SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Face Recognition Using Viola Jones And Eigenface Algorithm

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Anand Singh	Exam No: B121094204
Erarica Mehra	Exam No: B121094214
Saundarya Dorle	Exam No: B121094212

Under The Guidance of

Prof. Ishwar Kalbandi

DEPARTMENT OF COMPUTER ENGINEERING
Padmashree Dr. D.Y. Patil Institute of Engineering, Management and Research Akurdi, Pune-411044

Academic Year 2015-2016

Pad Dr.D.Y.Patil Institute of Engineering, Management and Research, Akurdi
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Face Recognition Using Viola Jones And Eigenface Algorithm

Submitted by

Anand Singh	Exam No: B121094204
Erarica Mehra	Exam No: B121094214
Saundarya Dorle	Exam No: B121094212

is a bonafide work carried out by Students under the supervision of Prof. Ishwar Kalbandi and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Prof. Ishwar Kalbandi
Internal Guide
Dept. of Computer Engg.

Prof. P.P. Shevatakar
H.O.D
Dept. of Computer Engg.

Dr. A. V. Patil
Principal
Pad Dr.D.Y.Patil Institute of Engineering, Management and Research, Akurdi

Sign of Internal Examiner

Sign of External Examiner

PROJECT APPROVAL SHEET

A Project Title

FACE RECOGNITION USING VIOLA JONES AND EIGENFACE ALGORITHM

is successfully completed by

Anand Singh	Exam No: B121094204
Erarica Mehra	Exam No: B121094214
Saundarya Dorle	Exam No: B121094212

at

DEPARTMENT OF COMPUTER ENGINEERING

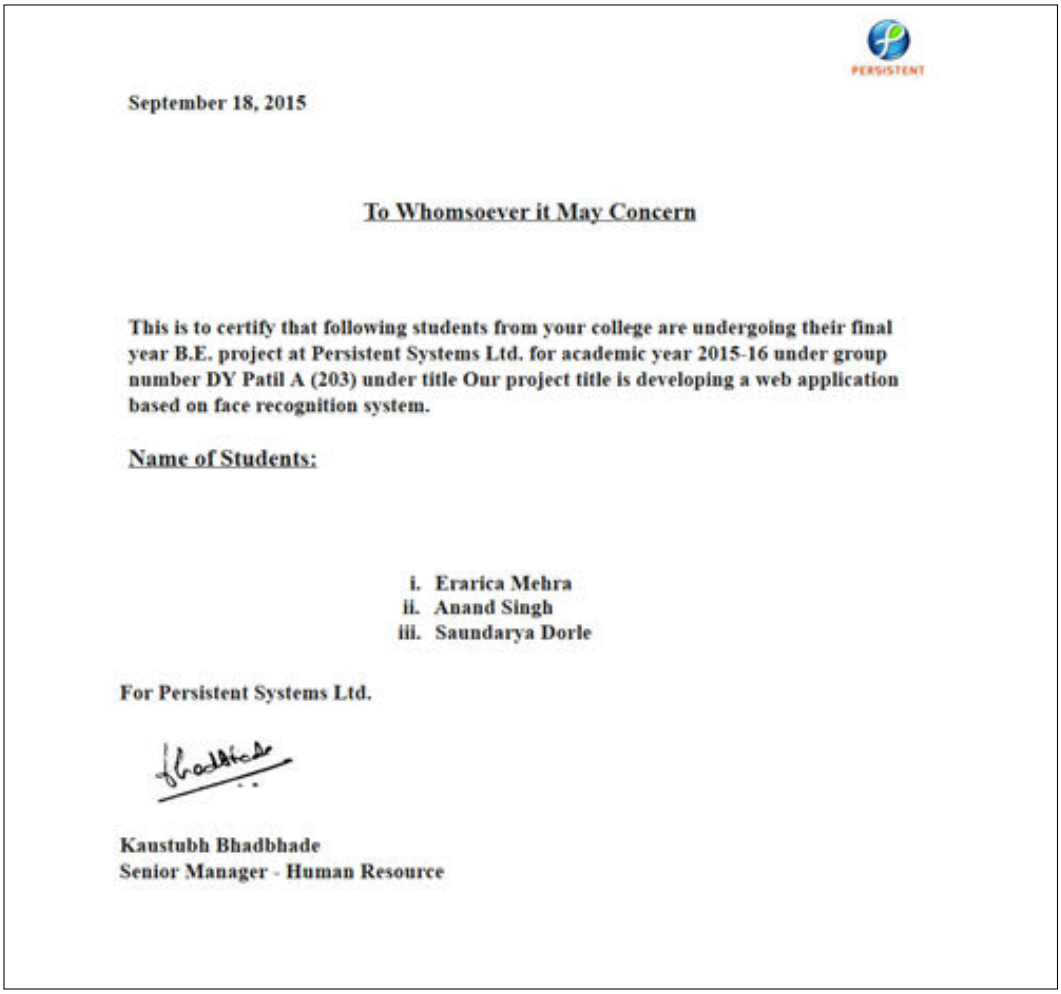
(Padmashree Dr. D.Y. Patil Institute of Engineering, Management and Research, Akurdi)

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2015-2016

Prof. Ishwar Kalbandi	Prof. P.P.Shevtakar
Internal Guide	H.O.D
Dept. of Computer Engg.	Dept. of Computer Engg.

Sponsorship Letter



PERSISTENT

September 18, 2015

To Whomsoever it May Concern

This is to certify that following students from your college are undergoing their final year B.E. project at Persistent Systems Ltd. for academic year 2015-16 under group number DY Patil A (203) under title Our project title is developing a web application based on face recognition system.

Name of Students:

i. Erarica Mehra
ii. Anand Singh
iii. Saundarya Dorle

For Persistent Systems Ltd.

Kaustubh Bhadbhade
Senior Manager - Human Resource

Project Completion Letter

March 31, 2016

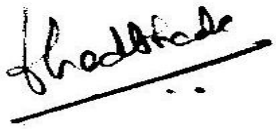
To Whomsoever it May Concern

This is to certify that following students have completed their final year B.E. Project, under title Information On the Move: Use of Mobile Information System For Police Forces at Persistent Systems Ltd. for academic year 2015-16.

Name of Students:

i. Erarica Mehra
ii. Anand Singh
iii. Saundarya Dorle

For Persistent Systems Ltd.



Kaustubh Bhadbhade
SeniorManager - Human Resource

Abstract

Face recognition system is a computer application capable of verifying or identifying a person from a digital image. One of the ways to do this is to compare facial features from an image from a database. It is typically used in security systems and is similar with fingerprint or eye iris recognition system.

Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data.

Recognition algorithms can be divided into two main approaches, geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances.

Face is a complex multidimensional structure and needs a good computing techniques for recognition. Our approach treats face recognition as a two-dimensional recognition problem. In this scheme face recognition is done by Principal Component Analysis (PCA). Face images are projected onto a face space that encodes best variation among known face images. The face space is defined by eigenface which are eigen vectors of the set of faces, which may not correspond to general facial features such as eyes, nose, lips. The eigenface approach uses the PCA for recognition of the images. The system performs by projecting pre extracted face image onto a set of face space that represent significant variations among known face images. Face will be categorized as known or unknown face after matching with the present database. If the user is new to the face recognition system then his/her template will be stored in the database else matched against the templates stored in the database. The variable reducing theory of PCA accounts for the smaller face space than the training set of face.

Acknowledgments

It gives us great pleasure in presenting the project report on ***'Face Recognition using Viola Jones and Eigenfaces Algorithm'****.*

I would like to take this opportunity to thank my internal guide ***Prof.Ishwar Kalbandi*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Prof. P.P. Shevataker****, Head of Computer Engineering Department, DYPIEMR for her indispensable support, suggestions.*

In the end our special thanks to ***Mr.Kaustubh Bhadbhade(Senoir Manager - Human Resource)*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.*

Anand Singh
Saundarya Dorle
Erarica Mehra
(B.E. Computer Engg.)

Contents

List of Figures

List of Tables

Chapter 1

Synopsis

1.1 Project Title

Face Recognition using Viola Jones and Eigenfaces Algorithm

1.2 Project Option

Industry Sponsored by Persistent System

1.3 Internal Guide

Prof. Ishwar Kalabandi

1.4 Sponsorship and External Guide

- Sponsored by Persistent Systems
- Enternal Guide- Mr. Kaustubh Bhadbhade (Senoir Manager - Human Resource)

1.5 Technical Keywords (As per ACM Keywords)

1. Standalone Application
   (a) Face Recognition System
      i. Feature extraction
      ii. Eigenvectors
      iii. Eigenfaces

1.6 Problem Statement

Police organizations are information-intensive and intelligence led organizations. New advanced services offer the promise that police officers can access the information they need to carry out their tasks more easily, when and where the information is needed. To aid the police service we have come up with an application that takes image of a suspect as input and compares the image to all the images from the database. If the suspect has a past criminal

background the system displays the details of his/her criminal record such as name, gender, address, crime detail, punishment,etc.

1.7 Abstract

Recognition system is a computer application capable of verifying or identifying a person from a digital image. One of the ways to do this is to compare facial features from an image from a database. It is typically used in security systems and is similar with fingerprint or eye iris recognition system.
Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw.These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.
Face is a complex multidimensional structure and needs a good computing techniques for recognition. Our approach treats face recognition as a two-dimensional recognition problem. In this scheme face recognition is done by Principal Component Analysis (PCA). Face images are projected onto a face space that encodes best variation among known face images. The face space is defined by eigenface which are eigenvectors of the set of faces, which may not correspond to general facial features such as eyes, nose, lips. The eigenface approach uses the PCA for recognition of the images. The system performs by projecting pre extracted face image onto a set of face space that represent significant variations among known face images. Face will be categorized as known or unknown face after matching with the present database. If the user is new to the face recognition system then his/her template will be stored in the database else matched against the templates stored in the database. The variable reducing theory of PCA accounts for the smaller face space than the training set of face.

1.8 Goals and Objectives

- To speed up investigation process by quick identification of suspects
- Gathering details of suspects
- Simplification of police investigation

1.9 Relevant mathematics associated with the Project

- System description: XP/Windows 7/ Windows 8/ Windows 10, RAM 512 MB or more, Hard Disk 100 GB, Pentium Processor
- Input: Image to be matched
- Output: Confirmation that the image has matched and details of matched suspect including
  - name
  - address
  - gender
  - date of birth
  - blood group
  - crime
- Functions : capture_ images(), crop_ faces_ for_ training(), face_ recognition()
- capture_ image(): This function uses the web camera to capture image in real time
- crop_ images_ for_ training(): This function detects the faces from the captured images, crops and saves the faces
- face_ recognition()- This function matches the faces from the database and displays the result

Success Conditions: Suspect identified and Criminal records are displayed

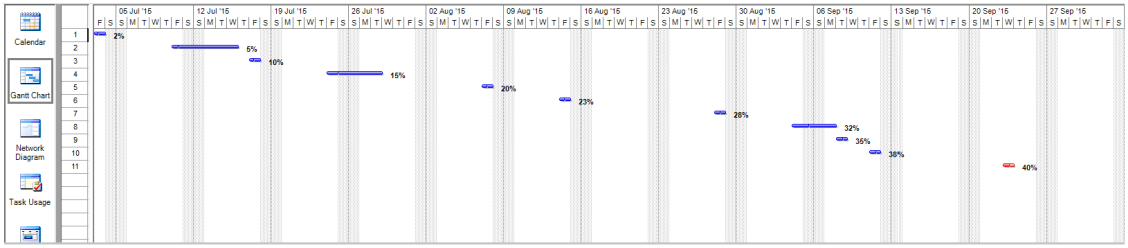
Failure Conditions: Image not recognized

1.10 Names of Conferences / Journals where papers can be published

- International Journal of Research in Science and Engineering (IJRISE)
- Interternational Conference on Emerging Trends in Engineering and Management Research (ICETEMR-16)

1.11 Plan of Project Execution

	Name	Duration	Work	Start	Finish	Resource Names	% Comple
1	Paper Discussion	1 day	8 hrs	Fri 3/7/15	Fri 3/7/15	Internet	2%
2	More Paper search	4 days	32 hrs	Fri 10/7/15	Wed 15/7/15	Internet	5%
3	Topic Finilization, Da	1 day	8 hrs	Fri 17/7/15	Fri 17/7/15	Project Guide	10%
4	Title, topic refined, c	3 days	24 hrs	Fri 24/7/15	Tue 28/7/15	Project Guide	15%
5	Synopsis checked v	1 day	8 hrs	Fri 7/8/15	Fri 7/8/15	Project Guide	20%
6	Discussion of proje	1 day	8 hrs	Fri 14/8/15	Fri 14/8/15	Project Guide	23%
7	I / O specification of	1 day	8 hrs	Fri 28/8/15	Fri 28/8/15	Project Guide	28%
8	Discussion on Modu	2 days	16 hrs	Fri 4/9/15	Mon 7/9/15	Project Guide	32%
9	Review1 of Synops	1 day	8 hrs	Tue 8/9/15	Tue 8/9/15	Project Guide	35%
10	Review 2 & Name c	1 day	8 hrs	Fri 11/9/15	Fri 11/9/15	Project Guide	40%



1.12 Review of Conference/Journal Papers supporting Project idea

Mobile Application for Police Officers By Harry Bouwman, Timber Haaker, Henny de Vos

Design and Implementation of 3G mobile police system by Baocai Zhong; Lu Niu; Hanbin Chen

Implementation and Benchmarking of Perceptual Image Hash Functions by Christoph Zauner

Perceptual Image Hashing via Feature Points: Performance Evaluation And Trade Offs by Vishal Monga, Brian L. Evans

Perceptual Image Hashing by Azhar Hadmi, William Puech

Face Recognition Using PCA and Eigen Face Approach by Abhishek Singh and Saurabh Kumar

An Efficient Method For Face Recognition Using Principal Component Analysis(PCA), Gunjan Dashor, Dr V. Cyril Raj

J.Zobel and A.Moffat, "' Inverted files for text search engines,"' ACM Computing Surveys, 2006

"International Conference on Computer Vision" by N.Kumar, A.C. Berg, P.N. Belhumeeur and S. K. Naayar

'Robust Real-Time Face Detection' PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA

Chapter 2

Technical Keywords

2.1 Area of Project

- Face detection using Viola Jones algorithm
- Feature Extraction using Eigen faces
- Face recognition

2.2 Technical Keywords

1. Standalone Application
   (a) Face Recognition System
      i. Feature extraction
      ii. Eigenvectors
      iii. Eigenfaces

Chapter 3

Introduction

3.1 Project Idea

Developing an application that takes image of a suspect as input and compares the image to all the images from the database. If the suspect has a past criminal background, the system displays the details of his/her criminal record such as name, gender, address, crime detail, punishment, etc.

3.2 Literature Survey

- Design and implement of 3G mobile police system by Baocai Zhong; Lu Niu; Hanbin Chen
  Police organizations are information-intensive and intelligence led organizations. New advanced services offer the promise that police officers can access the information they need to carry out their tasks more easily, when and where the information is needed. The main questions in this case are how to make information available to police officers , to ensure that the information matches the police officers specific context, and to increase the performance of the police organization.
- Face Recognition Using PC and Eigen Face Approach by Abhishek Singh, Saurabh Kumar Face is a complex multidimensional structure and needs a good computing techniques for recognition. Our approach treats face recognition as a two-dimensional recognition problem. In this scheme face recognition is done by Principal Component Analysis (PCA). Face images are projected onto a face space that encodes best variation among known face images. The face space is deffined by eigenface which are eigenvectors of the set of faces, which may not correspond to general facial features such as eyes, nose, lips. The eigenface approach uses the PCA for recognition of the images. The system performs by projecting pre extracted face image onto a set of face space that represent signifficant variations among known face images. Face will be categorized as known or unknown face after matching with the present database. If the

user is new to the face recognition system then his/her template will be stored in the database else matched against the templates stored in the database. The variable reducing theory of PCA accounts for the smaller face space than the training set of face.

- Robust Real-Time Face Detection' PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
  There are three main contributions of our face detection framework. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections. The first contribution of this paper is a new image representation called an integral image that allows for very fast feature evaluation. Viola and Jones use a set of features which are reminiscent of Haar Basis functions (though we will also use related filters which are more complex than Haar filters). In order to compute these features very rapidly at many scales we introduce the integral image representation for images (the integral image is very similar to the summed area table used in computer graphics . The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Haar-like features can be computed at any scale or location in constant time. The second contribution of this paper is a simple and efficient classifier that is built by selecting a small number of important features from a huge library of potential features using AdaBoost (Freund and Schapire, 1995). Within any image sub-window the total number of Haar-like features is very large, far larger than the number of pixels. In order to ensure fast classifi-cation, the learning process must exclude a large majority of the available features, and focus on a small set of critical features. Motivated by the work of Tieu and Viola (2000) feature selection is achieved using the AdaBoost learning algorithm by constraining each weak classifier to depend on only a single feature. As a result each stage of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. AdaBoost provides an effective learning algorithm

and strong bounds on generalization performance. The third major contribution of this paper is a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image.

