**FACE RECOGNITION ATTENDENCE SYSTEM**

**A PROJECT REPORT**

Submitted to

**Jawaharlal Nehru Technological University Kakinada, Kakinada**

in partial fulfillment for the award of the degree of

**Bachelor of Technology**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by

**M.JAHNAVI PRIYANKA(15KN1A0556) B.TEJASWINI (15KN1A0513)**

**CH.HARI KRISHNA(15KN1A0530) CH. SRI AJITH(15KN1A0529)**

Under the esteemed guidance of

Mr. A. HARI KRISHNA

**Assistant Professor, CSE Department**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**NRI INSTITUTE OF TECHNOLOGY**

**(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)**

**(Accredited by NAAC with ‘A’ Grade, ISO 9001 : 2015 Certified)**

**Pothavarappadu (V), Agiripalli (M), Krishna Dist, PIN: 521212, A.P, India.**

**2015-2019**

**NRI INSTITUTE OF TECHNOLOGY**

**(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)**

**(Accredited by NAAC with ‘A’ Grade, ISO 9001 : 2015 Certified)**

**Pothavarappadu (V), Agiripalli (M), Krishna Dist, PIN: 521212, A.P, India.**

Certificate

This is to certify that the Project entitled **“FACE RECOGNITION ATTENDENCE SYSTEM”** is a bonafide work carried out by **M.Jahnavi Priyanka(15KN1A0556),B.S.S.P.Tejaswini(15KN1A0513), Ch.Hari Krishna (15KN1A0530) and Ch.Sri Ajith (15KN1A0529)** in partial fulfillment for the award of degree of Bachelor of Technology in **Computer Science & Engineering** of **Jawaharlal Nehru Technological University, Kakinada** during the year 2018-2019.

**(**Mr. A. Hari krishna**) ( Dr. K.V. Sambasiva Rao)**

**Project Guide Head of the Department**

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

I take this opportunity to thank all who have rendered their full support to my work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of my project cannot be expressed with a few words for their valuable suggestions.

I am grateful to my Project Guide Mr. A. Hari Krishna**, Assistant Professor** for rendering the valuable suggestions and extended his support to complete the project successfully.

I am expressing my heartfelt thanks to **Head of the Department, Dr. K.V. Sambasiva Rao** garu for his continuous guidance for completion of my Project work.

I am thankful to the **Principal, Dr. C. Naga Bhaskar** garu for his encouragement to complete the Project work.

I am extending my sincere and honest thanks to the **Chairman, Dr. R. Venkata Rao** garu **& Secretary, Sri K. Sridhar** garu for their continuous support in completing the Project work.

Finally, I thank the Administrative Officer, Staff Members, Faculty of Department of CSE, NRI Institute of Technology and my friends, directly or indirectly helped us in the completion of this project.

**15KN1A0556**

**15KN1A0513**

**15KN1A0530**

**15KN1A0529**

**ABSTRACT**

Uniqueness or individuality of an individual is his face. In this project face of an individual is used for the purpose of attendance making automatically. Attendance of the student is very important for every college, universities and school. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is recorded. Time consumption for this purpose is an important point of concern. Assume that the duration for one subject is around 60 minutes or 1 hour & to record attendance takes 5 to 10 minutes. For every tutor this is consumption of time. To stay away from these losses, an automatic process is used in this project which is based on image processing. In this project face detection and face recognition is used. Face detection is used to locate the position of face region and face recognition is used for marking the understudy’s attendance. The database of all the students in the class is stored and when the face of the individual student matches with one of the faces stored in the database then the attendance is recorded. The image will be captured twice-once at the beginning of the class and once at the end to ensure that the student has attended the whole class. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential.

**INDEX**

1. **List of Figures i**
2. **List of Tables ii**
3. **Introduction 1**
   1. Introduction to project 1
   2. Problem Definition 2
   3. Solution for Problem Definition 2
   4. Process Diagram 3
4. **Literature Review 4**
5. **System Analysis 5**
   1. Existing System 5
   2. Proposed System 5
   3. Analysis Model 5
   4. Modules 6
6. **Feasibility study 7**
   1. Technical 7
   2. Economical 8
   3. Operational 8
7. **System Requirement Specification 9**
   1. Introduction 9
   2. Functional Requirements 10
   3. Non-Functional Requirements 10
   4. System Requirements  **11**
      1. Software Requirements 11
      2. Hardware Requirements 11
8. **System Design 12**
   1. UML Modeling Introduction 12
   2. High level Design 12
   3. Low level Design 13
   4. Class Diagram 14
   5. Use Case Diagram 15
   6. Activity Diagram 16
   7. Sequence Diagram 17
9. **Coding 18**
   1. Software Description (Technical Description) 18
   2. Sample Code 24
10. **Testing 28**
11. **Screen Shots 32**
12. **Conclusion 35**
13. **Future Enhancement 36**
14. **Bibliography 37**

