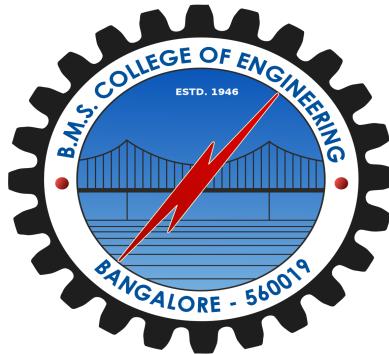


BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Bull Temple Road, Basavanagudi, Bangalore -560019

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



Title: Live Stream Multiple Face Recognition using Convolutional Neural Networks

Course – Project Work

Course Code –16IS8DCPRW

2020-21

Faculty In-charge:

Rashmi R

Submitted by:

Swapnil Hazarika - 1BM15IS084

Pranav Raj - 1BM15IS097

Moiz Rehman - 1BM14IS053

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**CERTIFICATE**

Certified that the project entitled “Live Stream Multiple Face Recognition using Convolutional Neural Networks” is a bonafide work carried out by Swapnil Hazarika(1BM15IS084), Moiz Rehman(1BM14IS053), and Pranav Raj(1BM15IS097) and is in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2020-21. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Guide

Rashmi R,
Assistant Professor,
Dept of ISE,
B.M.S. College of Engineering

Head of Department

Dr. M Dakshayini,
Professor and HOD,
Dept of ISE,
B.M.S. College of Engineering

Principal

Dr. B. V. Ravishankar
B.M.S. College of Engineering

External Viva

Name of the Examiners

Signature with Date

1.

2.

Table of Contents

TITLE	PAGE NUMBER
Table of Contents	2
List of Table	5
List of Figures	5
Abstract	7
Declaration by the Student batch and Guide	8
Acknowledgements	9

Chapter Number	Title	Page Number
1	Introduction	10
1.1	Overview	10
1.2	Problem Statement	14
1.3	Existing System	14
1.4	Objective	14
1.5	Outline of Report	14
1.6	Implementation Overview	15

1.6.1	Face Detection using CNN	16
2	Literature Survey	18
3	System Specifications	23
3.1	Hardware Configuration	23
3.2	Software Requirements	23
4	Tools and Programming Language Used	24
4.1	NetBeans IDE Framework	24
4.2	MySQL Database	25
4.3	Apache Maven	25
4.3.1	Maven's Objectives	26
4.4	Selection of Language	27
4.5	JAVA	27
4.5.1	Java Platform Discussions	28
5	System Design	29
5.1	High Level Design	29
5.1.1	Design Considerations	29
5.1.2	Development Methods	29
5.2	Architecture	32
5.3	Data Flow Diagram	33
5.4	Activity Block Diagram	34

6	Code Snippets	35
7	Testing	38
7.1	Types of Testing	38
7.1.1	Unit Testing of System	38
7.1.2	Test Cases	39
7.1.3	Integration Testing of System	42
7.1.4	Functional Testing of the System	43
7.1.5	System Testing	44
7.1.6	White-Box Testing of System	44
7.1.7	Black-Box Testing of System	44
7.1.8	Acceptance Testing	44
8	Results	45
9	Conclusion and Future Scope	48
10	References	49

List of Tables

Table Number	Description	Page Number
7.1	Test Case 1	39
7.2	Test Case 2	40
7.3	Test Case 3	40-41
7.4	Test Case 4	41
7.5	Integration for all Unit Testing	42

List of Figures

Figure Number	Description	Page Number
1.1	Domains of AI	13
1.2	Layers of CNN Network	13
1.3	Face Detection Process	17
4.1	NetBeans IDE	24
5.1	Waterfall Model	31
5.2	System Architecture	32
5.3	Data Flow Diagram (DFD)	33
5.4	Activity Block Diagram	34

6.1	Database Connectivity	35
6.2	Neural Network Analyser	35
6.3	Main Class	36
6.4	Learning	36
6.5	Recogniser	37
8.1-8.5	Output Screenshots	45-47

Abstract

Security concerns have grown tremendously over the years with the growth of technology, and it is very important that we safeguard our property from any worldly dangers. Surveillance as a form of security has been used since time immemorial, but it has constantly evolved with time, and with the advent of machine learning and artificial intelligence the scenario in the technological world was further ahead. Traditionally the recognition in live video required a large dataset of photos which compared them to the live stream, which is not the most efficient method. In this project we are trying to build a software that will be able to conduct multiple face recognitions using the Convolutional Neural Network(CNN) algorithm that will give us greater efficiency with a smaller dataset. In this project we are trying to get an efficient facial recognition system that can identify the person with the minimum training set of images and give us a good result.

DECLARATION

We, hereby declare that the Project Phase-1 work entitled “**Live Stream Multiple Face Recognition using Convolutional Neural Networks**” is a bonafide work and was carried out by us under the guidance of Prof. Rashmi R, Associate Professor, Department of Information Science and Engineering, B.M.S. College of Engineering, Bengaluru, in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi.

I further declare that, to the best of my knowledge and belief, this project has not been submitted either in part or in full to any other university for the award of any degree.

Candidate details:

SL. NO.	Student Name	USN	Student's Signature
1	Swapnil Hazarika	1BM15IS084	
2	Moiz Rehman	1BM14IS053	
3	Pranav Raj	1BM15IS097	

Place: Bengaluru

Date:

Certified that these candidates are students of the Department of Information Science and Engineering, B.M.S. College of Engineering. They have carried out the project work titled “**Live Stream Multiple Face Recognition using Convolutional Neural Networks**” as Project work for VII semester. It is in partial fulfillment for completing the requirement for the award of B.E. degree by VTU. The works are original and duly certify the same.

	Guide Name		Signature
	Rashmi R		

Date:

Acknowledgment

Foremost, We would like to express our sincere gratitude to our project mentor and advisor Rashmi R for our project and research, for her patience, motivation, enthusiasm, and immense knowledge. Her advice acted as a guiding light for this project on the path to success.

We duly acknowledge the contribution of researchers whose works are referenced in this project. Their works are the foundation upon which this project is built.

We also extend our gratitude to our HOD Dr. M Dakshayini for providing us the opportunity and the experience to carry out this research project.

We thank our principal, Dr. B. V. Ravi Shankar for his support by encouraging us to venture outside our comfort zone to carry out research for this project.

We extend our gratitude to BMS College of Engineering for giving us the opportunity to learn new concepts and explore different ideas.

Last but not the least, We also acknowledge the contribution of all the people who were anyhow, directly or indirectly contributed to the making of this project.

Chapter 1:

Introduction

1.1 Overview

One of the more popular methods of security in the recent decades can be said to be the range of biometric recognition systems that has popped up and within that group, facial recognition technology has become one of the hottest research hotspots in this area where visual technology in computers meets artificial intelligence. However, at the current level of technology as far as face recognition is concerned, it is pretty easily affected by internal or external differences, and it is often difficult for traditional face recognition methods to achieve ideal results, but, the emergence and the subsequent development of CNN completely changed and improved the ways of pattern recognition system.

Compared to using a standard manual feature extraction method, the CNN or convolutional neural network uses a relatively modern deep learning modal. When it comes to image processing, this has rather unique advantages taking into account the local connection weight sharing and it's various other characteristics. The training in this scenario is pretty important, which like in other machine learning algorithms directly affects the success of the network's training and it's final recognition rate.

Facial recognition is a fundamental research focalpoint, be it in the field of computer vision, pattern recognition or from the angle of machine learning or AI. It is also a fundamental step in research related to the face and technology, such as facial verification, facial recognition and tracking of a face. The aim of facial recognition is to detect human faces from videos or images and form the foundation for further studies in the field of this subject. The most classic face detection method would probably be the VJ facial recognition method, which was proposed by Viola and Jones in the year 2001, which used the various simple Haar features available and a cascade AdaBoost classifier to detect faces to gain efficient facial recognition performance in a real time scenario.

Following this, many researchers have over the years used more and more features to improve the detection accuracy, for instance, Local Binary Pattern (LBP), Scale-Invariant Feature Transform (SIFT), and Histogram of Oriented Gradient (HOG). Despite this, the performance of the face detector of this type will still rather drastically decrease with changes as the facial image set diversifies in any real world application. Apart from the waterfall structure, Felzenszwalb presented a DPM or a Deformable Part Model, which is a detection method based on HOG, using SVM as classifier, which could achieve a significant performance boost even when small samples are not fully labeled.

The main drawback here is the fact that the calculation is too complicated, and it is mainly based upon artificially designed features, and lacks stability. ACF [10] uses the aggregate channel function to detect faces in multiple views and has made great strides in the field of non-depth learning. In recent times, facial recognition methods based on CNN have multiple breakthroughs and have become the mainstream method for face recognition for real world applications. [1]Zhang et al. proposed to use a deep complex neural network for facial alignment and performed the recognition of facial attributes through a deep convolutional neural network. However, facial recognition research in recent times has majorly focused on the aspects of the facial area that are uncontrollable, for instance, exaggerated expressions or changes in posture or even facial occlusion. Faced with these many problems, it becomes rather difficult to make a good generalization based upon a single structural model of detection, which renders the model relatively less powerful in practical applications.

