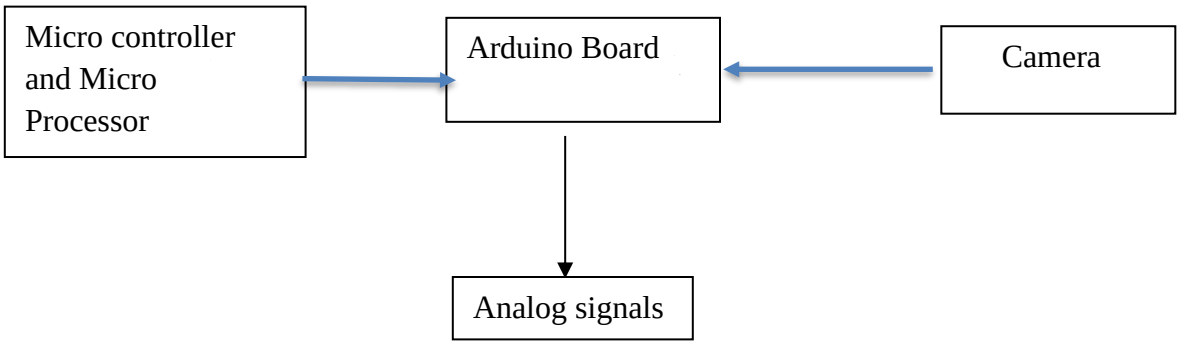


Fig 7.3.1 Device Initialization Module

7.3.2 Image Acquisition and Pre-Processing

It collects the input data (images of obstacles) and the data is pre-processed. Pre-processing involves reading the images, resizing the images, removing noise (De-noise) in the image. The obstacle on the way is detected through sensor, Camera gets triggered to capture the object and send to the cloud. In this module, it is defined as the action of retrieving an image from some source.

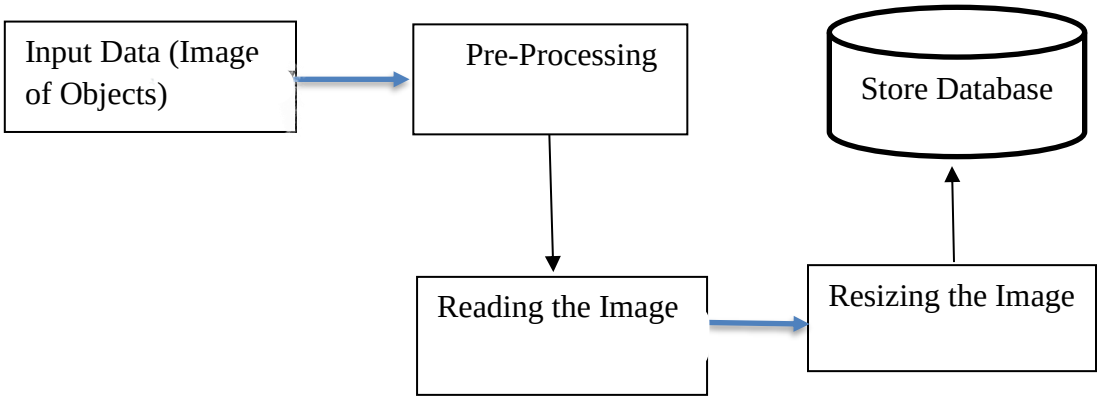


Fig 7.3.2 Image Acquisition and Pre-Processing

7.3.3 Image Enhancement and Feature Extraction

In this module Image enhancement is the process of adjusting images so that the results are more suitable image analysis. It involves removing the noise, sharpen, or brighten an image. Feature extraction is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval.

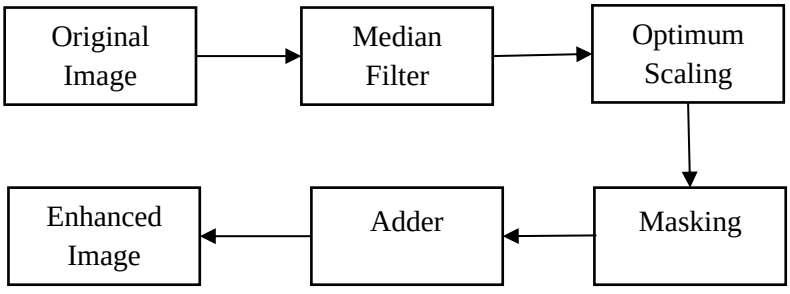


Fig 7.3.3 Image Enhancement and Feature Extraction

CHAPTER 8

SYSTEM TESTING AND MAINTENANCE

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.

8.1 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.

8.2 SYSTEM TESTING

Testing is the analysis of source/executable code and the controlled execution of executable code to reveal defects that compromise a Java program's executable integrity. Defects often lead to erratic behavior or the premature termination of an executing program.

The Software testing process commences once the program is created and the documentation and related data structures are designed. Software testing essential for correcting errors. Otherwise the program or the project is said to be not complete.

Software testing is a process of checking whether the developed system is working according to the original objectives and requirements. The system should be tested experimentally with test data so as to ensure that the system works according to the required specification. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. After the coding phase, computer programs are available that can be executed for testing purposes. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases.

Software Testing Fundamentals

Testing presents an interesting task for software engineers. Earlier in the software process, the engineer attempts to build software from an abstract concept to a tangible implementation. The engineer creates the series of test cases that are intended to "demolish" the software that has been build.

To test any program we need to have a description of its expected behavior and a method of determining whether the observed behavior conforms to the expected behavior for this we need a test -oracle. A test-oracle is a mechanism; different from the program itself that can be used to check the correctness of the output of the program for the test cases. Human-oracle is human beings who mostly compute by hand what the output of the program should be. Human-oracle can make mistake. So test -oracle is defined in the tool to automate testing and avoids mistakes.

The testing objectives are summarized in the following steps:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding an as-yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

Testing principles

All the tests should be traceable to customer requirement. Tests should be planned long before testing begins that is the test planning can bring as soon as the requirement model is complete.

Testing should begin "in the small" and progress towards testing "in the large". The first planned and executed generally focus on individual program modules. As testing progresses, testing shifts focus and attempt to find errors in integrated clusters of modules and ultimately in the entire system.

The number of path permutations for even a moderately sized program is exceptionally large. For this reason, it is possible to execute every combination of paths during testing. It is possible, however, to adequately cover program logic and to ensure that all conditions in the procedural design have been exercised. To be more effective, testing has highest probability of finding errors.

The following are the attributes of the good test:

- A good test has high probability of finding an error.
- A good test is not redundant.
- A good test should be “best of breed”.
- A good test should be neither too simple nor too complex.

TESTING STRATEGIES

System testing is stage of implementation which is aimed at ensuring that the system works accurately and efficient before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

The testing steps are:

- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing
- User Acceptance Testing

LEVELS OF TESTING

- This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client’s requirement. The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.

- By performing acceptance tests on an application, the testing team will reduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system.

- This test is the first stage of testing and will be performed amongst the teams (developer and QA teams). Unit testing, integration testing and system testing when combined together is known as alpha testing. During this phase, the following aspects will be tested in the application.

- This test is performed after alpha testing has been successfully performed. In beta testing, a sample of the intended audience tests the application. Beta testing is also known as **pre-release testing**. Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program a "real-world" test and partly to provide a preview of the next release.

8.3 TESTING PROCESS

8.3.1 Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. Unit testing specifies paths in the module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit.

According to the project, the given java source code is tested for used and unused variables, different java source codes are taken as input and variables are listed within each and every class and tested for their correctness.

The number of loops in the given java source code are checked for the correctness of line numbers. Memory consumed by the Java source code during the compilation and execution of coding is found.

8.3.2 Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets are conducted. The objective is to take unit tested modules and combine them test it as a whole. Thus, in the integration-testing step, all the errors uncovered are corrected for the next testing steps.