**LIST OF FIGURES**

**1: Process Diagram 3**

**2: Waterfall model (SDLC) 6**

**3: Class Diagram of the process 14**

**4: Use case diagram of the process 15**

**5: Activity Diagram of the process 16**

**6: Sequence diagram of the process 17**

**7: Spyder working environment (IDE) 20**

**8: HOG Algorithm 23**

**9: Registration page 32**

**10:Login page 32**

**11: Page after logging in successfully 32**

**12: Page to upload pictures to the server 33**

**13:After successful Face detection 33**

**14: Output Excel sheet 34**

**LIST OF TABLES**

**1: Test cases to test the software 31**

***CHAPTER-1***

***INTRODUCTION***

* 1. **Introduction to project:**

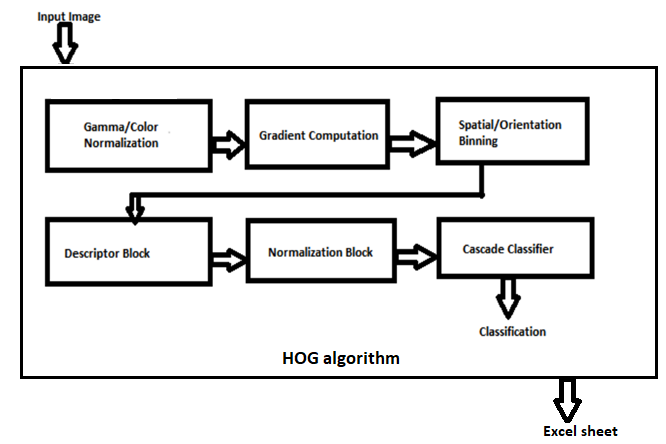
Face recognition is an important application of Image processing owing to its use in many fields. Identification of individuals in an organization for the purpose of attendance is one such application of face recognition. Maintenance and monitoring of attendance records plays a vital role in the analysis of performance of any organization. The purpose of developing attendance management system is to computerize the traditional way of taking attendance. Automated Attendance Management System performs the daily activities of attendance marking and analysis with reduced human intervention. The prevalent techniques and methodologies for detecting and recognizing face fail to overcome issues such as scaling, pose, illumination, variations, rotation, and occlusions. The proposed system aims to overcome the pitfalls of the existing systems and provides features such as detection of faces, extraction of the features, detection of extracted features, and analysis of students' attendance. The system integrates techniques such as image contrasts, integral images, color features and cascading classifier for feature detection. The system provides an increased accuracy due to use of a large number of features (Shape, Colour, LBP, wavelet, Auto-Correlation) of the face. Faces are recognized using Euclidean distance and k-nearest neighbor algorithms. Better accuracy is attained in results as the system takes into account the changes that occur in the face over the period of time and employs suitable learning algorithms. The system is tested for various use cases. We consider a specific area such as classroom attendance for the purpose of testing the accuracy of the system. The metric considered is the percentage of the recognized faces per total number of tested faces of the same person.

Face recognition consists of two steps, in first step faces are detected in the image and then these detected faces are compared with the database for verification. A number of methods have been proposed for face detection i.e. HOG algorithm, the HAAR algorithm, and various OpenCV algorithms. The efficiency of face recognition algorithm can be increased with the fast face detection algorithm. In all the above methods HOG is most efficient. Our system utilized this algorithm for the detection of faces in the class room image. Face recognition techniques can be divided into two types Appearance based which use texture features that is applied to whole face or some specific Regions, other is Feature based which uses geometric features like mouth, nose, eyes, eye brows, cheeks and Relation between them. Statistical tools such as Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), Kernel Methods, and Neural Networks, Eigen-faces have been used for construction of face templates. Illumination invariant algorithm is utilized for removing the lighting effect inside the class room.

* 1. **Problem definition:**

Maintaining the attendance is very important in all the institutes for checking the performance of Students. Every institute has its own method in this regard. Some are taking attendance manually using the old paper or file based approach and some have adopted methods of automatic attendance using some biometric techniques. But in these methods students have to wait for long time in making a queue at time they enter the college. Many biometric systems are available but the key authentications are same is all the techniques. Every biometric system consists of enrolment process in which unique features of a person is stored in the database and then there are processes of identification and verification. These two processes compare the biometric feature of a person with previously stored template captured at the time of enrollment. Biometric templates can be of many types like Fingerprints, Eye Iris, Face, Hand Geometry, Signature, Gait and voice.