- Toward a Practical Face Recognition System:Robust Alignment and Illumination by Sparse Representation by Andrew Wagner, Student Member, IEEE, John Wright, Member, IEEE
  Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. Recognition algorithms can be divided into two main approaches,geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances.
- Implementation and Benchmarking of Perceptual Image Hash Functions by Christoph Zauner
  A perceptual image hash function should have the property that two images that look the same to the human eye map to the same hash value, even if the images have different digital representations, e.g. separated by a large distance in mean squared error. An immediately obvious application for a perceptual image hash is identification/search of images in large databases.

Chapter 4

Problem Definition and scope

4.1 Problem Statement

New advanced services offer the promise that police officers can access the information they need to carry out their tasks more easily. To aid the police service we have come up with an application that takes image of a suspect as input and compares the image to all the images from the database and displays the details of his/her criminal record such as name, gender, address, crime, etc.

4.1.1 Goals and objectives

Goal and Objectives:

- The objective of the project is to speed up investigation process by quick identification of suspects, gathering details based on available digital evidence and simplification of police investigation

Input: It takes image as an input. The web camera captures the image at real time

Output: If the image matches, it displays the criminal record of the suspect.

4.1.2 Statement of scope

-
- The scope of the project can be summarized as identifying suspects/criminals and displaying their criminal records that is name, gender, address, crime detail, punishment and pending cases. It takes input in the form of image which are captured at real time. The features are extracted of the captured image and compared against the extracted features of the images already stored. The Euclidian Distance is computed and if the difference is within the threshold value, the image is matched and results are displayed.

4.2 Software context

4.3 Major Constraints

- Quality of camera and resolution are major constraints.
- If the distance of the person from the camera is more, image captured will not be trained properly.
- Images should be captured from same resolution camera only. Camera should not be changed.
- The database consists of images only.
- The extracted features are stored in a separate file. Computation on the extracted features is not possible.

4.4 Outcome

- Outcome of the project: Confirmation that the image has matched and details of matched suspect including
  - name
  - address
  - gender
  - date of arrest
  - blood group
  - crime

4.5 Applications

- Applications of our Project is for the police forces for quicker identification of criminals. Since the project is on face recognition it can be put to multiple uses, for example in companies, offices, banks for allowing access to authorized persons. It is an alternative to fingerprint recognition and iris recogniton.

4.6 Software Resources Required

Platform : Visual Studio 2012

1. Operating System: Windows 7/ Windows 8/ Windows 10
2. Programming Language: cpp

4.7 Hardware Resources Required

Platform :

1. RAM: 2GB
2. Hard Disk: 80GB

Chapter 5

Project Plan

5.1 Project Estimates

Use Waterfall model and associated streams derived from assignments 1,2, 3, 4 and 5(Annex A and B) for estimation.

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

5.1.1.2 Time Estimates

5.1.2 Project Resources

5.2 Risk Management w.r.t. NP Hard analysis

This section discusses Project risks and the approach to managing them.

5.2.1 Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [**?**]. Please refer table 5.1 for all the risks. You can refereed following risk identification questionnaire.

1. Have top software and customer managers formally committed to support the project?
2. Are end-users enthusiastically committed to the project and the system/product to be built?
3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end-users have realistic expectations?
6. Does the software engineering team have the right mix of skills?
7. Are project requirements stable?
8. Is the number of people on the project team adequate to do the job?
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built? .....

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Description 1	Low	Low	High	High
2	Description 2	Low	Low	High	High

Table 5.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.2: Risk Probability definitions [**?**]

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.3: Risk Impact definitions [**?**]

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk ID	1
Risk Description	Description 1
Category	Development Environment.
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Strategy
Risk Status	Occurred

Risk ID	2
Risk Description	Description 2
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Better testing will resolve this issue.
Risk Status	Identified

Risk ID	3
Risk Description	Description 3
Category	Technology
Source	This was identified during early development and testing.
Probability	Low
Impact	Very High
Response	Accept
Strategy	Example Running Service Registry behind proxy balancer
Risk Status	Identified

5.3 Project Schedule

5.3.1 Project task set

- Task 1: Topic Selection
- Task 2: Discussion of the Topic with project guide
- Task 3: Find out IEEE paper related to topic
- Task 4: Checked the abstract of project
- Task 5: Make synopsis and give first review
- Task 6: Give second review
- Task 7: First journal paper published (ijrise)
- Task 8: Discussion about project implementation
- Task 9: Start the coding
- Task 10: Review on module
- Task 11: Second Journal Paper published (ijates)

5.3.2 Timeline Chart

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Auto**	**Design**	**13 days**	**Fri 22/1/16 08:00**	**Tue 9/2/16 17:00**		
7	**Auto**	**Development**	**32.98 days**	**Fri 22/1/16 08:00**	**Tue 8/3/16 16:52**		
8	Auto	Review functional specifications	1 day	Fri 22/1/16 08:00	Fri 22/1/16 17:00	6	Project Guide,Team member
9	Auto	Identify modular/tiered	1 day	Mon 25/1/16 08:00	Mon 25/1/16 17:00	8	project Guide
10	Auto	Assign	1 day	Tue 26/1/16 08:00	Tue 26/1/16 17:00	9	Team member
11	Auto	Develop code	15 days	Wed 27/1/16 08:00	Tue 16/2/16 17:00	10	Project Guide,Team member
12	Auto	Developer testing (primary	15 days	Tue 16/2/16 16:52	Tue 8/3/16 16:52	11FS-0.02 days	Project Guide,Team member
13	Auto	Development	0 days	Tue 8/3/16 16:52	Tue 8/3/16 16:52	12	
14	**Auto**	**Testing**	**59.98 days**	**Fri 22/1/16 08:00**	**Thu 14/4/16 16:52**		
15	Auto	Develop unit test plans using product	4 days	Fri 22/1/16 08:00	Wed 27/1/16 17:00	6	Team member
16	Auto	Develop integration test plans using	4 days	Fri 22/1/16 08:00	Wed 27/1/16 17:00	6	Team member
17	**Auto**	**Unit Testing**	**15 days**	**Tue 8/3/16 16:52**	**Tue 29/3/16 16:52**		
24	**Auto**	**Integration Testing**	**12 days**	**Tue 29/3/16 16:52**	**Thu 14/4/16 16:52**		
30	**Auto**	**Documentation**	**1 day**	**Fri 22/1/16 08:00**	**Fri 22/1/16 17:00**		
31	Auto	Review Help documentation	1 day	Fri 22/1/16 08:00	Fri 22/1/16 17:00		Team member

5.4 Team Organization

5.4.1 Team structure

Member name	role
Erarica Mehra	Team leader
Saundarya Dorle	Team member
Anand Singh	Team member

Table 5.4: Team Structure

Chapter 6

Software requirement specification

6.1 Introduction

6.2 Software Description - opencv with Visual Studio 2012

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level.

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++ and C++/CLI, VB.NET and support for other languages such as Python, Ruby,etc. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS.

6.2.1 Features

- opencv OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel's research center in Nizhny Novgorod (Russia), later supported by Willow Garage and now maintained by Itseez.The library is cross-platform and free for use under the open-source BSD license. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C, Perl, Ch, and Ruby have been developed to encourage adoption by a wider audience.

All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

- Code editor: The code editor is used for all supported languages. The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Visual Studio features background compilation (also called incremental compilation). As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic but has now been expanded for all included languages.
- Debugger: Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. If source code for the running process is available, it displays the code as it is being run. If source code is not available, it can show the disassemble. The Visual Studio debugger can also create memory dumps as well as load them later for debugging. Multithreaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment crashes.

  The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses). Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line (of source code) at a time. It can either step into functions to debug inside it, or step over it, i.e., the execution of the function body isn't available for manual inspection.
- Designer: Visual Studio includes a host of visual designers to aid in the development of applications. These tools include:
  1. Windows Forms Designer The Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box, grid view, etc.) can be bound to data sources

like databases or queries. Data-bound controls can be created by dragging items from the Data Sources window onto a design surface.[31] The UI is linked with code using an event-driven programming model. The designer generates either C or VB.NET code for the application.

2. Class designer The Class Designer is used to author and edit the classes (including its members and their access) using UML modeling. The Class Designer can generate C and VB.NET code outlines for the classes and methods. It can also generate class diagrams from hand-written classes.
3. Data designer The data designer can be used to graphically edit database schemas, including typed tables, primary and foreign keys and constraints. It can also be used to design queries from the graphical view. Mapping designer From Visual Studio 2008 onwards, the mapping designer is used by LINQ to SQL to design the mapping between database schemas and the classes that encapsulate the data. The new solution from ORM approach, ADO.NET Entity Framework, replaces and improves the old technology.
4. Data Explorer Data Explorer is used to manage databases on Microsoft SQL Server instances. It allows creation and alteration of database tables (either by issuing T-SQL commands or by using the Data designer). It can also be used to create queries and stored procedures, with the latter in either T-SQL or in managed code via SQL CLR. Debugging and IntelliSense support is available as well.
5. Server Explorer The Server Explorer tool is used to manage database connections on an accessible computer. It is also used to browse running Windows Services, performance counters, Windows Event Log and message queues and use them as a data source.

- Extensibility Visual Studio allows developers to write extensions for Visual Studio to extend its capabilities. These extensions "plug into" Visual Studio and extend its functionality. Extensions come in the form of macros, add-ins, and packages. Macros represent repeatable tasks and actions that developers can record pro-grammatically for saving, replaying, and distributing. Macros, however, cannot implement new commands or create tool windows. They are written using Visual Basic and are not compiled. Add-Ins provide access to the Visual Studio object model and can interact with the IDE tools. Add-Ins can be used to implement new functionality and can add new tool windows.

6.2.2 Overview of responsibilities of Developer

The application is developed keeping in mind the accuracy of the algorithm so that it gives correct match and does not give more than one match other wise the purpose of the system fails.

6.3 Usage Scenario

This section provides various usage scenarios for the system to be developed.

6.3.1 User profiles

User: The sole user of the system is a police officer(admin) who has correct user-name and password to log into the system.

Criminal Database: contains criminal records. Admin has the responsibility to update and maintain the database

---

6.3.2 Use Case View

Use Case Diagram.

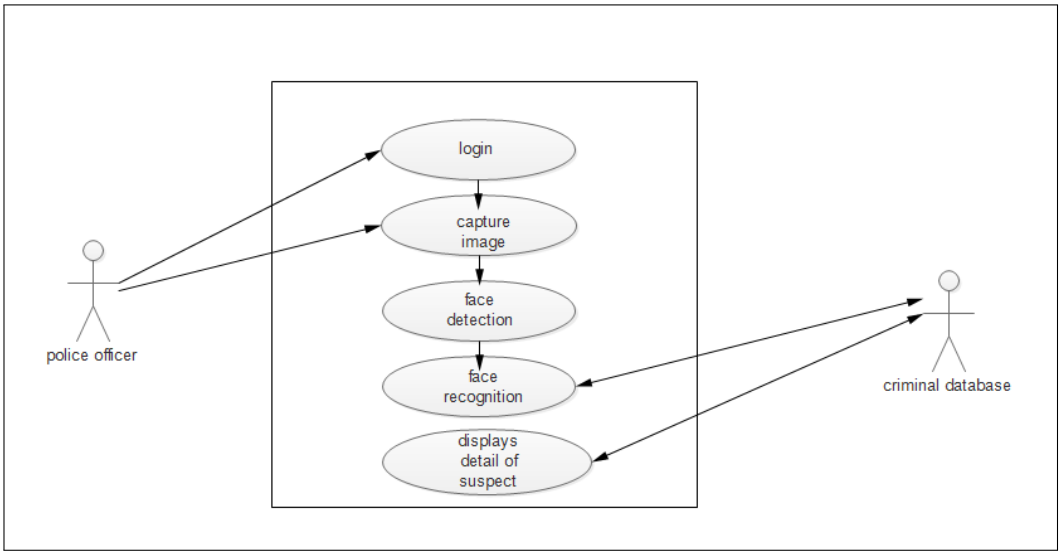


Figure 6.1: Use case diagram

6.4 Functional Model and Description

6.4.1 Data Flow Diagram

6.4.1.1 Level 0 Data Flow Diagram

Refer 6.2

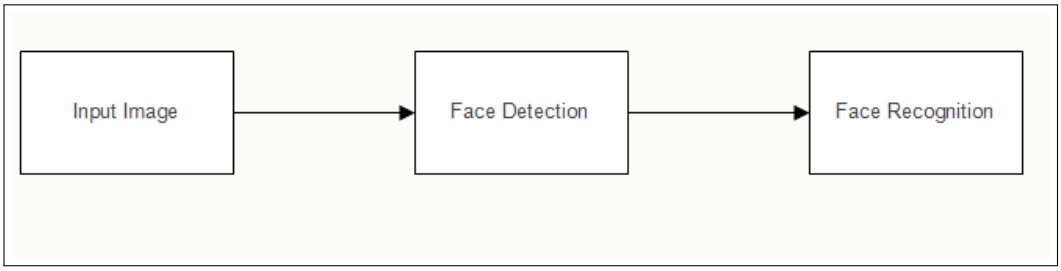


Figure 6.2: DFD level 0 diagram

6.4.1.2 Level 1 Data Flow Diagram

Refer 6.3

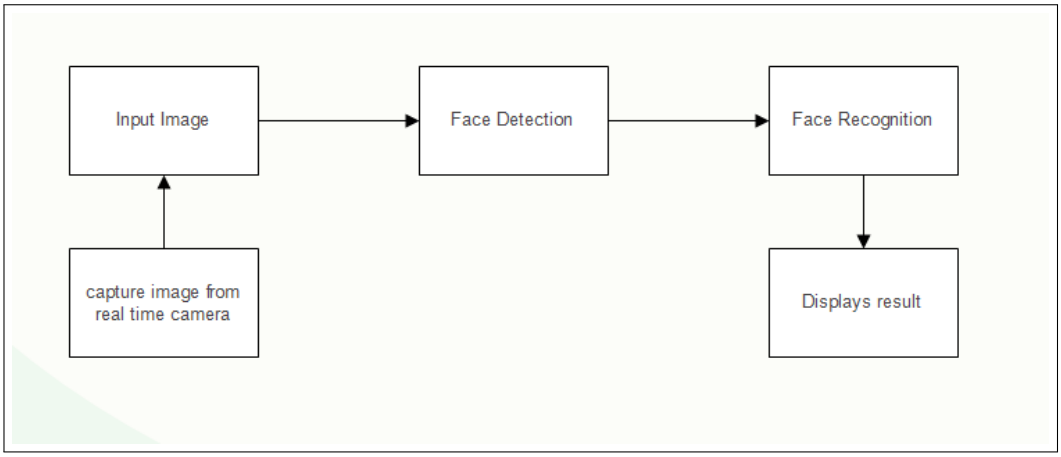


Figure 6.3: DFD level 1 diagram

6.4.1.3 Level 2 Data Flow Diagram

Refer 6.4

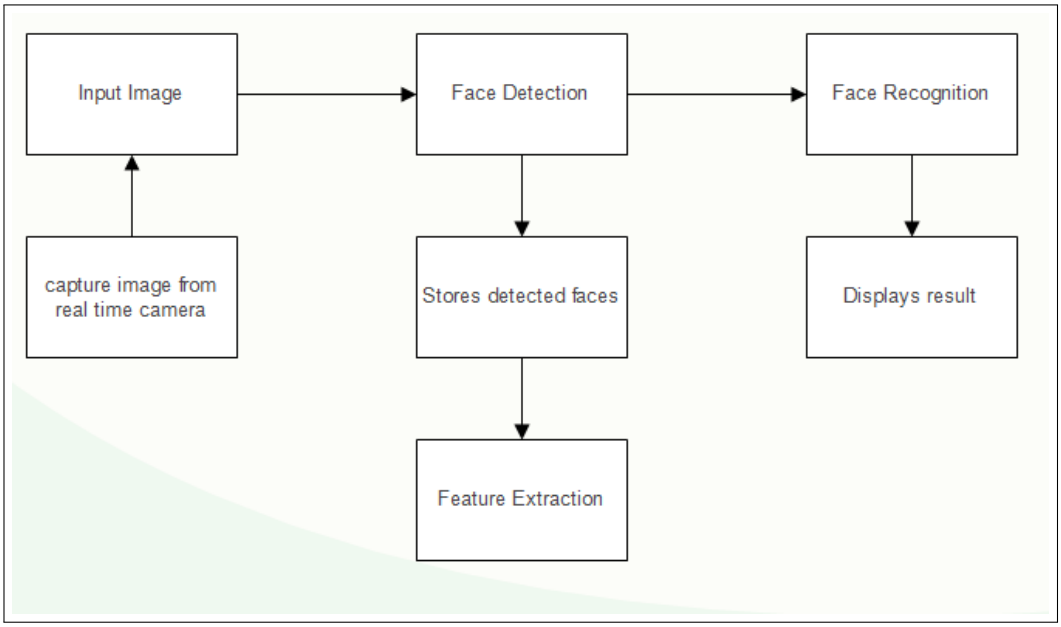


Figure 6.4: DFD level 2 diagram

6.4.2 Activity Diagram:

- The Activity diagram represents the steps taken in the execution of the application