To fill in this gap, a chain of innovative deep-learning methods have emerged over the years recently. Advanced CNN effectively provides solutions to the above mentioned problems by using an innovative structure that can record the live video while recognizing faces.

AI or artificial intelligence, sometimes also referred to as machine intelligence, can be said to be intelligence shown by the machines, as opposed to natural intelligence shown by humans or other creatures. In the field of AI research, it is defined as the study of an “intelligent agent” that is any device which is aware of its environment and takes actions according to it to maximize its

chances of reaching its objective. The term "artificial intelligence" is also applicable when a machine mimics the "cognitive" functionalities that we, humans associate with our brain, for instance "learning" or "problem solving".

Artificial intelligence or AI is the study of how the machine emulates the human brain's thought process and the learning, decision making, and operating processes and applying it while solving problems. Finally, this research creates intelligent software systems. The aim of AI is to upgrade computational functionalities relating to human knowledge, such as, learning, reasoning and even problem solving. Intelligence is invisible and intangible and in this case it is composed of:

- Reasoning
- Learning
- Perception
- Linguistic Intelligence
- Problem Solving

Apart from the objectives of AI research that can be seen above, there are also other long-term goals in the general field of the intelligence sector as AI covers several domains as can be seen in Fig1.1 below.

Over the past few decades, methods and ideas regarding pattern classification have evolved and have become a behemoth in the fields of artificial intelligence research as it comprises an essential component in many of the real-world applications of this subject. ANN or Artificial neural networks and fuzzy logic are the two widely used models when it comes to pattern classification in today's world and is used to build an efficient model that retains the power, the various researchers in this field have introduced various hybrid models that combine both of them.

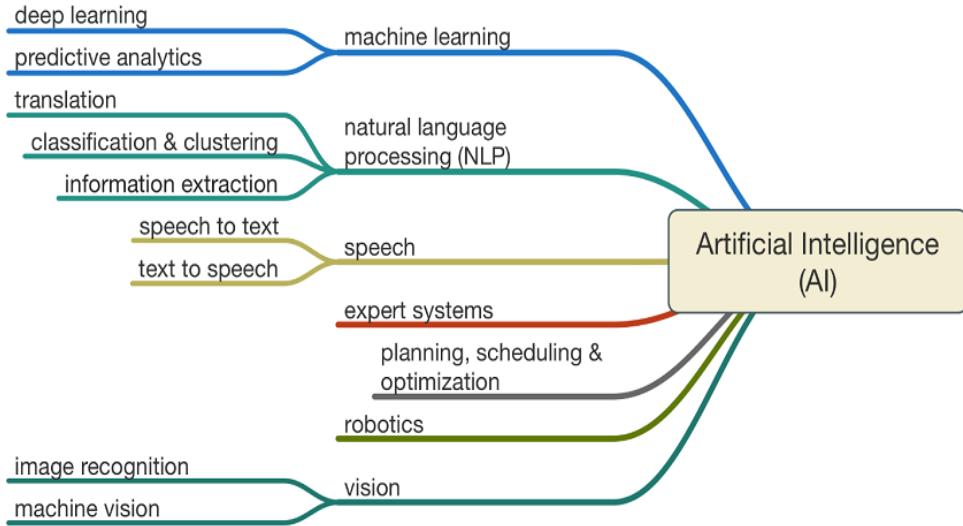


Fig 1.1 : Domains of AI

ANNs or Artificial Neural Networks are one of the most useful methods when it comes to handling the pattern classification problems. It has various capabilities that ensures it stands out from the other available methods, such as, handling nonlinear problems or being able to learn and handle a large data sample set.

The CNN network has three layers, namely: input layer, hidden layer, and output layer as can be seen below in Fig1.2. It has the ability to absorb the various information found online, and most importantly, there is no need to re-train the network in question in case any further new information is made available. This network possesses various features making it an effective network when addressing problems such as in pattern classification.

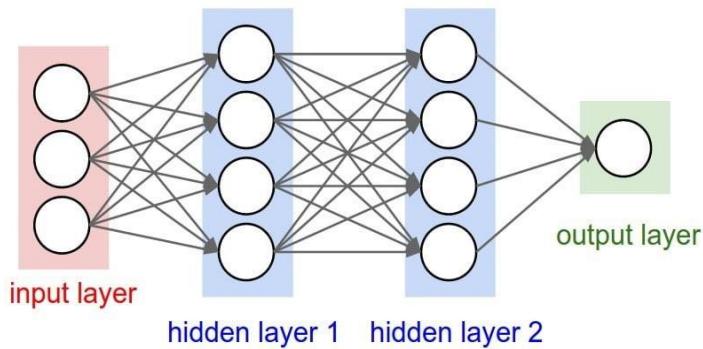


Fig 1.2 : Layers of CNN Network

1.2 Problem Statement

To develop a robust real-time system for detecting and recognizing a single face in different environments. The system can detect and recognize a face among multiple faces in a real-time video sequence. With the real-time video, the frames are extracted and the face is recognized with the help of a training set. The name of the person is displayed as the result.

1.3 Existing System

The existing system detects a single face using Machine Learning Neural Network and hedge cutting techniques. As there will be more edges detected by processing high dimension images, it degrades the performance of the System.

Some of the disadvantages in the current system are:

1. Lack of performance
2. More complex
3. Processing Is sequential stage by stage.
4. Time-consuming

1.4 Objective

Our goal is to set up a system for automatic face detection and recognition that works efficiently and simply in a real-time video and under varying poses, the difficult part of which is to handle face rotations in depth. The simulation is done in OpenCV using CNN. The code is tested for different environments.

1.5 Outline of Report

Starting with the Introduction to face detection and recognition we have included the Literature Survey and the challenges. The proposed methodologies are mentioned thereof. The software requirements and the configuration of OpenCV for the workspace environments are clearly

specified. The flow of the project is then followed which is represented by a professional flow diagram. Implementation of offline as well as online training and recognition, assisted with code description for the ease of reading is given. The validation of the project is proved with various test cases. The experimental results are shown along with the help of snapshots. The report is concluded along with discussion and future works at the end.

1.6 Implementation Overview

Considering an image from a live video or even a file, the face detector verifies the location of each image and then classifies it as "face" or "no face". The classification then assumes that the face aspect ratio is fixed, such as 50x50 pixels. This classifier further uses the data that has been stored in the XML file to determine the way to classify the location of every image.

haarcascade_frontalface_default.xml is the XML file trained as the classifier to hold the data for detecting faces.

The procedure follows four steps to perform the face detection:

- **Initializing the face detector**

The XML file is loaded as a classifier, the memory buffer is allocated to store the detected faces, expanded as the need arises.

- **Running the detector**

The detector is run which requires the image, the XML file, and the initialized memory buffer.

- **Viewing the results**

The display window is created to show the results and the detection rectangles are drawn to the in-memory images that are loaded from the file.

- **Releasing the resources**

The resources that are held by the input image, the XML data, and the memory buffer are then released, if multiple faces are detected, then the resources are not released until all the faces are detected.

When we call the face detector one thing that happens "behind the scenes" is that each of the positive facial regions generates multiple hits from the Haar detector. The area of the face itself generates the largest rectangular clusters and these overlap to a great extent. In addition, there is also a small detection to the left (of the observer) and two larger detections slightly above and to the left of the host's face group.

Isolated detections are generally false detections, so it makes sense to dismiss them. It also makes sense to combine multiple detections of each area of the face into one detection in some way. OpenCV performs these two operations before returning to the list of faces that were detected. The merging step first groups the rectangles with a lot of overlap and then locates the rectangle that can be said to be the average of the group. Then it proceeds to replace all the rectangles found in the group with it.

Between all the large groupings and the isolated rectangles, there are also smaller clusters that may face each other, or it may be a false detection. The minimum neighbor threshold ensures to avoid it and sets the cutoff level for discarding or retaining rectangular groups based upon the number of original detections in the group. Therefore, if we find that any face is missing, we can lower this threshold.

1.6.1 Face Detection using CNN

A Deep Convolutional Neural Network is the one in which the connectivity pattern between its neurons is modeled to copy an animal's visual cortex and is a class of neural networks and this is most commonly applied in the real world while analyzing the various visual imageries.

- **Convolutional Neural Network (ConvNet/CNN)** is a deep learning algorithm that takes an input image, processes it, and then proceeds to classify it into specific categories. An important advantage here is that the pre-processing required here for ConvNet is much lower when compared to other classification algorithms. Technically, to train and test a model using a deep learning CNN model, each image taken as an input passes through a series of convolution layers with filters (Kernels), that pools fully connected layers (FC).

- **Convolution Layer:** While extracting from an input image, it is the first layer that works on it. It learns the characteristics of the image using small squares of input data and preserves the relationship between pixels. It is more of a mathematical operation taking two inputs such as an image matrix and a kernel (filter). Convolution of an image with different filters can perform various operations such as edge detection, blur, and sharpen by applying its filters.
- **Non Linearity (ReLU):** It stands for Rectified Linear Unit for a non-linear operation and its output is $f(x) = \max(0, x)$. The main purpose of ReLU is to introduce non-linearity into ConvNet. The real-world data want or even need the ConvNet to learn non-negative linear values. There are also other nonlinear functions like tan h or sigmoid that can be used instead of ReLU.
- **Pooling:** Convolutional networks might also include various local or global pooling layers to streamline its underlying computation. Pooling layers reduce the dimensionality of the data by combining the output of one layer of the cluster of neurons into a single neuron in the next layer. The local pooling combines small clusters, generally 2×2 . The global grouping acts on all the neurons of the convolutional layer. In addition, the grouping can calculate the maximum or average value. The max pooling uses the maximum value of each group of neurons in the previous layer. The average grouping uses the average value of each cluster of neurons in the previous layer.
- **Fully connected layer:** This layer, as the name suggests, connects every neuron present in the layer in question to every neuron in another layer. It can be said to be similar to the traditional MLP or the multi-layer perceptron neural network. The flattened matrix then proceeds through a fully connected layer for the classification of the images.