8.3.3 Validation Testing

The outputs that come out of the system are as a result of the inputs that go in to the system. So, for the correct and the expected outputs the inputs that go in to the system should be correct and proper. So this testing is done to check if the inputs are correct and they are validated before it goes in to the system for processing.

8.3.4 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2-ways-one is on screen and another is printed format.

8.3.5 System Testing

A system testing does not test the software but rather the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specifications, and system documentation.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

8.3.6 Performance Testing

Performance testing is designed to test the run-time performance of software within the context of an integrated system. It requires both hardware and software instrumentation. It is often necessary to measure resource utilization in an exacting fashion.

8.3.7 Procedure Testing

Determine the clarity of the documentation on operation and the user of the system by having users do exactly what manual request. In case of this project work system testing and unit testing are mainly used.

Before going for testing, first we have to decide the type of testing. For this impact system unit testing is carried out. Before going for testing, the following things are taken into consideration.

- To ensure that information properly flows in and out of the program.
- To find out whether the local data structures maintains its integrity during all steps in an algorithm execution.
- To ensure that the module operates properly at boundaries established to limit or restrict processing.
- To find out whether all statements in the module have been executed at least once.
- To find out whether error-handling paths are working correctly or not.

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system is by using test data errors are again uncovered by using above testing steps and corrections are also noted for future use.

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person using this data as a way to partially test the system. It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all

combinations or formats that can enter the system.

This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Quality assurance consisting of the auditing and reporting functions of management for the system. The goal of quality assurance is to provide management with the data necessary to be informed about the product quality, thereby gaining insight and confidence that product quality is meeting its goals. For the confirmation of quality of the application the functional and performance requirements and characteristics are documented and reviewed a per the management decision.

The risk is an unwanted event that has negative consequences. Project will engage in risk management to understand and control the risk on their projects. We can distinguish risks from other project events by looking for three things.

- A loss associated with the event
- The likelihood that the event will occur
- The degree to which we can change the outcome.

8.4 SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into working system. This is the most crucial stage in achieving a new successful system and in giving confidence of the new system for the users that it will work efficiently.

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operations of the system. In many organizations someone who will not be operating it, will commission the software development project. The people who are not sure that the software is meant to make their job easier. In the initial stage they doubt their software but we have to ensure that the resistance does not build up as:

- The active user must be aware of the benefits of using the system.
- Their confidence in the software is buildup.
- Proper guidance be imparted to the user so that he is comfortable in using the application.

The implementation procedures involves careful planning, investigation of the current system and the constraints on implementation, design of methods to achieve the changeover, an evaluation of change over methods. Initially preliminary implementation plan is prepared to schedule and manage many different activities that must be completed for a successful system implementation.

The preliminary plan serves as a basis for the initial scheduling and assignment of resources to important implementation activities. The preliminary plan has been updated throughout the implementation phase in order to reflect the current state. A complete implementation plan includes the following items: selection of quality personnel, system training plan, system test plan, equipment installation pan, system conversion plan, and overall implementation plan. Apart from planning major task of preparing the implementation procedures are education and training to the users. The more complex the system being implemented, the more involved be the system's analysis and design effort required just for implementation. An implementation coordinating committee based on polices of individual organization is appointed.

The implementation process begins with preparing a plan for implementation of the system. According to this plan the activities have been carried out; discussions have been made regarding the equipment and resources. According to the above plan the necessary equipment has to be acquired to implement the new system.

To achieve the objective and benefits expected from computer based system it is essential for the people who will be involved then in understanding the overall system and its effect on the organization, and in being able to carry out effectively their specific tasks. As system become more complex the need for education and training is more and more important.

Training the user is one of the most important jobs of the developer. User are informed of hoe the whole system works its objectives, new document etc. For this purpose system and user manuals were prepared. In system manuals, details about the system, which were used to develop, were specified. In user manuals, data flow diagram, menu and screen formats are given. The user for the system is shown the screens and they are thought how to operate the system.

8.5 SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which a software product performs useful work. After the system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle.

CHAPTER 9

SAMPLE CODING

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.OleDb;
using System.Speech;
using System.Speech.Synthesis;
using AForge.Imaging;
using CnetSDK.OCR.Trial;
namespace ImgProImpl
{

public partial class Form1 : Form
   {
      OleDbConnection cn = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data
Source=dbase.mdb");
      Canny CannyData;
      OpenFileDialog oDlg;
      SaveFileDialog sDlg;
double zoomFactor = 1.0;
private MenuItem cZoom;
      ImageHandler imageHandler = new ImageHandler();
string fp = "";
public Form1()
      {
InitializeComponent();
fp = "D:\\Image\\";
oDlg = new OpenFileDialog(); // Open Dialog Initialization
         oDlg.RestoreDirectory = true;
         oDlg.InitialDirectory = "D:\\";
         oDlg.FilterIndex = 1;
         oDlg.Filter = "jpg Files (*.jpg)|*.jpg|gif Files (*.gif)|*.gif|png Files (*.png)|*.png |bmp Files
(*.bmp)|*.bmp";

      }
private void button1_Click(object sender, EventArgs e)
```

```
        {
if (DialogResult.OK == oDlg.ShowDialog())
            {
                imageHandler.CurrentBitmap = (Bitmap)Bitmap.FromFile(oDlg.FileName);
                imageHandler.BitmapPath = oDlg.FileName;
this.AutoScroll = true;
                this.AutoScrollMinSize = new
Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width * zoomFactor),
Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
                //menuItemImageInfo.Enabled = true;
                ImageInfo imgInfo = new ImageInfo(imageHandler);
imgInfo.Show();
imageHandler.SaveBitmap(fp + "img1.jpg");
imageHandler.SaveBitmap(fp + "img6.jpg");
                pictureBox1.Image = System.Drawing.Image.FromFile(fp + "img1.jpg");
            }
        }
private void Form1_Load(object sender, EventArgs e)
        {
            panel6.Visible = false;
            panel5.Visible = false;
            panel2.Visible = false;
            panel3.Visible = false;
            panel4.Visible = false;
        }
private void button2_Click(object sender, EventArgs e)
        {
            this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetColorFilter(ImageHandler.ColorFilterTypes.Red);
this.Invalidate();
            this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img2.jpg");
            pictureBox2.Image = System.Drawing.Image.FromFile(fp + "img2.jpg");
imageHandler.ResetBitmap();
            this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
            this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetColorFilter(ImageHandler.ColorFilterTypes.Green);
this.Invalidate();
            this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img3.jpg");
            pictureBox3.Image = System.Drawing.Image.FromFile(fp + "img3.jpg");
imageHandler.ResetBitmap();
```

```
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
        this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetColorFilter(ImageHandler.ColorFilterTypes.Blue);
this.Invalidate();
        this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img4.jpg");
        pictureBox4.Image = System.Drawing.Image.FromFile(fp + "img4.jpg");
imageHandler.ResetBitmap();
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
        this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetGrayscale();
this.Invalidate();
        this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img5.jpg");
        pictureBox5.Image = System.Drawing.Image.FromFile(fp + "img5.jpg");
imageHandler.ResetBitmap();
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
        this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetContrast(20.0);
this.Invalidate();
        this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img8.jpg");
        pictureBox6.Image = System.Drawing.Image.FromFile(fp + "img8.jpg");
imageHandler.ResetBitmap();
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
        panel3.Visible = true;
        pictureBox7.Image = System.Drawing.Image.FromFile(fp + "img6.jpg");
    }
private void button3_Click(object sender, EventArgs e)
    {
if (File.Exists(fp + "img11.jpg"))
        {
File.Delete(fp + "img11.jpg");
        }
pictureBox9.Image.Save(fp + "img11.jpg");
        imageHandler.CurrentBitmap = (Bitmap)Bitmap.FromFile(fp + "img11.jpg");
```