* 1. **Solution for problem Definition:**
* Our system uses the face recognition approach for the automatic attendance of students in the class room environment without human intervention.
* Our system utilized HOG algorithm for the detection of faces in the class room image.
* Face recognition consists of two steps, in first step faces are detected in the image and then these detected faces are compared with the database for verification.
* When the face is detected, our system automatically plots the attendance for that particular candidate.
  1. **Process Diagram:**

****

**Fig. 1: Process Diagram**

***CHAPTER-2***

***LITERATURE REVIEW***

**2.1 LITERATURE REVIEW:**

**Paper Name**: A design and implementation of a wireless iris recognition attendance management system.

**Author:** Seifedine Kadry, Khaled Smaili.

Iris recognition verification is one of the most reliable personal identification methods in biometrics. With the rapid development of iris recognition verification, a number of its applications have been proposed until now including time attendance system etc. In this paper, a wireless iris recognition attendance management system is designed and implemented using Daugman’s Algorithm. This system based biometrics and wireless technique solves the problem of spurious attendance and the trouble of laying the corresponding network. It can make the users’ attendances more easily and effectively. .

**Paper Name**: Robust Face Recognition via Adaptive Sparse Representation.

**Author**: Jing Wang, Canyi Lu, Meng Wang, Member, IEEE, Peipei Li.

Sparse Representation (or coding) based Classification (SRC) has gained great success in face recognition in recent years. However, SRC emphasizes the 7 sparsity too much and overlooks the correlation information which has been demonstrated to be critical in real-world face recognition problems. Besides, some work considers the correlation but overlooks the discriminative ability of sparsity. Different from these existing techniques, in this paper, we propose a framework called Adaptive Sparse Representation based Classification (ASRC) in which sparsity and correlation are jointly considered. Specifically, when the samples are of low correlation, ASRC selects the most discriminative samples for representation, like SRC; when the training samples are highly correlated, ASRC selects most of the correlated and discriminative samples for representation, rather than choosing some related samples randomly. In general, the representation model is adaptive to the correlation structure, which benefits from both norm and norm. Extensive experiments conducted on publicly available data sets verify the effectiveness and robustness of the proposed algorithm by comparing it with state-of-the-art methods.

***CHAPTER-3***

***SYSTEM ANALYSIS***

**3.1 EXISTING SYSTEM:**

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. So an automatic attendance system can solve all above problems. There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique. Although it is automatic and a step ahead of traditional method it fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking.

**3.2 PROPOSED SYSTEM:**

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions.

**3.3 ANALYSIS MODEL:**

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model. The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: Requirement Analysis, Project Planning, System design, Detail design, Coding, Unit testing, System integration & testing. Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done. WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

****

**Fig.2 Waterfall model (SDLC)**

**3.4 MODULES:**

* **Input image:** The image is captured by any phone camera having resolution 13mp.
* **HOG algorithm:** This is the main algorithm used in our project for face detection purpose. After detecting faces, it plots the attendance for the person.
* **Excel sheet:** This is the output sheet which provides the present’s list. This excel sheet is connected to the college’s database for automated generation of the attendance.

***CHAPTER-4***

***FEASIBILITY STUDY***

**4. FEASIBILITY STUDY:**

Feasibility Study is a high level capsule version of the entire process intended to answer a number of questions like: What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? Feasibility study is conducted once the problem clearly understood. Feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system.

The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:

* Technical Feasibility.
* Economical Feasibility.
* Operational Feasibility.

In this phase, we study the feasibility of all proposed systems, and pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows.

**4.1 TECHNICAL FEASIBILITY**

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not. Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained. The system can be feasible because of the following grounds.

* All necessary technology exists to develop the system.
* This system is flexible and it can be expanded further.
* This system can give guarantee of accuracy, ease of use, and reliability.
* Our project is technically feasible because, all the technology needed for our

project is readily available.

**4.2. ECONOMICAL FEASIBILITY**

In this step, we verify which proposal is more economical. We compare the financial benefits of the new system with the investment. The new system is economically feasible only when the financial benefits are more than the investments and expenditure.

Economical Feasibility determines whether the project goal can be within the resource limits

allocated to it or not. It must determine whether it is worthwhile to process with the entire project or whether the benefits obtained from the new system are not worth the costs. Financial benefits must be equal or exceed the costs. In this issue, we should consider:

* The cost to conduct a full system investigation.
* The cost of h/w and s/w for the class of application being considered.
* The development tool.
* The cost of maintenance etc.

Our project is economically feasible because the cost of development is very minimal when

compared to financial benefits of the application.