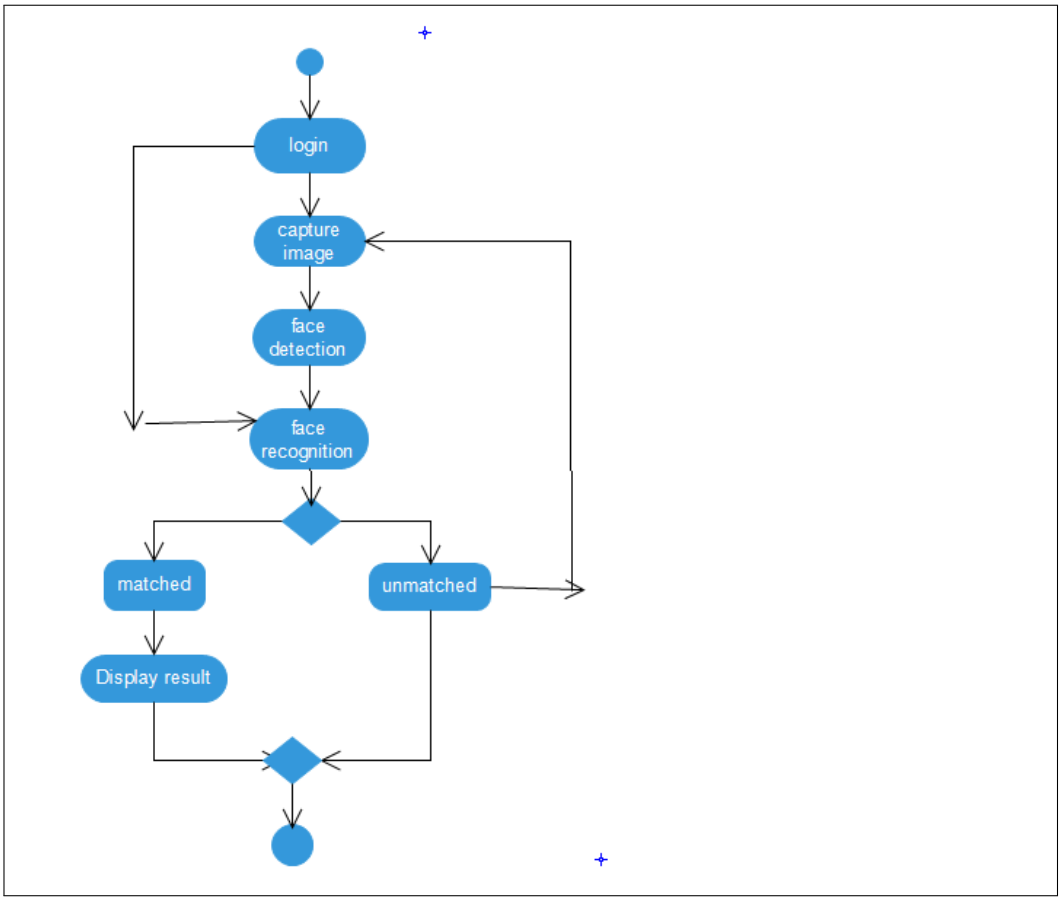


Figure 6.5: Activity diagram

6.4.3 Non Functional Requirements:

- Performance Requirements: The higher the accuracy, the lower the error rate of analyzing. The accuracy in turn depends on the camera and environment conditions. So the camera and persons distance from the camera should be proper. Resolution of camera should not change for better performance.

- Software quality attributes: A simple GUI is provided to interact with the application.
  - Correctness: The system works correctly and efficiently if it is used in the way mentioned in the user manual
  - Reliability: The system provides a very easy-to-use interface to the user. Using the system is easy and does not involve any complicated procedures. The system is user friendly and self-explanatory.
  - Integrity: The system maintains its integrity based on its use.
  - Maintainability: The system is easy to maintain. Regular maintenance of the database is essential for ensuring the system gives optimal performance.

6.4.4 Sequence Diagram:

Sequence Diagram: A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Fig.6.6 example shows the sequence of events taking place while running the application

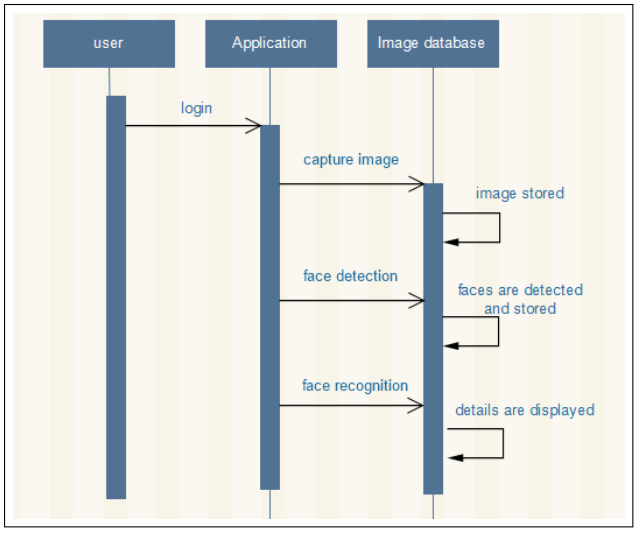


Figure 6.6: Sequence diagram

Chapter 7

Detailed Design Document using Appendix A and B

7.1 Introduction

This document specifies the design that is used to solve the problem of Product.

7.2 Architectural Design

Viola and Jones Algorithm : A statistical approach originally for the task of human face detection and tracking. Employed 3 techniques :

- Haar-like features: Each Haar-like feature consists of two or three connected black and white rectangles. Each feature results in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle.

- Integral image: If we consider all possible parameters of the haar features like position, scale and type we end up calculating about 160,000+ features in this window. Since it is clear that huge number of these rectangular haar features have to be evaluated each time Viola Jones have come up with a neat technique to reduce the computation rather than summing up all pixel values under the black and white rectangles every time. They have introduced the concept of integral image to find the sum of all pixels under a rectangle with just 4 corner values of the integral image.

- AdaBoost Learning algorithm: As stated previously there can be approximately 160,000 + feature values within a detector at 24x24 base resolution which need to be calculated. But it is to be understood that only few set of features will be useful among all these features to identify a face. AdaBoost is a machine learning algorithm which helps in finding only the best features among all these 160,000+ features. After these features are found, a weighted combination of all these features in used in evaluating and deciding any given window has a face or not.
- Cascade Classifiers: For fast processing, the algorithm should concentrate on discarding non-faces quickly and spend more on time on probable face regions. Hence a single strong classifier formed out of linear combination of all best features is not a good to evaluate on each window because of computation cost. Therefore, a cascade classifier is used which is composed of stages each containing a strong classifier. So all

the features are grouped into several stages where each stage has certain number of features. Each stage determines whether a given sub window is definitely not a face or may be a face. A given sub window is immediately discarded as not a face if it fails in any of the stage

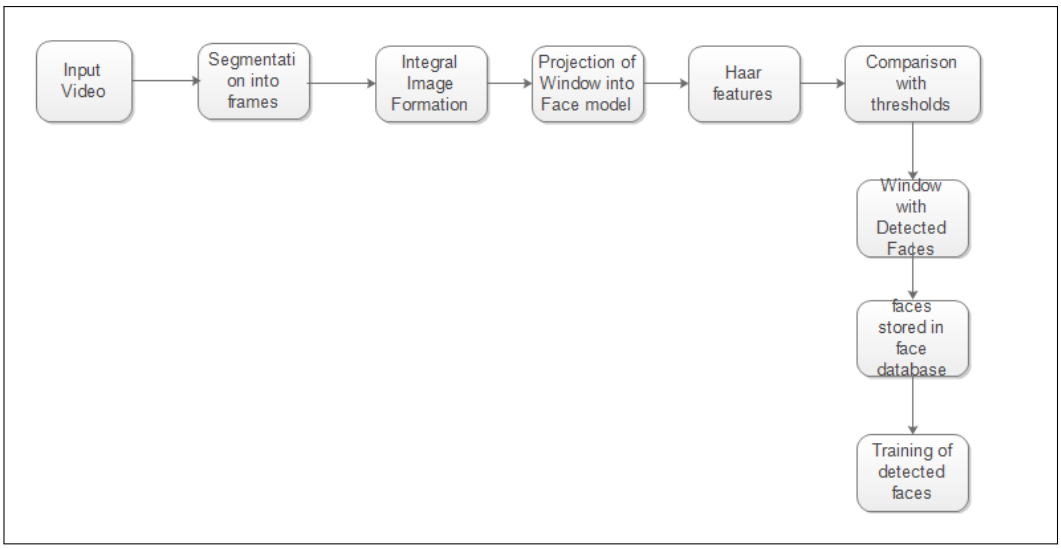


Figure 7.1: Architecture1 diagram

We wish to find Principal Components of the distribution of faces, or the Eigen vectors of the covariance matrix of the set of face images. Each image location contributes to each Eigen vector, so that we can display the Eigen vector as a sort of face. Each face image can be represented exactly in terms of linear combination of the Eigen faces. The number of possible Eigen faces is equal to the number of face image in the training set. The faces can also be approximated by using best Eigen face, those that have the largest Eigen values, and which therefore account for most variance between the set of face images. The primary reason for using fewer Eigen faces is computational efficiency.

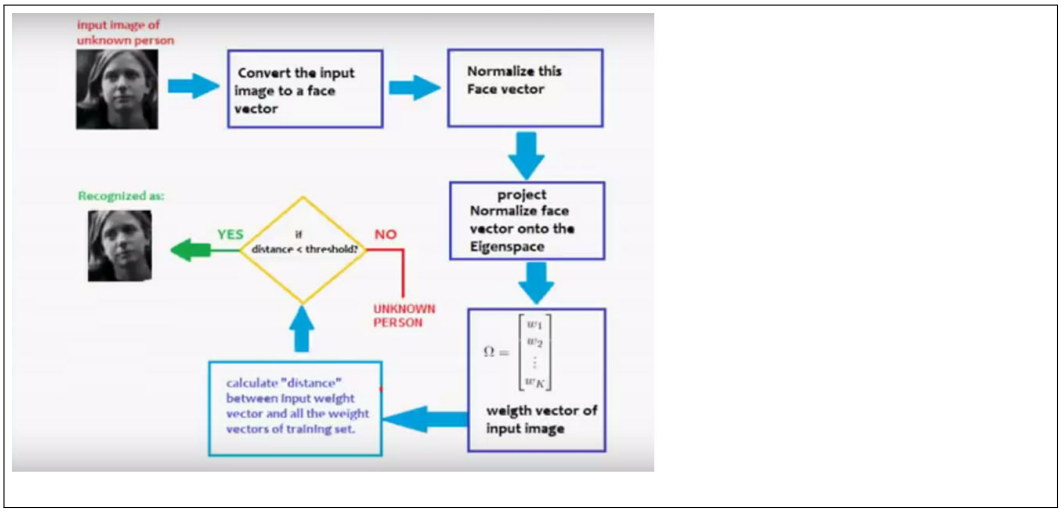


Figure 7.2: Architecture1 diagram

7.3 Component Design

7.3.1 Class Diagram

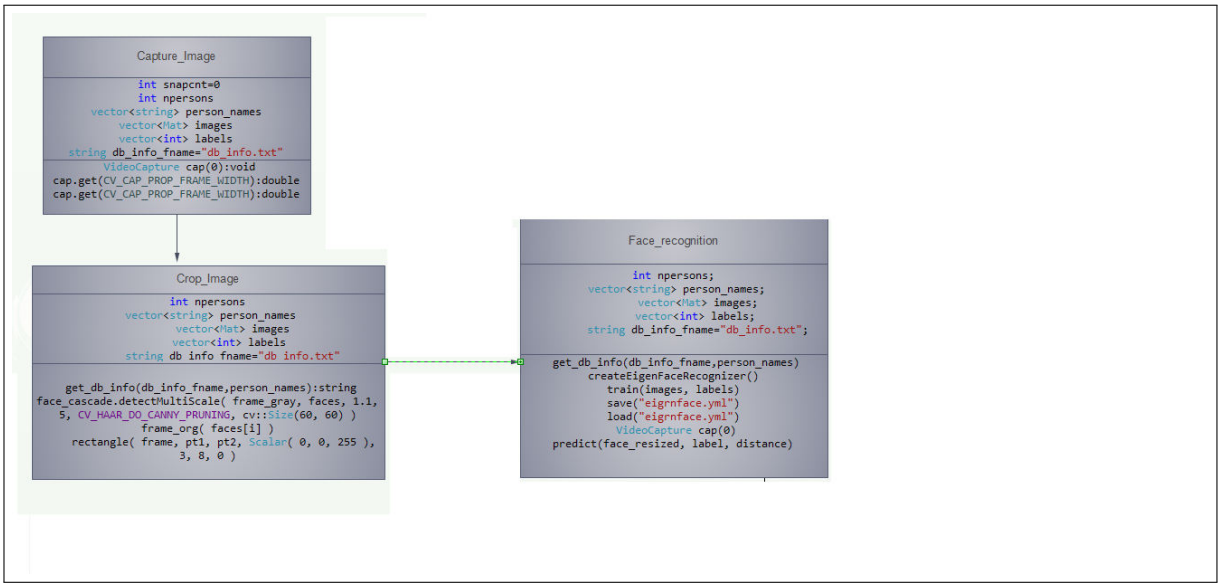


Figure 7.3: Class Diagram

Chapter 8

Project Implementation

8.1 Introduction

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. PCA is a variable reduction procedure and useful when obtained data have some redundancy. This will result into reduction of variables into smaller number of variables which are called Principal Components which will account for the most of the variance in the observed variable.// Problems arise when we wish to perform recognition in a high-dimensional space. Goal of PCA is to reduce the dimensionality of the data by retaining as much as variation possible in our original data set. On the other hand dimensionality reduction implies information loss. The best low-dimensional space can be determined by best principal components.// The major advantage of PCA is using it in eigenface approach which helps in reducing the size of the database for recognition of a test images. The images ar stored as their feature vectors in the database which are found out projecting each and every trained image to the set of Eigen faces obtained. PCA is applied on Eigen face approach to reduce the dimensionality of a large data set.

8.2 Tools and Technologies Used

Tools used: Visual Studio with opencv
OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C, Perl, Ch, and Ruby have been developed to encourage adoption by a wider audience.
Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

8.3 Methodologies/Algorithm Details

Eigenface approach is adequate and efficient method to be used in face recognition due to its simplicity, speed and learning capability. Eigen faces are a set of Eigen vectors used in the Computer Vision problem of human face recognition. They refer to an appearance based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner.
The Eigen faces are Principal Components of a distribution of faces, or equivalently, the Eigen vectors of the covariance matrix of the set of the face images, where an image with N by N pixels is considered a point in N 2 dimensional space. Previous work on face recognition ignored the issue of face stimulus, assuming that predeffined measurement were relevant and sufficient. This suggests that coding and decoding of face images may give information of face images emphasizing the signifficance of features. These features may or may not be related to facial features such as eyes, nose, lips and hairs. We want to extract the relevant information in a face image, encode it efficiently and compare one face encoding with a database of faces encoded similarly. A simple approach to extracting the information content in an image of a face is to somehow capture the variation in a collection of face images.
We wish to find Principal Components of the distribution of faces, or the Eigen vectors of the covariance matrix of the set of face images. Each image location contributes to each Eigen vector, so that we can display the Eigen vector as a sort of face. Each face image can be represented exactly in terms of linear combination of the Eigen faces. The number of possible Eigen faces is equal to the number of face image in the training set. The faces can also be approximated by using best Eigen face, those that have the largest Eigen values, and which therefore account for most variance between the set of face images. The primary reason for using fewer Eigen faces is computational efficiency.

8.3.1 Algorithm 1/Pseudo Code

8.3.2 Algorithm 2/Pseudo Code

Chapter 9

Software Testing

9.1 Type of Testing Used

- Unit testing :Unit testing is a software testing method where individual units of source code are tested after their completion. A unit is the smallest testable part of any system. It can be a procedure, a function, entire class or interface. Unit is the basis of component testing and is often performed as a part of white box testing. Units can be tested in isolation and independently.
  - Benefits of unit testing
  - Early detection of software detects
  - Provides scope for change
  - Simplifies integration testing
  - Valuable documentation
  - Good designing
- Integration testing: Integration testing starts as soon as two r more components are available for integration. Integration testing requires additional information about the communication and interaction between the modules. This involves testing of internal as well as external interfaces. Integration testing can be top down or bottom up.
- System Testing: When testing is performed on complete integrated system capable of performing all the intended operations, the testing is called system testing. System testing checks for all intended features of the system.. The system is evaluated against all requirement specifications. The system is tested against all functional as well as nonfunctional requirements of the system.

