ATTENDANCE REGISTRATION USING FACIAL RECOGNITION

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

By

LITHWIN R (Reg No - 39110571)
LOGESH K (Reg No -39110573)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE ANDTECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL- 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **LITHWIN R (39110571) and LOGESH K (39110573)** who carried out the Project Phase-2 entitled **"ATTENDANCE REGISTRATION USING FACIAL RCOGNITION"** under my supervision from JANUARY 2023 to APRIL 2023

Internal Guide

Dr. A. MARY POSONIA M.E., Ph.D.

Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

Submitted for Viva voce Examination held on 24.04.2023

Internal Examiner **External Examiner**

DECLARATION

I, **LITHWIN R (39110571) ,**hereby declare that the Project Phase-2 Report entitled done by me under the guidance of **Dr.A. MARY POSONIA, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 24.04.2023

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, **Professor & Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D., Professor & Head,** Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Mary posonia M.E., Ph.D. Professor**, Department of Computer Science and Engineering, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Face recognition is one of the most useful applications for image processing and plays a crucial role in the technical field. For authentication purposes, specifically in the context of student attendance, face recognition is a pressing issue. The process of recognizing students using face bio-statistics based on high-definition monitoring and other computer technologies is known as an attendance system using face recognition. The goal of developing this system is to digitize the old method of taking attendance by calling names and keeping pen-and-paper records. The current methods for taking attendance are time-consuming and arduous. Manual recording makes it simple to alter attendance records. Both the current biometric systems and the conventional method for taking attendance are susceptible to proxies. As a result, it is suggested that this paper address all of these issues. Har classifiers, KNN, CNN, SVM, generative adversarial networks, and Gabor filters are utilized in the proposed system. Attendance reports will be created and saved in Excel format following face recognition. The system is put through its paces under a variety of conditions, including lighting, head movements, and varying distances between the student and the cameras. The overall complexity and accuracy are determined after extensive testing.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

We address the problem of 3-D face tracking, which is a fundamental problem in the context of human computer interaction. The input consists in a video stream acquired by a low-cost webcam pointing to the user's face. Our efforts focus on real-time and robust full 3-D face tracking (6 DOF), in the presence of facial expression changes, wide pose variations, and partial occlusions. To this end, we make very few assumptions compared to most of the state-of-the-art approaches. The distance to the camera should comply with the requirements of a facial interactive application, but can vary according to the focal length used. Also, we require a near frontal pose at initialization. The proposed approach includes the use of a generic 3-D face model to track in real-time the 3-D head motion of users and produce accurate and textured face models. As illustrated in Fig. 1, our application is composed of three main modules: initial 3-D model fitting, 3-D head tracking, and re-acquisition.The applications of the 3-D tracker and 3-D model are not discussed here. We need a 3- D model, which can either be a generic model, or a specific one retrieved from a database. In the initial 3-D modeling step, the model is warped orthogonally to the focal axis in order to fit to the user's face in an input image, by matching 2-D facial landmarks extracted at run time. The initial 3-D reconstruction of the face model is a critical step.

Face detection is a computer technology that is being applied for many different applications that require the identification of human faces in digital images or video. It can be regarded as a specific case of object-class detection, where the task is to find the locations and sizes of all objects in an image that belongs to a given class. The technology is able to detect frontal or near-frontal faces in a photo, regardless of orientation, lighting conditions or skin colour.

CHAPTER 2

LITERATURE SURVEY

Yen-Lin Chen et al (2019) accurate camera and robust facial capture using a single rgbd camera This paper presents an automatic and robust approach that accurately captures high- quality 3D facial performances using a single RGBD camera. The key of our approach is to combine the power of automatic facial feature detection and image- based 3D non rigid registration techniques for 3D facial reconstruction. In particular, we develop a robust and accurate image-based non rigid registration algorithm that incrementally deforms a 3D template mesh model to best match observed depth image data and important facial features detected from single RGBD images. The whole process is fully automatic and robust be-cause it is based on single frame facial registration frame-work. The ability to accurately capture 3D facial performances has many applications including animation, gaming, human-computer interaction, security, and telepresence.

Adrian et al (2016) face detection with a 3D model This paper presents a part-based face detection approach where the spatial relationship between the face parts is represented by a hidden 3D model with six parameters. The computational complexity of the search in the six dimensional pose space is addressed by proposing meaningful 3D pose candidates by image-based regression from detected face key point locations. The 3D pose candidates are evaluated using a parameter sensitive classifier based on difference features relative to the 3D pose. A compatible subset of candidates is then obtained by non-maximal suppression. Experiments on two standard face detection datasets show that the proposed 3D model based approach obtains results comparable to or better than state of the art. Face recognition has been a hot research area for its wide range of applications

Jangmoo choi et al (2020)) real gime 3D face- tracking and modeling from a webcam This paper infer a 3-D face model from a single frontal image using automatically extracted 2-D landmarks and deforming a generic 3-D model. Then, for any input image, we extract feature points and track them in 2-D. Given these correspondences, sometimes noisy and incorrect, we robustly estimate the 3-D head pose using PnP and a RANSAC process. As the head moves, we dynamically add new feature points to handle a large range of poses.

Sergey tulyakov et al (2019) facecept real time 3D tracking and analysis This paper presents an open source cross platform technology for 3D face tracking and analysis. It contains a full stack of components for complete face understanding: detection, head pose tracking, facial expression and action unit's recognition. Given a depth sensor, one can combine FaceCept3D modules to fulfill a specific application scenario. Key advantages of the technology include real time processing speed and ability to handle extreme head pose variations. There is one important constraint shared by all these scenarios when solving the above-mentioned tasks: non- invasiveness,i.e.the solution must not hinder the naturalness of the subject's behavior. Consequently, the vision sensors are typically placed out of the direct sight of the subject. FaceCept3D is motivated by challenges arising from these types of scenarios and is able to successfully address them in a unified, open source and cross-platform solution. Additionally, our system can be deployed in a much broader spectrum of applications (e.g. Those cases for which the face is not fully visible to the sensor), being able to maintain state-of-the-art performance.

N. Bayramoglu et al (2017) development of real time face recognition system using opencv This paper represents A real-time, GUI based automatic Face detection and recognition system is developed in this project. It can be used as access control system by registering the staff or students of an organization with their faces, and later it will recognize the people by capturing their images with faces, when they are entering or leaving the premises. The system is implemented on a desktop with a Graphical User Interface, Initially it detects the faces in the images that are grabbed from a web camera. All the tools and operating, used to develop this system like Ubuntu, open Face, Python ..., are open source tools. This real time GUI based face recognition system is developed using Open source tool Open face. Open Face is the face recognition tool developed by Carnegie Mellon University, using Open CV. Open Face, consists in a broader Prospective, three phases: Detection, Feature extraction, and Recognition. The dimensionality of face image is reduced by the Histogram of Oriented Gradients (HOG) and this algorithm is developed to detect frontal views of faces. After detecting the face part of image, extract the 128 face features for the given image by using a Deep Neural Network algorithm and the recognition is done by the Support Vector machine (SVM)

L. Jurjević et al (2020) Realtime data acquisition based on OpenCV for close-range photogrammetry applications This paper represents the Development of the technology in the area of the cameras, computers and algorithms for 3D the reconstruction of the objects from the images resulted in the increased popularity of the photo gramme try. Algorithms for the 3D model reconstruction are so advanced that almost anyone can make a 3D model of photographed object. The main goal of this paper is to examine the possibility of obtaining 3D data for the purposes of the close-range photo gramme try applications, based on the open source technologies. All steps of obtaining 3D point cloud are covered in this paper.

Special attention is given to the camera calibration, for which two-step process of calibration is used. Both, presented algorithm and accuracy of the point cloud are tested by calculating the spatial difference between referent and produced point clouds. During algorithm testing, robustness and swiftness of obtaining 3D data is noted, and certainly usage of this and similar algorithms has a lot of potential in the real-time application. That is the reason why this research can find its application in the architecture, spatial planning, protection of cultural heritage, forensic, mechanical engineering, traffic management, medicine and other sciences.

Kristjan Krips et al (2019) creating a real module from a video by a single camera This paper represents that It has been shown by that a 3D model can be constructed from a video made by a stereo camera. However, stereo cameras are yet not widely available and thus in this paper we show how a 3D model could be constructed from a video created by a single camera. As creating a 3D model with a single camera is more di- cult than with a stereo camera we assume that the videos used for modelling are taken in strict conditions. We propose two different approaches for solving this problem. The results of this paper show that these methods can be used to create a 3D model. However, the results are visually not comparable to real 3D models, both of these methods

2.1 INFERENCE OF LITERATURE SURVEY

Facial recognition technology has been a hot research area in recent years due to its wide range of applications, such as security, human-computer interaction, animation, gaming, telepresence, and more. There are several approaches to implementing facial recognition using a single RGBD camera, 3D models, and open-source technologies. Yen-Lin Chen et al (2019) presented an automatic and robust approach that accurately captures high-quality 3D facial performances using a single RGBD camera. The key to their approach is to combine the power of automatic facial feature detection and image-based 3D non-rigid registration techniques for 3D facial reconstruction. Adrian et al (2016) presented a part-based face detection approach where the spatial relationship between the face parts is represented by a hidden 3D model with six parameters. The computational complexity of the search in the six-dimensional pose space is addressed by proposing meaningful 3D pose candidates by image-based regression from detected face key point locations. Jangmoo Choi et al (2020) infer a 3D face model from a single frontal image using automatically extracted 2D landmarks and deforming a generic 3D model. Then, for any input image, they extract feature points and track them in 2D. Given these correspondences, sometimes noisy and incorrect, they robustly estimate the 3D head pose using PnP and a RANSAC process .Sergey Tulyakov et al (2019) presented an open-source cross-platform technology for 3D face tracking and analysis. It contains a full stack of components for complete face understanding: detection, head pose tracking, facial expression and action unit's recognition. Given a depth sensor, one can combine FaceCept3D modules to fulfill a specific application scenario.N. Bayramoglu et al (2017) developed a real-time, GUI-based automatic face detection and recognition system using open source tools like OpenCV and Open Face. The system can be used as an access control system by registering the staff or students of an organization with their faces, and later it will recognize the people by capturing their images with faces when they are entering or leaving the premises'. Jurjević et al (2020) examined the possibility of obtaining 3D data for the purposes of the close-range photogrammetry applications based on open-source technologies. All steps of obtaining 3D point cloud are covered in their paper, and special attention is given to the camera calibration.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 GENERAL REQUIREMENTS

SOFTWARE REQUIRED

Open CV (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

Open CV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe

APPLICATIONS

OpenCV's application areas include:

- 2D and 3D feature toolkits.
- Egomotion estimation.
- Facial recognition system.
- Gesture recognition.
- Human–computer interaction (HCI).
- Mobile robotics.
- Motion understanding.
- Object identification.
- Segmentation and recognition.
- Stereopsis stereo vision: depth perception from 2 cameras.
- Structure from motion (SFM).
- Motion tracking.
- Augmented reality.