**4.3. OPERATIONAL FEASIBILITY**

In this step, we verify different operational factors of the proposed systems like manpower, time etc., whichever solution uses less operational resources, is the best operationally feasible solution. The solution should also be operationally possible to implement. Operational Feasibility determines if the proposed system satisfied user objectives could be fitted into the current system operation. The present system Smart Traffic Control can be justified as operationally feasible based on the following grounds.

* The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements.
* The clients have been involved in the planning and development of the system.
* The proposed system will not cause any problem under any circumstances.

Our project is operationally feasible because the time requirements and personnel requirements are satisfied. We are a team of four members and we worked on this project for three working months.

***CHAPTER-5***

***SYSTEM REQUIREMENT SPECIFICATION***

**5.1 Introduction:**

A Software Requirements specification (SRS) – a requirements specification for a software

system- is a complete description of behavior of a system to be developed. It includes a set of cases that describe all he interactions users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are

requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

**System Requirements Specification:**

It is a collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Projects are subject to three sorts of require elements

* Business requirements describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For

instance, process requirements could specify methodologies that must be followed, and

constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the

activities that will be performed to satisfy a product requirement. For example, a maximum

development cost requirement (a process requirement) may be imposed to help achieve a

maximum sales price requirement ( a product requirement) a requirement that the product be

maintainable ( a product requirement) often is addressed by imposing requirements to follow

particular development styles.

A system engineering, a requirement can be a description of what a system must do, referred to as Functional Requirement. This type of requirement specifies something that the delivered

system must be able to do. Another type of requirement specifies something about the system

itself, and how well it performs its functions. Such requirements are often called Non-functional requirements, or ‘Performance requirements’ or ‘Quality of service requirements’. Examples of such requirements include usability, availability, reliability, supportability, testability and maintainability.

A collection of requirements define the characteristics or features of the desired system. A

‘good’ list of requirements as far as possible avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called “implementation bias” or “solution engineering”. However,

implementation constraints on the solution may validly be expressed by the future owner, for

example for required interfaces to external systems; for interoperability with other systems; and for commonality with other owned products.

**5.2 Functional Requirements:**

The Functional Requirements Specification gives the operations and activities that a system

must be able to perform. Functional requirements should include functions performed by

specific screens, outlines of work-flows performed by the system, and other business or

compliance requirements the system must meet. It also depends upon the type of software,

expected users and the type of system where the software is used.

**5.3 Non-Functional Requirements:**

In systems engineering, a non-functional requirement is a requirement that specifies

criteria that can be used to judge the operation of a system, rather than specific behaviors.

They are contrasted with functional requirements that define specific behavior or

functions.

**5.4 SYSTEM REQUIREMENTS:**

**5.4.1 HARDWARE REQUIREMENT:**

* Processor : Pentium –IV.
* Speed : 1.1 GHz.
* RAM : 512 MB (min).
* Hard Disk : 40 GB.
* Key Board : Standard Windows Keyboard.
* Mouse : Two or Three Button Mouse.
* Monitor : LCD/LED.
* Camera : Any smart phone camera with 13mp or above.

**5.4.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP.
* Coding Language : Java/J2EE.
* Database : My SQL.

***CHAPTER-6***

***SYSTEM DESIGN***

**6.1 INTRODUCTION:**

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

System Design is

* A creative process
  + No cook book solutions
* Goal driven
  + We create a design for solving some problem
* Constraint driven
  + By the function to be served and the constructions which are possible
* Good designs can be recognized
  + Simple, coherent, adequately meets requirements, adaptable.

System design transforms the analysis model by:-

* Defining the design goals of the project
* Decomposing the system into smaller subsystems
* Selection of off-the- shelf and legacy components
* Mapping subsystems to hardware
* Selection of persistent data management infrastructure
* Selection of access control policy
* Selection of global control flow mechanism
* Handling of boundary conditions

**6.2 HIGH LEVEL DESIGN**

**System Design**

Understanding bigger application with its external interfaces is called system design

**Subsystem Design**

Understanding bigger system into smaller independent working system is called

subsystem design

**6.3 LOW LEVEL DESIGN**

**UML DIAGRAMS**

* UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.
* The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.
* The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
* The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
* The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

**1.** Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

**2.** Provide extendibility and specialization mechanisms to extend the core concepts.

**3.** Be independent of particular programming languages and development process.

**4.** Provide a formal basis for understanding the modeling language.

**5.** Encourage the growth of OO tools market.

**6.** Support higher level development concepts such as collaborations, frameworks, patterns and components.

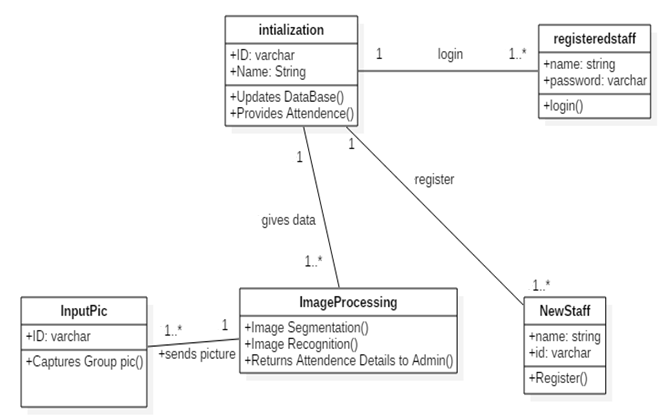
**7.** Integrate best practices.