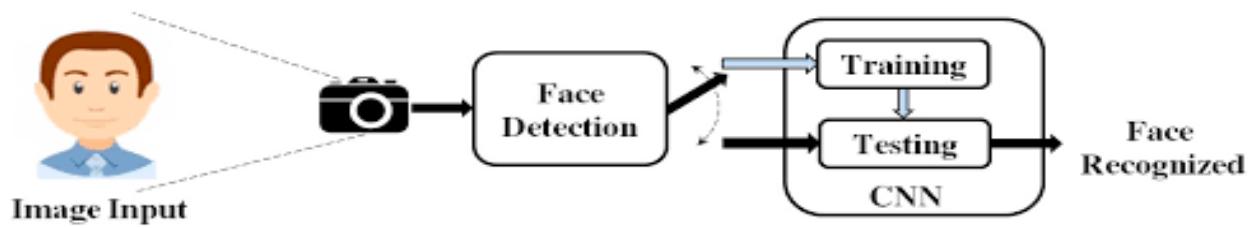


Fig 1.3 : Face Detection Process

Chapter 2:

Literature Survey

[1] Face Recognition Using Depth Images Base Convolutional Neural Network

- Facial recognition which is Because of the benefits of 3D information and its insensitivity, based on depth photographs has been extensively researched.to ambient light.
- Due to the rapidly developed biometric identification technology, people have high hopes for the accuracy of identification.
- Facial Security systems, video conferencing, and human-machine interface all require recognition. In an uncontrolled environment where facial photos have complex and vast features, extraction of distinctive traits and elimination of the influence of some inevitable aspects is a fundamental stage in various facial recognition algorithms.intrapersonal variations, such as facial expressions, postures, decorations and lighting.
- Because the RGB photos are affected by light, other parameters such as Kinect-based depth images, near infrared (NIR) images, and hyperspectral images were employed for face detection (HSI).

[2] Technical Analysis of CNN-Based Face Recognition System—A Study.

- Facial recognition is an important security system that has received more attention in recent years, especially in research as well as in industry.
- The facial recognition system is an automated feature that properly identifies or validates the pre-training of a certain label based on the label's digital image, which creates a bank of face traits.

- Facial recognition throughout the last decade, has resulted in the study of improved algorithms and mathematical modelling of linked functions..
- Facial recognition systems are paired with another validation system to improve the privilege's correctness.

[3] Thermal face recognition using a convolutional neural network.

- Face recognition technology is rapidly evolving (FR), more and more attention is paid to thermal face recognition.
- In the structure each node of each layer performs a direct calculation and serves as an input for the nodes of the following level in a multilayer neural network..
- Thermal images are first to be taken from the database of RGB-D-T faces. After that, a CNN method was devised for automatically recognising features from thermal facial data. In comparison to other recent approaches like LBP(Local Binary Patterns), HOG (Oriented Gradient Histogram) and invariable moments, the experimental results have demonstrated the proposed CNN architecture for face recognition has a higher recognition rate.

[4] Face Recognition System Based on CNN.

- With the tremendous advances in computer vision as well as artificial intelligence in recent years, face recognition is increasingly being used in our daily life.
- Face recognition has become one of the most prominent biometric recognition technologies, as well as one of the key research subjects in computer vision and artificial intelligence over the years.
- However, it has its drawbacks as it is easy to be influenced by differences, whether internal or external, and furthermore, traditional facial recognition methods are usually difficult to achieve the required results.

- This research paper has proposed a facial recognition algorithm based on an upgraded convolutional neural network, as well as demonstrating that the new algorithm can be applied to a particular problem dataset in the most efficient way.

[5] Enhanced Face Recognition System Based on Deep CNN

- Techniques on the most important study fields in biometric technology are facial recognition and this research article introduces a face recognition system (FR).
- The Viola-Jones face detection technique, face image enhancement with adaptive histogram equalisation (AHE) algorithm, and learning function for classification are the three steps of the system..
- Convolutional neural network (CNN) architectures VGG16, ResNet50 were used to study the functions and subsequent classification of the proposed system.
- Experimental work carried out with the The Yale B database has been expanded, as has the CMU PIE face database. The proposed approach's reliability and efficiency are demonstrated in comparison to previous methods in both databases. The Resnet50 architecture seen here reached 97, 23% and 98, 38%, respectively.

[6] Motion Detection and Tracking using Background Subtraction and Consecutive Frames Difference Method

- This document was submitted by the authors, mainly with security considerations in mind, to counter the growing methodologies used by various thieves' sources. Video surveillance and monitoring is an ideal method to solve this problem, and for this a new methodology for detecting motion in a real-time video stream environment has been developed.
- The motion detection system was developed for real-time applications. To identify motion from video frames, background subtraction and frame difference methods were utilised, and motion was recognised in real-time video in this system.

- Motion detection is a type of when motion is detected, a software monitoring system tells a security camera to begin recording an event, or displays motion detection using a graphical technique or by triggering an alarm..
- Background motion detection, also known as pixelwise, is a simple motion detection approach that uses a fixed camera to compare the current image with a reference or background image.

[7] Face Detection and Recognition System using Digital Image Processing

- The whole procedure for authentication of any face data was divided into two stages: at the first stage, face detection is performed quickly, except for those cases when the object is quite far away, after which the second stage begins, in which the face is recognized as a person.
- EigenFace Method (EFM): Kohonen initiated the implementation of eigenvectors for a face recognition problem using a simple neural network; for recognizing a human face in an aligned and normalized position.
- Because no. The number of Eigenfaces used is limited in PCA conversion, so the system did not have more than 90% accuracy for both manual as well as automatic face recognition.

[8] Security System using Motion Detection and Face Recognition

- The imaging power can be used to enhance the role of CCTV cameras in alerting the respective owners or police officers when an intruder is detected on the owners' premises.
- The footage captured by the camera is analyzed to detect any motion, and if motion is detected over a period of time, the intruder's face is matched against an authorized face database provided by the site owner using face recognition techniques.

- The project demonstrates huge potential in the aspect of image processing, not only in the computer vision industry, but also in the field of intrusion detection, which could be utilized for other purposes in the future.

[9] An Application of Face Recognition System using Image Processing and Neural Networks.

- Proposed to denote a self-organizing map (SOM) for measuring image similarity, we load face images associated with areas of interest into a neural network.
- At the end of the training phase, each neural unit is tuned to a specific prototype of the face image.
- Facial recognition is then performed using a probabilistic decision rule, this scheme offers very promising results for identifying faces associated with changes in lighting, as well as poses and facial expressions.
- The SOM method was trained on images from a single database, and the novelty of this work lies in the integrity of a face recognition system as a computer application to automatically identify or verify a person from a picture or video frame from a video source.
- We can train the system to recognize a small group of people. This is why the application of a multilayer perceptron (MLP) neural network (NN) has been studied for this task.

[10] VGGFace2: A Dataset for Recognising Faces across Pose and Age.

- Concurrent with the rapid development of deep convolutional neural networks (CNNs), a great deal of effort has been made recently to collect datasets in large-scale to support these data-demanding models.
- DATA SET OVERVIEW - where removal of label noise is done by human annotators has 800; 000 images, which is approximately 305 images per identity.

- The dataset is split into two parts, one for teaching with 8631 grades and the other for assessing (testing) with 500 grades.

Chapter 3:

System Specifications

3.1 Hardware Configuration

- **Hardware:** Dual Core
- **Hard Disk:** 250 GB
- **Speed:** 1.4 GHz
- **RAM:** 1GB
- **Key Board:** Standard Keyboard
- **TouchPad:** Button Mouse
- **Monitor:** LED

3.2 Software Requirements

- **Operating System:** Windows
- **IDE:** Net Beans 8.0.2
- **Technology:** Java8
- **User Interface:** AWT, Swings, and Applets
- **Software Tool Kit:** JDK8

Chapter 4:

Tools and Programming Languages Used

4.1 NetBeans IDE Framework

In the field of computer programming, a software structure is basically an abstraction in which software that provides general functionality could be selectively modified with some user-written code, thus providing software for a specific application. A software environment is a general-purpose, reusable software platform utilized to develop applications, products, and solutions.

The software environment includes utility programs, compilers, code libraries, toolkits, and APIs which bring together all of the different components to help you develop a project or solution.