9.2 Test Cases and Test Results

The various test cases are given in the following table:

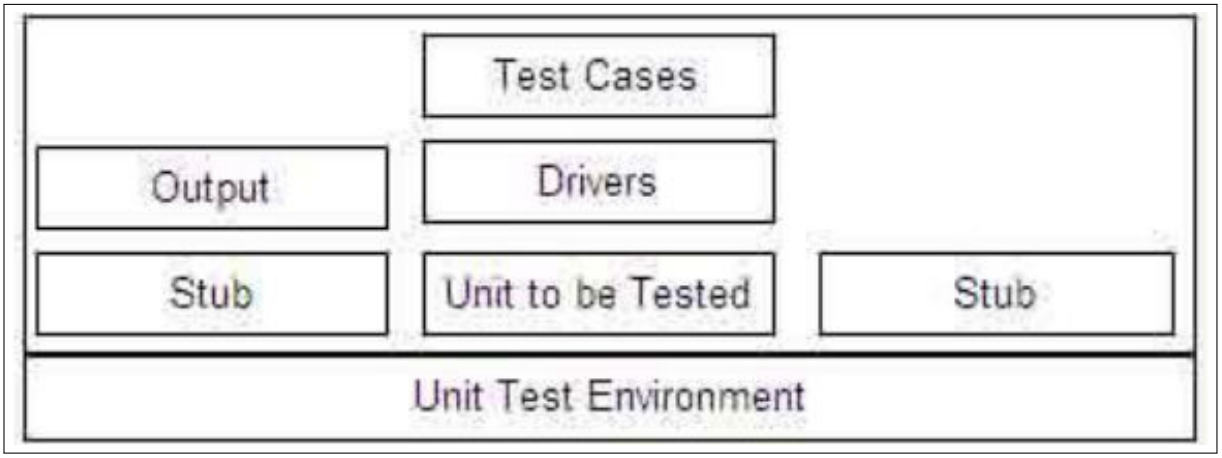


Figure 9.1: Unit testing

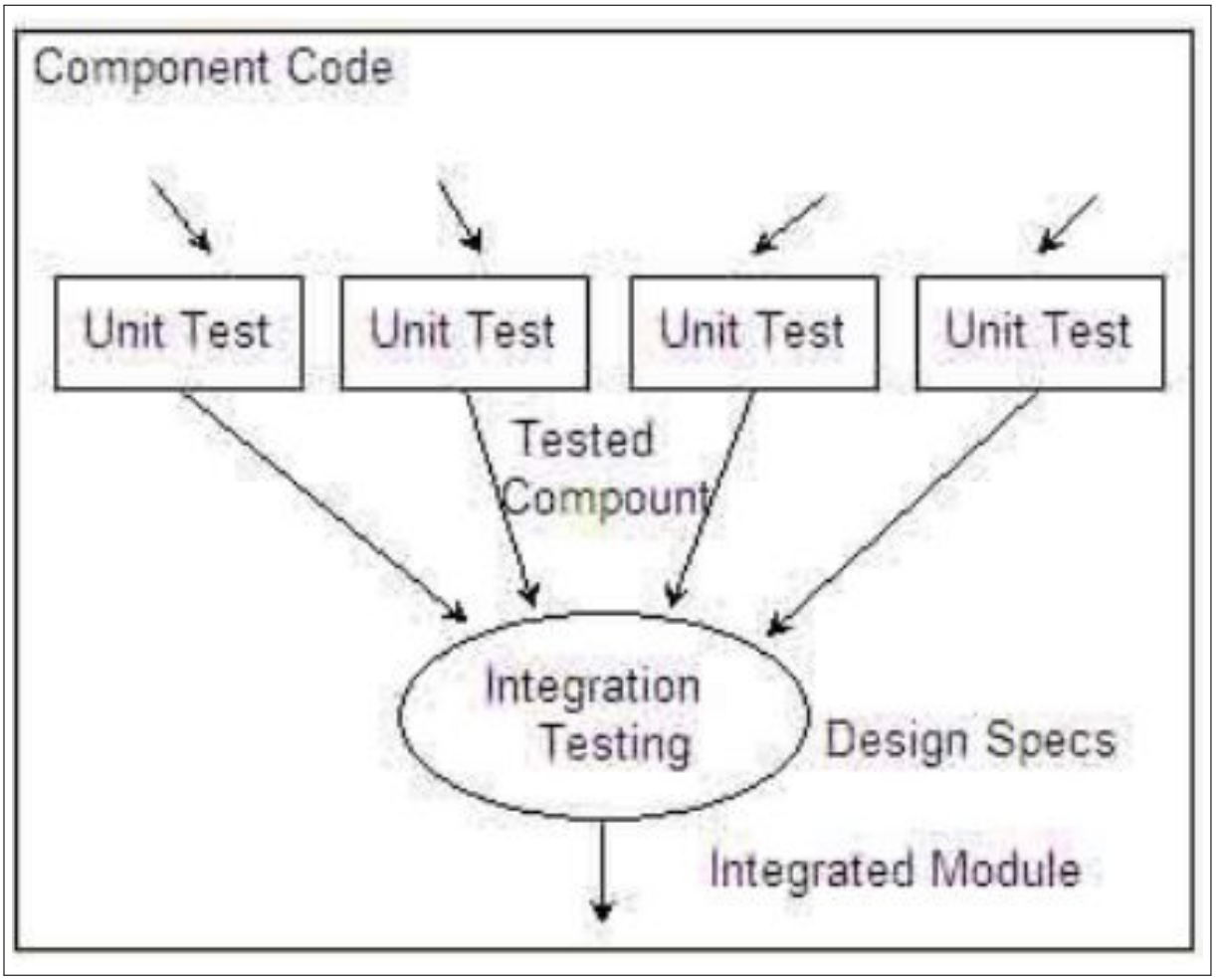


Figure 9.2: Integration testing

STEPS	TEST PARAMETER	TEST CONDITION	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Login	Valid username and password	Successful Login	Successful Login	Pass
		Invalid username and password	Incorrect username or password	Incorrect username or password	Pass
2	Camera	Accessible	Should open the input video window and ready to capture the image	Opens the input video window and ready to capture the image	Pass
		Inaccessible		Error in Camera	Fail
3	Presence of Face	Face Present	Should detect the face, crop the face correctly and save in database	Detects the face, crops the face and saves in database	Pass
		Face Absent	Should capture the image , cannot detect the face ,so no face saved	Capture the image does not detect the face, saves error image	Fail
4	Training of Database	Face Present	Should crop the face correctly and save in database	Crops the face correctly and saves in database	Pass
		Face Absent	No faces detected	Error in training	Fail
5	Face Recognition	If face matched	Should display the details of matched face	Displays the details of matched face	Pass
		If face is unmatched	Should display 'No previous record found'	Displays 'No previous record found'	Pass

Figure 9.3: Testing table

Chapter 10

Results

10.1 Screen shots

Outputs and Snap shots of the application are given below:

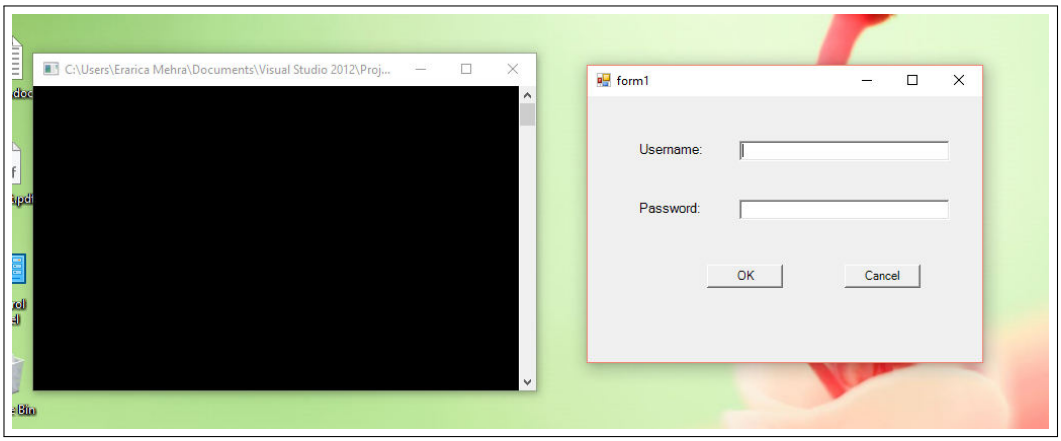


Figure 10.1: Use case diagram

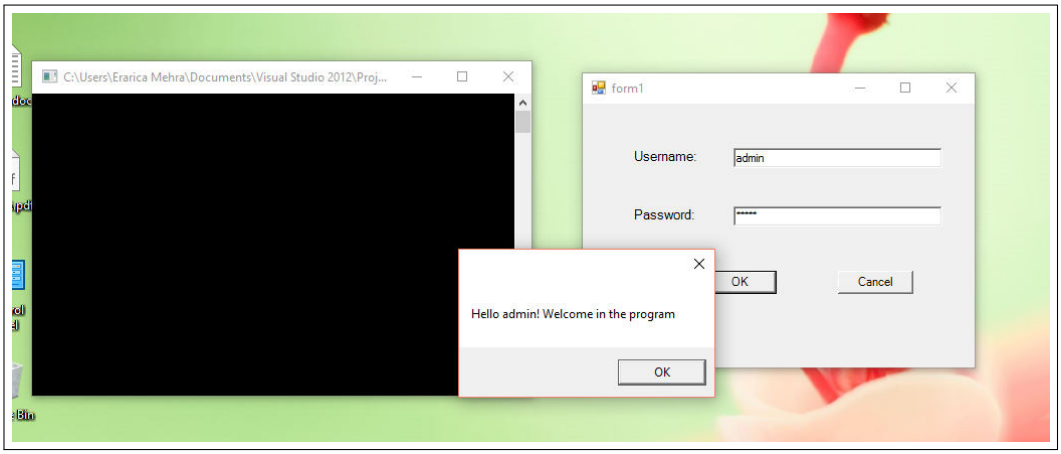


Figure 10.2: Use case diagram

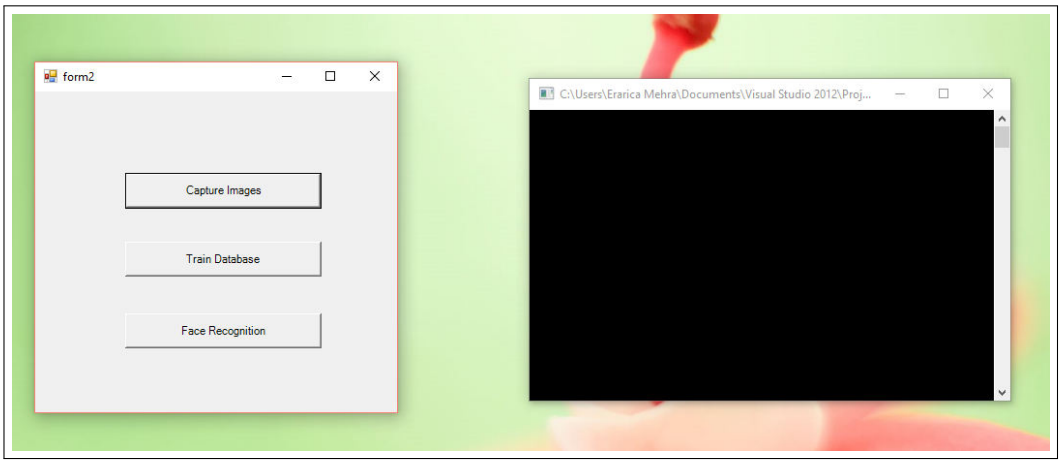


Figure 10.3: Use case diagram

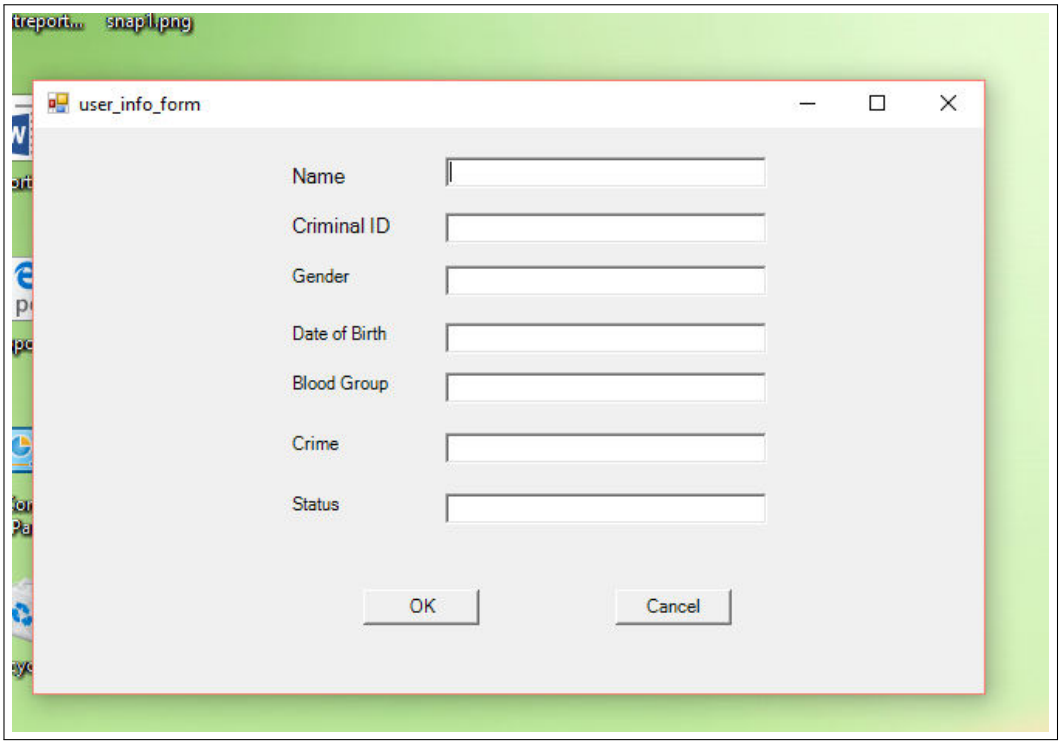


Figure 10.4: Use case diagram

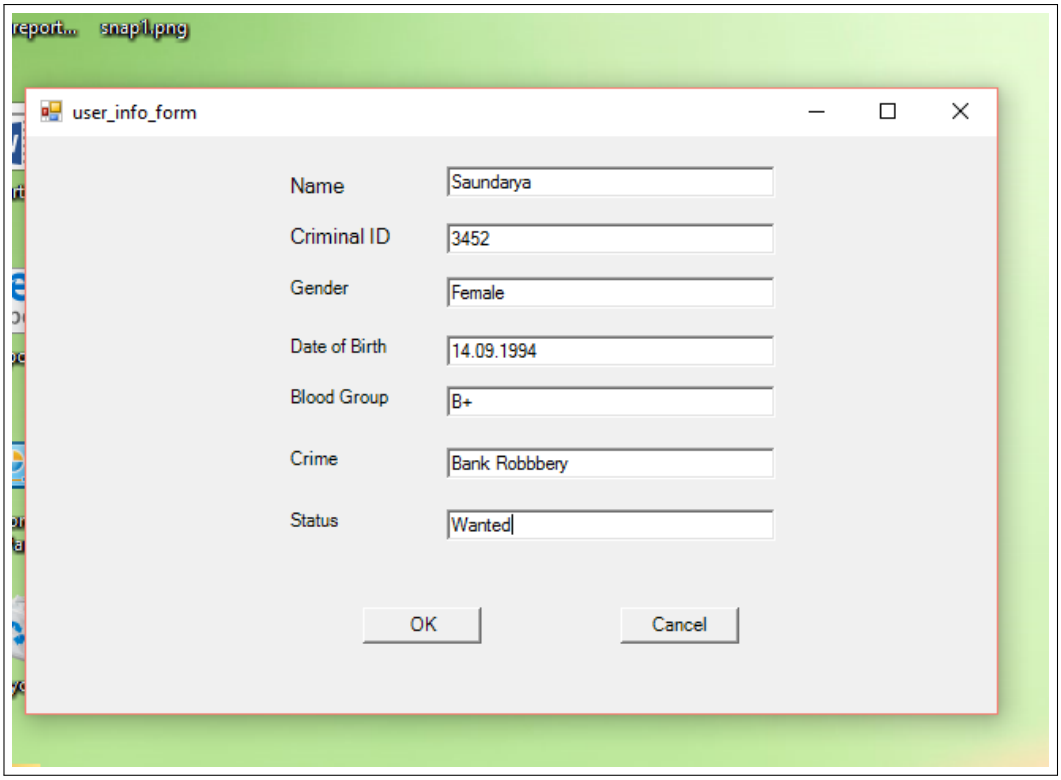


Figure 10.5: Use case diagram

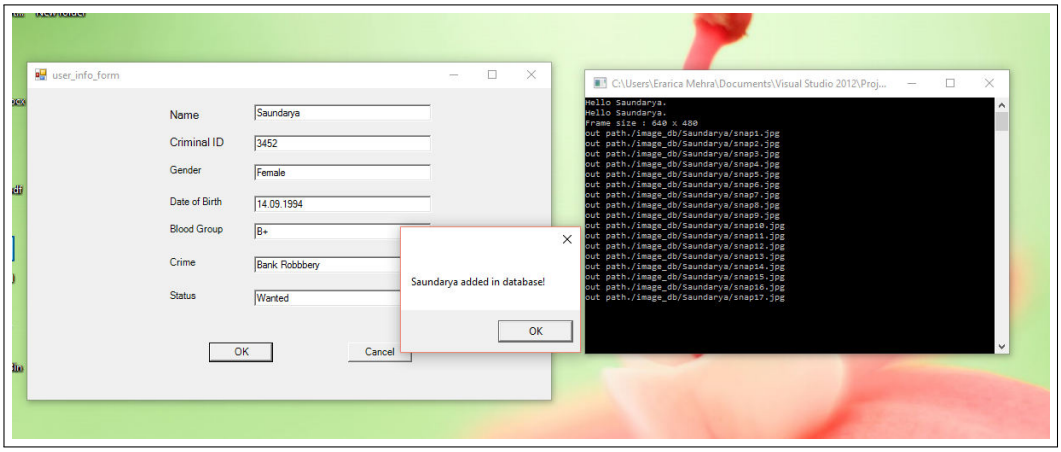


Figure 10.6: Use case diagram

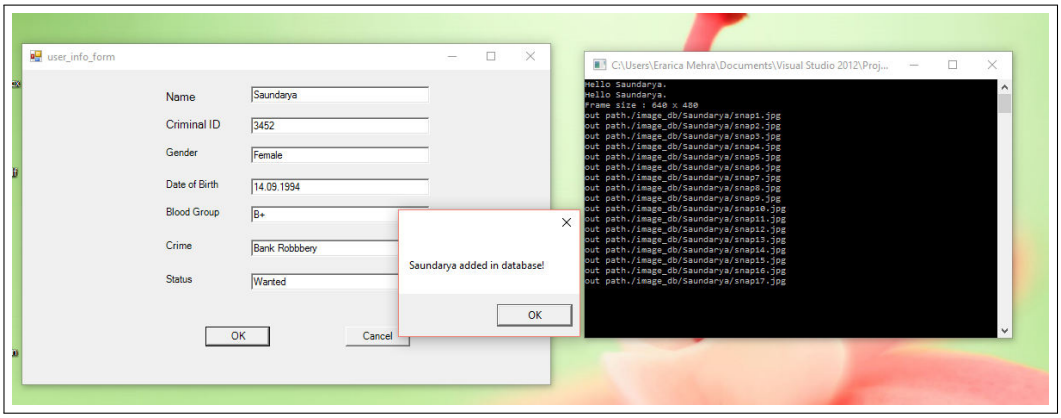


Figure 10.7: Use case diagram

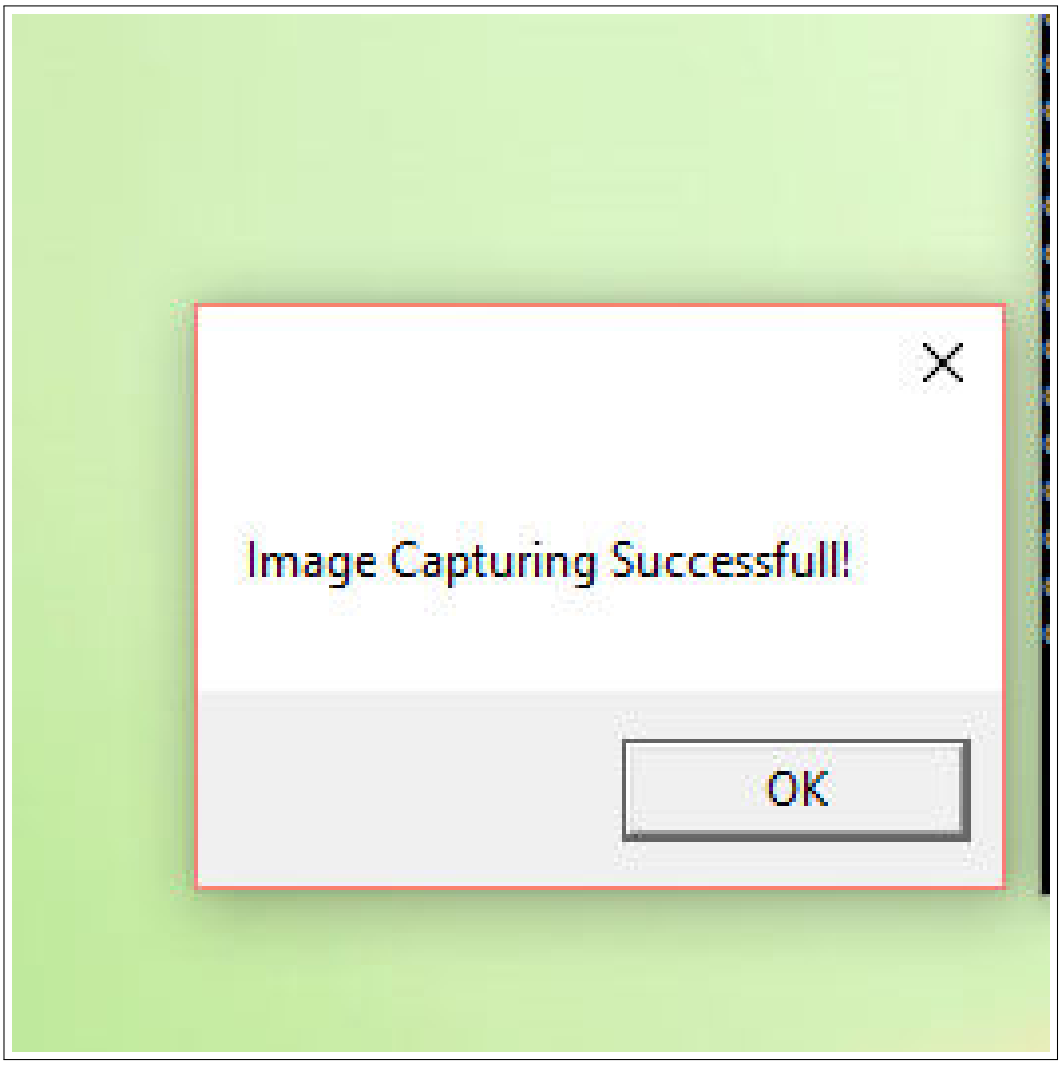


Figure 10.8: Use case diagram

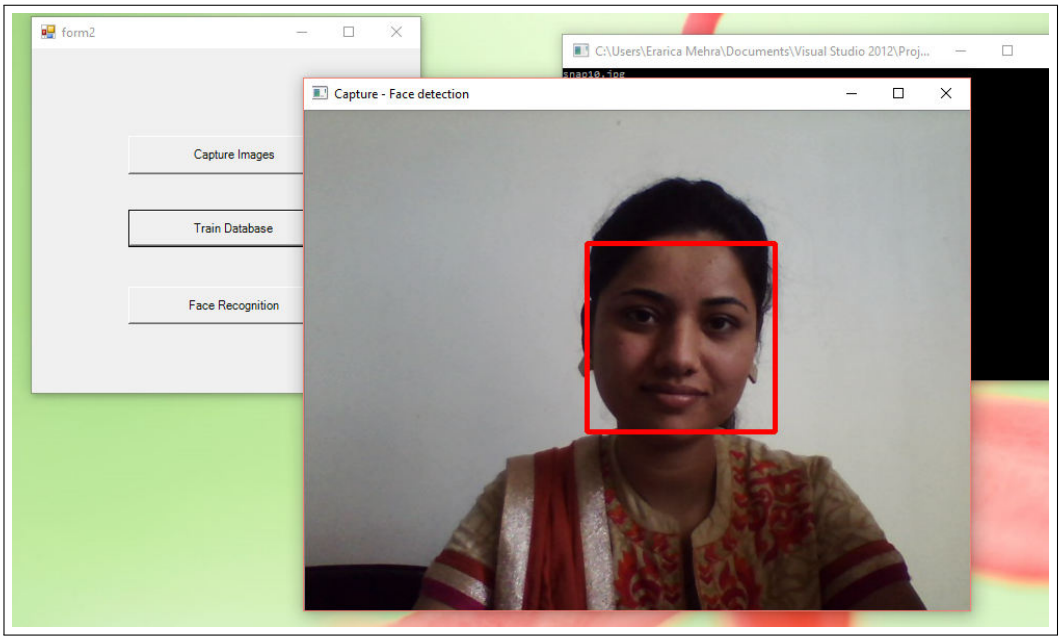


Figure 10.9: Use case diagram

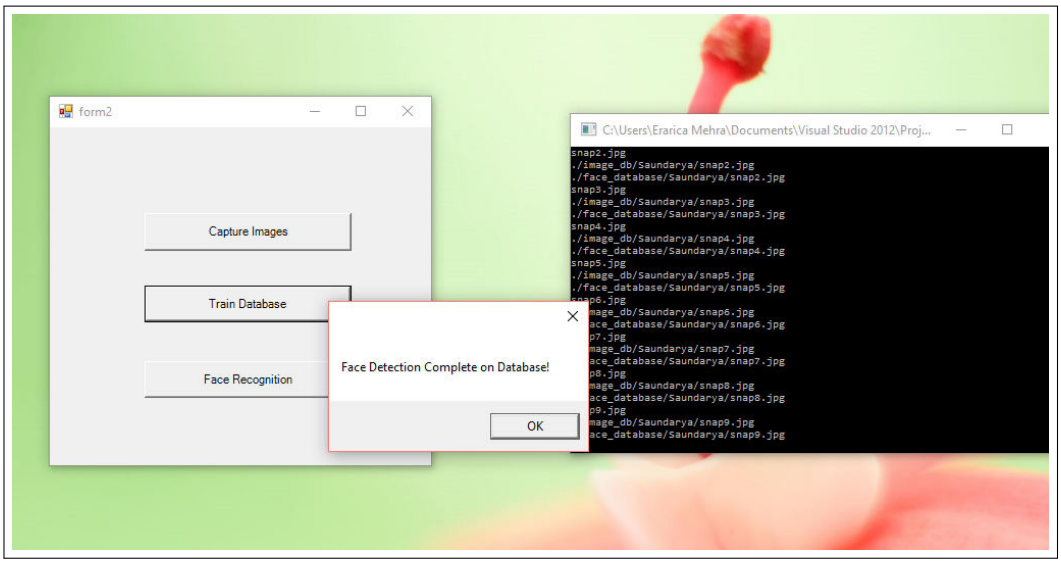


Figure 10.10: Use case diagram

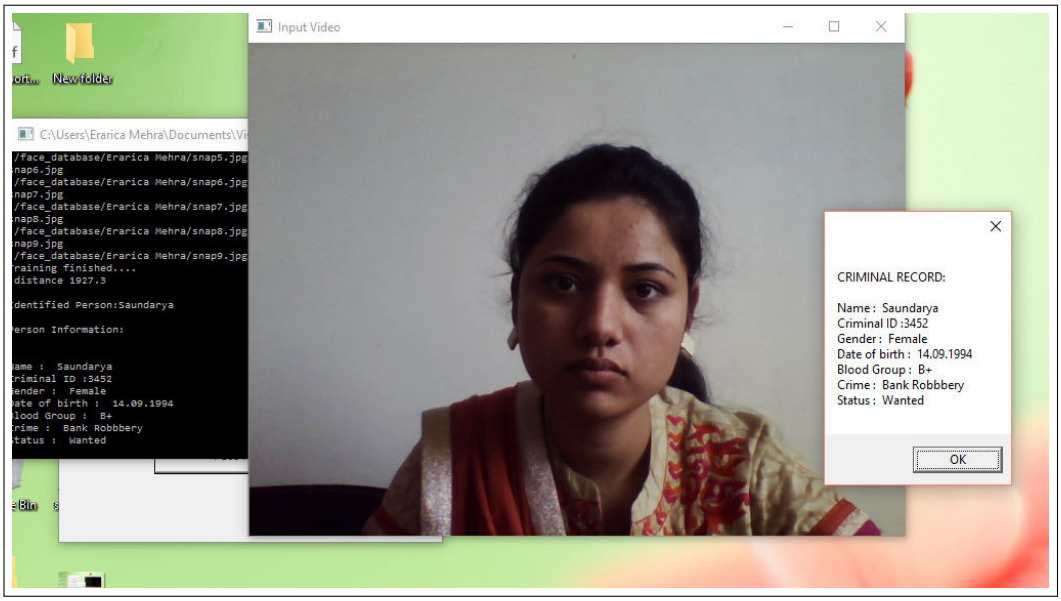


Figure 10.11: Use case diagram

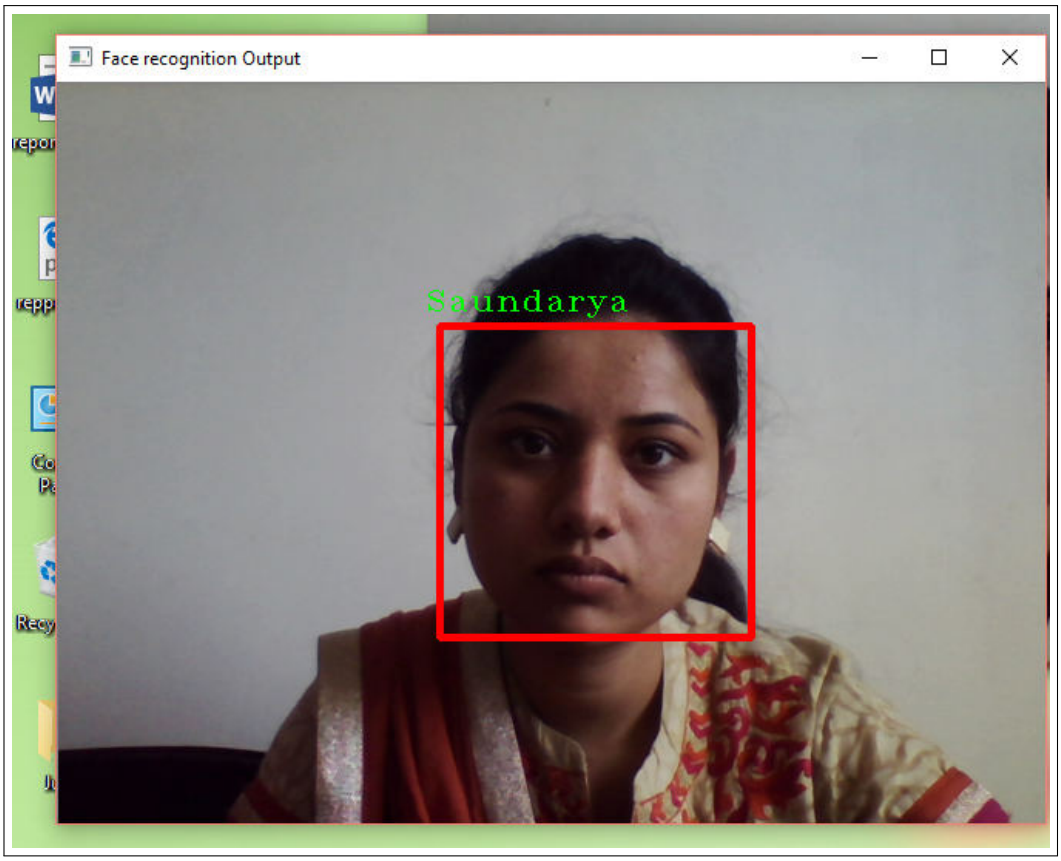


Figure 10.12: Use case diagram

Chapter 11

Deployment and Maintenance

11.1 Installation and un-installation

Following are the steps to configure and install opencv

- Step1:Download windows version of opencv from www.opencv.org
- Step2:Extract into a folderC:
- Step3:Set system Environment Variable
  Right Click On My Computer and select properties -Advanced System Settings - Environment Variable
  Click New button in User Variable Window and set following:
  variable name: OPENCV _ DIR
  value:C:8612 (You change path where u kept opencv)
  Then Edit Path in System Variable ;
- Step4: Configure opencv with Visual studio Project
  Select :File - Project - Then Press Next- Select Empty Project
  Right click on project and go to properties. Select Release in configuration, then Select - C/C++.

Chapter 12

Conclusion and Future Scope

12.1 Conclusion and Future Scope

In this project we can successfully conclude new advanced services that promise that police ocers can access the information they need to carry out their tasks more easily. To aid the police service we have come up with an application that takes image of a suspect as input and compares the image to all the images from the database and displays the details of his/her criminal record such as name, gender, address, crime, etc. It is adequate and efficient method to be used in face recognition due to its simplicity, speed and learning capability. Eigen faces are a set of Eigen vectors used in the Computer Vision problem of human face recognition. They refer to an appearance based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner.
Client server Application: The project can be extended as a client server application where the client can send images to the server and receive the details after successful face recognition. Database Management: The admin can access the information from the database by using appropriate queries.