**6.4 Class diagram:**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

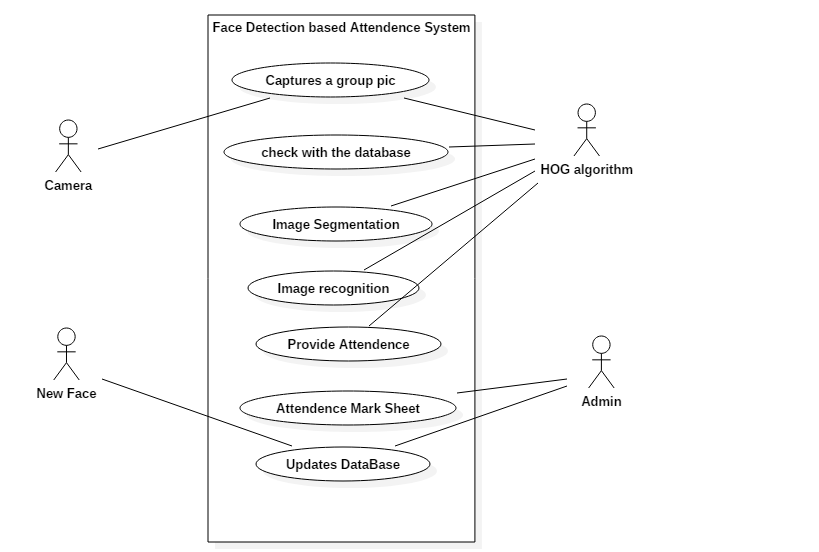
Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

****

**Fig.3: Class Diagram of the process**

**6.5 Use case Diagram:**

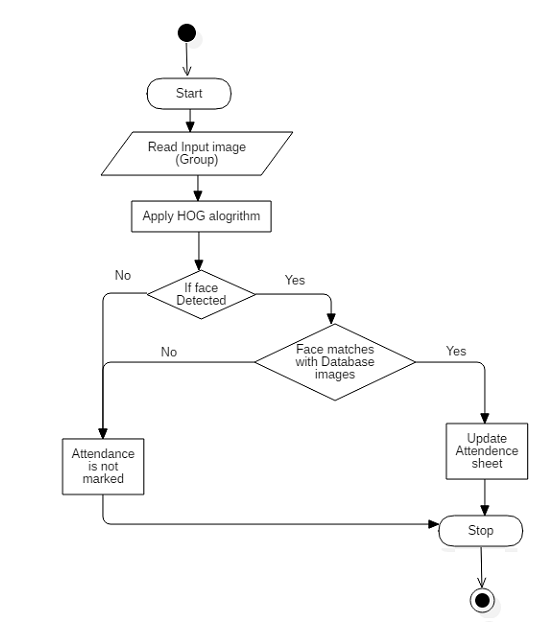
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

****

**Fig.4: Use case diagram of the process**

**6.6 Activity Diagram:**

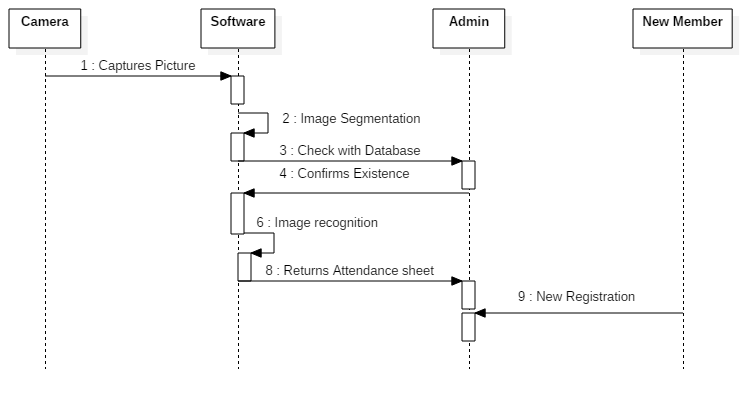
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc



**Fig.5: Activity Diagram of the process**

**6.7 Sequence Diagram:**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.



**Fig.6: Sequence diagram of the process**

***CHAPTER-7***

***CODING***

**7.1 Software Description**

**OpenCV:**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many start-ups such as Applied Minds, Video Surf, and Zeitera, that make extensive use of OpenCV.