To support some of the above areas, Open CV includes a statistical machine learning library that contains:

- Boosting.
- Decision tree learning.
- Gradient boosting trees.
- Expectation-maximization algorithm.
- k-nearest neighbor algorithm.
- Naive Bayes classifier.
- Artificial neural networks.
- Random forest.
- Support vector machine (SVM).

HARDWARE REQUIREMENTS

If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

A CUDA - based GPU interface has been in progress since September 2010.

An OpenCL-based GPU interface has been in progress since October 2012, documentation for version 2.4.13.3 can be found at docs.opencv.org.

OS SUPPORT

Open CV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. Open CV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10. The user can get official releases from SourceForge or take the latest sources from GitHub. Open CV uses CMake

3.2 REQUIREMENT OF PROGRAMMING KNOWLEDGE

Open CV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell, and Ruby have been developed to encourage adoption by a wider audience.

Since version 3.4, Open CV.js is a JavaScript binding for selected subset of Open CV functions for the web platform. All of the new developments and algorithms in Open CV are now developed in the C++ interface.

PYTHON (PROGRAMMING LANGUAGE)

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open-source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

- The proposed system is intended to improve the fault of an existing manual system by allowing for the attendance of the various organizations.
- Student information is added manually by a supervisor. There are two distinct aspects to our system.
- This project's major goal is to provide an innovative method for organizations to track student attendance through an attendance taking system. and the second is the project's back end, which is composed of logic and is built using Python machine learning.

Advantages OF PROPOSED SYSTEM

- When there is a clear line of demarcation between classes, SVM performs reasonably well.
- SVM uses less memory and performs better in high-dimensional spaces.

4.2 RECOGNITION ALGORITHMS CAN BE DIVIDED INTO TWO MAIN APPROACHES

1. **Geometric:** Is based on geometrical relationship between facial landmarks, or in other words the spatial configuration of facial features. That means that the main geometrical features of the face such as the eyes, nose and mouth are first located and then faces are classified on the basis of various geometrical distances and angles between features.
2. **Photo-metric stereo:** Used to recover the shape of an object from a number of images taken under different lighting conditions. The shape of the recovered object is defined by a gradient map, which is made up of an array of surface normal (Zhao and Chellappa, 2006).

POPULAR RECOGNITION ALGORITHMS INCLUDE

1. Principal Component Analysis using Eigen-faces, (PCA)

2. Linear Discriminate Analysis,
3. Elastic Bunch Graph Matching using the Fisher-face algorithm.

4.4 SYSTEM ARCHITECTURE DIAGRAM

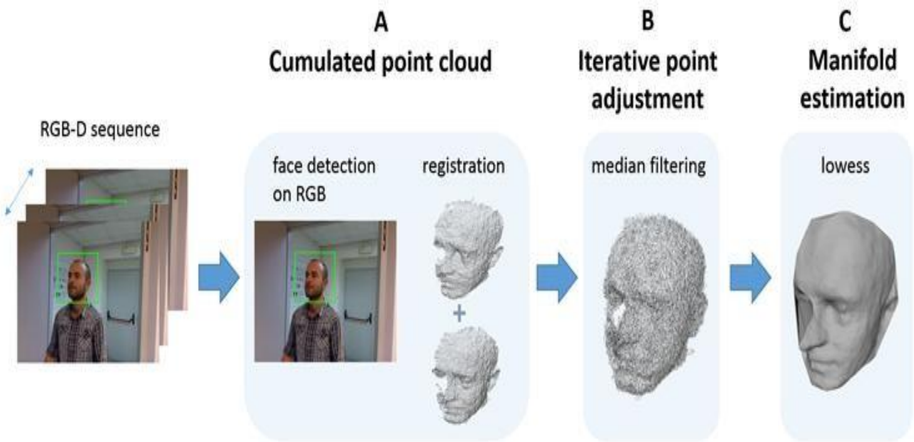


Fig 4.4: Architecture Diagram

This component processes the facial images captured by the attendance system and matches them with the registered user's facial recognition data in the database.

These components handle the attendance and user management logic respectively.

They communicate with the database and Facial Recognition API to retrieve and store user attendance and user data.

This component provides an interface for users to interact with the system. It allows users to view their attendance and update their personal information.

It communicates with the Attendance System to retrieve user attendance and update user information.

The attendance register system using facial recognition works by capturing the facial images of users, processing them using the Facial Recognition API, and storing attendance and user data in the database.

4.5 FEATURE SEARCHING METHODS

VIOLA JONES METHOD

Paul Viola and Michael Jones presented an approach for object detection which minimizes computation time while achieving high detection accuracy. Paul Viola and Michael Jones proposed a fast and robust method for face detection which is 15 times quicker than any technique at the time of release with 95% accuracy at around 17 fps.The technique relies on the use of simple Haar-like features that are evaluated quickly through the use of a new image representation. Based on the concept of an ―Integral Image‖ it generates a large set of features and uses the boosting algorithm Ada Boost to reduce the over complete set and the introduction of a degenerative tree of the boosted classifiers provides for robust and fast interference. The detector is applied in a scanning fashion and used on gray-scale images, the scanned window that is applied can also be scaled, as well as the features

Gabor featured method

Sharif et al proposed an Elastic Bunch Graph Map (EBGM) algorithm that successfully implements face detection using Gabor filters. The proposed system applies 40 different Gabor filters on an image. As a result of which 40 images with different angles and orientation are received. Next, maximum intensity points in each filtered image are calculated and mark them as fiducially points. The system reduces these points in accordance to distance between them. The next step is calculating the distances between the reduced points

Eigen faces Method

An early example of employing eigen vectors in face recognition was done by Kohonen in which a simple neural network is demonstrated to perform face recognition for aligned and normalized face images. Kirby and Sirovich suggested that images of faces can be linearly encoded using a modest number of basis images. The idea is arguably proposed first by Pearson in 1901 and then by HOTELLING in 1933 .Given a collection of n by m pixel training.

Images represented as a vector of size m X n, basis vectors spanning an optimal subspace are determined such that the mean square error between the projection of the training images onto this subspace and the original images is minimized.They call the set of optimal basis vectors Eigen pictures since these are simply the eigen vectors of the covariance matrix computed from the vectored face images in the training set.Experiments with a set of 100 images show that a face image of 91 X 50 pixels can be effectively encoded using only 50 Eigen pictures.

STATISTICAL APPROCHS

Support Vector Machine (SVM*)*

SVMs were first introduced Osuna et al. for face detection. SVMs work as a new paradigm to train polynomial function, neural networks, or radial basis function (RBF) classifiers.SVMs works on induction principle, called structural risk minimization, which targets to minimize an upper bound on the expected generalization error. An SVM classifier is a linear classifier where the separating hyper plane is chosen to minimize the expected classification error of the unseen test patterns. In Osunaet al. developed an efficient method to train an SVM for large scale problems, and applied it to face detection. Based on two test sets of 10,000,000 test patterns of 19 X 19 pixels, their system has slightly lower error rates and runs approximately30 times faster than the system by Sung and Poggio . SVMs have also been used to detect faces and pedestrians in the wavelet domain

CONSTELLATION METHOD

All methods discussed so far are able to track faces but still some issue like locating faces of various poses in complex background is truly difficult. To reduce this difficulty investigator form a group of facial features in face-like constellations using more robust modelling approaches such as statistical analysis. Various types of face constellations have been proposed by Burl et al. . They establish use of statistical shape theory on the features detected from a multiscale Gaussian derivative filter. Huang et al. also apply a Gaussian filter for per-processing in a framework based on image feature analysis.

NEURAL NETWORK

Neural networks gaining much more attention in many pattern recognition problems, such as OCR, object recognition, and autonomous robot driving. Since face detection can be treated as a two class pattern recognition problem, various neural network algorithms have been proposed. The advantage of using neural networks for face detection is the feasibility of training a system to capture the complex class conditional density of face patterns. However, one demerit is that the network architecture has to be extensively tuned (number of layers, number of nodes, learning rates, etc.) to get exceptional performance. In early days most hierarchical neural network was proposed by Agui et al. [43]. The first stage having two parallel sub networks in which the inputs are filtered intensity values from an original image. The inputs to the second stage network consist of the outputs from the sub networks and extracted feature values. An output at the second stage shows the presence of a face in the input region.Propp and Samal developed one of the earliest neural networks for face detection . Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.Feraud and Bernier presented a detection method using auto associative neural networks . The idea is based on which shows an auto associative network with five layers is able to perform a nonlinear principal component analysis. One auto associative network is used to detect frontal-view faces and another one is used to detect faces turned up to 60 degrees to the left and right of the frontal view. After that Lin et al. presented a face detection system

.CHAPTER 5

IMPLEMENTATION DETAILS

5.1 MODULES

A facial recognition-based attendance registration system typically consists of several key modules that work together to capture, process, and verify the identities of individuals based on their facial features. Here are some common modules in an attendance registration system using facial recognition:

Image Acquisition: This module captures images of individuals' faces using cameras or other imaging devices. The images are usually taken in real-time as individuals approach the registration point.

Preprocessing: The acquired images are then preprocessed to enhance their quality and reduce noise. This may involve tasks such as resizing, cropping, and normalization to ensure consistent image quality for further processing.

Face Detection: In this module, facial detection algorithms are used to locate and extract faces from the preprocessed images. These algorithms can detect faces based on certain facial features, such as eyes, nose, and mouth, and create a bounding box around the detected faces.

Feature Extraction: Once the faces are detected, the next step is to extract relevant features from the facial images. These features are usually represented as numerical values or vectors that capture the unique characteristics of an individual's face, such as the shape, texture, and color of the facial features.

Face Matching: In this module, the extracted facial features are compared with the stored facial templates or references in a database to identify potential matches. Various algorithms such as eigenfaces, local binary patterns (LBP), or deep neural networks (DNNs) can be used for face matching.