Chapter 13

References

13.1 References

- Mobile Application for Police Officers By Harry Bouwman, Timber Haaker, Henny de Vos
- Design and Implementation of 3G mobile police system by Baocai Zhong; Lu Niu; Hanbin Chen
- Implementation and Benchmarking of Perceptual Image Hash Functions by Christoph Zauner
- Perceptual Image Hashing via Feature Points: Performance Evaluation And Trade Offs by Vishal Monga, Brian L. Evans
- Perceptual Image Hashing by Azhar Hadmi, William Puech
- Face Recognition Using PCA and Eigen Face Approach by Abhishek Singh and Saurabh Kumar
- An Efficient Method For Face Recognition Using Principal Component Analysis(PCA), Gunjan Dashor, Dr V. Cyril Raj
- J.Zobel and A.Moffat, "' Inverted files for text search engines,"' ACM Computing Surveys, 2006
- "International Conference on Computer Vision" by N.Kumar, A.C. Berg, P.N. Belhumeeur and S. K. Naayar
- 'Robust Real-Time Face Detection' PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA

Annexure A

Laboratory assignments on Project Analysis of Algorithmic Design

Assignment 1

Title: Develop the problem under consideration and justify feasibility using concepts of Knowledge canvas and IDEA Matrix.

Assessment:
Developing problem using knowledge canvas and IDEA Matrix.

CANVAS:
1. It is used to improve the communication in the team.
2. It is strategic management and lean startup template for develop new business models.
3. It consists of 12-elements that gives an overview of the project.

Project:
Project define the problem statement as FACE RECOGNITION USING VIOLA JONES AND EIGENFACE ALGORITHM

Trust Model.
1. Our project is unique as it solves the problem of identification of suspects using an efficient face recognition algorithm.
2. It has specific end result i.e. provides criminal record of the suspect if present in the database

A] Goal setting Element:

Purpose:
Its purpose is to provide an efficient and accurate algorithm for face recognition to speed up investigation process by extracting details of criminals, suspects or missing persons from the database. The algorithm used here is Viola Jones for face detection and Eigenface for face recognition.

Scope:
A] The project takes input in the form of images captured in real time
B] Detects faces from the captured images
C] Crops and saves the images during training
D] Performs face recognition
E] Displays details of matched persons

Success criteria:
The goal to be achieved is successful matching of the image and displaying the details

Outcomes:
Successful matching of image
Successful display of criminal record

B] Timeframes:
Milestones: It is phase from finding partners to upto detailed project planning

Sr No.	Task Name	Start	Finish
1	Information Gathering	3/7/2015	7/7/2015
2	Requirement analysis	10/7/2015	22/7/2015
3	Literature Survey	24/7/2015	31/7/2015
4	Problem Statement Definition	7/8/2015	21/8/2015
5	Define Specification	24/8/2015	4/9/2015
6	Project Planning	7/9/2015	25/9/2015
7	Detailed Design	28/9/2015	23/10/2015
8	Risk Analysis	25/10/2015	20/11/2015
9	Software Development	1/12/2015	29/1/2016
10	Testing and QA	1/2/2016	19/2/2016
11	Final Delivery	24/2/2016	18/3/2016

Actions:
The link to outcomes of solving all good reputation problem is provided by necessary action as create the plan, specify the need, build the workspace, the plan, develop backend, implement the system, and test the plan, final deliver the product.

C] People:

Team:
The participants for our project are: 3-Team members

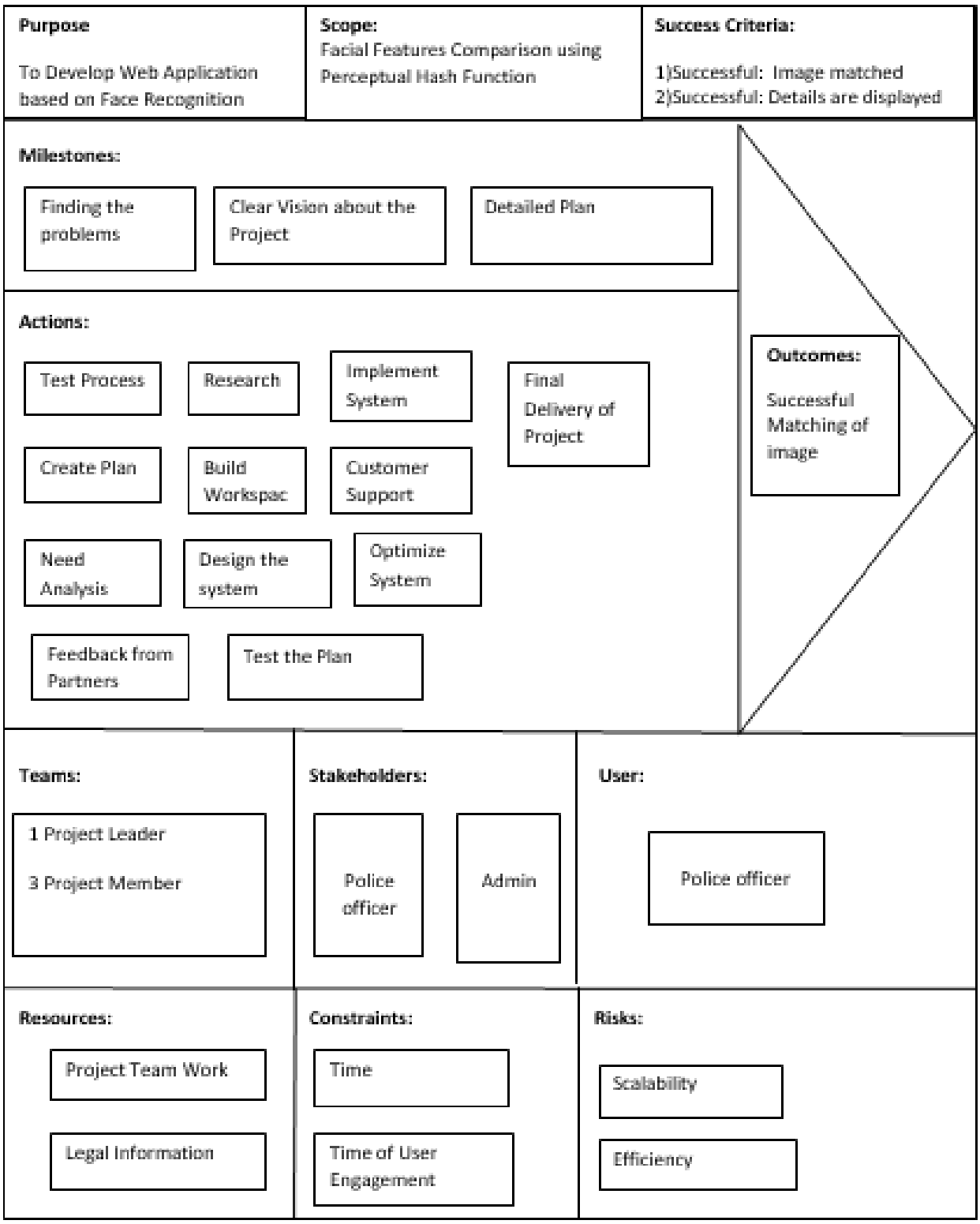
Stakeholder
The people get affected by the project are
A] Police officer: is the user of the project
.
User
The benefit of end result is to the User or police officer

Constraints
These are basically the limitation set up by the environment as time as well as Engagement of user.

1. Quality of camera and resolution are major constraints.
2. If the distance of the person from the camera is more, image captured will not be trained properly. Images should be captured from same resolution camera only. Camera should not be changed. The database consists of images only.
3. The extracted features are stored in a separate file. Computation on the extracted features is not possible.

Risk
The condition that can have negative impact on project outcomes are scalability and efficiency.
CANVAS REPRSENTATION:
Project name: Face recognition using Viola Jones and Eigenface algorithm

Idea matrix:
_ is nothing but making the idea multiply.
_ Short time force your brain to think of an idea in every cell.
_ Its aim is to generate the more ideas.


Purpose
To Develop Web Application based on Face Recognition
Scope:
Facial Features Comparison using Perceptual Hash Function
Success Criteria:
1)Successful: Image matched
2)Successful: Details are displayed
Milestones:
Finding the problems
Clear Vision about the Project
Detailed Plan
Actions:
Test Process
Research
Implement System
Final Delivery of Project
Create Plan
Build Workspac
Customer Support
Need Analysis
Design the system
Optimize System
Feedback from Partners
Test the Plan
Outcomes:
Successful Matching of image
Teams:
1 Project Leader
3 Project Member
Stakeholders:
Police officer
Admin
User:
Police officer
Resources:
Project Team Work
Legal Information
Constraints:
Time
Time of User Engagement
Risks:
Scalability
Efficiency

Assignment No 2

1 Title

Project workstation selection,installation along with setup and installation report preparations.

2 Workstation Selection

2.1 Visual Studio

Introducton:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studiouses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level. Visual Studio supports dffierent programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specfic service exists. Built-in languages include C, C++ and C++/CLI,VB.NET and support for other languagessuch as Python, Ruby,etc. It also supports XML/XSLT, HTML/XHTML,JavaScript and CSS.

2.2 Opencv

Introduction:

opencv OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel's research center in Nizhny Novgorod (Russia),later supported by Willow Garage and now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license. OpenCV is written in C++ and its primary interface is in C++,but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C, Perl, Ch, and Ruby have been developed to encourage adoption by a wider audience.

Installation

1. How to Install Eclipse 4.2 (JUNO) for Java

1.1 For Windows

Step1: Download windows version of opencv from www.opencv.org

Step2: Extract into a folder C:\
opencv 3.0.0

Step3: Set system Environment Variable
Right Click On My Computer and select properties >> Advanced System Settings >> Environment Variable
opencv envirenment1

opencv envirenment2

Click New button in User Variable Window and set following variable name: OPENCV_DIR value:C:\opencv\build\x86\vc12 (You change path where u kept opencv…)
opencv envirenment3

Then Edit Path in System Variable
;%OPENCV_DIR%\bin
opencv envirenment4

Step4: Configure opencv with Visual studio Project
Select :File >> Project
Then Press Next
>> Select Empty Project
Right click on project and go to properties
Select Release in configuration
then Select >>C/C++
Additional include directories : $(OPENCV_DIR)\..\..\include
opencv 3.0.0 installation3

then Select >>Linker >> General
Additional include directories: $(OPENCV_DIR)\lib

Linker >> Input
Edit Additional Dependencies
opencv_ts300d.lib opencv_world300d.lib

and also follow above steps for Debug Configurations

Assignment No 3

Title:

Programming of the project functions, interfaces as per 1st Term term-work submission using corrective actions recommended in Term-I assessment of Term-work.

Programming of UI

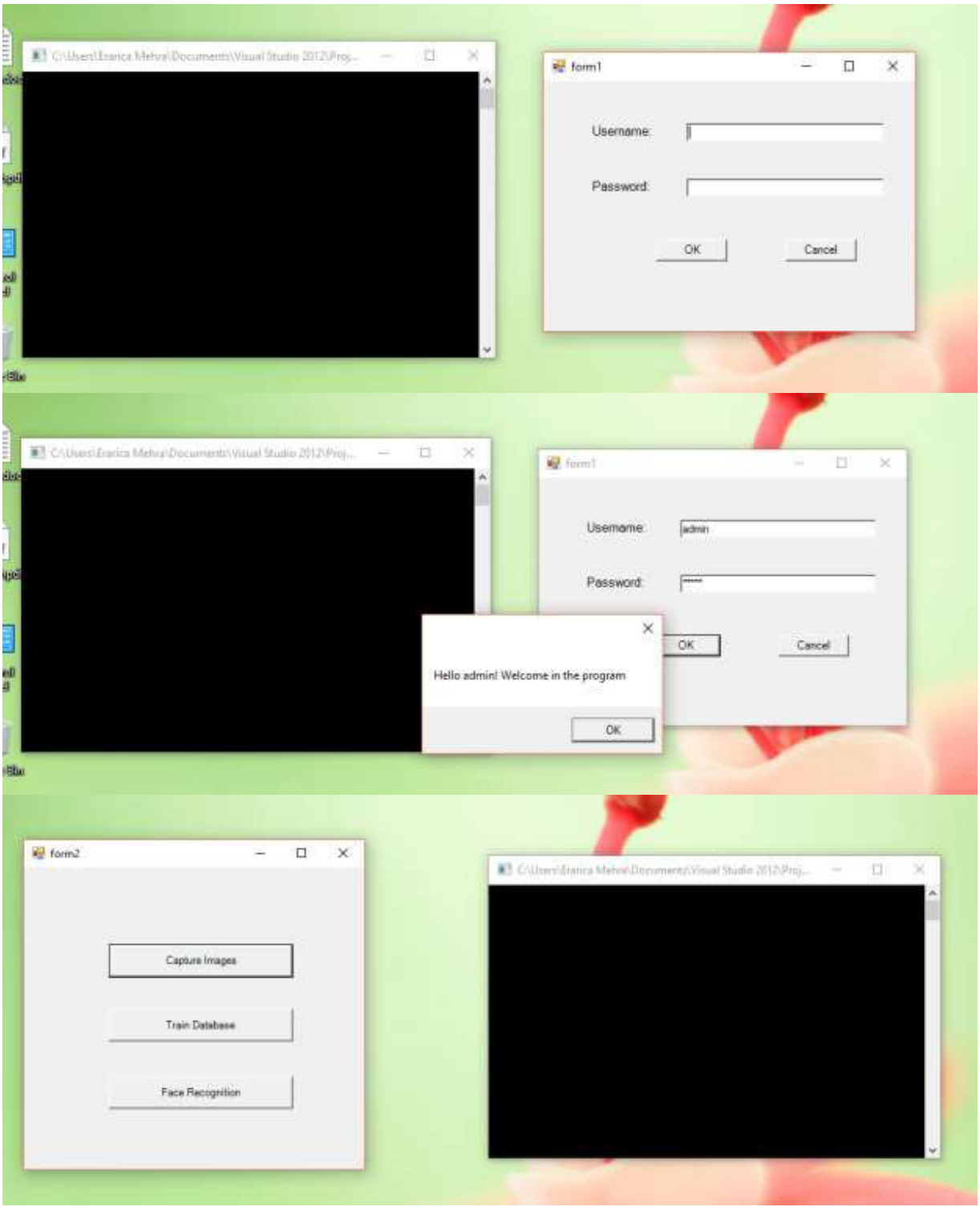
```
// ConsoleApplication2.cpp : main project file.

#include "stdafx.h"

#include "form1.h"

//#include"user_info_form.h";
#include "form2.h"
//#include "user_info_form.h"
using namespace ConsoleApplication2;  // put namespace of your project
here
//using namespace System;
[STAThreadAttribute]
int main(array<System::String ^> ^args)
{
    //Console::WriteLine(L"Hello World");
      Application::Run(gcnew form1());
      Application::Run(gcnew form2());

    return 0;
}


// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#pragma once

// TODO: reference additional headers your program requires here

/*header files*/


 void capture_images()
 {

 }
```