OpenCV’s deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a template interface that works seamlessly with STL containers.

OpenCV is becoming widely known in computer vision field for its library that mainly built for computer vision computation. Another notable feature is the library is cross-platform, which is usable in Windows, Linux and MacOS. Such high portability reduces burdens and works if port of system is needed. Nowadays, OpenCV has huge collection of popular computer vision computation algorithm, and optical flow is one of them.

**What is Image Processing?**

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

The two types of methods used for Image Processing are Analog and Digital Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing.

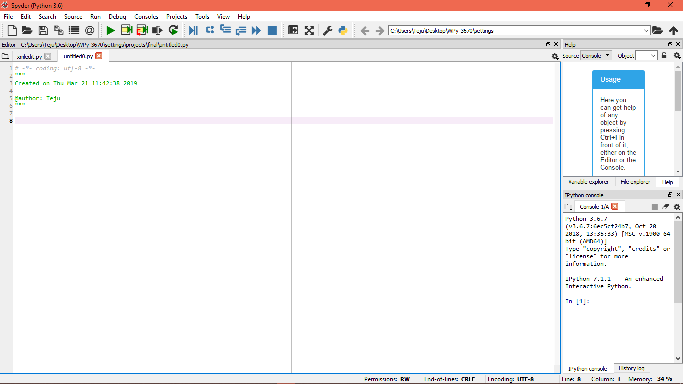
**Spyder:**

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection and beautiful visualization capabilities of a scientific package.

Furthermore, Spyder offers built-in integration with many popular scientific packages, including NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and more.

Beyond its many built-in features, Spyder’s abilities can be extended even further via first- and third-party plugins.

Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console or advanced editor, in your own software.



**Fig.7: Spyder working environment (IDE)**

**Histogram of Oriented Gradients-HOG algorithm:**

The HOG is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

**Algorithm implementation:**

### Step-1:

### Gradient computation:

The first step of calculation in many feature detectors in image pre-processing is to ensure normalized color and gamma values. As Dalal and Triggs point out, however, this step can be omitted in HOG descriptor computation, as the ensuing descriptor normalization essentially achieves the same result. Image pre-processing thus provides little impact on performance. Instead, the first step of calculation is the computation of the gradient values. The most common method is to apply the 1-D centered, point discrete derivative mask in one or both of the horizontal and vertical directions. Specifically, this method requires filtering the color or intensity data of the image with the following filter kernels:

***[-1, 0, 1] and [-1, 0, 1] T{\displaystyle [-1,0,1]{\text{ and }}[-1,0,1]^{\top }.\,}***

### Step-2:

### Orientation binning:

### The second step of calculation is creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is “unsigned” or “signed”.

### Step-3:

### Descriptor blocks:

### To account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then the concatenated vector of the components of the normalized cell histograms from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks. R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram. In the Dalal and Triggs human detection experiment, the optimal parameters were found to be four 8x8 pixels cells per block (16x16 pixels per block) with 9 histogram channels.

### Step-4:

### Block normalization:

### Let be the non-normalized vector containing all histograms in a given block, be its  *k*-norm for {\displaystyle k={1,2}}and {\displaystyle e} be some small constant.

### Then the normalization factor can be one of the following:

### 

### L2-hys: L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing

### 

### 

### Step-5:

### Object recognition:

### HOG descriptors may be used for object recognition by providing them as features to a machine learning algorithm. Dalal and Triggs used HOG descriptors as features in a support vector machine(SVM); however, HOG descriptors are not tied to a specific machine learning algorithm.

### hog.png

### Fig.8: HOG Algorithm

**Sample Code:**

**//HOG Algorithm**

new\_picture = face\_recognition.load\_image\_file(filename)

image = cv2.imread(filename)

rgb = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

boxes = face\_recognition.face\_locations(rgb,model='hog')

count=0;

for face\_encoding in face\_recognition.face\_encodings(new\_picture):

counte=4

deviation=0.62

while counte>1:

results = face\_recognition.compare\_faces(feature, face\_encoding, deviation)

counte=results.count(True)

deviation=deviation-0.02

print(counte)

name='' if True in results:

first\_match\_index = results.index(True)

name = names[first\_match\_index]

if not name == '':

cv2.rectangle(rgb,(boxes[count][1],boxes[count][0]),(boxes[count][3],boxes[count][2]), (255, 0, 0), 2)

cv2.putText(rgb,name, (boxes[count][3]+10, boxes[count][0]), 0, 0.5, (0, 255, 0), 2, cv2.LINE\_4)

sheet.cell(column=int(re.search(r'\d+',filename[len(filename)-5:len(filename)]).group())+1, row=first\_match\_index+2, value="present")