Attendance Recording: After successful face matching, the attendance of the individual is recorded in a database along with the timestamp and other relevant information, such as the date and location of the registration.

User Interface: A user interface module provides an interface for users to interact with the attendance registration system. This may include displaying registration results, providing feedback to users, and allowing administrators to manage the system, such as adding or deleting registered individuals.

Database Management: This module handles the storage and retrieval of facial templates or references, as well as attendance records. It may include functions such as database creation, data insertion, data deletion, and data retrieval.

Security and Privacy: As facial recognition systems may involve sensitive biometric data, security and privacy modules are essential to protect the data and ensure compliance with relevant regulations. This may include encryption of data, access control, and data privacy measures.

System Integration: This module ensures that all the components of the attendance registration system, including hardware, software, and networking components, are integrated and work seamlessly together.

These are some of the common modules that are typically involved in an attendance registration system that uses facial recognition technology. The specific implementation and functionalities of each module may vary depending on the requirements and design of the system.

5.2 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING

1. IMAGE ACQUISITION

This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling etc.

2. IMAGE ENHANCEMENT

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as, changing brightness & contrast etc.

3. IMAGE RESTORATION

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

4. COLOR IMAGE PROCESSING

Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modelling and processing in a digital domain etc.

5. WAVELETS AND MULTIRESOLUTION PROCESSING

Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

6. COMPRESSION

Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to
compress data.

7. MORPHOLOGICAL PROCESSING

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

8. SEGMENTATION

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

9. REPRESENTATION AND DESCRIPTION

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

10. OBJECT RECOGNITION

Recognition is the process that assigns a label, such as, “vehicle” to an object based on its descriptors

As CNN based learning algorithm shows better performance on the classification issues, the rich labeled data could be more useful in the training stage. 3D object classification and pose estimation is a jointed mission aiming at separate different posed apart in the descriptor form.

In the training stage, we prepare 2D training images generated from our module with their class label and pose label. We fully exploit the information lies in their labels by using a triplet and pair-wise jointed loss function in CNN training.

As CNN based learning algorithm shows better performance on the classification issues, the rich labeled data could be more useful in the training stage. 3D object classification and pose estimation is a jointed mission aiming at separate different poses apart in the descriptor form.

In the training stage, we prepare 2D training images generated from our module with their class label and pose label. We fully exploit the information that lies in their labels by using a triplet and pair-wise jointed loss function in CNN training.

Both class and pose label are in consideration in the triplet loss. The loss score will be smaller when features from the same class and same pose is more similar and features from different classes or different poses will lead to a much larger loss score.

This loss is also jointed with a pair wise component to make sure the loss is never be zero and have a restriction on the model scale.

About the training and feature extraction process, it is a rough implementation by using Open CV and Caffe from the idea of Paul Wohlhart. The principal purpose of this API is constructing a well labeled database from .ply models for CNN training with triplet loss and extracting features with the constructed model for prediction or other purpose of pattern recognition, algorithms into two main Class:

Open CV is a library for real-time computer vision. It has very powerful functions that make the art of processing images and getting information about them easy. In this post, we will review some of the functions that we used for making a 3D-reconstruction from an image in order to make an autonomous robotic arm.
Open CV uses a pinhole camera model. This model works by projecting 3D points onto the image plane using a perspective transformation.
There are some functions of Open CV that help us accomplish our goal. These functions work with a chess board model to calibrate the model, so the first thing is to get a chess board and take some photos of it. We took several

photographs in order to get a better calibration.

It has come to my attention that most 3D reconstruction tutorials out there are a bit lacking. Don't get me wrong they're great, but they're fragmented or go too deep into the theory or a combination of both. Worse yet they use specialized datasets (like Tsukuba) and this is a bit

of a problem when it comes to using the algorithms for anything outside those datasets (because of parameter tuning).

I believe that the cool thing about 3D reconstruction (and computer vision in general) is to reconstruct the world around you, not somebody else's world (or dataset). This tutorial is a humble attempt to help you recreate your own world using the power of Open CV.

Simply put this tutorial will take you from scratch to point cloud USING YOUR OWN PHONE CAMERA and pictures. So without further ado,

5.3 The Steps Required for 3D Reconstruction:

There are many ways to reconstruct the world around but it all reduces down to getting an actual depth map.
A depth map is a picture where every pixel has depth information (instead of color information). It is normally represented like a grayscale picture.

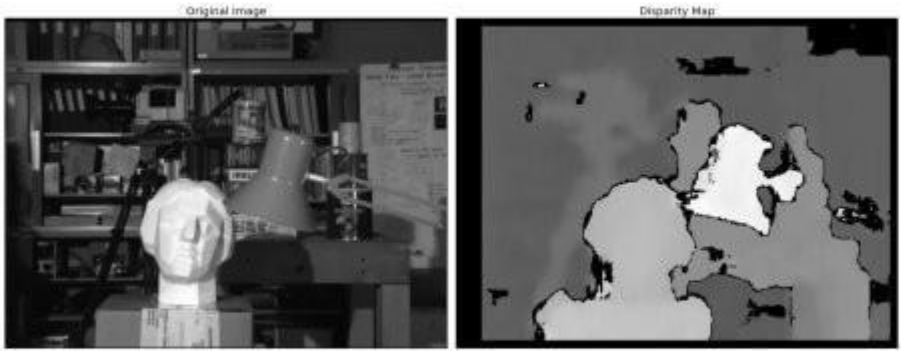


Fig 6.4: Original Image and Depth Map.

As mentioned before there are different ways to obtain a depth map and these depend on the sensor being used. A type of sensor could be a simple camera (from now on called RGB camera in this text) but it is possible to use others like LiDAR or infrared or a combination.

The type of sensor will determine the accuracy of the depth map. In terms of accuracy it normally goes like this: LiDAR > Infrared > Cameras. Depth maps can also be colorized to better visualize depth.

Fig 6.5: Image Processing.

Depending on the kind of sensor used, there are more or less steps required to actually get the depth map. The Kine-ct camera for example uses infrared sensors combined with RGB cameras and as such you get a depth map right away (because it is the information processed by the infrared sensor).

But what if you don’t have anything else but your phone camera?. In this case you need to do stereo reconstruction. Stereo reconstruction uses the same principle your brain and eyes use to actually understand depth.

The gist of it consists in looking at the same picture from two different angles, look for the same thing in both pictures and infer depth from the difference in position. This is called stereo matching.

In order to do stereo matching it is important to have both pictures have the exact same characteristics. Put differently, both pictures shouldn’t have any distortion. This is a problem because the lens in most cameras causes distortion. This means that in order to accurately do stereo matching one needs to know the optical centers and focal length of the camera.
In most cases this information will be unknown (especially for your phone camera) and this is why stereo 3D reconstruction requires the following steps:

1. **Camera calibration**: Use a bunch of images to infer the focal length and optical centers of your camera

2. **Un-distort images**: Get rid of lens distortion in the pictures used for reconstruction

3. **Feature matching**: Look for similar features between both pictures and build a depth map

4. **Re-project points**: Use depth map to re-project pixels into 3D space.

5. **Build point cloud**: Generate a new file that contains points in 3D space for visualization.

6. Build mesh to get an actual 3D model (outside of the scope of this tutorial, but coming soon in different tutorial).

Step 1 only needs to be executed once unless you change cameras. Steps 2–5 are required every time you take a new pair of pictures…and that is pretty much it.

The actual mathematical theory (the why) is much more complicated but it will be easier to tackle after this tutorial since you will have a working example that you can experiment with by the end of it.

Development of the technology in the area of the cameras, computers and algorithms for 3D the reconstruction of the objects from the images resulted in the increased popularity of the photo grammetry. Algorithms for the 3D model reconstruction are so advanced that almost anyone can make a 3D model of photographed object. The main goal of this paper is to examine the possibility of obtaining 3D data for the purposes of the close-range photo grammetry applications, based on the open-source technologies. All steps of obtaining 3D point cloud are covered in this paper. Special attention is given to the camera calibration, for which two-step process of calibration is used. Both, presented algorithm and accuracy of the point cloud are tested by calculating the spatial difference between referent and produced point clouds. During algorithm testing, robustness and swiftness of obtaining 3D data is noted, and certainly usage of this and similar algorithms has a lot of potential in the real-time application. That is the reason why this research can find its application in the architecture, spatial planning, protection of cultural heritage, forensic, mechanical engineering between training and testing images.

DIFFERENT APPROACHES OF FACE RECOGNITION

There are two predominant approaches to the face recognition problem: Geometric (feature based) and photo-metric (view based). As researcher interest in face recognition continued, many different algorithms were developed, three of which have been well studied in face recognition literature.

FACE DETECTION SYSTEM CAN BE DIVIDED INTO THE FOLLOWING STEPS

1. **Pre-Processing:** To reduce the variability in the faces, the images are processed before they are fed into the network. All positive examples that is the face images are obtained by cropping images with frontal faces to include only the front view. All the cropped images are then corrected for lighting through standard algorithms.
2. **Classification:** Neural networks are implemented to classify the images as faces or non faces by training on these examples. We use both our implementation of the neural network and the neural network toolbox for this task. Different network configurations are experimented with to optimize the results.
3. **Localization:** The trained neural network is then used to search for faces in an image and if present localize them in a bounding box. Various Feature of Face on which the work has done on:- Position Scale Orientation Illumination. Face detection is a computer technology that determines the location and size of human face in arbitrary (digital) image. The facial features are detected and any other objects like trees, buildings, and bodies etc are ignored from the digital image. It can be regarded as a

   ‗specific' case of object-class detection, where the task is finding the location and sizes of all objects in an image that belong to a given class. Face detection, can be regarded as a more ‗general' case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). Basically, there are two types of approaches to detect facial part in the given image i.e. feature base and image base approach. Feature base approach tries to extract features of the image and match it against the knowledge of the face features. While image base approach tries to get best match

   qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps.

   The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).

5.4 FUNDAMENTAL STEPS IN IMAGE PROCESSING

Fundamental steps in image processing are

1. **Image acquisition**: to acquire a digital imag
2. **Image pre-processing**: to improve the image in ways that increases the chances for success of the other processes.
3. **Image segmentation**: to partitions an input image into its constituent parts of objects.
4. **Image segmentation**: to convert the input data to a from suitable for computer processing.
5. **Image description**: to extract the features that result in some quantitative information of interest of features that are basic for differentiating one class of objects from another.
6. **Image recognition**: to assign a label to an object based on the information provided by its description.