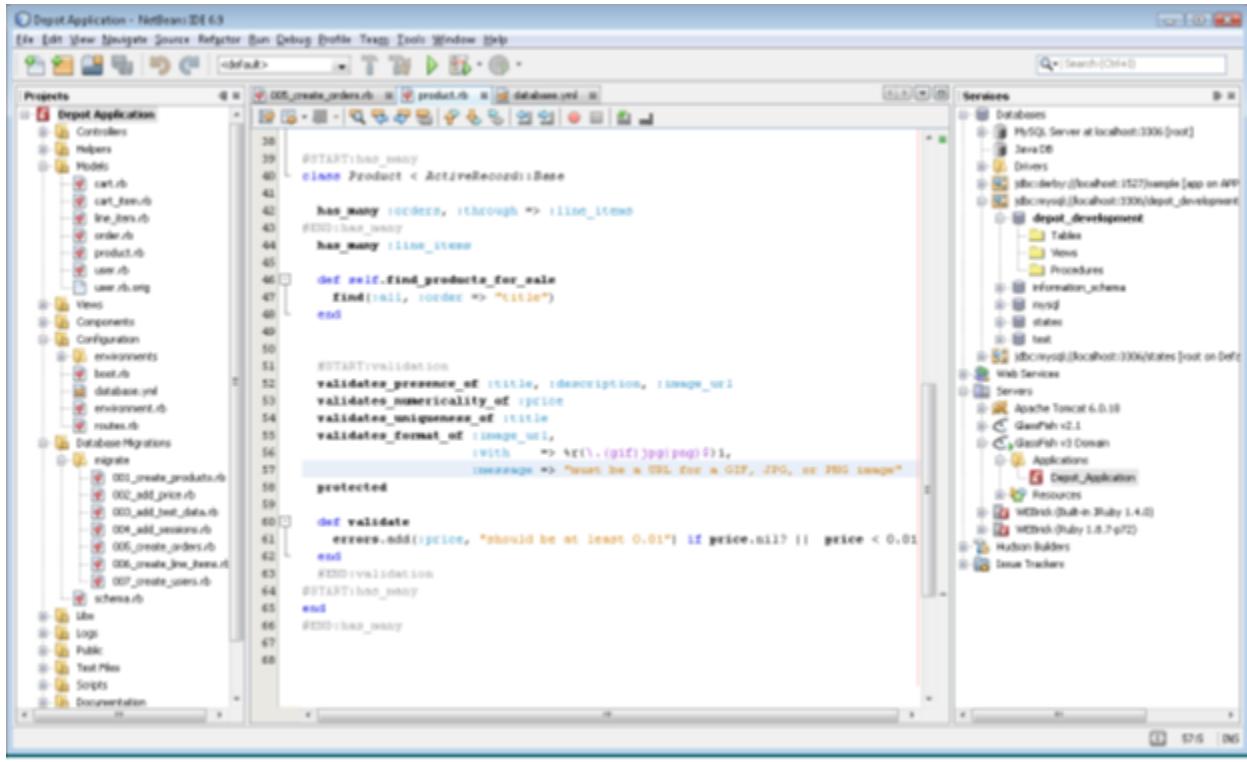


Fig 4.1: NetBeans IDE

NetBeans is a type of integrated development environment (IDE) for developing primarily in Java, but also in other languages such as PHP, C / C ++, and HTML5. It is also an application platform for Java based applications or other applications. It is written in Java and also

runs on any popular OS like Windows, OS X, Linux, Solaris, etc. that support a JVM. It also allows us to develop various applications from modules which are a collection of components of modular software.

The applications that are based upon this platform (also including the IDE of NetBeans itself) hence, it can also be further extended by any 3rd party developers that choose to use it. NetBeans IDE is what is called an open source IDE. It supports through the development of each and every type of applications of Java that can be said to be out of the box. Other features also include the Ant-based project system, also the Maven support, refactoring, version control.

All IDE functions were provided by the various available modules and each of these modules provides pretty well-defined functionalities such as the Java language support, or editing, or even the version control support for CVS and SVN. It also contains all the modules we possibly need for development of the Java file in one sweep of a download, allowing the user to get started immediately.

4.2 MySQL Database

It can be said to be an open source relational database management system (DBMS). Its name was designed by combining the "My", from the name of the daughter of co-founder Michael Widenius, and "SQL", an acronym for Structured Query Language. The source code has been made available under the terms of the GNU General Public License as well as various private agreements by the MySQL Development Project.

4.3 Apache Maven

Maven is a word derived from the Hebrew word for knowledge accumulator, and this project was started to simplify the building processes of the Jakarta Turbine project. There were also many other projects, each having their own set of Ant build files, which were a bit different from each other. The JARs have been verified with CVS. We need a standard way to create projects, a clear description of what the project is made of, a simple way to publish project information, and a method to share JAR files across multiple projects.

The resultant now is a powerful tool that could be used to create and manage any project based on Java. It was created with the hope that the developers created something that would simplify the day-to-day work of the many Java developers significantly and generally help in understanding the wide range of Java-based projects.

4.3.1 Maven's Objectives

The main goal of Maven is to enable the developer to understand the complete state of development in no time. To achieve this goal, Maven addresses several problem areas:

- Simplifying the building process
- Provide a unified build system
- Provide quality information regarding the project
- Encourage the development of better practices

While using Maven does not obviate the need for the underlying mechanisms, Maven protects developers from many details.

Provide a uniform build system

Maven creates projects using its POM or the Project Object Model and some of its plugins. By looking at a single Maven project, we learned how Maven projects are built, and this saved time when navigating many projects that we would ever need to.

Provide quality project information

Maven gives useful information about a project, which is partly taken from our POM and partly generated from the sources of our project. Maven can provide:

- Change of log created directly from its source control

- Cross-referencing of sources
- Mailing lists which are managed by the project in question
- The various dependencies used by the project
- Unit test reports including coverage

4.4 Selection of Language

The use phase should perfectly portray the plan report in a sane dialect of programming, keeping in mind the ultimate goal of accomplishing the fundamental last and correct element. The JAVA programming dialect is used for usage reasons. Advanced java helps more in building web applications. Servlets and JSPs (Java Server Pages) are used to build our enterprise.

4.5 JAVA

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

- Simple, based on the OOP standard and designed in such a way that people can easily learn if they know the basic concept of C ++.
- Object Oriented Everything is treated like an object and can be easily stretched as it is based on the concepts of object model.
- A distributed process of creating an object or product available for use by the customer and business users. By direct means of work or by indirect means of work with the help of intermediaries. The three parts of the marketing mix are product and price.
- This multi-threaded feature allows you to write plans that can perform multiple tasks at the same time. This blueprint allows programmers to hypothesize about smoothly working cooperative systems.
- Dynamic was designed to adapt to your environment. Java based programs pass more runtime assertions that can be used to validate and resolve objects at runtime.

The non-architecture compiler generates and constructs independent object code that makes the bytecode executable on many platforms, involving the java runtime system.

- The portable operation-specific aspects of the requirement make Java portable.
- High performance use of Just In Compilers increases productivity.
- Reliability: Java eliminates error related situations by highlighting mostly compile-time errors and run-time checks.
- Security: The security feature allows you to design virus-free systems.

4.5.1 Java Platform Discussion

The Java stage is the things or environment in which the framework runs. Windows 2000, Linux, Solaris, and Mac OS are by far the most understandable steps. Java stages are shown as a mixture of desktop and hardware. It is not the same as the other stages in that it is just organized programming that continues to run on top of the other transfer-based stages.

Java modules:

- Java Virtual Machine (JVM)
- Java Application Programming Interface (JAPI)

Chapter 5:

System Design

5.1 High-Level Design

The purpose of this design phase here is to use the requirement document and plan a solution to overcome the problem that the project seeks to solve. This step is the first one when it comes to moving to the domain of the solution for the question here. At this step, we create a block diagram of the system to help us understand how the system works. Here we divide the problem into modules.

This section represents the following:

- Design Considerations: This section covers various issues, which have the need to be resolved before even attempting to develop a complete version of the design solution.
- Development methods

5.1.1 Design Considerations

As stated above this part of designing needs to happen even before developing the design solution. The system design may be the key factor that affects the software quality and influences the later stages as well (like testing and maintenance). The purpose of designing a system is to determine the modules and its specifications that need to be included in the system, and how they tend to interact with each other and produce the needed results. By the end of this stage, all the major data structures, major system modules, also the file and output formats and their specifications are needed to be determined.

5.1.2 Development Methods

The waterfall lifecycle model is being used for the research for the development of this project. The waterfall model can be defined as an activity-centered lifecycle model that was developed by Royce.

This waterfall model is walked through in a gradual way to meet all the needed requirements of the activities before the designing of the activity is started. The entire design of the project is divided into many smaller tasks in the order of priority or precedence, and these tasks are designed one by one to ensure that they work perfectly. Once one of these small tasks is completed, the next one can be started, which is dependent on the first task that is completed. Every step after completion is checked to ensure that the task is working properly, error-free and meets all the necessary requirements.

This model was selected for this project mainly for two reasons. Reason one simply being simplicity of the modal, by using the waterfall model here the entirety of the project can be broken down into much smaller activities which can in turn be converted to code rather easily and following this as it is combined, the complete code for the project in question can be obtained. And the next reason would be the step of verification that is rendered essential by the model as it would lead to ensuring that the task is freed before any other task that may be dependent on it. Therefore, this reduces the chances of having any error remaining somewhere farther higher up in the task hierarchy.