#worksheet.write(first\_match\_index+1, int(re.search(r'\d+',

filename[len(filename)-5:len(filename)]).group()),"present")

#print(filename+'detected--'+str(int(re.search(r'\d+',filename[len(filename)-5:len(filename)]).group())))

print('detected')

else:

print('detect failed')

count=count+1;

wb.save(filename ='test.xlsx')

bgr = cv2.cvtColor(rgb, cv2.COLOR\_RGB2BGR)

cv2.imwrite('detected.jpg',bgr)

cv2.imshow('image',bgr)

**training.py :**

onlyfiles = [f for f in listdir(individuals) if isfile(join(individuals, f))]

data=[]

stringall='';

loop=0

for individual in onlyfiles:

print(loop)

loop=loop+1

loadiamge = face\_recognition.load\_image\_file(individuals+individual)

face\_encoding = face\_recognition.face\_encodings(loadiamge)[0]

file = open("encodings.txt","a")

stringall='';

for x in face\_encoding:

stringall=stringall+str(x)+",\n"

file.write(stringall)

file.close()

file = open("encodingnames.txt","a")

file.write(individual+",")

file.close()

data.append(face\_encoding)

**For App:**

<application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 android:name="com.lfr.app.AppController">  
 <activity  
 android:name=".LoginActivity"  
 android:label="@string/app\_name">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".RegisterActivity"  
 android:label="@string/app\_name"  
 android:launchMode="singleTop"  
 android:windowSoftInputMode="adjustPan" />  
 <activity  
 android:name=".MainActivity"  
 android:label="@string/app\_name"  
 android:launchMode="singleTop" />

<activity  
 android:name=".UploadActivity"  
 android:label="@string/app\_name"  
 android:launchMode="singleTop" />  
  
 </application>

***CHAPTER-8***

***TESTING***

**TESTING:**

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

***TYPES OF TESTS***

***Unit Testing***

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the

application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

***Integration Testing***

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

***Functional Test***

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

* Valid Input : identified classes of valid input must be accepted.
* Invalid Input : identified classes of invalid input must be rejected.
* Functions : identified functions must be exercised.
* Output : identified classes of application outputs must be exercised.
* Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

***System Test***

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

***White Box Testing***

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

***Black Box Testing***

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

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

**7.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as

two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

● All user views must work properly.

● The distance and values should be clearly visible.

● There toast messages should be able to help the user understand the problem.

**Features to be tested**

● Verify that the distance and signal states are accurate.

● The coordinates checking must be then at periodic time intervals.

● The Bluetooth device list page should be able connect the application to the chosen

Bluetooth module.

**7.2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant Participation by the end user. It also ensures that the system meets the functional requirements.

**Test cases:**

|  |  |  |
| --- | --- | --- |
| **Test case ID** | **Test case condition** | **Output** |
| Case 1 | On giving invalid Login credentials | Displays a toast message as ‘Invalid User’ |
| Case 2 | On giving valid Login credentials | Allows the user to access the app |
| Case 3 | Registration of an existing user | Displays a toast message as ‘User already exists’ |
| Case 4 | Uploading a picture with the name as the date of that day | Sends the picture to the server |
| Case 5 | Uploading a picture with already existing date | Shows an error message |
| Case 6 | If all the faces are detected and recognized | Plots present for the people |
| Case 7 | If not detected | Blank space will be displayed |

**Table 1: Test cases to test the software**

***CHAPTER-9***

***SCREEN SHOTS***

### Graphical User Interface:

### Screenshot_20190312-111053.png Screenshot_20190312-111018.png

### Fig.9: Registration page Fig.10: Login page

### Screenshot_20190312-111112.png

### Fig.11: Page after logging in successfully

### Uploading pictures to server:



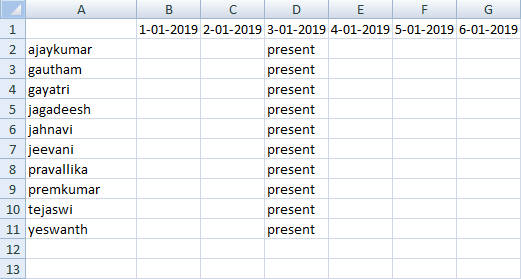
**Fig.12: Page to upload pictures to the server**