A SIMPLE IMAGE FORMATION MODEL

Image are denoted by two-dimensional function f(x, y).f(x, y) may be characterized by 2 components:

1. The amount of source illumination i(x, y) incident on the scene.
2. The amount of illumination reflected r(x, y) by the objects of the scene.

3. f(x, y) = i(x, y)r(x, y), where 0 < i(x,y) < and 0 < r(x, y) < 1.

Format name	Full name	Description	Recognized extensions
TIFF	Tagged Image File Format	A flexible file format supporting a variety image compression standards including JPEG	.tif, .tiff
JPEG	Joint Photographic Experts Group	A standard for compression of images of photographic quality	.jpg, .jpeg
GIF	Graphics Interchange Format	Frequently used to make small animations on the internet	.gif
BMP	Windows Bitmap	Format used mainly for simple uncompressed images	.bmp
PNG	Portable Network Graphics	Compresses full color images with trasparency(up to 48bits/p	.png

TABLE 5.4. 1: Image formation model

Face Recognition Using Geometrical Features

This technique involves computation of a set of geometrical features such as nose width and length, mouth position and chin shape, etc. from the picture of the face we want to recognize. This set of features is then matched with the features of known individuals. A suitable metric such as Euclidean distance (finding the closest vector) can be used to find the closest match. Most pioneering work in face recognition was done using geometric features (Kanade, 1973), although Craw et al. (1987) did relatively recent work in this area.

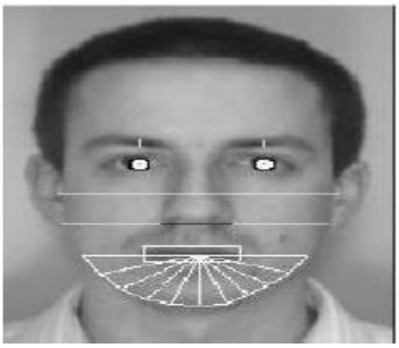


Fig 6.1: Face Recognition Using Geometric Feature.

The advantage of using geometrical features as a basis for face recognition is that recognition is possible even at very low resolutions and with noisy images (images with many disorderly pixel intensities). Although the face cannot be viewed in detail its overall geometrical configuration can be extracted for face recognition. The technique's main disadvantage is that automated extraction of the facial geometrical features is very hard. Automated geometrical feature extraction based recognition is also very sensitive to the scaling and rotation of a face in the image plane (Brunelli and Poggio, 1993). This is apparent when we examine Kanade's(1973) results where he reported a recognition rate of between 45-75 % with a database of only 20 people. However if these features are extracted manually as in Goldstein et al. (1971), and Kaya and Kobayashi (1972) satisfactory results may be obtained.

Face Recognition Using Template Matching

This is similar the template matching technique used in face detection, except here we are not trying to classify an image as a 'face' or 'non-face' but are trying to recognize a face.

Fig 6.2: Face Recognition Using Template Matching.

Whole face, eyes, nose and mouth regions which could be used in a template matching strategy.The basis of the template matching strategy is to extract whole facial regions (matrix of pixels) and compare these with the stored images of known individuals. Once again Euclidean distance can be used to find the closest match. The simple technique of comparing Grey-scale intensity values for face recognition was used by Baron (1981). However there are far more sophisticated methods of template matching for face recognition. These involve extensive pre-processing and transformation of the extracted Grey-level intensity values. For example, Turk and Pentland (1991a) used Principal Component Analysis, sometimes known as the eigen faces approach, to pre-process the gray-levels and Wiskott et al. (1997) used Elastic Graphs encoded using Gabor filters to pre-process the extracted regions. An investigation of geometrical features versus template matching for face recognition by Brunelli and Poggio (1993) came to the conclusion that although a feature based strategy may offer higher recognition speed and smaller memory requirements, template based techniques offer superior recognition accuracy

5.5 INPUT AND OUTPUT DESIGN

INPUT DESIGN

Facial recognition technology can be a useful tool for attendance registration as it can identify individuals based on their unique facial features. Here are some input design considerations for implementing facial recognition technology in attendance registration:

Camera placement: The camera is positioned in a way that captures clear images of people's faces as they enter the designated area. Consider factors like lighting and angles to ensure that the camera can capture high-quality images.

Enrollment process: Before using facial recognition for attendance registration, individuals need to enroll their faces in the system. The enrollment process should be simple and user-friendly, and provide clear instructions for capturing and saving images.

Database management: The system maintains an up-to-date database of registered faces. Consider implementing a system that can automatically update the database as new faces are enrolled or existing faces change over time.

Error handling: The system is able to handle errors that may arise, such as incorrect or mismatched face recognition. Consider implementing a system that provides feedback to the user and prompts them to try again.

Security and privacy: Facial recognition technology raises concerns about privacy and security. The system implements appropriate security measures to protect the data and ensure that it is only accessible by authorized personnel.

Accessibility: The system is designed to accommodate individuals with disabilities or other accessibility needs. Consider factors like camera height and placement, as well as providing alternative registration methods for individuals who cannot use facial recognition technology

OUTPUT DESIGN

Output design is an important aspect of attendance registration using facial recognition technology. Here are some considerations for designing the output of such a system:

Attendance reports: The system should generate accurate and detailed attendance reports that can be easily accessed by authorized personnel. The reports should provide information on the time and date of attendance, as well as the name and image of the individual.

Notification alerts: The system is designed to send automatic notifications to the appropriate personnel when a person enters or exits the designated area. This feature can help ensure that attendance is accurately recorded and can be used for security purposes.

User interface: The output is presented in a user-friendly interface that can be easily navigated. Consider factors such as font size, color contrast, and layout to ensure that the interface is accessible to all users.

Real-time updates: The system is designed to provide real-time updates on attendance, allowing authorized personnel to monitor attendance as it occurs. This feature can be especially useful in large events or settings where attendance is critical.

CHAPTER 6
RESULT AND DISCUSSION

Facial recognition technology has the potential to revolutionize the way attendance is taken in schools and other institutions. In this system, a camera captures the faces of individuals and uses computer algorithms to match those faces with a database of known individuals. This can be used to take attendance automatically, without the need for manual registration or identification.

One major advantage of using facial recognition for attendance registration is that it is fast and efficient. It can take attendance of a large number of students in a short amount of time. This can save valuable instructional time and make it easier for teachers and staff to keep track of who is present and who is absent.

Another advantage of facial recognition is that it is more accurate than traditional methods of attendance taking, such as calling out names or having students sign in. It can eliminate errors and reduce the risk of fraud or impersonation. Additionally, it can provide real-time data on attendance, which can be used for tracking and analysis.

However, there are also some concerns associated with facial recognition technology. One concern is privacy. The use of facial recognition raises questions about who has access to the data and how it is being used. Additionally, there are concerns about the accuracy of the technology, particularly in identifying people of different races, ages, and genders.

CHAPTER 7
CONCLUSION & FUTURE WORK

CONCLUSION

The purpose of reducing the errors that occur in the traditional attendance taking system has been achieved by implementing this automated attendance system. In this paper, face recognition system have been presented using deep learning which exhibits robustness towards recognition of the users with accuracy of 98.3% . The result shows the capability of the system to cope with the change in posing and projection of faces. From face recognition with deep learning, it has been determined that during face detection, the problem of illumination is solved as the original image is turned into a HOG representation that captures the major features of the image regardless of image brightness. In the face recognition method, local facial landmarks are considered for further processing. After which faces are encoded which generates 128 measurements of the captured face and the optimal face recognition is done by finding the person’s name from the encoding. The result is then used to generate an excel sheet, the PDF of which is sent to the students and professors on weekly interval. This system is convenient to the user and it gives better security.

FUTURE WORK

The project will have a huge future. The undertaking can be executed on intranet in future. Because it is extremely adaptable in terms of expansion, the project can be updated in the near future if the need arises. The client is now able to manage and, as a result, run the entire project in a much improved, accurate, and error-free manner because the proposed database Space Manager software is ready and fully functional. Coming up next are the future extension for the task. Ø End of specific understudy take out possible participation. Individual attendance system with a photo and student login based on a bar code reader

REFERENCES

[1] V. Blanz and T. Vetter. Face recognition based on fitting a 3dmorphable model.IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(9):1063–1074, 2003.2

[2] P. Breuer, K.-I. Kim, W. Kienzle, B. Scholkopf, and V. Blanz.Automatic 3d face reconstruction from single images or video. InProc. of the 8th IEEE International Conference on Automatic Face Gesture Recognition (FG ’08), pages 1–8, 2008.2

[3] Chaoyang, L. Wang, Y. Wang, F. Matsushita, K., and Soong.Binocular photo metric stereo acquisition and reconstruction for 3d talking head applications. In Inter speech 2013 sub-mission, 2013.3

[4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 2d modeling ofindoor environments.Proc. of International Symposium on Experimental Robotics, 2010.1

[5] M. Hernandez, J. Choi, and G. Medioni. Laser scan quality 3-d face modeling using a low-cost depth camera.Proc.of the 20th European Signal Processing Conference (EU-
SIPCO), pages 1995–1999, 2012.1,2,3,4,5,7

[6]C. Mayer, M. Wimmer, M. Eggers, and B. Radig, “Facial Expression Recognition with 3D Deformable Models.” Second International Conferences on Advances in Computer-Human Interactions, pp 26-31, 2009.

[7]V. Blanz, and T. Vetter, “Face Recognition Based on Fitting a 3D Morph-able Model”, IEEE Trans. PAMI, vol. 25, no. 9, pp. 1063-1074, 2003.

[8]X. Lu, and A. Jain, “Deformation Modeling for Robust 3D Face Matching.” Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, No. 8, pp. 1347-1356, 2008.

[9]B. Moghaddam, J. Lee, F. Fister, and R. Machiraju, “Model-Based 3D Face Capture with Shape-from-Silhouettes.” Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2003.