```
        imageHandler.BitmapPath = fp + "img11.jpg";
imageHandler.SetContrast(50.0);
        this.AutoScroll = true;
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();

        panel6.Visible = true;
this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetInvert();
this.Invalidate();
        this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img10.jpg");
        pictureBox12.Image = System.Drawing.Image.FromFile(fp + "img10.jpg");
imageHandler.ResetBitmap();
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
    }
private void button4_Click(object sender, EventArgs e)
    {
float TH, TL, Sigma;
int MaskSize;
        TH = (float)Convert.ToDouble("30.0");
        TL = (float)Convert.ToDouble("10.0");
        MaskSize = Convert.ToInt32("5");
        Sigma = (float)Convert.ToDouble("1.5");
        CannyData = new Canny((Bitmap)pictureBox3.Image, TH, TL, MaskSize, Sigma);
        pictureBox8.Image = CannyData.DisplayImage(CannyData.GNL);
        pictureBox9.Image = CannyData.DisplayImage(CannyData.GNH);
        panel4.Visible = true;
        this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetBrightness(100);
this.Invalidate();
        this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img9.jpg");
imageHandler.ResetBitmap();
        this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
        }
private void button5_Click(object sender, EventArgs e)
    {
        panel6.Visible = false;
```

```
MessageBox.Show("Classification has been done", "Alert", MessageBoxButtons.OK,
MessageBoxIcon.Information);
      }
public void classify1()
      {
string fn = "";
int flag = 0;
         System.Drawing.Bitmap sourceImage = (Bitmap)Bitmap.FromFile(fp + "img9.jpg");

         DirectoryInfo dir = new DirectoryInfo(@"D:\Dataset\Datas");
FileInfo[] fi = dir.GetFiles();
         System.Drawing.Bitmap sourceIma = (Bitmap)Bitmap.FromFile(fp + "img1.jpg");
         System.Drawing.Bitmap clon = sourceIma.Clone(new Rectangle(0, 0, sourceIma.Width,
sourceIma.Height), PixelFormat.Format24bppRgb);
foreach (FileInfo f in fi)
         {
            System.Drawing.Bitmap templat = (Bitmap)Bitmap.FromFile(f.FullName.ToString());
            ExhaustiveTemplateMatching tm = new ExhaustiveTemplateMatching(0.999f);
            Bitmap clone = sourceImage.Clone(new Rectangle(0, 0, sourceImage.Width,
sourceImage.Height), PixelFormat.Format8bppIndexed);
            Bitmap template = templat.Clone(new Rectangle(0, 0, templat.Width, templat.Height),
PixelFormat.Format8bppIndexed);
TemplateMatch[] matchings = tm.ProcessImage(clone, template);
            BitmapData data = clone.LockBits(new Rectangle(0, 0, clone.Width, clone.Height),
ImageLockMode.ReadWrite, clone.PixelFormat);
foreach (TemplateMatch m in matchings)
            {
flag = 0;
Drawing.Rectangle(data, m.Rectangle, Color.White);
               Graphics graphics = Graphics.FromImage((System.Drawing.Image)clon);
               Pen whitePen = new Pen(Color.White, 0);
               OleDbCommand cm = new OleDbCommand("select * from dbtab where fname='" +
f.Name + "'", cn);
cn.Open();
               OleDbDataReader dr = cm.ExecuteReader();
while (dr.Read())
               {
whitePen = new Pen(Color.Red, 2);
int x = m.Rectangle.X + 6;
int y = m.Rectangle.Y + 6;
int n = Convert.ToInt32(dr.GetString(1));
int nm = Convert.ToInt32(dr.GetString(2));
                  Rectangle rect = new Rectangle(x, y, n, nm);
graphics.DrawRectangle(whitePen, rect);
graphics.Dispose();
               textBox2.Text = dr.GetString(3);
               }
```

```
cn.Close();
            }
clone.UnlockBits(data);
         }
          pictureBox11.Image = clon;
          panel5.Visible = true;
      }
public void classify2()
      {
string fn = "";
int flag = 0;
          System.Drawing.Bitmap sourceImage = (Bitmap)Bitmap.FromFile(fp + "img9.jpg");
          DirectoryInfo dir = new DirectoryInfo(@"D:\Dataset\Datas");
FileInfo[] fi = dir.GetFiles();
          System.Drawing.Bitmap sourceIma = (Bitmap)Bitmap.FromFile(fp + "img1.jpg");
          System.Drawing.Bitmap clon = sourceIma.Clone(new Rectangle(0, 0, sourceIma.Width,
sourceIma.Height), PixelFormat.Format24bppRgb);
foreach (FileInfo f in fi)
          {
            System.Drawing.Bitmap templat = (Bitmap)Bitmap.FromFile(f.FullName.ToString());
             ExhaustiveTemplateMatching tm = new ExhaustiveTemplateMatching(0.999f);
             Bitmap clone = sourceImage.Clone(new Rectangle(0, 0, sourceImage.Width,
sourceImage.Height), PixelFormat.Format8bppIndexed);
             Bitmap template = templat.Clone(new Rectangle(0, 0, templat.Width, templat.Height),
PixelFormat.Format8bppIndexed);
TemplateMatch[] matchings = tm.ProcessImage(clone, template);
             BitmapData data = clone.LockBits(new Rectangle(0, 0, clone.Width, clone.Height),
ImageLockMode.ReadWrite, clone.PixelFormat);
foreach (TemplateMatch m in matchings)
             {
Drawing.Rectangle(data, m.Rectangle, Color.White);
                Graphics graphics = Graphics.FromImage((System.Drawing.Image)clon);
                Pen whitePen = new Pen(Color.Red, 2);
if (f.Name.IndexOf('m') >= 0)
                {
int x = m.Rectangle.X;
int y = m.Rectangle.Y;
int n = clon.Height;
int nm = clon.Width;
                  n = 50;
nm = 50;
                  Rectangle rect = new Rectangle(x, y, n, nm);
graphics.DrawRectangle(whitePen, rect);
graphics.Dispose();
                }
             }
clone.UnlockBits(data);
```

```
            }
            pictureBox10.Image = clon;
            panel5.Visible = true;
        }
private void button6_Click(object sender, EventArgs e)
        {
                                OcrEngine OCRLibrary = new OcrEngine();
                        // Set the absolute path of tessdata.
            OCRLibrary.TessDataPath = "D:/image/";
                        // Set the target text language.
            OCRLibrary.TextLanguage = "eng";
            // Recognize text from image file.
string Imagetext = OCRLibrary.PerformOCR("D://Image//img8.jpg");
        }
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {
            this.Cursor = Cursors.WaitCursor;
imageHandler.RestorePrevious();
imageHandler.SetBrightness(100);
this.Invalidate();
            this.Cursor = Cursors.Default;
imageHandler.SaveBitmap(fp + "img9.jpg");
imageHandler.ResetBitmap();
            this.AutoScrollMinSize = new Size(Convert.ToInt32(imageHandler.CurrentBitmap.Width *
zoomFactor), Convert.ToInt32(imageHandler.CurrentBitmap.Height * zoomFactor));
this.Invalidate();
        }
private void button7_Click_1(object sender, EventArgs e)
        {
            SpeechSynthesizer speak = new SpeechSynthesizer();
speak.SpeakAsync(textBox1.Text);
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing.Imaging;
using System.Drawing;
namespace ImgProImpl
{
class Canny
    {
public int Width, Height;
public Bitmap Obj;
public int[,] GreyImage;
```