**Face detection:**



**Fig.13: After successful Face detection**

**Output Excel sheet:**



**Fig.14: Output Excel Sheet**

***CHAPTER-10***

***CONCLUSION***

**CONCLUSION:**

There may be various types of lighting conditions, seating arrangements and environments in various classrooms. Most of these conditions have been tested on the system and system has shown 100% accuracy for most of the cases. There may also exist students portraying various facial expressions, varying hair styles, beard, spectacles etc. All of these cases are considered and tested to obtain a high level of accuracy and efficiency. Thus, it can be concluded from the above discussion that a reliable, secure, fast and an efficient system has been developed replacing a manual and unreliable system. This system can be implemented for better results regarding the management of attendance and leaves. The system will save time, reduce the amount of work the administration has to do and will replace the stationery material with electronic apparatus and reduces the amount of human resource required for the purpose. Hence a system with expected results has been developed but there is still some room for improvement

The face detection and recognition algorithms were studied thoroughly taking number of the test from different varying condition images. For face detection combination of RGB and HSV model algorithm is used. For face recognition principal component analysis method is used. Attendance of the student are marked using the recognized face of every individual student and the data is stored in an attendance sheet. The attendance of every student marked automatically by recognizing their face with the face present in the data base.

***CHAPTER-11***

***FUTURE ENHANCEMENT***

**FUTURE ENHANCEMENT:**

* Currently, the system has reached the accuracy level up to 80% for partial and dense images. It can further be improved to obtain higher accuracy levels.
* Further work can be done on this project to alert the student by sending SMS regarding the attendance. For this purpose GSM module can be used. SMS alert can be given to the parent of the student.
* Further, 2 or more IP cameras can be employed and each image can be processed separately. The results of these can be merged to obtain better results and accuracy in denser classrooms.

***CHAPTER-12***

***BIBILIOGRAPHY***

**BIBILIOGRAPHY**

[1]. Arulogun O. T., Olatunbosun, A., Fakolujo O. A., and Olaniyi, O. M. RFID-Based Students Attendance Management System. International Journal of Scientific & Engineering Research Volume 4, Issue 2, February-2013.

[2]. Chitresh Saraswat and Amit Kumar. An Efficient Automatic Attendance System using Fingerprint Verification Technique. International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 264-269. International Journal of Advances in Engineering & Technology, July, 2014. ©IJAET ISSN: 22311963 979 Vol. 7, Issue 3, pp. 974-979

[3]. Jobin J., Jiji Joseph, Sandhya Y.A, Soni P. Saji, Deepa P.L.. Palm Biometrics Recognition and Verification System. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 2, August 2012.

[4]. Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Rudra Pal. Study of Implementing Automated Attendance System Using Face Recognition Technique. International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012.

[5]. Yogesh Tayal, Ruchika Lamba, Subhransu Padhee. Automatic Face Detection Using Color Based Segmentation. International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012.

[6]. Omaima N. A. AL-Allaf. Review Of Face Detection Systems Based Artificial Neural Networks Algorithms. The International Journal of Multimedia & Its Applications (Ijma) Vol.6, No.1, February 2014.

[7]. Deepak Ghimire and Joonwhoan Lee. A Robust Face Detection Method Based on Skin Color and Edges. J Inf Process Syst, Vol.9, No.1, March 2013.

[8]. Sanjay Kr. Singh, D. S. Chauhan, Mayank Vatsa, Richa Singh. A Robust Skin Color Based Face Detection Algorithm. Tamkang Journal of Science and Engineering, Vol. 6, No. 4, pp. 227-234 (2003). [9]. Viola P. & Jones J...Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154, 2004.

[10]. Mayank Agarwal, Nikunj Jain, Mr. Manish Kumar and Himanshu Agrawal. Face Recognition Using Eigen Faces and Artificial Neural Network. International Journal of Computer Theory and Engineering, Vol. 2, No. 4, August, 2010.

[11]. Bayan Ali Saad Al-Ghamdi, Sumayyah Redhwan Allaam. Recognition of Human Face by Face Recognition System using 3D. Journal of Information & Communication Technology Vol. 4, No. 2, (Fall 2010) 27-34. [12]. Hafiz Imtiaz and Shaikh Anowarul Fattah. A Face Recognition Scheme Using Wavelet - Based Dominant Features. Signal & Image Processing: An International Journal (SIPIJ) Vol.2, No.3, September 2011. [13]. Jagadeesh H S, Suresh Babu K, and Raja K B. DBC based Face Recognition using DWT. Signal & Image Processing: An International Journal (SIPIJ) Vol.3, No.2, April 2012.

[14]. Aruna Bhat. Medoid Based Model For Face Recognition Using Eigen And Fisher Faces. International Journal of Soft Computing, Mathematics and Control (IJSCMC), Vol. 2, No. 3, August 2013.

[15]. Kandla Arora. Real Time Application of Face Recognition Concept. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-5, November 2012.