Some of its unique traits are:

- It can be implemented for basically all sizes of projects.
- It leads to a clearer and much more solid approach to software development.
- In this model, testing becomes inherent in every phase.

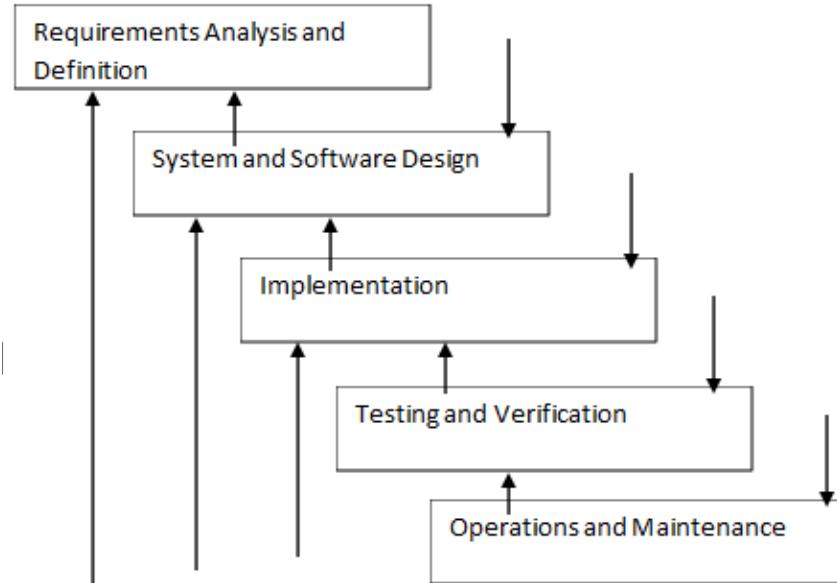


Fig 5.1: Waterfall Model

The waterfall model consists of the following stages, namely:

- **Requirement Analysis and Definition:** At this stage, the problem of the project is to be specified and identified followed by the desired objectives and its constraints.
- **System and Software Design:** At this level, the system specifications are developed into forming a software representation. The concern here is with the software architecture, data structure, algorithm details and also the interface representations.
- **Implementation:** Here, the designs are developed into the software that will be used.
- **Unit, Integration and System Testing:** This stage which includes testing ensures that the errors are identified and the software specification requirement is met. Following this stage the software may be delivered.
- **Operations and Maintenance:** At this level, the software may be updated as and when required to meet the changing and evolving needs, adapting to the changes in the external environment, correcting the various errors or oversights that might have previously gone undetected in the testing phase, further enhancing and ensuring the efficiency of our software.

5.2 Architecture

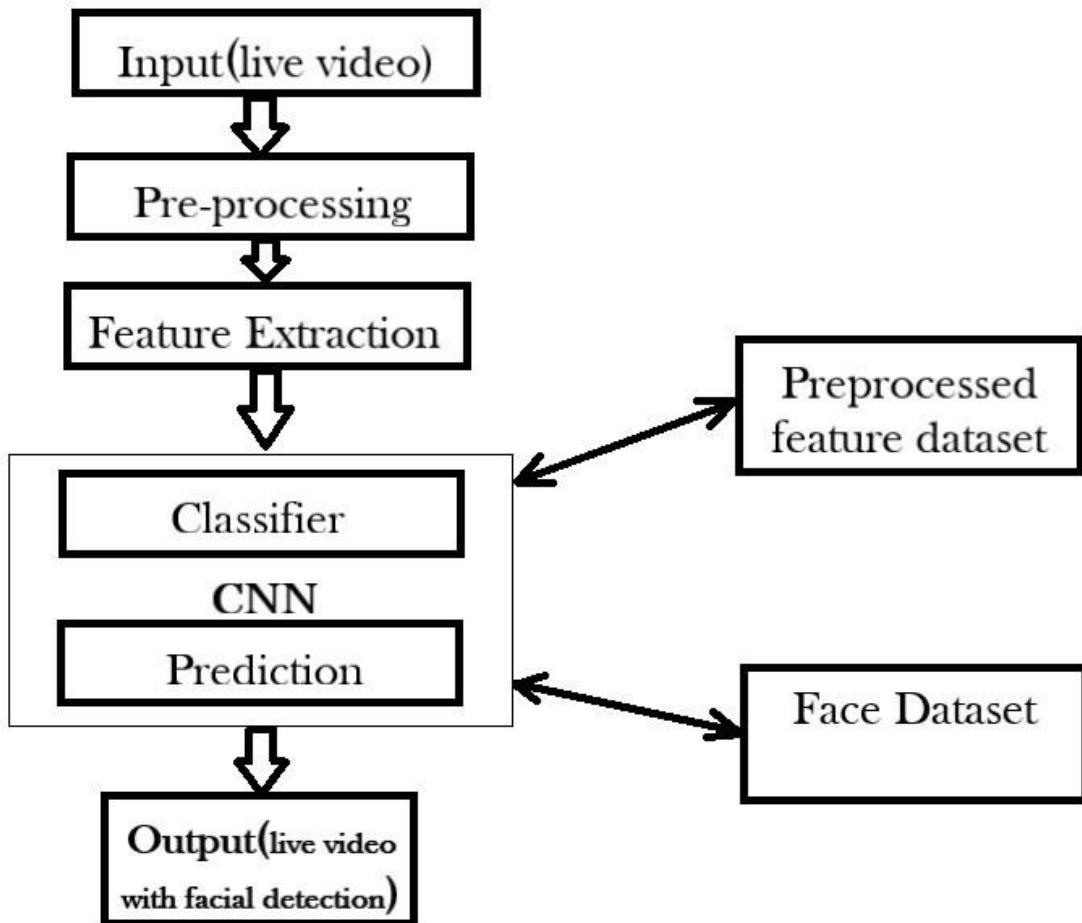


Fig 5.2: System Architecture

5.3 Data Flow Diagram

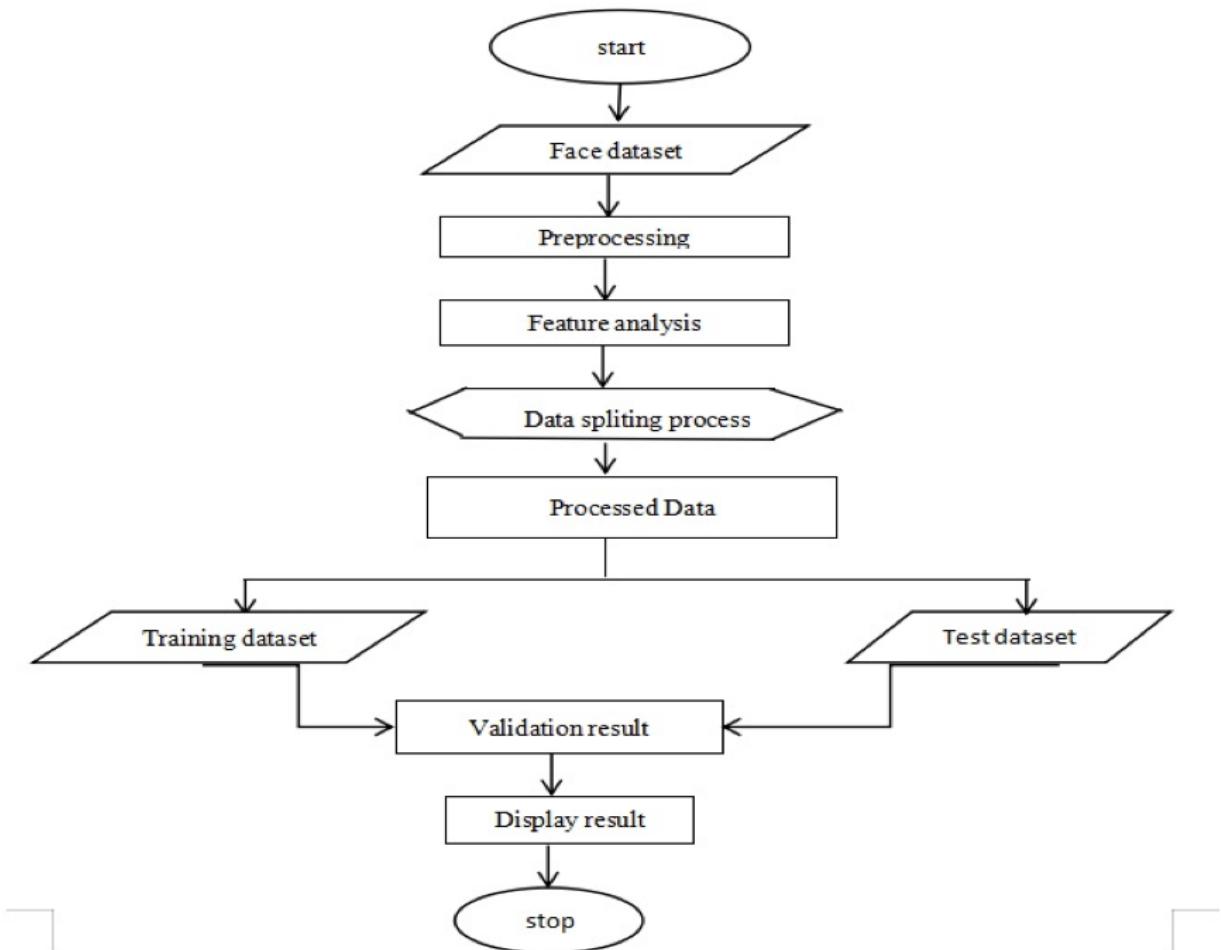


Fig 5.3: Data Flow Diagram (DFD)