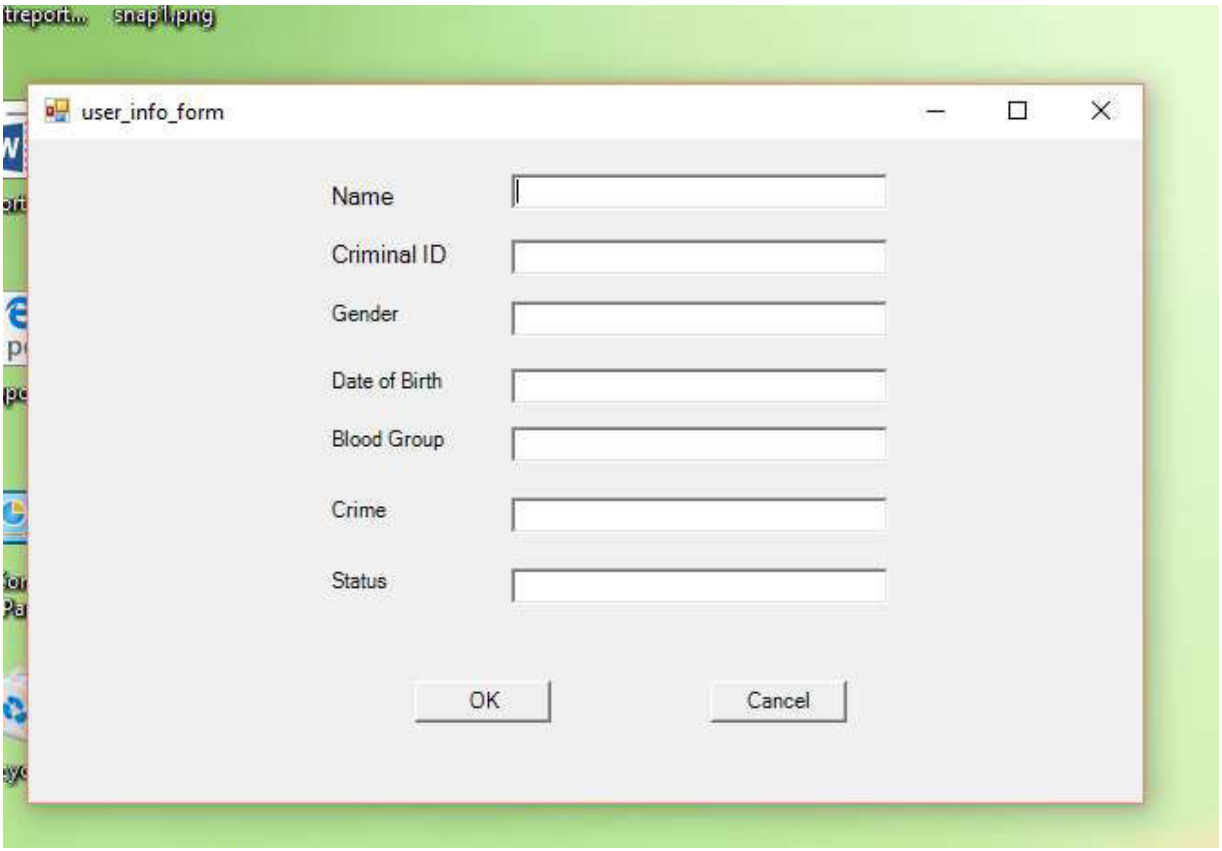
```
 void crop_faces_for_training()
 {
 }

 void face_recognition()
 {

            bool bSuccess = cap.read(frame); // read a new frame from
video
         if (!bSuccess) //if not success, break loop
        {
             cout << "Cannot read a frame from video stream" << endl;
             break;
        }



   //-- 3. Apply the classifier to the frame
       //if( !frame.empty() )
       //{ detectAndDisplay( frame,model1,person_names);
        // }
      // else
      // {
      //       printf(" --(!) No captured frame -- Break!");
       //  }
      //   waitKey(0);
          //imshow("Input Video", frame); //show the frame in "MyVideo"
window
      char c = waitKey(33);


        if (c == 'c')
       {

  Mat frame_gray;
  CvPoint pt1, pt2;
  cvtColor( frame, frame_gray, CV_BGR2GRAY );
  equalizeHist( frame_gray, frame_gray );

  //-- Detect faces
         faces.clear();
  face_cascade.detectMultiScale( frame_gray, faces, 1.1, 5,
CV_HAAR_DO_CANNY_PRUNING, cv::Size(90, 90));

  string Pname = "";
  for( size_t i = 0; i < faces.size(); i++ )
  {


    // Find the dimensions of the face,and scale it if necessary
            pt1.x = faces[i].x;
```

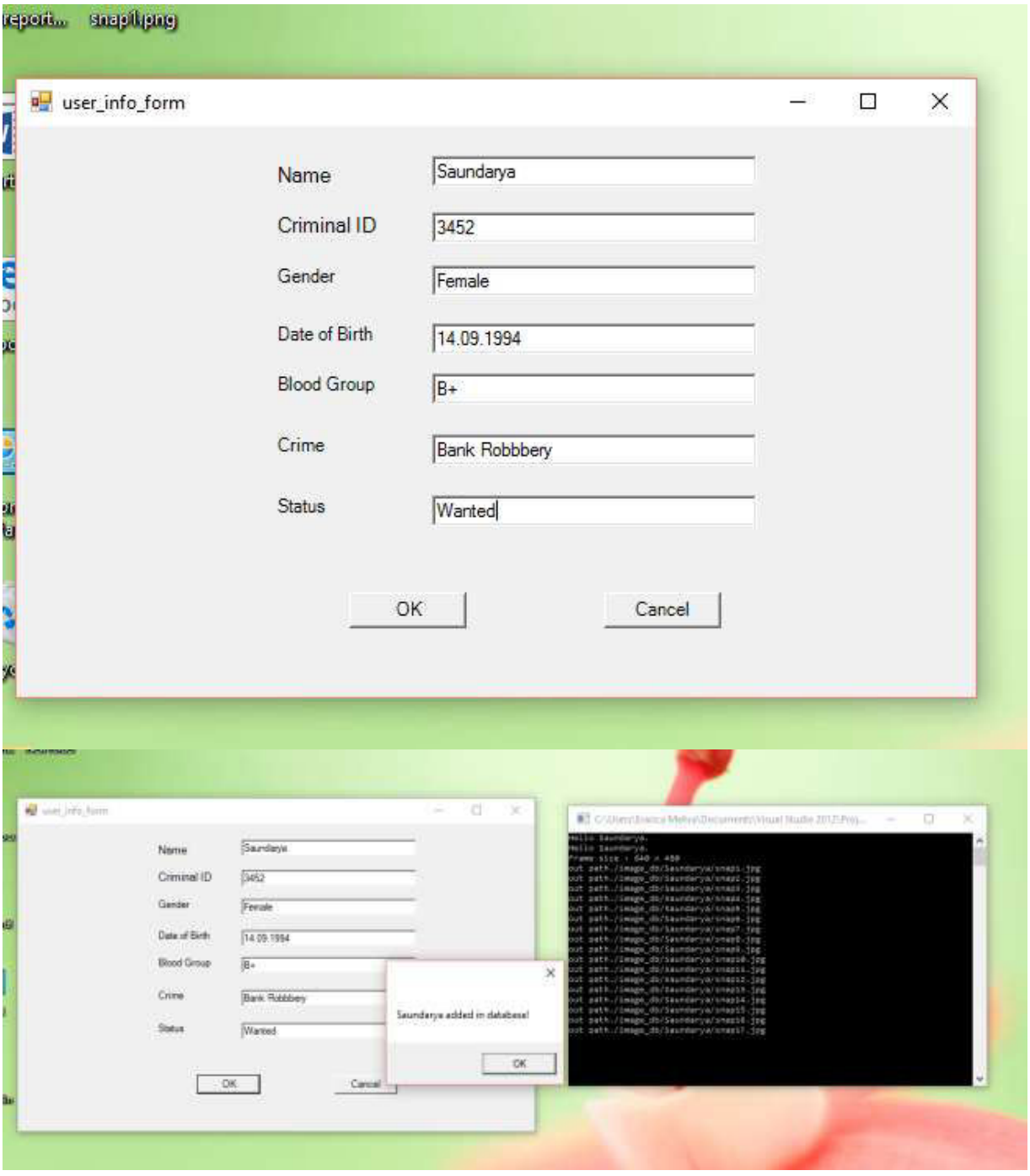
```
            pt2.x = (faces[i].x + faces[i].width);
            pt1.y = faces[i].y;
            pt2.y = (faces[i].y + faces[i].height);

            // Draw the rectangle in the input image
           rectangle( frame, pt1, pt2, Scalar( 0, 0, 255 ), 3, 8, 0 );
    Mat faceROI = frame_gray( faces[i] );
     Mat face_resized;
cv::resize(faceROI, face_resized, cv::Size(img_width, img_height),
1.0, 1.0, INTER_CUBIC);
 //recognizing what faces detected
                      int label = -1; double distance = 50000;
                      model1->predict(face_resized, label, distance);

                      cout << " distance " << distance << endl;

                      //drawing green rectagle in recognize face
                //    rectangle(frame, faces[i], CV_RGB(0, 255, 0), 1);
                      string text = "Detected";

                      if (distance<2800)
                      {

                            string msgtxt1="";
                  Pname=person_names[label];
                  cout<<"\nIdentified Person:"<<Pname<<endl;
                  cout<<"\nPerson Information:"<<endl;
                  string fname="info_files/"+Pname+".txt";
                   string STRING;
      ifstream infile;
      infile.open(fname);
       if (infile.is_open()){
         while(!infile.eof()) // To get you all the lines.
         {
               getline(infile,STRING); // Saves the line in STRING.
                  msgtxt1=msgtxt1+"\n"+ STRING;
               cout<<"\n"<<STRING; // Prints our STRING.
         }
       }
      infile.close();
      //system ("pause");

      System::String^ msgtxt = gcnew System::String(msgtxt1.c_str());
MessageBox::Show("CRIMINAL RECORD:"+ msgtxt);

                      }else
                      {Pname="Unknown";

MessageBox::Show("No previous record found !");

                      }
                      string text1 = format("Person is  = %s", Pname);
```

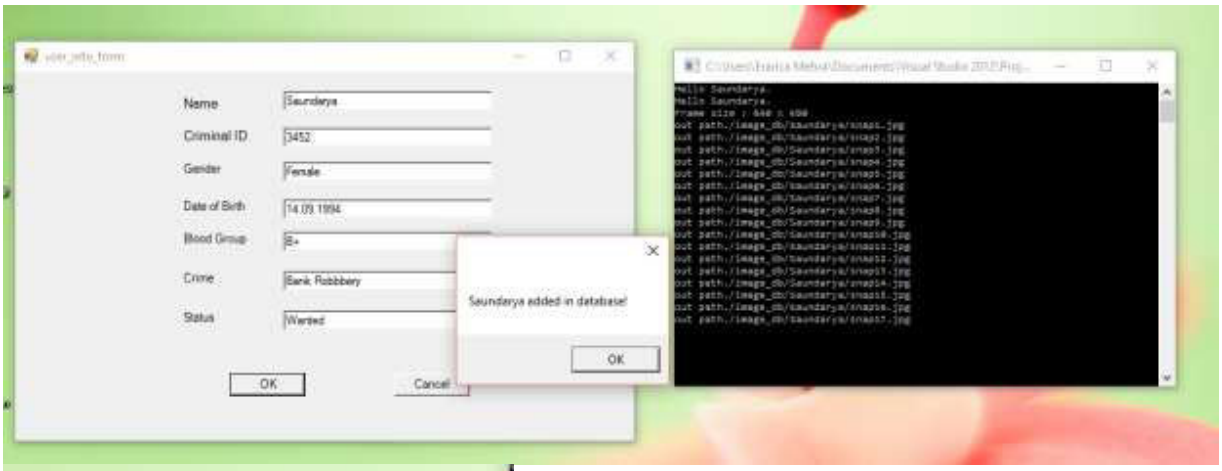
```
                int pos_x = max(faces[i].tl().x - 10, 0);
                    int pos_y = max(faces[i].tl().y - 10, 0);
                    //putText(frame, text, Point(pos_x, pos_y),
FONT_HERSHEY_COMPLEX_SMALL, 1.0, CV_RGB(0, 255, 0), 1.0);
                    putText(frame, Pname, cv::Point(pos_x, pos_y),
FONT_HERSHEY_COMPLEX_SMALL, 1.0, CV_RGB(0, 255, 0), 1.0);


  }


  //-- Show what you got
 imshow( "Face recognition Output", frame );

           }
           else if (c==27)
           {
break;
           }
           else
           {imshow("Input Video", frame); //show the frame in
"MyVideo" window
           }
       //if( (char)c == 27 ) {  return 0;}
    }
      destroyAllWindows();

 }
```

Screenshots of output


form1
Username:
Password:
OK
Cancel
admin
Hello admin! Welcome in the program
form2
Capture Images
Train Database
Face Recognition


user_info_form
Name
Criminal ID
Gender
Date of Birth
Blood Group
Crime
Status
OK
Cancel


report...
snap1.png
user_info_form
Name
Saundarya
Criminal ID
3452
Gender
Female
Date of Birth
14.09.1994
Blood Group
B+
Crime
Bank Robbbery
Status
Wanted
OK
Cancel
Saundarya added in database!
OK


Name
Saundarya
Criminal ID
3452
Gender
Female
Date of Birth
14.09.1994
Blood Group
B+
Crime
Bank Robbbery
Status
Wanted
OK
Cancel
Saundarya added in database!
OK

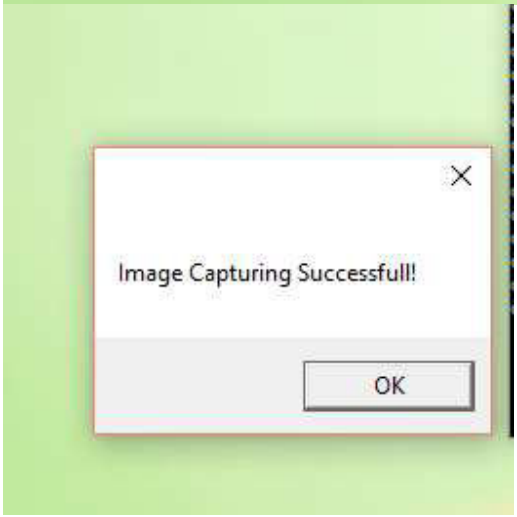
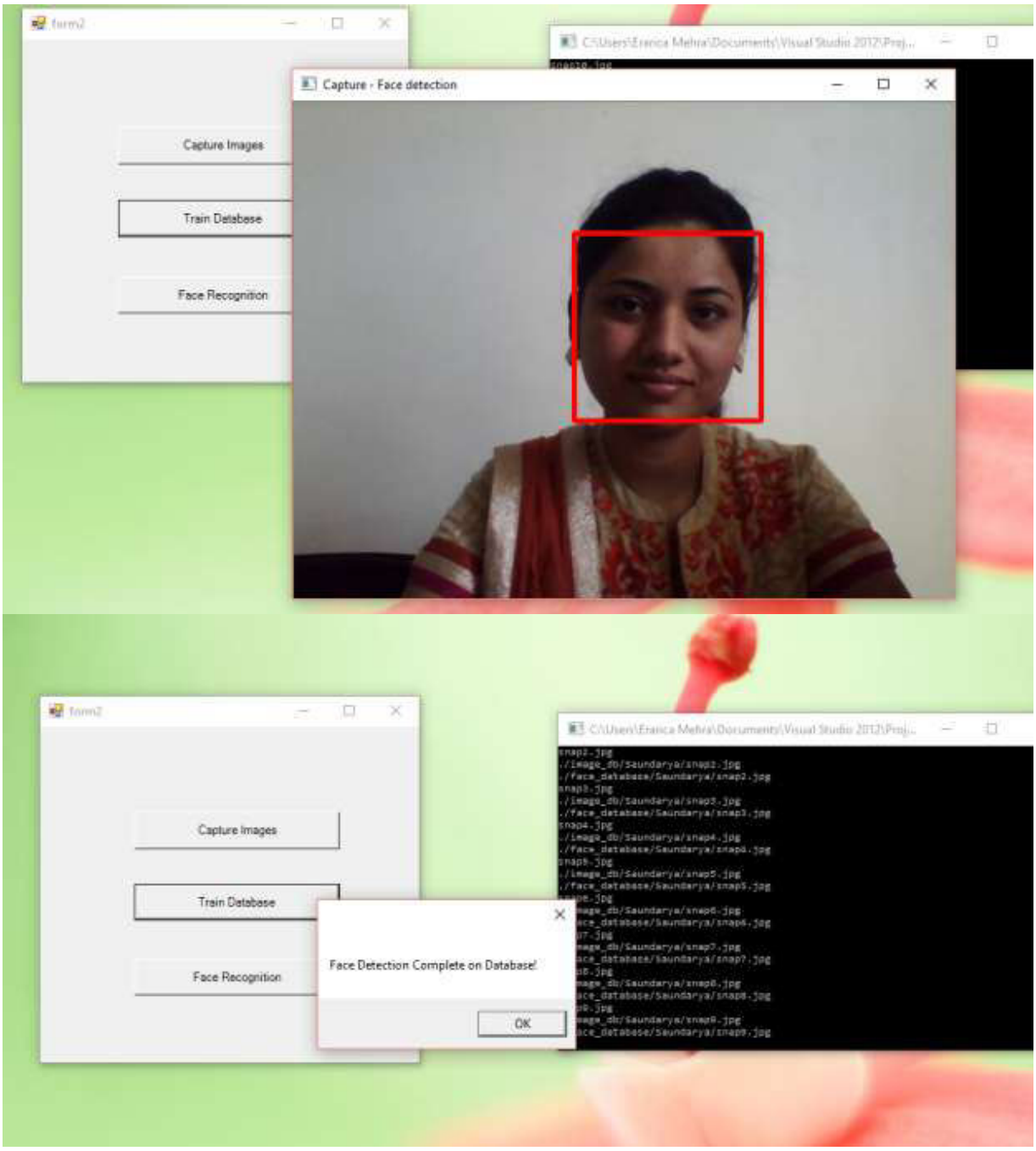
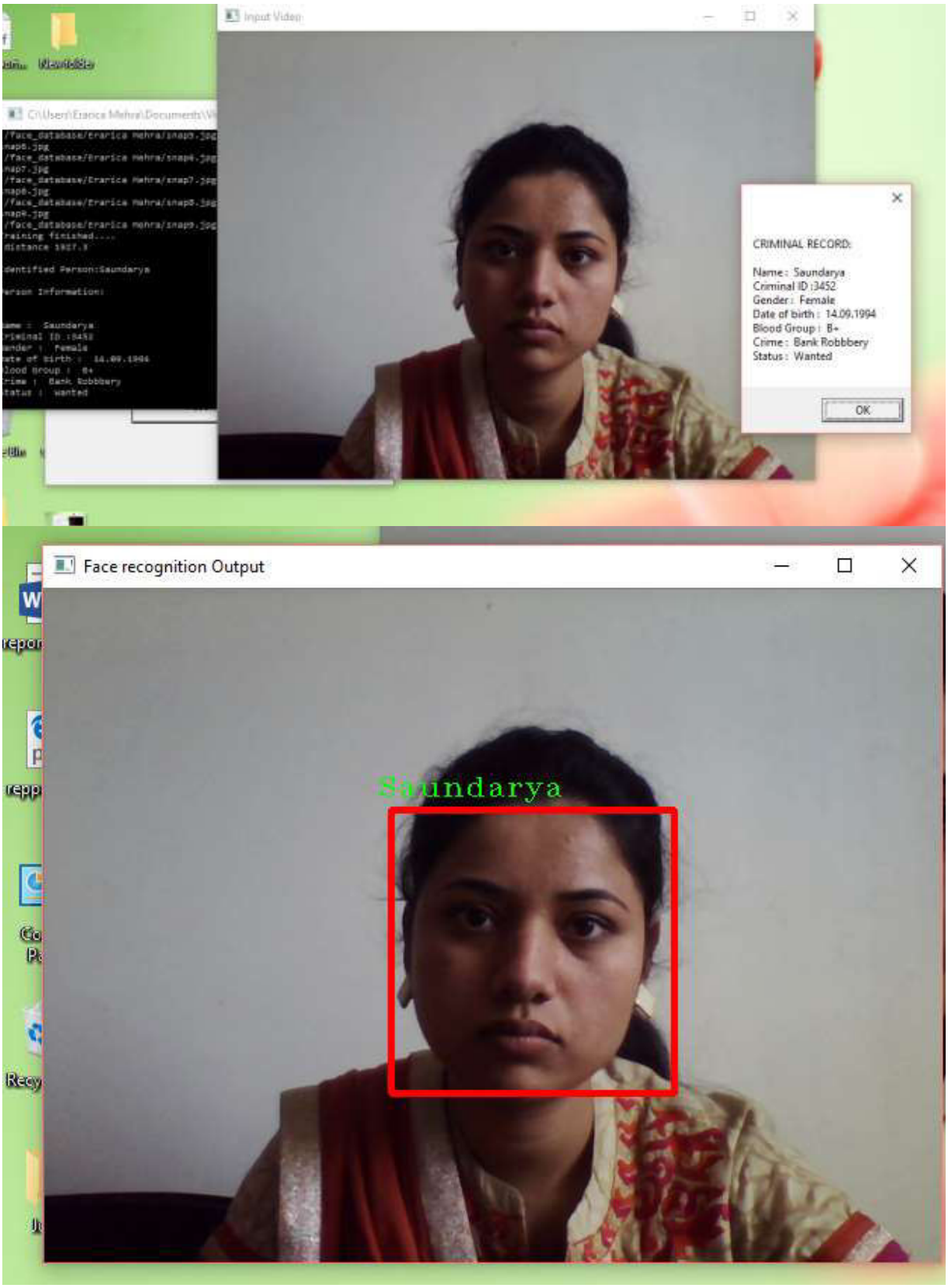