APPENDIX

A) SOURCE CODE

IMPORTING

```
import tkinter as tk from tkinter import ttk
from tkinter import messagebox as mess import tkinter.simpledialog as tsd
import cv2,os import csv
import numpy as np from PIL import Image import pandas as pd import datetime
import time
```

FUNCTIONS

```
def assure_path_exists(path): dir = os.path.dirname(path) if not os.path.exists(dir):
os.makedirs(dir)

def tick():
time_string = time.strftime('%H:%M:%S') clock.config(text=time_string)
clock.after(200,tick)


def contact():
mess._show(title='Contact us', message="Please contact us on :
'xxxxxxxxxxxxx@gmail.com' ")


def check_haarcascadefile():
exists = os.path.isfile("haarcascade_frontalface_default.xml") if exists:
pass else:
mess._show(title='Some file missing', message='Please contact us for help')
window.destroy()


 def save_pass():
    assure_path_exists("TrainingImageLabe
    l/")
    exists1 =
    os.path.isfile("TrainingImageLabel\psd.tx
    t") if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
```

```
        key = tf.read()
    else:
        master.destroy()
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password
below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try
again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered
successfully!!')
            return
    op = (old.get())
    newp= (new.get())
    nnewp = (nnew.get())
    if (op == key):
        if(newp == nnewp):
            txf = open("TrainingImageLabel\psd.txt", "w")
            txf.write(newp)
        else:
            mess._show(title='Error', message='Confirm new password again!!!')
            return
    else:
        mess._show(title='Wrong Password', message='Please enter correct old password.')
        return
    mess._show(title='Password Changed', message='Password changed successfully!!')
    master.destroy()



def
    change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")
    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold
    ')) lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')
    old.place(x=180,y=10)
    lbl5 = tk.Label(master, text='    Enter New Password', bg='white', font=('times', 12, ' bold
    ')) lbl5.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold
'),show='*')
    new.place(x=180, y=45)
    lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))
```

```
    lbl6.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold
'),show='*')
    nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel",
command=master.destroy ,fg="black" ,bg="red" ,height=1,width=25 , activebackground =
"white" ,font=('times', 10, ' bold '))
    cancel.place(x=200, y=120)
    save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3ece48",
height = 1,width=25, activebackground="white", font=('times', 10, ' bold '))
    save1.place(x=10, y=120)
    master.mainloop()



def psw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password
below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try
again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered
successfully!!')
            return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
        TrainImages()
    elif (password == None):
        pass
    else:
        mess._show(title='Wrong Password', message='You have entered wrong password')


def clear():
    txt.delete(0,
    'end')
    res = "1)Take Images >>> 2)Save Profile"
    message1.configure(text=res)


def clear2():
    txt2.delete(0,
    'end')
```

```
   res = "1)Take Images >>> 2)Save Profile"
   message1.configure(text=res)



def TakeImages():
   check_haarcascadefile()
   columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
   assure_path_exists("StudentDetails/"
   )
   assure_path_exists("TrainingImage/")
   serial = 0
   exists                                                     =
   os.path.isfile("StudentDetails\StudentDetails.csv")          if
   exists:
      with      open("StudentDetails\StudentDetails.csv",      'r')      as
         csvFile1: reader1 = csv.reader(csvFile1)
         for   l   in   reader1:
            serial  =  serial  +
            1
      serial  =  (serial  //
      2) csvFile1.close()
   else:
      with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
         writer = csv.writer(csvFile1)
         writer.writerow(columns)
         serial = 1
      csvFile1.close()
   Id = (txt.get())
   name =
   (txt2.get())
   if ((name.isalpha()) or (' ' in name)):
      cam = cv2.VideoCapture(0)
      harcascadePath = "haarcascade_frontalface_default.xml"
      detector = cv2.CascadeClassifier(harcascadePath)
      sampleNum = 0
      while (True):
         ret, img = cam.read()
         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
         faces = detector.detectMultiScale(gray, 1.3, 5)
         for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
            # incrementing sample number
            sampleNum = sampleNum + 1
            # saving the captured face in the dataset folder TrainingImage
            cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' +
str(sampleNum) + ".jpg",
                     gray[y:y + h, x:x +
            w]) # display the frame
            cv2.imshow('Taking Images',
         img) # wait for 100 miliseconds
         if cv2.waitKey(100) & 0xFF == ord('q'):
            break
         # break if the sample number is morethan 100
```

```
elif sampleNum > 100:
```

```
            break
        cam.release()
        cv2.destroyAllWindows(
        )
        res = "Images Taken for ID : " + Id
        row = [serial, '', Id, '', name]
        with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
        message1.configure(text=res)
    else:
        if (name.isalpha() == False):
            res = "Enter Correct name"
            message.configure(text=res
            )


def TrainImages():
    check_haarcascadefile()
    assure_path_exists("TrainingImageLabel/")
    recognizer =
    cv2.face_LBPHFaceRecognizer.create()
    harcascadePath =
    "haarcascade_frontalface_default.xml" detector =
    cv2.CascadeClassifier(harcascadePath) faces, ID =
    getImagesAndLabels("TrainingImage")
    try:
        recognizer.train(faces, np.array(ID))
    except:
        mess._show(title='No Registrations', message='Please Register someone first!!!')
        return
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
    message.configure(text='Total Registrations till now : ' + str(ID[0]))



def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faces = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy
        array imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        ID = int(os.path.split(imagePath)[-1].split(".")[1])
```

```
        faces.append(imageNp)
        Ids.append(ID)
    return faces, Ids



def TrackImages():
    check_haarcascadefile()
    assure_path_exists("Attendance/")
    assure_path_exists("StudentDetails/")
    for k in tv.get_children():
        tv.delete(k
    ) msg = ''
    i = 0
    j = 0
    recognizer = cv2.face.LBPHFaceRecognizer_create() #
cv2.createLBPHFaceRecognizer()
    exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")
    if exists3:
        recognizer.read("TrainingImageLabel\Trainner.yml")
    else:
        mess._show(title='Data Missing', message='Please click on Save Profile to reset
data!!')
        return
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);

    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
    exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists1:
        df = pd.read_csv("StudentDetails\StudentDetails.csv")
    else:
        mess._show(title='Details Missing', message='Students details are missing, please
check!')
        cam.release()
        cv2.destroyAllWindows()
        window.destroy()
    while True:
        ret, im = cam.read()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.2, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
            serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
            if (conf < 50):
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
                timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
```

```
            ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
            ID = str(ID)
            ID = ID[1:-1]
            bb = str(aa)
            bb = bb[2:-
            2]
            attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

        else:
            Id = 'Unknown'
            bb = str(Id)
        cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
    cv2.imshow('Taking Attendance', im)
    if (cv2.waitKey(1) ==
        ord('q')): break
  ts = time.time()
  date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
  exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
  if exists:
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
        writer = csv.writer(csvFile1)
        writer.writerow(attendance)
    csvFile1.close()
  else:
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
        writer = csv.writer(csvFile1)
        writer.writerow(col_names)
        writer.writerow(attendance)
    csvFile1.close()
  with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:
    reader1 = csv.reader(csvFile1)
    for lines in reader1:
        i = i + 1
        if (i > 1):
            if (i % 2 != 0):
                iidd = str(lines[0]) + ' '
                tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))
  csvFile1.close()
  cam.release()
  cv2.destroyAllWindows()
```

USED STUFFS

```
global key
key = ''

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-
%Y') day,month,year=date.split("-")

mont={'01':'January',
    '02':'February',
```

```
    '03':'March',
    '04':'April',
    '05':'May',
    '06':'June',
    '07':'July',
    '08':'August',
    '09':'September',
    '10':'October',
    '11':'November',
    '12':'December'
    }
```

GUI FRONT-END

```
window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
window.title("Attendance System")
window.configure(background='#262523')

frame1 = tk.Frame(window, bg="#00aeff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#00aeff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance
System" ,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+" | ",
fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, '
bold '))
clock.pack(fill='both',expand=1)
tick()

head2 = tk.Label(frame2, text="                       For New Registrations
                       ", fg="black",bg="#3ece48"
,font=('times', 17, ' bold ') )
head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text="                       For Already Registered
                       ", fg="black",bg="#3ece48"
,font=('times', 17, ' bold ') )
```

head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter
ID",width=20 ,height=1 ,fg="black" ,bg="#00aeff" ,font=('times', 17, ' bold ')) lbl.place(x=80,
y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold
')) txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name",width=20 ,fg="black" ,bg="#00aeff" ,font=('times',
17, ' bold '))
lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images >>> 2)Save
Profile" ,bg="#00aeff" ,fg="black" ,width=39 ,height=1, activebackground =
"yellow" ,font=('times', 15, ' bold '))
message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black" ,width=39,height=1,
activebackground = "yellow" ,font=('times', 16, ' bold '))
message.place(x=7, y=450)

lbl3 = tk.Label(frame1,
text="Attendance",width=20 ,fg="black" ,bg="#00aeff" ,height=1 ,font=('times', 17, ' bold '))
lbl3.place(x=100, y=115)

```
res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
   with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
      reader1 = csv.reader(csvFile1)
      for l in reader1:
         res = res + 1
   res = (res // 2) - 1
   csvFile1.close()
else:
   res = 0
message.configure(text='Total Registrations till now : '+str(res))
```

MENUBAR

```
menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)
```

TREEVIEW ATTENDANCE TABLE

```
tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ='ID')
tv.heading('name',text
='NAME') tv.heading('date',text
='DATE') tv.heading('time',text
='TIME')
```