5.4 Activity Block Diagram

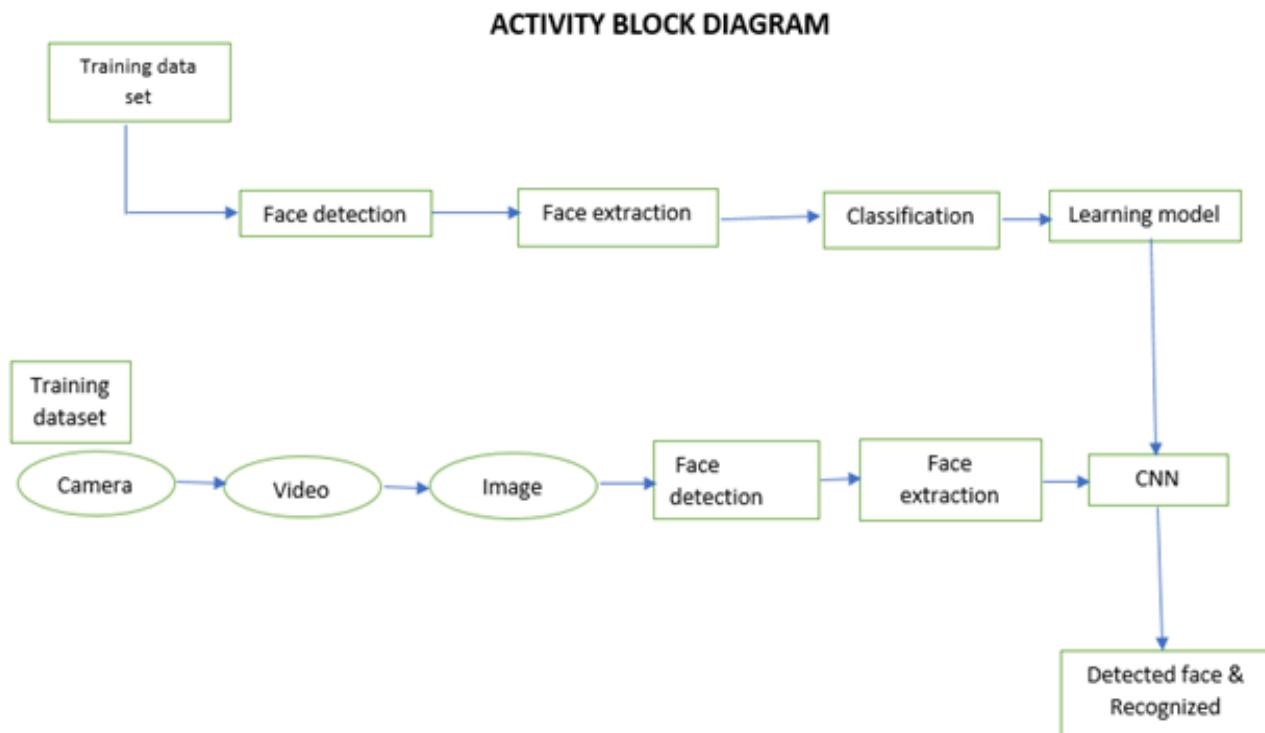


Fig 5.4: Activity Block Diagram

Chapter 6:

Code Snippets

```

21
22     public boolean init() throws SQLException {
23         try {
24             Class.forName("com.mysql.jdbc.Driver");
25
26             try {
27                 this.con = DriverManager.getConnection("jdbc:mysql://localhost:3306/" + Database_name, Database_user,
28                                                 Database_pass);
29             } catch (SQLException e) {
30
31                 System.out.println("Error: Database Connection Failed ! Please check the connection Setting");
32
33                 return false;
34
35             }
36
37         } catch (ClassNotFoundException e) {
38
39             e.printStackTrace();
40
41             return false;
42
43         }
44
45         return true;
46     }
47

```

Fig 6.1: Database Connectivity

```

40
41     public NeuralNetwork (float[][] inputs, int[] output, INeuralNetworkCallback neuralNetworkCallback) {
42         bOut = Utils.randFloat(-0.5f, 0.5f);
43         this.neuralNetworkCallback = neuralNetworkCallback;
44         // Default transfer function
45         this.transferFunction = new SigmoidFunction();
46         // Default result parser
47         this.resultParser = new BinaryResultParser();
48
49         this.inputs = inputs;
50         this.outputs = output;
51
52         this.nElements = output.length;
53         try {
54             this.dimension = inputs[0].length;
55         } catch (ArrayIndexOutOfBoundsException e) {
56             neuralNetworkCallback.failure(Error.ZERO_INPUT_ELEMENTS);
57         }
58
59         // Default num neurons = dimension
60         this.neurons = dimension;
61     }
62

```

Fig 6.2: Neural Network Analyser

```

16
17 public class Main extends Application {
18
19
20     @Override
21     public void start(Stage primaryStage) {
22         int i=0;
23         try {
24             BorderPane root = (BorderPane)FXMLLoader.load(getClass().getResource("Sample.fxml"));
25             Scene scene = new Scene(root,1350,720);
26
27             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
28             primaryStage.getIcons().add(new Image("logo.png"));
29             primaryStage.setTitle("Face Recognition System Using Machine Learning");
30             if(i==1)
31             {
32                 NN m = new NN();
33                 m.neuralNetwork();
34             }
35
36             primaryStage.setScene(scene);
37             primaryStage.show();
38         } catch(Exception e) {
39             e.printStackTrace();
40         }
41     }
42
43     public static void main(String[] args) {
44         launch(args);
45
46     }
47 }
48 }
```

Fig 6.3: Main Class

```

67     public void startLearning(){
68         try {
69             if (inputs.length != outputs.length)
70                 throw new NotSameInputOutputSizeException();
71             if (inputs.length == 0)
72                 throw new ZeroInputElementException();
73
74             HiddenLayerNeuron hiddenLayerNeuron = new HiddenLayerNeuron (neurons, dimension);
75             bias = hiddenLayerNeuron.getBias();
76             vWeights = hiddenLayerNeuron.getVWeights();
77             wWeights = hiddenLayerNeuron.getWWeights();
78
79             new NeuralNetworkThread().run();
80
81         } catch (NotSameInputOutputSizeException e) {
82             neuralNetworkCallback.failure(Error.NOT_SAME_INPUT_OUTPUT);
83         } catch (ZeroInputDimensionException e) {
84             neuralNetworkCallback.failure(Error.ZERO_INPUT_DIMENSION);
85         } catch (ZeroInputElementException e) {
86             neuralNetworkCallback.failure(Error.ZERO_INPUT_ELEMENTS);
87         } catch (ZeroNeuronsException e) {
88             neuralNetworkCallback.failure(Error.ZERO_NEURONS);
89         }
90     }
```

Fig 6.4: Learning

```
public int recognize(IplImage faceData) {  
  
    Mat faces = cvarrToMat(faceData);  
  
    cvtColor(faces, faces, CV_BGR2GRAY);  
  
    IntPointer label = new IntPointer(1);  
    DoublePointer confidence = new DoublePointer(0);  
  
    this.faceRecognizer.predict(faces, label, confidence);  
  
    int predictedLabel = label.get(0);  
  
    //System.out.println(confidence.get(0));  
  
    //Confidence value less than 60 means face is known  
    //Confidence value greater than 60 means face is unknown  
    if(confidence.get(0) > 60)  
    {  
        //System.out.println("-1");  
        return -1;  
    }  
  
    return predictedLabel;  
}  
}
```

Fig 6.5: Recogniser

Chapter 7:

Testing

The inspiration driving is to find defaults in the errand. The testing is the explanation behind attempting to discover every lack or deficiency in a working thing. It gives the best way to check the suitability of sections, sub-assemblies, and a completed thing. It is the method of working with the arrangement of confirming that the item framework meets its chucks and the customer covets and does not unacceptably miss the mark. There are characteristic sorts of tests. Every test category discourses a precise testing need.

7.1 Types of Testing

1. Unit
2. Integration
3. Functional
4. System
5. White-Box
6. Black-Box
7. Acceptance

7.1.1 Unit Testing of System

Testing fuses the sketch of examinations that endorse that the inner venture reason is working truly and that the program inputs extensive yields. All inside code and branches ought to be insisted. It is utilized to test the individual programming unit of an application. Unit testing is done before the breaker. It is the central testing that depends on heaps of its change and is intruding.

Unit testing guarantees that each exceptional strategy for an occupational system performs correctly to the recorded purposes of interest and covers obviously depicted inputs and output results.

This performs indispensable tests part by part level and tests a specific occupational framework, application, structure, and course of action. Unit testing is for the most part decided as mutual code and unit test of the thing lifecycle.

- ❖ Test plan and method
 - It will be performed physically and practically and will be composed in point of interest.
- ❖ Objectives
 - Field accesses essentially work properly.
 - Folios must be triggered from the recognized link.
 - The entrance display, responses must not be late.
- ❖ Features
 - Link's should take you to the correct page.
 - Confirm that the accesses are of the format.
 - No same entries should be allowed.