```
        //Gaussian Kernel Data
int[,] GaussianKernel;
int KernelWeight;
int KernelSize = 5;
float Sigma = 1;   // for N=2 Sigma =0.85  N=5 Sigma =1, N=9 Sigma = 2    2*Sigma = (int)N/2
        //Canny Edge Detection Parameters
float MaxHysteresisThresh, MinHysteresisThresh;
public float[,] DerivativeX;
public float[,] DerivativeY;
public int[,] FilteredImage;
public float[,] Gradient;
public float[,] NonMax;
public int[,] PostHysteresis;
int[,] EdgePoints;
public float[,] GNH;
public float[,] GNL;
public int[,] EdgeMap;
public int[,] VisitedMap;
public Canny(Bitmap Input)
        {
           // Gaussian and Canny Parameters
           MaxHysteresisThresh = 20F;
           MinHysteresisThresh = 10F;
           Obj = Input;
           Width = Obj.Width;
           Height = Obj.Height;
           EdgeMap = new int[Width, Height];
           VisitedMap = new int[Width, Height];

ReadImage();
DetectCannyEdges();
return;
        }
public Canny(Bitmap Input, float Th, float Tl)
        {
           // Gaussian and Canny Parameters
           MaxHysteresisThresh = Th;
           MinHysteresisThresh = Tl;
          Obj = Input;
           Width = Obj.Width;
           Height = Obj.Height;
           EdgeMap = new int[Width, Height];
           VisitedMap = new int[Width, Height];
ReadImage();
DetectCannyEdges();
return;
        }
```

```
public Canny(Bitmap Input, float Th, float Tl, int GaussianMaskSize, float
SigmaforGaussianKernel)
      {
         // Gaussian and Canny Parameters

         MaxHysteresisThresh = Th;
         MinHysteresisThresh = Tl;
         KernelSize = GaussianMaskSize;
         Sigma = SigmaforGaussianKernel;
         Obj = Input;
         Width = Obj.Width;
         Height = Obj.Height;

         EdgeMap = new int[Width, Height];
         VisitedMap = new int[Width, Height];
ReadImage();
DetectCannyEdges();
return;
      }
public Bitmap DisplayImage()
      {
int i, j;
         Bitmap image = new Bitmap(Obj.Width, Obj.Height);
         BitmapData bitmapData1 = image.LockBits(new Rectangle(0, 0, Obj.Width, Obj.Height),
                        ImageLockMode.ReadOnly, PixelFormat.Format32bppArgb);
unsafe
         {
byte* imagePointer1 = (byte*)bitmapData1.Scan0;

for (i = 0; i < bitmapData1.Height; i++)
            {
for (j = 0; j < bitmapData1.Width; j++)
               {
                  // write the logic implementation here
imagePointer1[0] = (byte)GreyImage[j, i];
imagePointer1[1] = (byte)GreyImage[j, i];
imagePointer1[2] = (byte)GreyImage[j, i];
imagePointer1[3] = (byte)255;
                  //4 bytes per pixel
                  imagePointer1 += 4;
               }//end for j

               //4 bytes per pixel
               imagePointer1 += (bitmapData1.Stride - (bitmapData1.Width * 4));
            }//end for i
         }//end unsafe
```

```
image.UnlockBits(bitmapData1);
return image;// col;
      }     // Display Grey Image

public Bitmap DisplayImage(float[,] GreyImage)
      {
int i, j;
int W, H;
         W = GreyImage.GetLength(0);
         H = GreyImage.GetLength(1);
         Bitmap image = new Bitmap(W, H);
         BitmapData bitmapData1 = image.LockBits(new Rectangle(0, 0, W, H),
                           ImageLockMode.ReadOnly, PixelFormat.Format32bppArgb);
unsafe
         {
byte* imagePointer1 = (byte*)bitmapData1.Scan0;

for (i = 0; i < bitmapData1.Height; i++)
            {
for (j = 0; j < bitmapData1.Width; j++)
               {
                  // write the logic implementation here
imagePointer1[0] = (byte)((int)(GreyImage[j, i]));
imagePointer1[1] = (byte)((int)(GreyImage[j, i]));
imagePointer1[2] = (byte)((int)(GreyImage[j, i]));
imagePointer1[3] = (byte)255;
                  //4 bytes per pixel
                  imagePointer1 += 4;
               }   //end for j
               //4 bytes per pixel
               imagePointer1 += (bitmapData1.Stride - (bitmapData1.Width * 4));
            }//End for i
         }//end unsafe
image.UnlockBits(bitmapData1);
return image;// col;
      }     // Display Grey Imag

public Bitmap DisplayImage(int[,] GreyImage)
      {
int i, j;
int W, H;
         W = GreyImage.GetLength(0);
         H = GreyImage.GetLength(1);
         Bitmap image = new Bitmap(W, H);
         BitmapData bitmapData1 = image.LockBits(new Rectangle(0, 0, W, H),
                           ImageLockMode.ReadOnly, PixelFormat.Format32bppArgb);
unsafe
```

```
        {

byte* imagePointer1 = (byte*)bitmapData1.Scan0;

for (i = 0; i < bitmapData1.Height; i++)
            {
for (j = 0; j < bitmapData1.Width; j++)
                {
                   // write the logic implementation here
imagePointer1[0] = (byte)GreyImage[j, i];
imagePointer1[1] = (byte)GreyImage[j, i];
imagePointer1[2] = (byte)GreyImage[j, i];
imagePointer1[3] = (byte)255;
                   //4 bytes per pixel
                   imagePointer1 += 4;
                }   //end for j
                //4 bytes per pixel
                imagePointer1 += (bitmapData1.Stride - (bitmapData1.Width * 4));
            }//End for i
        }//end unsafe
image.UnlockBits(bitmapData1);
return image;// col;
      }      // Display Grey Image

private void ReadImage()
      {
int i, j;
         GreyImage = new int[Obj.Width, Obj.Height];  //[Row,Column]
         Bitmap image = Obj;
         BitmapData bitmapData1 = image.LockBits(new Rectangle(0, 0, image.Width,
image.Height),
                           ImageLockMode.ReadOnly, PixelFormat.Format32bppArgb);
unsafe
         {
byte* imagePointer1 = (byte*)bitmapData1.Scan0;

for (i = 0; i < bitmapData1.Height; i++)
            {
for (j = 0; j < bitmapData1.Width; j++)
                {
GreyImage[j, i] = (int)((imagePointer1[0] + imagePointer1[1] + imagePointer1[2]) / 3.0);
                   //4 bytes per pixel
                   imagePointer1 += 4;
                }//end for j
                //4 bytes per pixel
                imagePointer1 += bitmapData1.Stride - (bitmapData1.Width * 4);
            }//end for i
```

```
        }//end unsafe
image.UnlockBits(bitmapData1);
return;
      }
private void GenerateGaussianKernel(int N, float S, out int Weight)
      {
float Sigma = S;
float pi;
pi = (float)Math.PI;
int i, j;
int SizeofKernel = N;
float[,] Kernel = new float[N, N];
        GaussianKernel = new int[N, N];
float[,] OP = new float[N, N];
float D1, D2;
        D1 = 1 / (2 * pi * Sigma * Sigma);
        D2 = 2 * Sigma * Sigma;

float min = 1000;

for (i = -SizeofKernel / 2; i <= SizeofKernel / 2; i++)
        {
for (j = -SizeofKernel / 2; j <= SizeofKernel / 2; j++)
          {
Kernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j] = ((1 / D1) * (float)Math.Exp(-(i * i + j * j) / D2));
if (Kernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j] < min)
min = Kernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j];

          }
        }
int mult = (int)(1 / min);
int sum = 0;
if ((min > 0) && (min < 1))
        {

for (i = -SizeofKernel / 2; i <= SizeofKernel / 2; i++)
          {
for (j = -SizeofKernel / 2; j <= SizeofKernel / 2; j++)
            {
Kernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j] = (float)Math.Round(Kernel[SizeofKernel / 2 + i,
SizeofKernel / 2 + j] * mult, 0);
GaussianKernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j] = (int)Kernel[SizeofKernel / 2 + i,
SizeofKernel / 2 + j];
sum = sum + GaussianKernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j];
            }