SCROLLBAR

```
scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)
```

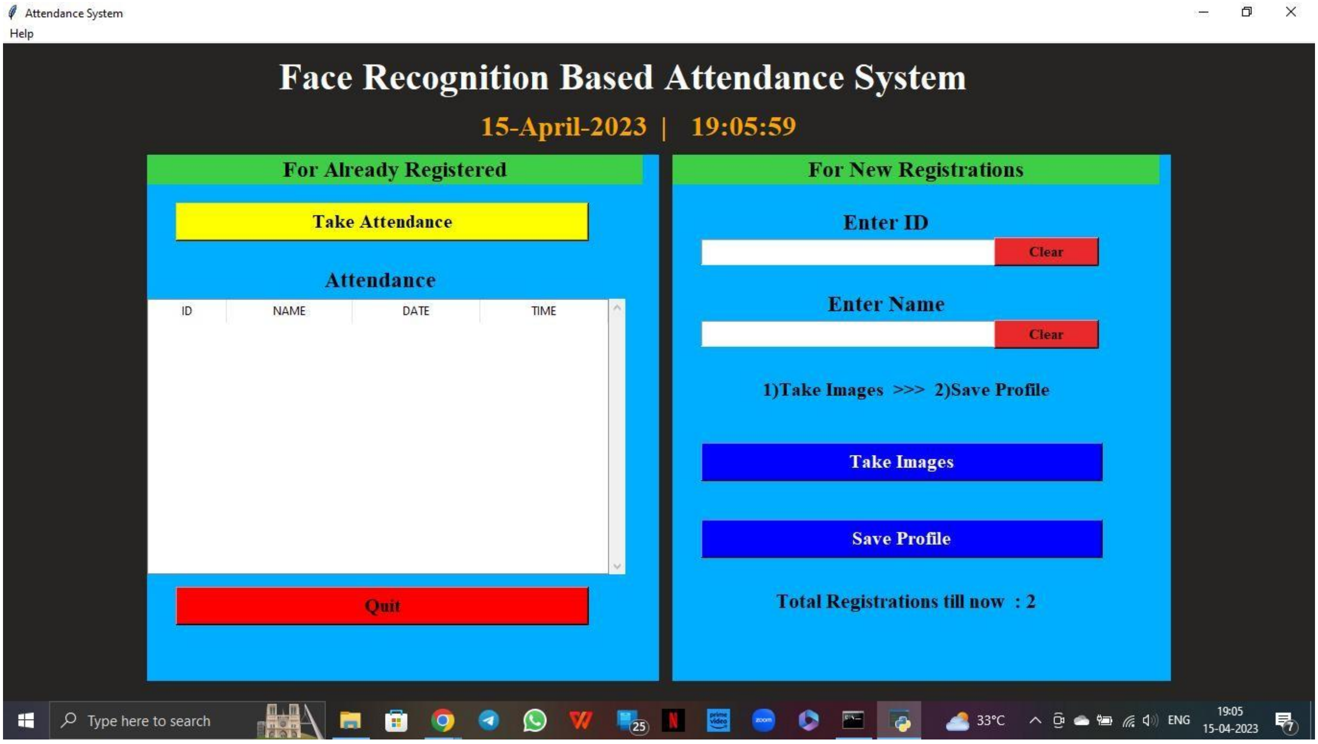
BUTTONS

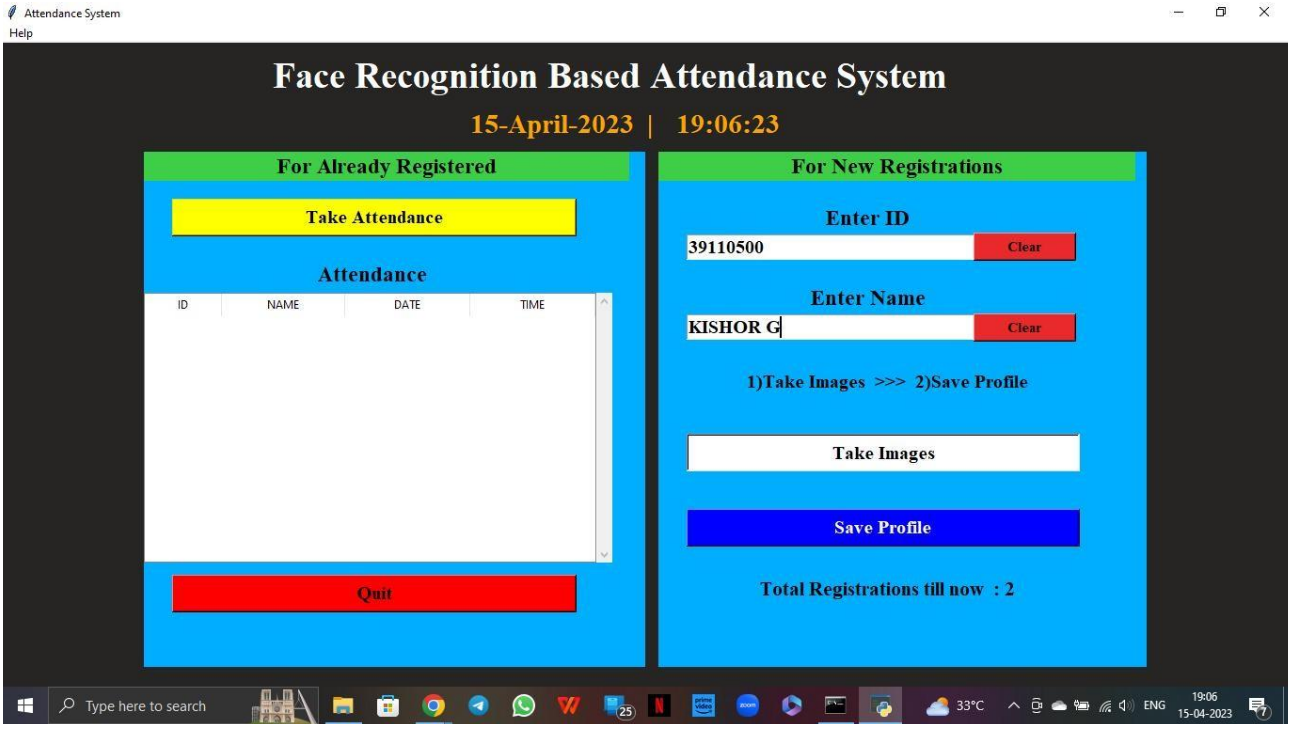
```
clearButton = tk.Button(frame2, text="Clear",
command=clear ,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground =
"white" ,font=('times', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear",
command=clear2 ,fg="black" ,bg="#ea2a2a" ,width=11 , activebackground =
"white" ,font=('times', 11, ' bold '))
clearButton2.place(x=335, y=172)
takeImg = tk.Button(frame2, text="Take Images",
command=TakeImages ,fg="white" ,bg="blue" ,width=34 ,height=1, activebackground =
"white" ,font=('times', 15,
' bold '))
takeImg.place(x=30,
y=300)
trainImg = tk.Button(frame2, text="Save Profile",
command=psw ,fg="white" ,bg="blue" ,width=34 ,height=1, activebackground =
"white" ,font=('times', 15, ' bold '))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take Attendance",
command=TrackImages ,fg="black" ,bg="yellow" ,width=35 ,height=1, activebackground
= "white" ,font=('times', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit",
command=window.destroy ,fg="black" ,bg="red" ,width=35 ,height=1, activebackground =
"white" ,font=('times', 15, ' bold '))
quitWindow.place(x=30, y=450)
```

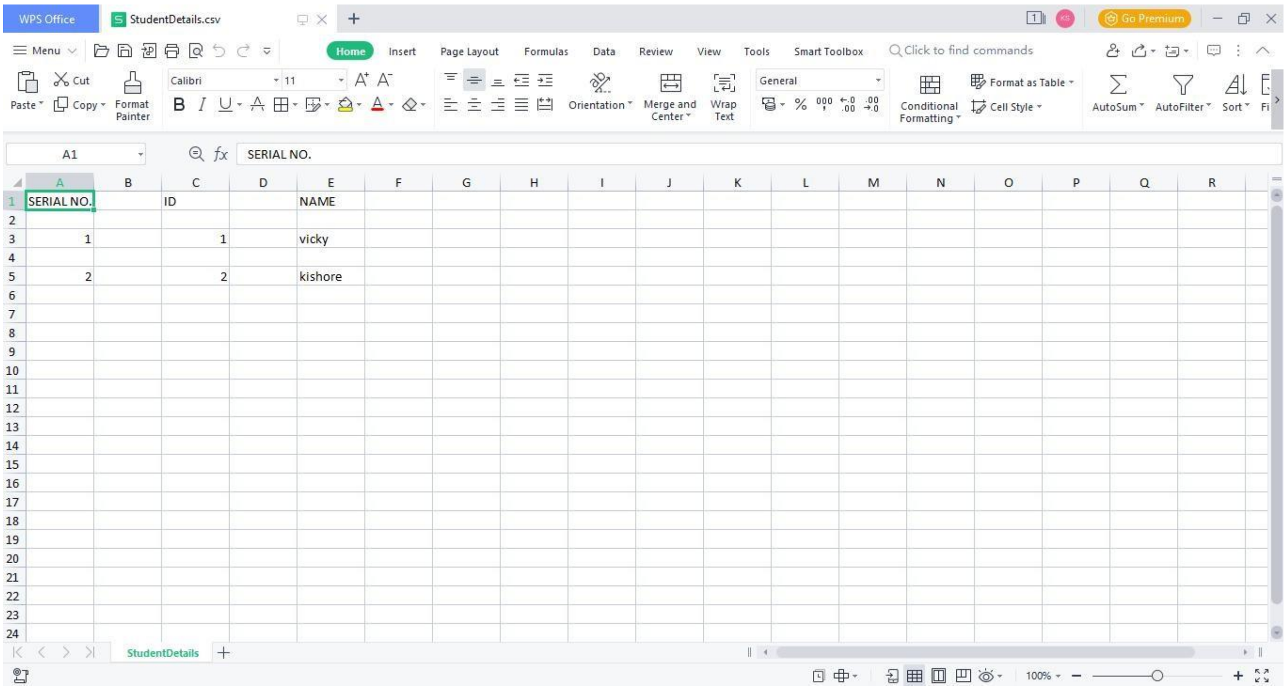
END

```
window.configure(menu=menubar)
window.mainloop()
```

B) SCREENSHOTS






StudentDetails.csv
SERIAL NO.
ID
NAME
1
1
vicky
2
2
kishore
StudentDetails

C) RESEARCH PAPER

ATTENDANCE REGISTRATION USING FACIAL RECOGNITION

Lithwin R
Computer Science
Sathyabama Institute of Science and Technology
Chennai, India
lithwinrajesh@gmail.com

Logesh K
Computer Science
Sathyabama Institute of Science and Technology
Chennai,India
1968logesh@gmail.com

Dr. Mary Posonia A
Asst. Professor
Computer Science
Sathyabama Institute of Science and Technology
Chennai,India
maryposonia.cse@sathyabama.ac.in

Abstract—The system should be able to identify college students' faces in the lecture hall with a 30% accuracy rate. The extensive range of college students that are featured in the GUI will automatically be displayed by the device. With a 30% accuracy rate, recognize pupils from a database of saved faces by comparing their images to those in the database. The machine must be designed such that students' faces may be recognized from a photograph that has their faces in the database. To reach the desired popularity via the lecture's end, the imaging devices must run for 10 minutes.Per hour and per lesson, those five names will be registered. The methodology for the applicable system's characteristic enables acquisition of the device's precision to within 20%. We make sure that the fine must not exceed 20%. For the advanced device to operate and provide access to its capabilities, a graphical user interface may be user-friendly

Keywords—Facerecognition,RecognitionAlgorithm,AttendanceRegistration,MATLAB,OpenC

INTRODUCTION

We resolve the 3D face tracking issue, which is a major issue in the setting of human-computer interaction. The input consists of a video stream acquired from a less expensive webcam that is directed at the subject's face. Our focus is on real-time full 3-D facial tracking (6 DOF) even when there are face changes, a wide range of positions, and partial occlusion. In contrast to the most recent processes, we only make a small number of assumptions up until this point. The distance from the camera should be sufficient for interactive face software, but it may vary based on the focal length being used. In addition, we commonly require an anal role all through initialization. The proposed approach entails the use of an everyday 3D face version to music customers' real-time 3-d head actions and generate correct face models and textures.