7.1.2 Test Cases

Name of the test	Face Reading with activated webcam
Test Description	Press on the power button to activate the webcam to see your face
Sample Input	Press on the power button
Expected Output	Webcam should be activated showing face
Actual result/Remarks	Same as expected

Passed (?)	Pass
------------	------

Table 7.1 Test Case 1

Name of the test	Face Saving
Test Description	Enter the details and press on face save button
Sample Input	Click on the face save button
Expected Output	Face and details should be saved in folder and database
Actual result/Remarks	Same as expected
Passed (?)	Pass

Table 7.2 Test Case 2

Name of the test	Face recognition
Test Description	Click on the Face recognize button to recognize a face
Sample Input	Show your face on the webcam and click on the face recognition button

Expected Output	Face should be recognize
Actual result/Remarks	Same as expected
Passed (?)	Pass

Table 7.3 Test Case 3

Name of the test	Face Motion Detection
Test Description	Click on the motion detection button to face detect your face motion on the webcam
Sample Input	Show your face on the webcam and click on the motion detection button
Expected Output	Motion detection pop up should be opened showing motion detected
Actual result/Remarks	Same as expected
Passed (?)	Pass

Table 7.4 Test Case 4

7.1.3 Integration Testing of System

Integration testing is needed to test urged programming sections to grasp whether they truly continue running as one structure. Exchange off particularly away to uncover the matters that climb up the mix of pieces. This testing is event driven and it is even more worried about the possibly critical aftereffect relating to the screens or even fields. These tests show that paying little personality to ensure the pieces were satisfactory, as displayed by unit testing.

The mix of sections is correct and solid. Programming testing can be said to be the testing of multiple programming parts made on a single stage to go on disillusion that was brought on by crossing points of various mishappenings. The attempt of testing is to watch that parts or programming application e.g. pieces in a thin framework or programming application at the association level to the interface without lurch.

Mix testing is deliberate when it comes to test joined programming portions to make sure of whether they may continue running as a framework. Combination testing is a way to uncover the various possible issues that may possibly rise up out of the mix of the many sections.

Testing can be said to be more stressed with the basic consequence of screen fields. Combination tests display that in spite of the way that the sections were solely gratification, as can be seen by up by viable unit testing. The programming blend testing can be said to be the coordination testing of two or more joined programming sections on an alone stage to ensure the frustrations realized by interface disfigurements.

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

Test Results: This test case passed successfully. No defects met.

Name of the test	Test End to End Face Detection and Face Recognition System
Test Description	Test case to show end to end face detection and recognition also motion detection
Sample Input	Power button is clicked, Click on the face save button, Click on the face recognition button, Click on the Motion detection button
Expected Output	Webcam should be activated, Face is saved with details, Face is recognized and motion is detected
Actual result/Remarks	Same as expected
Passed (?)	Pass

Table 7.5 Integration of all Unit Testing

7.1.4 Functional Testing of the system

Functional tests give exact establishes that limits make an effort shown by the occupational and specific necessities, framework certification, and customer manual. Affiliation and availability of functional tests revolve around necessities, key limits, or uncommon trials. Besides, effective allowance identifying with perceived occupational process streams, data fields, predefined structures, and techniques must be measured for testing. Additional perceived and convincing estimation of current tests is determined.

7.1.5 System Testing

This type of testing sees to it that the entire fused programming structure has met the various necessities. It further tests a setup to make sure of the known results. An instance of structure testing would be the arranged structure blend test. System testing relies upon techniques that focus on the various pre-driven strategy blend centers and associations.

7.1.6 White-Box Testing of System

Here the invention analyzer has all the data of internal mechanisms, structure and even product. It is additionally used to test parts that can't possibly come from a discovery level.

7.1.7 Black-Box Testing of System

The item here is attempted with unknown data of the internal working, structure, or lingo of the module attempted. Disclosure tests, which are the most diverse sorts of tests, are to be made from an indisputable basis record.

7.1.8 Acceptance Testing

This type of testing is a risky period of testing interest conducted by the end client. It additionally assures that the framework experiences the utilitarian prerequisites.

Results for the tests: All the trials said above have been passed effectively. No deformities has been experienced.

Chapter 8:

Results

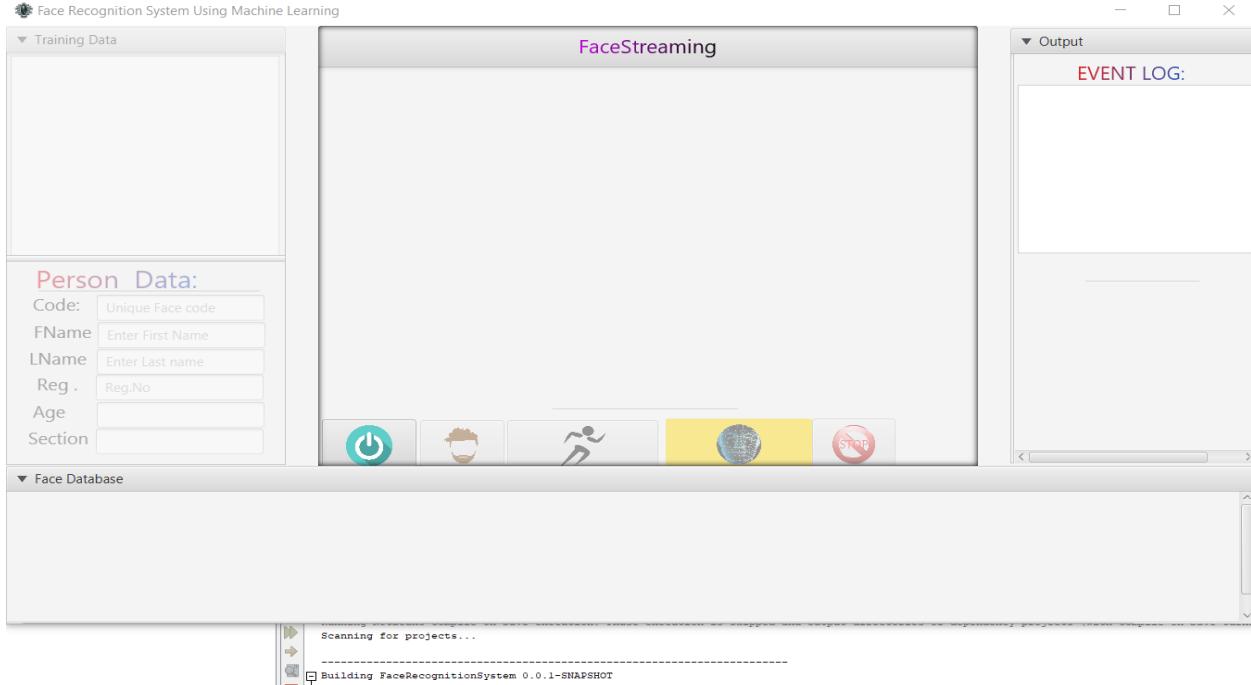


Fig 8.1 : Opening Page

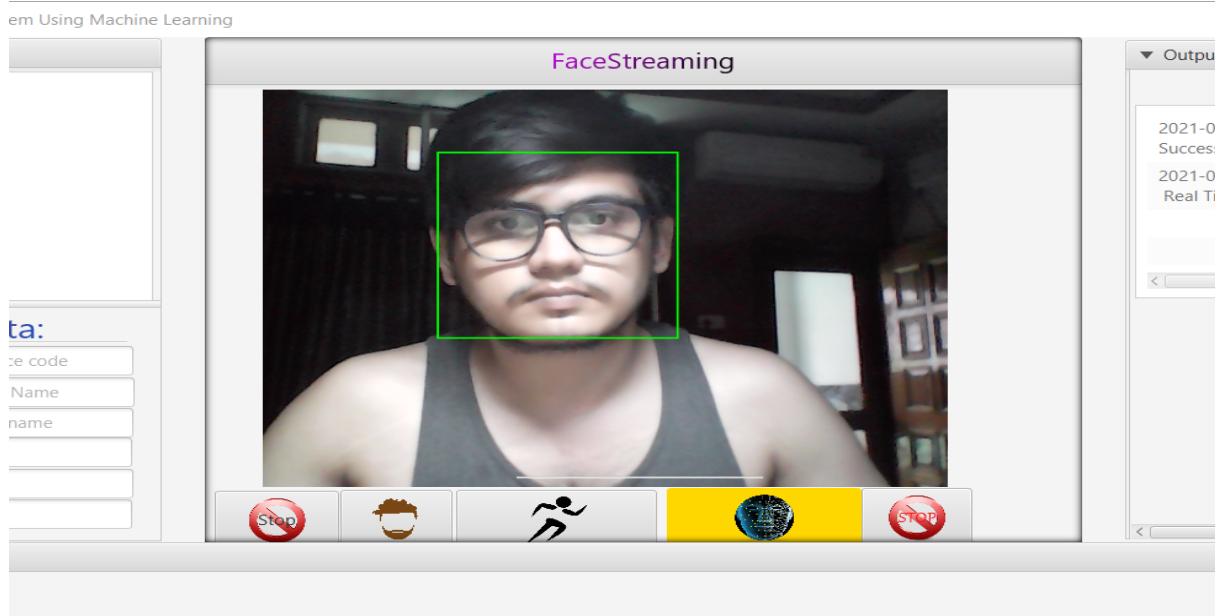


Fig 8.2 : Lifestreaming and recognizing the face.