          }
```

```
        }
else
        {
sum = 0;
for (i = -SizeofKernel / 2; i <= SizeofKernel / 2; i++)
            {
for (j = -SizeofKernel / 2; j <= SizeofKernel / 2; j++)
                {
Kernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j] = (float)Math.Round(Kernel[SizeofKernel / 2 + i,
SizeofKernel / 2 + j], 0);
GaussianKernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j] = (int)Kernel[SizeofKernel / 2 + i,
SizeofKernel / 2 + j];
sum = sum + GaussianKernel[SizeofKernel / 2 + i, SizeofKernel / 2 + j];
                }
            }
        }
        //Normalizing kernel Weight
        Weight = sum;
return;
    }
private int[,] GaussianFilter(int[,] Data)
    {
GenerateGaussianKernel(KernelSize, Sigma, out KernelWeight);

int[,] Output = new int[Width, Height];
int i, j, k, l;
int Limit = KernelSize / 2;
float Sum = 0;
        Output = Data; // Removes Unwanted Data Omission due to kernel bias while convolution
for (i = Limit; i <= ((Width - 1) - Limit); i++)
        {
for (j = Limit; j <= ((Height - 1) - Limit); j++)
            {
                Sum = 0;
for (k = -Limit; k <= Limit; k++)
                {
for (l = -Limit; l <= Limit; l++)
                    {
                        Sum = Sum + ((float)Data[i + k, j + l] * GaussianKernel[Limit + k, Limit + l]);
                    }
                }
Output[i, j] = (int)(Math.Round(Sum / (float)KernelWeight));
            }
        }
return Output;
    }
```

```
private float[,] Differentiate(int[,] Data, int[,] Filter)
      {
int i, j, k, l, Fh, Fw;

          Fw = Filter.GetLength(0);
          Fh = Filter.GetLength(1);
float sum = 0;
float[,] Output = new float[Width, Height];

for (i = Fw / 2; i <= (Width - Fw / 2) - 1; i++)
          {
for (j = Fh / 2; j <= (Height - Fh / 2) - 1; j++)
             {
sum = 0;
for (k = -Fw / 2; k <= Fw / 2; k++)
                {
for (l = -Fh / 2; l <= Fh / 2; l++)
                   {
sum = sum + Data[i + k, j + l] * Filter[Fw / 2 + k, Fh / 2 + l];
                   }
                }
Output[i, j] = sum;

             }

          }
return Output;

      }

private void DetectCannyEdges()
      {
          Gradient = new float[Width, Height];
          NonMax = new float[Width, Height];
          PostHysteresis = new int[Width, Height];
          DerivativeX = new float[Width, Height];
          DerivativeY = new float[Width, Height];
          //Gaussian Filter Input Image
          FilteredImage = GaussianFilter(GreyImage);
          //Sobel Masks
int[,] Dx = {{1,0,-1},
                    {1,0,-1},
                    {1,0,-1}};
int[,] Dy = {{1,1,1},
                    {0,0,0},
                    {-1,-1,-1}};
```

```
        DerivativeX = Differentiate(FilteredImage, Dx);
        DerivativeY = Differentiate(FilteredImage, Dy);
int i, j;

        //Compute the gradient magnitude based on derivatives in x and y:
for (i = 0; i <= (Width - 1); i++)
        {
for (j = 0; j <= (Height - 1); j++)
            {
Gradient[i, j] = (float)Math.Sqrt((DerivativeX[i, j] * DerivativeX[i, j]) + (DerivativeY[i, j] *
DerivativeY[i, j]));
            }
        }
        // Perform Non maximum suppression:
        // NonMax = Gradient;

for (i = 0; i <= (Width - 1); i++)
        {
for (j = 0; j <= (Height - 1); j++)
            {
NonMax[i, j] = Gradient[i, j];
            }
        }
int Limit = KernelSize / 2;
int r, c;
float Tangent;
for (i = Limit; i <= (Width - Limit) - 1; i++)
        {
for (j = Limit; j <= (Height - Limit) - 1; j++)
            {
if (DerivativeX[i, j] == 0)
                Tangent = 90F;
else
                Tangent = (float)(Math.Atan(DerivativeY[i, j] / DerivativeX[i, j]) * 180 / Math.PI);
//rad to degree

                //Horizontal Edge
if (((-22.5 < Tangent) && (Tangent <= 22.5)) || ((157.5 < Tangent) && (Tangent <= -157.5)))
                {
if ((Gradient[i, j] < Gradient[i, j + 1]) || (Gradient[i, j] < Gradient[i, j - 1]))
NonMax[i, j] = 0;
                }
                //Vertical Edge
if (((-112.5 < Tangent) && (Tangent <= -67.5)) || ((67.5 < Tangent) && (Tangent <= 112.5)))
                {
if ((Gradient[i, j] < Gradient[i + 1, j]) || (Gradient[i, j] < Gradient[i - 1, j]))
NonMax[i, j] = 0;
```

```
                }

                //+45 Degree Edge
if (((-67.5 < Tangent) && (Tangent <= -22.5)) || ((112.5 < Tangent) && (Tangent <= 157.5)))
                {
if ((Gradient[i, j] < Gradient[i + 1, j - 1]) || (Gradient[i, j] < Gradient[i - 1, j + 1]))
NonMax[i, j] = 0;
                }

                //-45 Degree Edge
if (((-157.5 < Tangent) && (Tangent <= -112.5)) || ((67.5 < Tangent) && (Tangent <= 22.5)))
                {
if ((Gradient[i, j] < Gradient[i + 1, j + 1]) || (Gradient[i, j] < Gradient[i - 1, j - 1]))
NonMax[i, j] = 0;
                }

            }
        }

        //PostHysteresis = NonMax;
for (r = Limit; r <= (Width - Limit) - 1; r++)
        {
for (c = Limit; c <= (Height - Limit) - 1; c++)
            {

                PostHysteresis[r, c] = (int)NonMax[r, c];
            }

        }

        //Find Max and Min in Post Hysterisis
float min, max;
min = 100;
max = 0;
for (r = Limit; r <= (Width - Limit) - 1; r++)
for (c = Limit; c <= (Height - Limit) - 1; c++)
            {
if (PostHysteresis[r, c] > max)
                {
max = PostHysteresis[r, c];
                }

if ((PostHysteresis[r, c] < min) && (PostHysteresis[r, c] > 0))
                {
min = PostHysteresis[r, c];
                }
            }
```

```
        GNH = new float[Width, Height];
        GNL = new float[Width, Height]; ;
        EdgePoints = new int[Width, Height];

for (r = Limit; r <= (Width - Limit) - 1; r++)
        {
for (c = Limit; c <= (Height - Limit) - 1; c++)
            {
if (PostHysteresis[r, c] >= MaxHysteresisThresh)
                {

                   EdgePoints[r, c] = 1;
                   GNH[r, c] = 255;
                }
if ((PostHysteresis[r, c] < MaxHysteresisThresh) && (PostHysteresis[r, c] >=
MinHysteresisThresh))
                {

                   EdgePoints[r, c] = 2;
                   GNL[r, c] = 255;

                }

            }

        }

HysterisisThresholding(EdgePoints);

for (i = 0; i <= (Width - 1); i++)
for (j = 0; j <= (Height - 1); j++)
            {
EdgeMap[i, j] = EdgeMap[i, j] * 255;
            }

return;
      }

private void HysterisisThresholding(int[,] Edges)
      {
int i, j;
int Limit = KernelSize / 2;
for (i = Limit; i <= (Width - 1) - Limit; i++)
for (j = Limit; j <= (Height - 1) - Limit; j++)
            {
if (Edges[i, j] == 1)
```

```
                {
EdgeMap[i, j] = 1;

                }
            }
for (i = Limit; i <= (Width - 1) - Limit; i++)
        {
for (j = Limit; j <= (Height - 1) - Limit; j++)
            {
if (Edges[i, j] == 1)
                {
EdgeMap[i, j] = 1;
Travers(i, j);
VisitedMap[i, j] = 1;
                }
            }
        }
return;
    }
private void Travers(int X, int Y)
    {
if (VisitedMap[X, Y] == 1)
        {
return;
        }