Image Capturing Successfull!
OK


form2
Capture Images
Train Database
Face Recognition
Capture - Face detection
C:\Users\Eranca Mehra\Documents\Visual Studio 2012\Proj...
form2
Capture Images
Train Database
Face Recognition
Face Detection Complete on Database!
OK
C:\Users\Eranca Mehra\Documents\Visual Studio 2012\Proj...


Input Video
CRIMINAL RECORD:
Name : Saundarya
Criminal ID :3452
Gender : Female
Date of birth : 14.09.1994
Blood Group : B+
Crime : Bank Robbbery
Status : Wanted
OK
Face recognition Output
Saundarya

Annexure B

Laboratory assignments on Project Quality and Reliability Testing of Project Design

ASSIGNMENT 4

1.Selenium Testing Tool

Selenium is a portable software testing framework for web applications. Selenium provides a record/playback tool for authoring tests without learning a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including Java, C#, Groovy,Perl, PHP, Python and Ruby. The tests can then be run against most modern web browsers. Selenium deploys on Windows, Linux, and Macintosh platforms. It is open-source software, released under the Apache 2.0 license, and can be downloaded and used without charge.

1.1 Selenium IDE

Selenium IDE is a complete integrated development environment (IDE) for Selenium tests. It is implemented as a Firefox Add-On, and allows recording, editing, and debugging tests. It was previously known as Selenium Recorder. Selenium-IDE was originally created by Shinya Kasatani and donated to the Selenium project in 2006. It is little-maintained and is compatible with Selenium RC, which was deprecated.

Scripts may be automatically recorded and edited manually providing auto completion support and the ability to move commands around quickly. Scripts are recorded in Selenese, a special test scripting language for Selenium. Selenese provides commands for performing actions in a browser (click a link, select an option), and for retrieving data from the resulting pages.

1.2 Selenium Grid

Selenium Grid is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the hub. Tests contact the hub to obtain access to browser instances. The hub has a list of servers that provide access to browser instances (Web Driver nodes), and lets tests use these instances. Selenium Grid allows running tests in parallel on multiple machines, and to manage different browser versions and browser configurations centrally (instead of in each individual test).

The ability to run tests on remote browser instances is useful to spread the load of testing across several machines, and to run tests in browsers running on different platforms or operating systems. The latter is particularly useful in cases where not all browsers to be used for testing can run on the same platform.

2.RELIBILITY TESTING

- Reliability Testing is about exercising an application so that failures are discovered

and removed before the system is deployed. The purpose of reliability testing is to

determine product reliability, and to determine whether the software meets the customer's reliability requirements.

- According to ANSI, Software Reliability is defined as: the probability of failure-free

software operation for a specified period of time in a specified environment. Software

Reliability is not a direct function of time. Electronic and mechanical parts may become "old" and wear out with time and usage, but software will not rust or wearout during its life cycle. Software will not change over time unless intentionally

changed or upgraded.

- Reliability refers to the consistency of a measure. A test is considered reliable if we

get the same result repeatedly. Software Reliability is the probability of failure-free

software operation for a specified period of time in a specified environment. Software

Reliability is also an important factor affecting system reliability.

- Reliability testing will tend to uncover earlier those failures that are most likely in

actual operation, thus directing efforts at fixing the most important faults.

- Reliability testing may be performed at several levels. Complex systems may be

tested at component, circuit board, unit, assembly, subsystem and system levels.

Software reliability is a key part in software quality. The study of software reliability can be

categorized into three parts:

1.Modeling

2.Measurement

3. Improvement

1. Modeling:

Software reliability modeling has matured to the point that meaningful results

can be obtained by applying suitable models to the problem. There are many models exist,

but no single model can capture a necessary amount of the software characteristics.

Assumptions and abstractions must be made to simplify the problem. There is no single model that is universal to all the situations.

2. Measurement:

Software reliability measurement is naive. Measurement is far from commonplace in software, as in other engineering field. “How good is the software, quantitatively?” As simple as the question is, there is still no good answer. Software reliability cannot be directly measured, so other related factors are measured to estimate software reliability and compare it among products. Development process, faults and failures found are all factors related to software reliability.

3. Improvement:

Software reliability improvement is hard. The difficulty of the problem stems from insufficient understanding of software reliability and in general, the characteristics of software. Until now there is no good way to conquer the complexity problem of software. Complete testing of a moderately complex software module is infeasible. Defect-free software product cannot be assured. Realistic constraints of time and budget severely limits the effort put into software reliability improvement.

3. TEST CASES

Input Activity:

STEPS	TEST PARAMETER	TEST CONDITION	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Login	Valid username and password	Successful Login	Successful Login	Pass
		Invalid username and password	Incorrect username or password	Incorrect username or password	Pass
2	Camera	Accessible	Should open the input video window and ready to capture the image	Opens the input video window and ready to capture the image	Pass
		Inaccessible		Error in Camera	Fail
3	Presence of Face	Face Present	Should detect the face, crop the face correctly and save in database	Detects the face, crops the face and saves in database	Pass
		Face Absent	Should capture the image , cannot detect the face ,so no face saved	Capture the image does not detect the face, saves error image	Fail
4	Training of Database	Face Present	Should crop the face correctly and save in database	Crops the face correctly and saves in database	Pass
		Face Absent	No faces detected	Error in training	Fail
5	Face Recognition	If face matched	Should display the details of matched face	Displays the details of matched face	Pass
		If face is unmatched	Should display 'No previous record found'	Displays 'No previous record found'	Pass

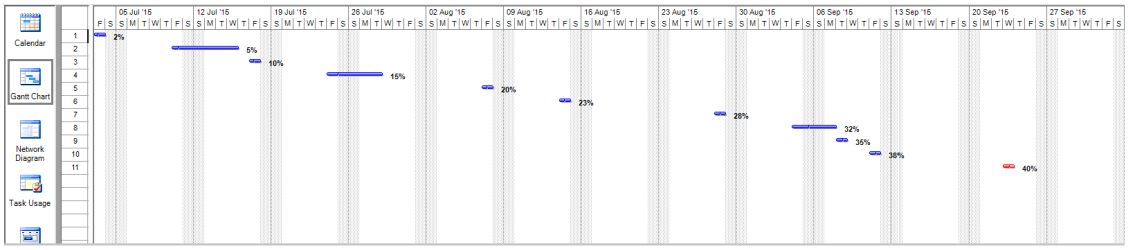
Output Activity:

Steps	Description	Output
1	Input Video	Face Detection, Cropping of face, Saving the face
2	Testing Video	Face Recognition , Criminal details i.e. Name, Date of birth, Gender ,Crime , Status etc.

Annexure C

Project Planner

	Name	Duration	Work	Start	Finish	Resource Names	% Comple
1	Paper Discussion	1 day	8 hrs	Fri 3/7/15	Fri 3/7/15	Internet	2%
2	More Paper search	4 days	32 hrs	Fri 10/7/15	Wed 15/7/15	Internet	5%
3	Topic Finilization, Da	1 day	8 hrs	Fri 17/7/15	Fri 17/7/15	Project Guide	10%
4	Title, topic refined, c	3 days	24 hrs	Fri 24/7/15	Tue 28/7/15	Project Guide	15%
5	Synopsis checked v	1 day	8 hrs	Fri 7/8/15	Fri 7/8/15	Project Guide	20%
6	Discussion of proje	1 day	8 hrs	Fri 14/8/15	Fri 14/8/15	Project Guide	23%
7	I / O specification of	1 day	8 hrs	Fri 28/8/15	Fri 28/8/15	Project Guide	28%
8	Discussion on Modu	2 days	16 hrs	Fri 4/9/15	Mon 7/9/15	Project Guide	32%
9	Review1 of Synops	1 day	8 hrs	Tue 8/9/15	Tue 8/9/15	Project Guide	35%
10	Review 2 & Name c	1 day	8 hrs	Fri 11/9/15	Fri 11/9/15	Project Guide	40%



Annexure D

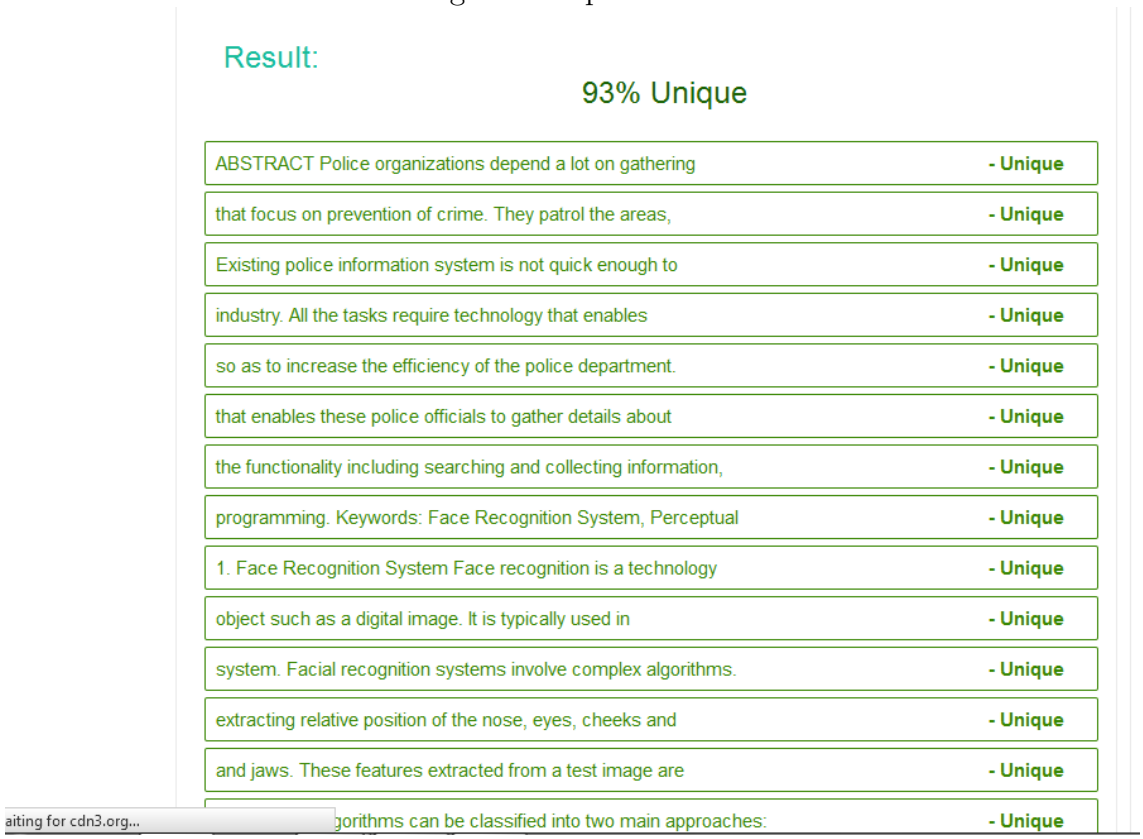
Reviewers Comments of Paper Submitted

1. Paper Title: Face Recognition using Viola Jones and Eigen Face Algorithm
2. Name of the Conference/Journal where paper submitted :International Journal of Research in Science and Engineering (IJRISE)
3. Paper accepted/rejected : Accepted
4. Review comments by reviewer :
5. Corrective actions if any :

Annexure E

Plagiarism Report

Plagiarism report



continuous hash vector. In the Quantization stage, the continuous	- Unique
Then the discrete hash vector is converted into the binary	- Unique
stage, the binary hash string is compressed and encrypted	- Unique
in a database. Perceptual hash functions offer excellent	- Unique
image. Perceptual hash function requires that only the hash	- Unique
to store the images (multimedia objects) in the database.	- Unique
advantage offered by perceptual hash functions is that if	- Unique
way, it still can be found in the database, thus increasing	- Unique
paper can successfully conclude that new advanced services	- Unique
they need more easily, to carry out their tasks. To aid	- Unique
that takes image of a suspect as input and compares the	- Unique
comparison by first converting the test image into its appropriate	- Unique

Originality Report

Original Work
Originality: 80%

This paper appears to be original, but be certain that any included text is properly cited.

The following web pages may contain content matching this document:

- http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1709989
- http://citeseerx.ist.psu.edu/showciting?cid=11727752
- http://users.ece.utexas.edu/~bevans/papers/2006/imageHashing/perceptualHashingTradeoffsIEEETIPDraft.pdf

A low originality percentage is indicative of plagiarized papers. Sometimes the score is lower due to long quotations within a document, so please make sure that you use proper citations if this is the case. For more information on our originality scoring process, click here.

Upgrade to premium to see which phrases were found to be un-original

Upload File

Annexure F

Information of Project Group Members

1. Name : Anand Singh
2. Date of Birth :17/01/1994
3. Gender : Male
4. Permanent Address :134A/1 Circular Road Allahabad
5. E-Mail : thakuranand818@gmail.com
6. Mobile/Contact No. :8698277384818@gmail.com
7. Placement Details :
8. Paper Published : yes

1. Name : Erarica Mehra
2. Date of Birth :15.07.1993
3. Gender : Female
4. Permanent Address : 97 White Wood, Malibu Towne, Gurgaon
5. E-Mail : irarikamehra@gmail.com
6. Mobile/Contact No. :7507049632
7. Placement Details : Placed in TCS
8. Paper Published : yes

1. Name :Saundarya Dorle
2. Date of Birth : 09.06.1994
3. Gender : Female
4. Permanent Address :Sant Krupa building, Adarsh Nagar, Hinjewadi, Pune-411057.
5. E-Mail : saundaryadorle@gmail.com
6. Mobile/Contact No. : 9579689045
7. Placement Details : placed in TCS
8. Paper Published : yes