Version healthy, three-D head tracking, and records reacquisition.
The usage of 3D hunter and 3D modeling isn't always discussed right here. We want a 3-D model, which could either be a typical version or a specific version retrieved from a database. At the beginning of the three-D model, the model is deformed orthogonally to the focal axis to match the consumer's face inside the input photograph to the 2D facial features extracted at runtime. The preliminary 3-D reconstruction of the face version is crucial.
Face detection is a pc era used in lots of exclusive packages that require the identity of human faces in pix or virtual pictures. One can think of the special object class detection as the process of identifying all objects in an image that belong to a specific category and determining their location and size.This can detect frontal or near-frontal faces in a photo, no matter orientation, light situations, or pores and skin coloration.
Locating human faces using an algorithm in an image (When you use this with face reputation, detecting human faces at the bottom is in full motion. On the other hand, face popularity refers to a biometric technology that outperforms reputation with a human visage by a considerable margin. It is there. Face. In this writing, I might not be recognized by face. I'll save that for a later blog post because right now I just want to show you how to use a Python camera to run a simple Face Detection programme
In their 2001 paper "Fast Object Detection Using Boosted Simple Feature Cascade," This technique uses characteristic-based cascaded classifiers.
If you want to remain on this rock megastar with face detection/recognition and device mastering, I advise you to save this PDF and read it as soon as the threat comes in. They presented the concept of cascade classes in this text. Open CV's Caesar Haar recognition
OpenCV includes both a detection and a hold. Open CV can be used to construct a custom classifier for any object, such as a car, plane, or other object .You can find all of its details here: Classifier for discipline cascades.
Here, we'll talk about innovation. There are already numerous facial, eye, smile, and other classifiers in OpenCV. contains the appointed The open cv/statistics/haar cascades/ subdirectory houses these

XML files. Let's use Opencv to create a flame monitor.
Growing hobby in a laptop vision within the last decade. Constantly doubling computing electricity each 13 months, face detection and popularity from the inner area to the popular studies area in computer vision and one of the first-rate and maximum of a success packages of picture analysis and algorithm-primarily based information. Due to the intrinsic nature of the problems, pc vision isn't only a subject of pc science, however additionally a topic of neuroscience and psychological research, specifically due to the overall consensus that advances in computational imaging and comprehension studies will provide perception into how our brain works, and vice versa.

LITERATURE OVERVIEW

[1] Jireh Jam and others This article demonstrates how face recognition and reputation algorithms can be applied to image processing to build a device that effectively finds and captures the faces of students in the front of the classroom. The front portion of a person or animal's head, from the forehead to the chin, is referred to as the face. (Oxford Dictionary). The face is crucial to human interactions because it gives away important details about a person or people. All people are capable of facial perception. The suggested remedy is to create a working prototype of a device that will enable Kingston University lecturers to control lecture rooms by sensing the front of An image captured inside the study area showing the faces of students. The 2nd part of the gadget may also be capable of carrying out facial popularity on a small part of the database. Recently, studies have been done and face popularity and detection systems have evolved. Some of them are utilized in social media structures, banking forums, authorities corporations, as an example. Metropolitan police, Facebook, and many others.

[2] Shraddha Shinde,Ms. Patil Priyanka

In this newsletter we recommend a gadget that should be utilized by faculty students to attend lectures. Our system has automatically registered the attendance of face reputation via continuous surveillance. Continuous surveillance improves performance for frequency estimation. This article is the primary evaluation related to offerings and facial reputation. It then shows the structure of the gadget, the plan, and the anticipated final results.

In this text we advocate a system that must beutilized by college students to attend lectures. Our gadget serves robotically using face recognition. In this paper, we recommend a method for estimating frequencies precisely by way of the usage of all the consequences of face reputation acquired during continuous remark. Continuous tracking frequency will increase the efficiency of the assessment. This article is the primary evaluation related to services and facial recognition. Next, our system presents the shape, strategy, and expected results.

[3] Aishwarya, Simran Raju Inamdar Dr. S. M. Mukane, Ankita Digambar Patil, and Vijay Kumar Patil Evaluating target market order frequency is not the simplest burdensome however additionally time-consuming. Due to the common presence of students in faculty, it's far usually possible to be a gift via a representative. It may be very difficult for teachers to manually pick out students who frequently skip instructions. Managing pupil attendance through traditional methods has come to be a venture in current years. In this settlement, a digital cam is fixed inside the college and takes a photograph, the face is removed, then it's miles recognized within the database, and sooner or later the listening is recorded. In this article, we proposed a conceptual model for an automated device of personnel using facial reputation.

[4] P. Visalakshi, Sushant Ashish Facial recognition is the excellent and continually evolving security feature. In this task, lecture room listening, usually monitored with the aid of the lecture room digital camera. A student attendance system is released and the primary time of the digital cam detects a face, the scholar's attendance is recorded. When a cold digital camera is used, If faces are captured at exceptionally high resolutions, it will be challenging to identify them. Using the OpenCV tool, this is accomplished. The use of local histograms allows for the identification of the visage. The classroom where the students sit has a digital camera. The students will be constantly followed by the digital camera in the footage.The most effective goals in the modern system are now to identify the possibility of college students attending the meeting as well as to improve the monitoring of study room lectures.

Facial recognition can be described as a method of identifying someone based totally on biometric information based totally on the assessment of a captured digital photograph or video with a stored report of the character in the query. In the early 90s, many algorithms for face popularity had been evolved, which result in an increase within the call for face reputation

PROPOSED WORK

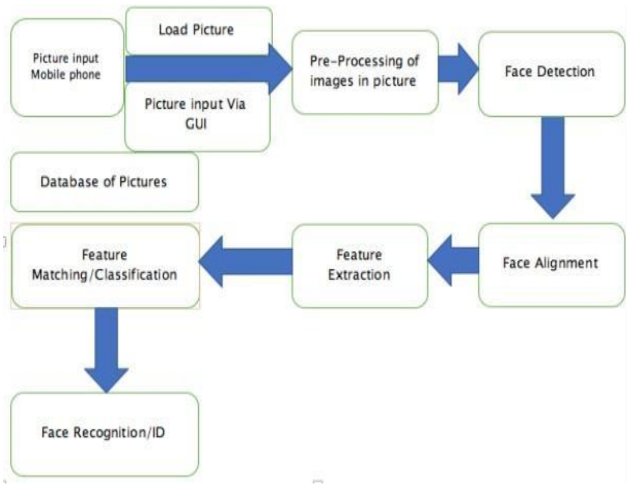


Fig 1. Architecture Diagram

At the core of the system is a facial recognition engine that analyzes images captured by a camera to identify individuals. The system may also include a database for storing employee information and attendance records, as well as an API or integration layer for connecting to other software systems within the organization. The system may also include additional hardware components such as cameras, sensors, and display screens. These components work together to create a seamless user experience for employees and administrators alike. Overall, the architecture of an attendance registration system using facial recognition is designed to be scalable, reliable, and secure, ensuring that organizations can effectively track employee attendance while maintaining the privacy and security of employee data.

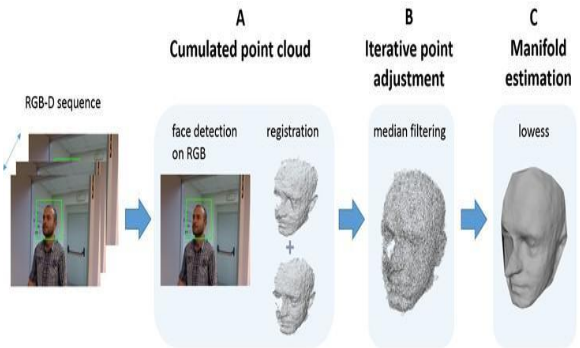


Fig 2. Data Flow Diagram

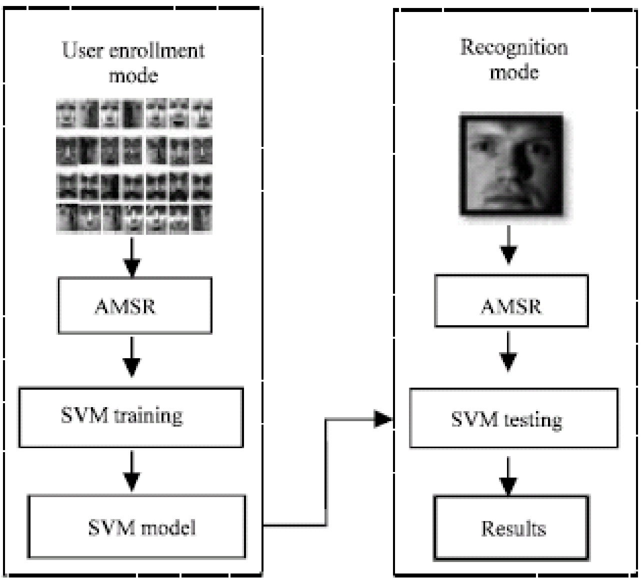
SOFTWARE MODULES

FACE DETECTION:

Face detection entails segmenting windows into collections of images, each containing a single face recognized history (clutter). This is difficult because, despite some facial similarities, there may be significant differences in the faces' age, and facial features. The issue is further complicated by the unique lighting circumstances, picture quality, and geometry. Along with the potential for omission, limited chance, and protection. The presence of any face in any lighting situation in opposition to any object would therefore be detectable by a truly accurate face detector. It is possible to divide the face detection project into stages. In the initial step, a type characteristic that accepts any image as input and outputs a binary value of either sure or not sure, is used. indicating whether or not there are any faces in the picture. The face localization problem, which is the second stage, asks users to input images of a face or faces in that region of the image as if they were in a container with a face on it. (x, y, width, and height).