        //1
if (EdgePoints[X + 1, Y] == 2)
        {
            EdgeMap[X + 1, Y] = 1;
            VisitedMap[X + 1, Y] = 1;
            Travers(X + 1, Y);
return;
        }
        //2
if (EdgePoints[X + 1, Y - 1] == 2)
        {
            EdgeMap[X + 1, Y - 1] = 1;
            VisitedMap[X + 1, Y - 1] = 1;
            Travers(X + 1, Y - 1);
return;
        }

        //3

if (EdgePoints[X, Y - 1] == 2)
        {
```

```
            EdgeMap[X, Y - 1] = 1;
            VisitedMap[X, Y - 1] = 1;
            Travers(X, Y - 1);
return;
        }

        //4

if (EdgePoints[X - 1, Y - 1] == 2)
        {
            EdgeMap[X - 1, Y - 1] = 1;
            VisitedMap[X - 1, Y - 1] = 1;
            Travers(X - 1, Y - 1);
return;
        }
        //5
if (EdgePoints[X - 1, Y] == 2)
        {
            EdgeMap[X - 1, Y] = 1;
            VisitedMap[X - 1, Y] = 1;
            Travers(X - 1, Y);
return;
        }
        //6
if (EdgePoints[X - 1, Y + 1] == 2)
        {
            EdgeMap[X - 1, Y + 1] = 1;
            VisitedMap[X - 1, Y + 1] = 1;
            Travers(X - 1, Y + 1);
return;
        }
        //7
if (EdgePoints[X, Y + 1] == 2)
        {
            EdgeMap[X, Y + 1] = 1;
            VisitedMap[X, Y + 1] = 1;
            Travers(X, Y + 1);
return;
        }
        //8

if (EdgePoints[X + 1, Y + 1] == 2)
        {
            EdgeMap[X + 1, Y + 1] = 1;
            VisitedMap[X + 1, Y + 1] = 1;
            Travers(X + 1, Y + 1);
return;
```

```
            }
            //VisitedMap[X, Y] = 1;
return;
        }

        //Canny Class Ends
    }
}
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using System.Drawing.Drawing2D;

namespace ImgProImpl
{
public class ImageHandler
    {
private string _bitmapPath;
private Bitmap _currentBitmap;
private Bitmap _bitmapbeforeProcessing;
private Bitmap _bitmapPrevCropArea;

public ImageHandler()
        {
        }
public Bitmap CurrentBitmap
        {
get
            {
if (_currentBitmap == null)
                    _currentBitmap = new Bitmap(1, 1);
return _currentBitmap;
            }
set { _currentBitmap = value; }
        }

public Bitmap BitmapBeforeProcessing
        {
get { return _bitmapbeforeProcessing; }
set { _bitmapbeforeProcessing = value; }
        }

public string BitmapPath
        {
get { return _bitmapPath; }
set { _bitmapPath = value; }
```

```
    }

public enum ColorFilterTypes
    {
      Red,
      Green,
      Blue
    };

public void ResetBitmap()
    {
if (_currentBitmap != null && _bitmapbeforeProcessing != null)
      {
        Bitmap temp = (Bitmap)_currentBitmap.Clone();
        _currentBitmap = (Bitmap)_bitmapbeforeProcessing.Clone();
        _bitmapbeforeProcessing = (Bitmap)temp.Clone();
}
    }

public void SaveBitmap(string saveFilePath)
    {
      _bitmapPath = saveFilePath;
if (System.IO.File.Exists(saveFilePath))
System.IO.File.Delete(saveFilePath);
      _currentBitmap.Save(saveFilePath);
    }

public void ClearImage()
    {
      _currentBitmap = new Bitmap(1, 1);
    }
public void RestorePrevious()
    {
      _bitmapbeforeProcessing = _currentBitmap;
    }

public void SetColorFilter(ColorFilterTypes colorFilterType)
    {
      Bitmap temp = (Bitmap)_currentBitmap;
      Bitmap bmap = (Bitmap)temp.Clone();
      Color c;
for (int i = 0; i < bmap.Width; i++)
      {
for (int j = 0; j < bmap.Height; j++)
        {
          c = bmap.GetPixel(i, j);
int nPixelR = 0;
```

```
int nPixelG = 0;
int nPixelB = 0;
if (colorFilterType == ColorFilterTypes.Red)
                {
nPixelR = c.R;
nPixelG = c.G - 255;
nPixelB = c.B - 255;
                }
else if (colorFilterType == ColorFilterTypes.Green)
                {
nPixelR = c.R - 255;
nPixelG = c.G;
nPixelB = c.B - 255;
                }
else if (colorFilterType == ColorFilterTypes.Blue)
                {
nPixelR = c.R - 255;
nPixelG = c.G - 255;
nPixelB = c.B;
                }

nPixelR = Math.Max(nPixelR, 0);
nPixelR = Math.Min(255, nPixelR);

nPixelG = Math.Max(nPixelG, 0);
nPixelG = Math.Min(255, nPixelG);

nPixelB = Math.Max(nPixelB, 0);
nPixelB = Math.Min(255, nPixelB);

bmap.SetPixel(i, j, Color.FromArgb((byte)nPixelR, (byte)nPixelG, (byte)nPixelB));
              }
          }
          _currentBitmap = (Bitmap)bmap.Clone();
      }

public void SetGamma(double red, double green, double blue)
      {
          Bitmap temp = (Bitmap)_currentBitmap;
          Bitmap bmap = (Bitmap)temp.Clone();
          Color c;
byte[] redGamma = CreateGammaArray(red);
byte[] greenGamma = CreateGammaArray(green);
byte[] blueGamma = CreateGammaArray(blue);
for (int i = 0; i < bmap.Width; i++)
          {
for (int j = 0; j < bmap.Height; j++)
```

```
            {
                c = bmap.GetPixel(i, j);
bmap.SetPixel(i, j, Color.FromArgb(redGamma[c.R], greenGamma[c.G], blueGamma[c.B]));
            }
        }
        _currentBitmap = (Bitmap)bmap.Clone();
    }

private byte[] CreateGammaArray(double color)
    {
byte[] gammaArray = new byte[256];
for (int i = 0; i < 256; ++i)
        {
gammaArray[i] = (byte)Math.Min(255, (int)((255.0 * Math.Pow(i / 255.0, 1.0 / color)) + 0.5));
        }
return gammaArray;
    }

public void SetBrightness(int brightness)
    {
        Bitmap temp = (Bitmap)_currentBitmap;
        Bitmap bmap = (Bitmap)temp.Clone();
if (brightness < -255) brightness = -255;
if (brightness > 255) brightness = 255;
        Color c;
for (int i = 0; i < bmap.Width; i++)
        {
for (int j = 0; j < bmap.Height; j++)
            {
                c = bmap.GetPixel(i, j);
int cR = c.R + brightness;
int cG = c.G + brightness;
int cB = c.B + brightness;

if (cR < 0) cR = 1;
if (cR > 255) cR = 255;

if (cG < 0) cG = 1;
if (cG > 255) cG = 255;

if (cB < 0) cB = 1;
if (cB > 255) cB = 255;

bmap.SetPixel(i, j, Color.FromArgb((byte)cR, (byte)cG, (byte)cB));
            }
        }
        _currentBitmap = (Bitmap)bmap.Clone();
```