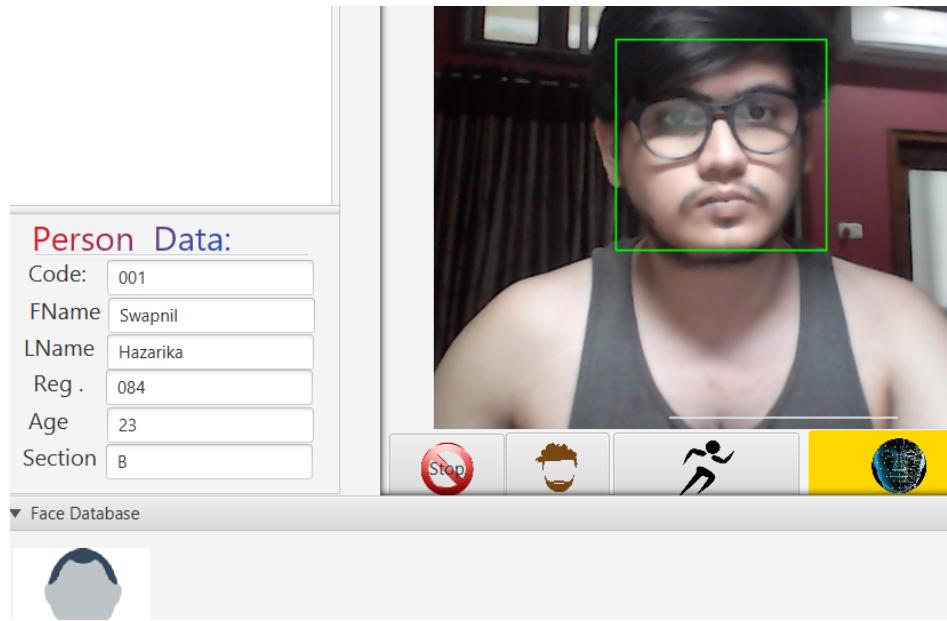


Fig 8.3 : Saving data of the person

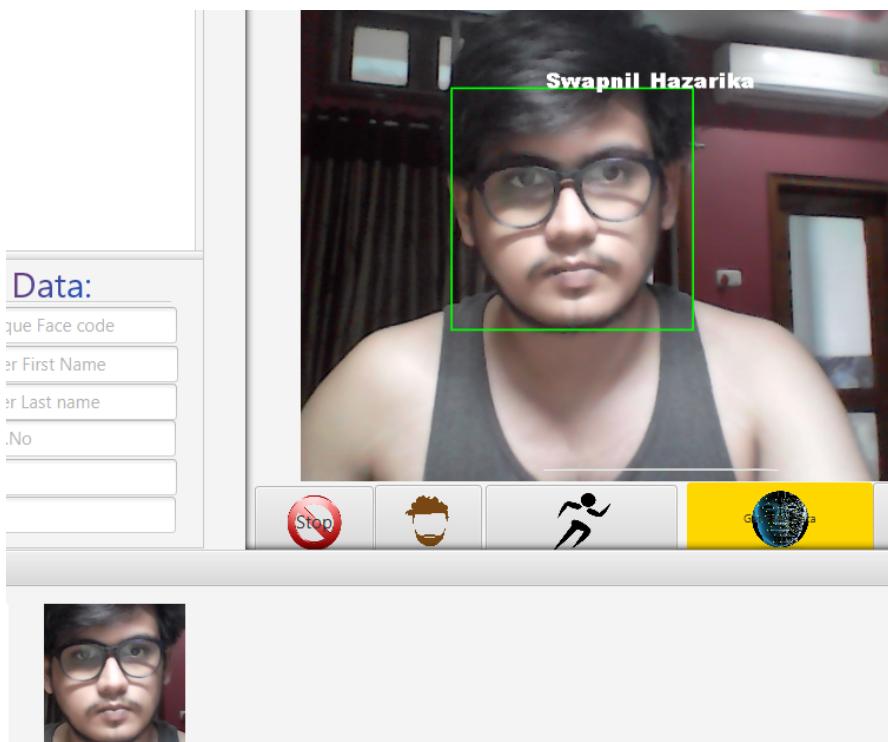


Fig 8.4 : Person recognized

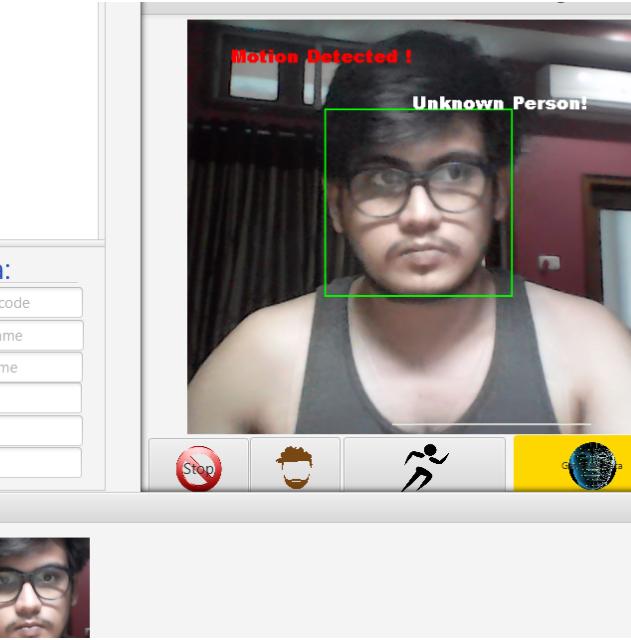


Fig 8.5 : Motion detected

Chapter 9:

Conclusion and Future Scope

We have provided an overview of the design and implementation of a real-time face recognition system. We offered a reasonably accurate face recognition system with excellent lead times. Although some aspects of the system are still under experimental development, the overall project has been successful as it was able to provide a robust experience in a limited environment. Face positioning and tracking is done in real time, and consistent face detection accuracy is recorded using video input. A dynamic demo system that performs real-time recognition from video input has also been implemented and has been found to operate at up to 2 frames per second with OpenCV with 75-80 ° C accuracy.

Convolutional neural networks are an important type of deep learning. Convolutional neural networks in image processing have unique advantages due to the separation of the weight of local connections and other properties. CNN training is essential for directly determining the success and ultimate recognition rate of network training.

We examined the number of model hierarchies, activation functions, outliers, and optimization algorithms. The model has been improved based on the original dataset, and finally the recognition speed has been greatly improved. The dataset used in this study can effectively represent the performance of the improved algorithm, albeit with some limitations because it is difficult to collect large amounts of data in practice.

The improved model still needs to be improved and the recognition accuracy may be improved in the future.

References

1. J. Chen, Z. Zhang, L. Yao, B. Li and T. Chen, "**Face Recognition Using Depth Images Base Convolutional Neural Network,**" *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2019, pp. 1-4, doi: 10.1109/CITS.2019.8862099.
2. D. Wang, H. Yu, D. Wang and G. Li, "**Face Recognition System Based on CNN,**" 2020 International Conference on Computer Information and Big Data Applications (CIBDA), 2020, pp. 470-473, doi: 10.1109/CIBDA50819.2020.00111.
3. G. Singh and A. K. Goel, "**Face Detection and Recognition System using Digital Image Processing,**" 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 348-352, doi: 10.1109/ICIMIA48430.2020.9074838.
4. D, Aju & Agarwal, Ashwani & Jain, Himanshu & Bhati, Divyanshu. (2017). "**Security System using Motion Detection and Face Recognition.**" Imperial Journal of Interdisciplinary Research (IJIR). 3.
5. Zhan, Wu & Peng, Min & Chen, T.. (2016). **Thermal face recognition using convolutional neural network.** 6-9. 10.1109/OPTIP.2016.7528489.
6. Ridha Ilyas, Bendjillali & Beladgham, Moh & Merit, Khaled & Miloud, Kamline. (2020). **Enhanced Face Recognition System Based on Deep CNN.** 1-6. 10.1109/ISPA48434.2019.8966797.
7. Sharma S., Kalyanam A., Shaik S. (2019) **Technical Analysis of CNN-Based Face Recognition System—A Study.** In: Behera H., Nayak J., Naik B., Abraham A. (eds) Computational Intelligence in Data Mining. Advances in Intelligent Systems and Computing, vol 711. Springer, Singapore. https://doi.org/10.1007/978-981-10-8055-5_49
8. Rathi, Rakesh & Choudhary, Manish & Tech, M & Chandra, Bhuwan. (2012). **An Application of Face Recognition System using Image Processing and Neural Networks.** International Journal of Computer Technology and Applications. 03.
9. Suresh DS, Lavanya M P, "**Motion Detection and Tracking using Background Subtraction and Consecutive Frames Difference Method**" International Journal of Research Studies in Science, Engineering and Technology Volume 1, Issue 5, August 2014.
10. Cao, Qiong & Shen, Li & Xie, Weidi & Parkhi, Omkar & Zisserman, Andrew. (2018). **VGGFace2: A Dataset for Recognising Faces across Pose and Age.** 67-74. 10.1109/FG.2018.00020.