SVM ALGORITHM:

For classification or regression analysis, SVM is a supervised learning algorithm. Its objective is to locate the hyperplane with the largest possible margin of separation between the data's various classes.w.x + b = 0 It can handle non-linearly separable data by using various types of kernel functions, such as linear, polynomial, RBF, and sigmoid. The margin is the distance between the hyperplane and the closest data points from each class..it is widely used in industry and research because of its high accuracy and robustness.it has some limitations, such as being sensitive to the algorithm's parameters and the selected kernel function..it can be computationally expensive for large datasets.it has many practical applications, such as in text classification, image classification, and bioinformatics.
distance = |w.x + b| / ||w|| for finding the distances.
MIN 1/2 ||w||^2 w.r.t y_i(w.x_i + b) ≥ 1 for all i actual formula for SVM Model.



Support Vector Machines have been successfully used in facial recognition by learning features that distinguish one face from another and classifying new facial images accordingly. SVMs are effective in handling high-dimensional feature spaces and non-linearly separable data, but challenges such as lighting variations, occlusions, and pose changes may still affect accuracy. SVMs remain a key component of facial recognition technology with continued research and development.

DATASETS:

Fig 4. Datasets

A dataset for attendance registration using facial recognition typically includes a large number of images of individuals along with labels indicating their identities. The images are captured in different lighting, poses, and facial expressions, with high quality and diverse representation of individuals. The dataset is used to train the facial recognition algorithm for accurate and reliable attendance registration.
IMPLEMENTATION.

1. Pre-processing: Before being fed to the network, images are pre-processed to reduce face variability. All significant examples, or photographs of faces, are obtained by cropping the front photographs of the face to only include the view. The use of general algorithms is then used to fix all cropped images.

2. Classification: By learning from those instances, neural networks are used to examine photos as either faces or non-faces. We utilize both neural network tools and our own neural network tools for this task. To improve results, try out various network settings.
The following metrics are often included in the categorization report:

Step 1: Load the MNIST dataset:
The dataset includes training and testing pictures .
Step 2: Preprocess the images:
Preprocess the photos in order to train the SVM model.standardizing pixel values to be between 0 and 1,To carry out these processes, use a method like preprocess().
Step 3: Define the SVM model:
Create a SVM model with a deep learning library like Keras or TensorFlow.
Normalize the pixel values of the training and testing sets
Step 4: Compile the model
Set the loss function, optimizer, and evaluation metric before compiling the model.utilise the
Train the linear SVM model using the training set:
Set the C parameter for the SVM
Fit the SVM to the training data
Step 5: Train the model
Fit the model to the training set using the fit() function, giving the number of epochs, batch size, and validation set.
Experiment with different hyperparameters such as learning rate and regularization strength to increase performance.
Step 6: Evaluate the model
Use the evaluate() function to assess the trained model on the testing set, which returns the loss and accuracy on the test set.
Do further evaluations, such as a confusion matrix and classification report, if desired, to examine the model's performance in each class.
Test the linear SVM model using the testing set:
Predict the labels of the testing data using the SVM
Compute the accuracy of the SVM on the testing data
Step 7: Make predictions
To create predictions on previously unseen photos, use the predict() function.
Visualize the predictions and their associated probabilities using approaches such as the confusion matrix, the ROC curve, or the precision-recall curve.
Print the accuracy of the SVM model

Class	(TP)	(FP)	(FN)	(TN)	Precision	Recall
True	27	63	63	147	0.3	0.7
False	147	63	63	27	0.7	0.3
Total	174	126	126	174	-	-

3. Localization: The features in the image are then located using a trained neural network, and if present, they are localized in a bounding box. The numerous facial characteristics that received treatment included: illumination of the scale role's direction. A laptop generation called facial recognition pinpoints the

location and length of a human visage in a random (digital) image. Along with other functions, such as those of wood, buildings, our bodies, and so forth, facial functions are recognized. are overlooked in the computer image. Consider the "unique" situation: finding the locations and measurements of all the objects in the image that fall under a specific elegance is a function of object class detection.

One could consider face recognition to be a particularly "popular" instance of face localization. The goal of face localization is to pinpoint the position and dimensions of a known range of faces. (normally one). To find the face in a given photo, there are essentially two types of methods: the characteristic-based strategy and the image-based approach. The function-based on-a-total method makes an effort to extract the picture's functions and match them with facial reputation. It attempts to achieve the ideal balance between instruction and check images using an image-based approach.

IMAGE:

The photograph represents the intensity function f(x, y) of a dimensional light, where (x, y) denotes the local coordinates and the value of f at any point (x, y) is proportional to the brightness or gray stage. Photo in that vicinity. Offers a virtual photo is a photograph of f(x,y) sampled in neighborhood coordinates and brightnesses. The arrays of this sort of virtual tool are called imaginary elements or elements.

IMAGE MODEL:

For a photo to be suitable for computation, a photo f(x, y) must be determined in each region and amplitude. Digitizing spatial coordinates (x, y) is called image sampling. The digitized amount is called grayscale.

With increasing spatial resolution and the variety of gray areas, the amount of data that needs to be stored and processed quickly grows.

For example: A 256x256 256-degree photo takes up sixty four KB of memory.

The lower-degree processing way acting primary operations on images, together with picture reading, resizing, rotating the image, changing RGB to gray stage, histogram equalization, and many others. The output image acquired after low-degree processing is the raw photograph. Medium-degree processing means that areas of interest are extracted from the output of a low-degree processed picture. The center level of the procedure revolves across the identification of barriers, i.e. Coasts this manner is known as segmentation. High-stage processing is set adding artificial intelligence to medium-stage signal processing.

STEPS IN IMAGE PROCESSING:

Step1: Image acquisition: Digital Image Acquisition

Step 2: Image pre-processing: Improving the image in order that the different methods are greater success.

Step 3: Image segmentation: divide the center picture into element elements.

Step 4: Image segmentation: convert the input facts into a form appropriate for laptop processing.

Step 5: Image description: to derive functions that yield some quantitative information about the qualities that are necessary to distinguish one sort of object from another.

Step 6: Image recognition: assigning a label to an item based on the information contained in its description.

RESULT AND DISCUSSION:-

A camera, a computer, and a display device can be incorporated into the attendance system. When a person arrives at the place where attendance is to be recorded, the camera is used to capture an image of their face. The picture is processed by the computer, which compares it to the facial information in the model. If a match is made, that person's attendance is recorded, and their identity appears on the screen. The first stage is to gather the facial information of each person whose attendance will be tracked. Once the facial information has been gathered, a facial recognition algorithm needs to be trained using it. The attendance system must be evaluated and improved after it is constructed.

Accuracy for No of Persons Using SVM

No of Persons	Accuracy	Precision
1	96%	0.745
2	90%	0.722
3	85%	0.672
4	76%	0.543
5	65%	0.875

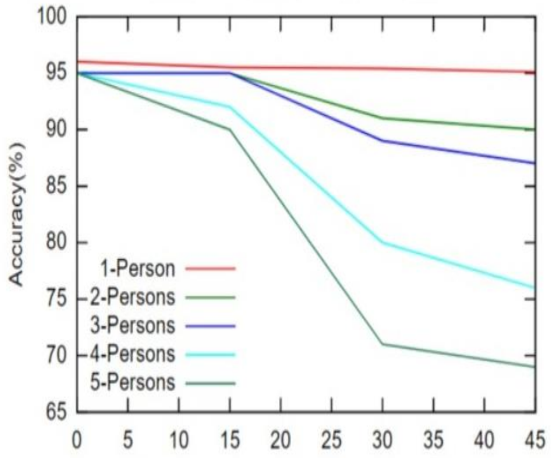


Fig 3. Accuracy Graph

The vertical Y-axis represents the facial recognition system's accuracy percentage, while the horizontal X-axis represents the number of people or subjects being recognized.

In conclusion, gathering facial data, training a facial recognition model, developing an attendance system, testing and improving the system, and ultimately deploying it are the steps involved in implementing attendance registration using facial recognition.

CONCLUSION:-

The reason for our task is to facilitate monitoring of the occasion. There are sincerely extra methods to become aware of human beings, consisting of using personal playing cards and amassing signatures at the hole event, however we found facial reputation to be extra practical than other conventional strategies. During the improvement method, we encountered numerous problems, whilst we determined that face recognition changed into very hard. First, a desire ought to be made: using a facial description or a facial recognition set of rules. We observed that this feature is more suitable for our utility. We additionally determined to apply the pictures without delay in our software, in preference to storing them in the database, just for testing purposes. As the quantity of users will increase, face reputation algorithms and databases may be without problems stuffed.

REFERENCES

T. Vetter and V. Blanz. Face identification using a 3D morphable model for fitting.IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9):1063–1074 (2002)

W. Kienzle, B. Scholkopf, P. Breuer, K.-I. Kim, and V. Blanz.automatic reconstruction of a 3D face from a single image or film. Pages 1–8 of the 2008 edition of the Proceedings of the 8th IEEE International Conference on Automatic Face Gesture Recognition (FG '08)Chaoyang, K. Matsushita, F. Matsushita, Y. Wang, L. Wang, and Soong.3D talking head apps using binocular photometric stereo acquisition and reconstruction. 2013's InInterspeech sub-mission

E. Herbst, X. Ren, D. Fox, P. Henry, M. Krainin, and M. Rgb-d mapping: dense 2D modelling of indoor areas using depth cameras.International Symposium on Experimental Robotics Proceedings, 2010

J. Choi, G. Medioni, and M. Hernandez. 3-D facial modelling in laser scan quality using a cheap depth camera.Pages 1995–1999 of the 2012 Proceedings of the 20th European Signal Processing Conference (EU-SIPCO).1,2,3,4,5,7

Facial Expression Recognition with 3D Deformable Models, C. Mayer, M. Wimmer, M. Eggers, and B. Radig. pp. 26–31 in Second International Conferences on Advances in Computer–Human Interactions, 2009.

T. Vetter and V. Blanz, "Face Recognition Based on Fitting a 3D Morphable Model," IEEE Trans. PAMI, volume 25, no. 9, 2003, pp. 1063–1074.

Deformation Modeling for Robust 3D Face Matching by X. Lu and A. Jain was published in Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, No. 8, 2008, pp. 1347–1356.

"Model-Based 3D Face Capture with Shape-from-Silhouettes," B. Moghaddam, J. Lee, F. Fister, and R. Machiraju. International Workshop on Analysis and Modeling of Faces and Gestures Proceedings of the IEEE, 2003.