```
        }

public void SetContrast(double contrast)
        {
            Bitmap temp = (Bitmap)_currentBitmap;
            Bitmap bmap = (Bitmap)temp.Clone();
if (contrast < -100) contrast = -100;
if (contrast > 100) contrast = 100;
contrast = (100.0 + contrast) / 100.0;
contrast *= contrast;
            Color c;
for (int i = 0; i < bmap.Width; i++)
            {
for (int j = 0; j < bmap.Height; j++)
                {
                    c = bmap.GetPixel(i, j);
double pR = c.R / 255.0;
pR -= 0.5;
pR *= contrast;
pR += 0.5;
pR *= 255;
if (pR < 0) pR = 0;
if (pR > 255) pR = 255;

double pG = c.G / 255.0;
                    pG -= 0.5;
                    pG *= contrast;
                    pG += 0.5;
                    pG *= 255;
if (pG < 0) pG = 0;
if (pG > 255) pG = 255;
double pB = c.B / 255.0;
pB -= 0.5;
pB *= contrast;
pB += 0.5;
pB *= 255;
if (pB < 0) pB = 0;
if (pB > 255) pB = 255;

bmap.SetPixel(i, j, Color.FromArgb((byte)pR, (byte)pG, (byte)pB));
                }
            }
            _currentBitmap = (Bitmap)bmap.Clone();
        }

public void SetGrayscale()
        {
```

```
            Bitmap temp = (Bitmap)_currentBitmap;
            Bitmap bmap = (Bitmap)temp.Clone();
            Color c;
for (int i = 0; i < bmap.Width; i++)
            {
for (int j = 0; j < bmap.Height; j++)
                {
                    c = bmap.GetPixel(i, j);
byte gray = (byte)(.299 * c.R + .587 * c.G + .114 * c.B);

bmap.SetPixel(i, j, Color.FromArgb(gray, gray, gray));
                }
            }
            _currentBitmap = (Bitmap)bmap.Clone();
        }

public void SetInvert()
        {
            Bitmap temp = (Bitmap)_currentBitmap;
            Bitmap bmap = (Bitmap)temp.Clone();
            Color c;
for (int i = 0; i < bmap.Width; i++)
            {
for (int j = 0; j < bmap.Height; j++)
                {
                    c = bmap.GetPixel(i, j);
bmap.SetPixel(i, j, Color.FromArgb(255 - c.R, 255 - c.G, 255 - c.B));
                }
            }
            _currentBitmap = (Bitmap)bmap.Clone();
        }
public void Resize(int newWidth, int newHeight)
        {
if (newWidth != 0 && newHeight != 0)
            {
                Bitmap temp = (Bitmap)_currentBitmap;
                Bitmap bmap = new Bitmap(newWidth, newHeight, temp.PixelFormat);

double nWidthFactor = (double)temp.Width / (double)newWidth;
double nHeightFactor = (double)temp.Height / (double)newHeight;

double fx, fy, nx, ny;
int cx, cy, fr_x, fr_y;
                Color color1 = new Color();
                Color color2 = new Color();
                Color color3 = new Color();
                Color color4 = new Color();
```

```
byte nRed, nGreen, nBlue;

byte bp1, bp2;

for (int x = 0; x < bmap.Width; ++x)
            {
for (int y = 0; y < bmap.Height; ++y)
                {

                    fr_x = (int)Math.Floor(x * nWidthFactor);
                    fr_y = (int)Math.Floor(y * nHeightFactor);
cx = fr_x + 1;
if (cx >= temp.Width) cx = fr_x;
cy = fr_y + 1;
if (cy >= temp.Height) cy = fr_y;
fx = x * nWidthFactor - fr_x;
fy = y * nHeightFactor - fr_y;
nx = 1.0 - fx;
ny = 1.0 - fy;

                    color1 = temp.GetPixel(fr_x, fr_y);
                    color2 = temp.GetPixel(cx, fr_y);
                    color3 = temp.GetPixel(fr_x, cy);
                    color4 = temp.GetPixel(cx, cy);

                    // Blue
                    bp1 = (byte)(nx * color1.B + fx * color2.B);

                    bp2 = (byte)(nx * color3.B + fx * color4.B);

nBlue = (byte)(ny * (double)(bp1) + fy * (double)(bp2));
                    // Green
                    bp1 = (byte)(nx * color1.G + fx * color2.G);

                    bp2 = (byte)(nx * color3.G + fx * color4.G);

nGreen = (byte)(ny * (double)(bp1) + fy * (double)(bp2));

                    // Red
                    bp1 = (byte)(nx * color1.R + fx * color2.R);

                    bp2 = (byte)(nx * color3.R + fx * color4.R);

nRed = (byte)(ny * (double)(bp1) + fy * (double)(bp2));

                    bmap.SetPixel(x, y, System.Drawing.Color.FromArgb(255, nRed, nGreen, nBlue));
                }
```

```
            }
            _currentBitmap = (Bitmap)bmap.Clone();
        }
    }

public void RotateFlip(RotateFlipType rotateFlipType)
    {
        Bitmap temp = (Bitmap)_currentBitmap;
        Bitmap bmap = (Bitmap)temp.Clone();
bmap.RotateFlip(rotateFlipType);
        _currentBitmap = (Bitmap)bmap.Clone();
    }
public void Crop(int xPosition, int yPosition, int width, int height)
    {
        Bitmap temp = (Bitmap)_currentBitmap;
        Bitmap bmap = (Bitmap)temp.Clone();
if (xPosition + width > _currentBitmap.Width)
width = _currentBitmap.Width - xPosition;
if (yPosition + height > _currentBitmap.Height)
height = _currentBitmap.Height - yPosition;
        Rectangle rect = new Rectangle(xPosition, yPosition, width, height);
        _currentBitmap = (Bitmap)bmap.Clone(rect, bmap.PixelFormat);
    }
public void DrawOutCropArea(int xPosition, int yPosition, int width, int height)
    {
        _bitmapPrevCropArea = (Bitmap)_currentBitmap;
        Bitmap bmap = (Bitmap)_bitmapPrevCropArea.Clone();
        Graphics gr = Graphics.FromImage(bmap);
        Brush cBrush = new Pen(Color.FromArgb(150, Color.White)).Brush;
        Rectangle rect1 = new Rectangle(0, 0, _currentBitmap.Width, yPosition);
        Rectangle rect2 = new Rectangle(0, yPosition, xPosition, height);
        Rectangle rect3 = new Rectangle(0, (yPosition + height), _currentBitmap.Width,
_currentBitmap.Height);
        Rectangle rect4 = new Rectangle((xPosition + width), yPosition, (_currentBitmap.Width -
xPosition - width), height);
gr.FillRectangle(cBrush, rect1);
gr.FillRectangle(cBrush, rect2);
gr.FillRectangle(cBrush, rect3);
gr.FillRectangle(cBrush, rect4);
        _currentBitmap = (Bitmap)bmap.Clone();
    }
public void RemoveCropAreaDraw()
    {
        _currentBitmap = (Bitmap)_bitmapPrevCropArea.Clone();
    }
public void InsertText(string text, int xPosition, int yPosition, string fontName, float fontSize, string
fontStyle, string colorName1, string colorName2)
```

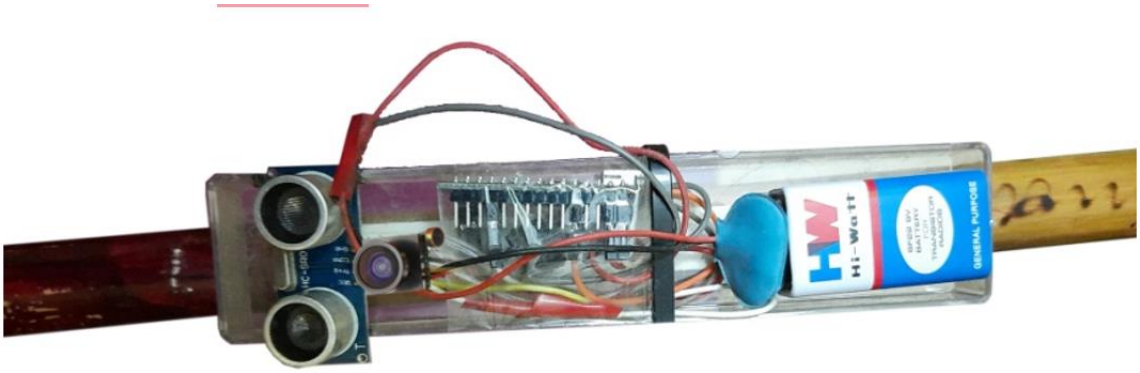
```
        {
            Bitmap temp = (Bitmap)_currentBitmap;
            Bitmap bmap = (Bitmap)temp.Clone();
            Graphics gr = Graphics.FromImage(bmap);
if (string.IsNullOrEmpty(fontName))
fontName = "Times New Roman";
if (fontSize.Equals(null))
fontSize = 10.0F;
            Font font = new Font(fontName, fontSize);
if (!string.IsNullOrEmpty(fontStyle))
            {
                FontStyle fStyle = FontStyle.Regular;
switch (fontStyle.ToLower())
                {
case "bold":
fStyle = FontStyle.Bold;
break;
case "italic":
fStyle = FontStyle.Italic;
break;
case "underline":
fStyle = FontStyle.Underline;
break;
case "strikeout":
fStyle = FontStyle.Strikeout;
break;
                }
font = new Font(fontName, fontSize, fStyle);
            }
if (string.IsNullOrEmpty(colorName1))
                colorName1 = "Black";
if (string.IsNullOrEmpty(colorName2))
                colorName2 = colorName1;
            Color color1 = Color.FromName(colorName1);
            Color color2 = Color.FromName(colorName2);
int gW = (int)(text.Length * fontSize);
gW = gW == 0 ? 10 : gW;
            LinearGradientBrush LGBrush = new LinearGradientBrush(new Rectangle(0, 0, gW,
(int)fontSize), color1, color2, LinearGradientMode.Vertical);
gr.DrawString(text, font, LGBrush, xPosition, yPosition);
            _currentBitmap = (Bitmap)bmap.Clone();
        }
public void InsertImage(string imagePath, int xPosition, int yPosition)
        {
            Bitmap temp = (Bitmap)_currentBitmap;
            Bitmap bmap = (Bitmap)temp.Clone();
            Graphics gr = Graphics.FromImage(bmap);
```

```
if (!string.IsNullOrEmpty(imagePath))
          {
             Bitmap i_bitmap = (Bitmap)Bitmap.FromFile(imagePath);
             Rectangle rect = new Rectangle(xPosition, yPosition, i_bitmap.Width, i_bitmap.Height);
gr.DrawImage(Bitmap.FromFile(imagePath), rect);
          }
          _currentBitmap = (Bitmap)bmap.Clone();
       }
public void InsertShape(string shapeType, int xPosition, int yPosition, int width, int height, string
colorName)
       {
          Bitmap temp = (Bitmap)_currentBitmap;
          Bitmap bmap = (Bitmap)temp.Clone();
          Graphics gr = Graphics.FromImage(bmap);
if (string.IsNullOrEmpty(colorName))
colorName = "Black";
          Pen pen = new Pen(Color.FromName(colorName));
switch (shapeType.ToLower())
          {
case "filledellipse":
gr.FillEllipse(pen.Brush, xPosition, yPosition, width, height);
break;
case "filledrectangle":
gr.FillRectangle(pen.Brush, xPosition, yPosition, width, height);
break;
case "ellipse":
gr.DrawEllipse(pen, xPosition, yPosition, width, height);
break;
case "rectangle":
default:
gr.DrawRectangle(pen, xPosition, yPosition, width, height);
break;

          }
          _currentBitmap = (Bitmap)bmap.Clone();
       }
    }
}
```

CHAPTER 10

PROJECT MODEL



It is intended to implement the system in a real-time context. The development of this innovative system in real time has been done using a Nano Arduino Kit with an ATMEGA 328 surface mountable chip. For this suggested work, 13 digital IO pins and 6 analogue IO pins have also been utilised. This real-time system, which is also linked to a 16MHz crystal oscillator and a camera device, may be used to detect obstacles. The oscillator is familiar with the buzzer sound and can detect obstructions in the blind person's path.

In the above diagram, the figure it shows the original image obtained from the device. Then the image can be enhanced when it required making analysis. The strong and weak edges should be identified to get the text in the image. Finally the obtained text can be sent to the blind person after performing the text to speech synthesis.

CHAPTER 11

CONCLUSION

11. CONCLUSION

The main objective after this paper to blueprint one system which becomes helpful to the visually impaired people by as long as one stick, which notice obstruction in front of the person, due to these characteristic, one smart tools for visually impaired people for under your own steam on the road or surrounding atmosphere through it we can give them rather sense of vision. Tracking, live video capturing and live video monitor is also implemented for the protection or safety purpose of the blind person. Track, live video capture, and live video monitoring is as well implement for the protection or safety cause of the blind being.

REFERENCES

[1] Abdel Ilah Nour Alshbatat, “Automated Mobility And Orientation System For Blind Or Partially Sighted People”,International Journal on Smart Sensing and Intelliegent System.

[2].Alejandro R. Garcia Ramirez and Renato Fonseca Livramento da Silvaetal (2012)“ Artificial EYE An Innovative Idea to Help the Blind”, Conference Proceeding of the International Journal of Engineering Development and Research(IJEDR), SRM University, Kattankulathur, pp.205-207, 2012.

[3] Chris Gearhart, Alex Herold, Dr. Brian Self, Dr. Charles Birdsong, Dr. Lynne Slivovsky, “Use Of Ultrasonic Sensors In The Development Of An Electronic Travel Aid”,IEEE Sensors Applications Symposium New Orleans, LA, USA - February 17-19, 2009.

[4] Jini.S, Swetha.P, Akshara.P.S, Jishnu S, Karthik Selvan, “Voice Maps for Visually Impaired with Obstacle Detection,” International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 4 Issue: 3., pp.14-16, March 2016.

[5] Jismi Johnson, NikhalRajan P, Nivya M Thomas, Rakendh C S, Sijo T Varghese, “Smart Walking Stick for Blind”, International Journal of Engineering Science Invention Research & Development; Vol. III, Issue IX, pp. 557-560, March 2017.

[6] Johann Borenstein and Iwan Ulrich, “The Guide Cane- A Computerized Travel Aidfor The Active Guidance Of Blind Pedestrians”, IEEE International Conference on Robotics and Automation, Albuquerque, NM, Apr. 21-27, 1997.

[7] Kher Chaitrali S., Dabhade Yogita A., Kadam Snehal K., Dhamdhere Swati D., Deshpande Aarti V., “An Intelligent Walking Stick for the Blind,” International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, pp.1057-1062, 2015.

[8] K. Ramarethinam, Mrs K. Thenkumari, [Assist.Prof], P.Kalaiselvan, “Navigation System for Blind People Using GPS & GSM Techniques ,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering , Vol. 3, Special Issue 2, pp.398-405, April 2014.

[9]Mohd Helmyabd Wahab and Amirul A. Talibetal , “A Review on an Obstacle Detection in Navigation of Visually Impaired”, International Organization of Scientific Research Journal of Engineering (IOSRJEN), Vol.3, No.1 pp. 01-06, January 2013.

[10] Mohd Helmy Wahab, Amirul A. Talib, Herdawatie A. Kadir, A.Noraziah, Roslina M. Sidek, “Smart Cane: Assistive Cane For Visually-Impaired People”,IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011.

[11] Sung Jae Kang, Young Ho, Kim, In Hyuk Moon, “Development Of An Intelligent Guide-Stick For The Blind”, IEEE International Conference on Robotics & Automation Seoul, Korea, May 21-26, 2001.

[12] Zul Azizi Hailani, Sakinah Jamaludin, “An Electronically Guided Walking Stick For The Blind” University Tenaga Nasional, Malaysia.