Automated Attendance Portal using RFID

and Face Recognition

**1.Prof. Anmol Jain 2. Anand Parashar 3. Antriksh Tyagi 4.  Devraj Gupta 5. Ansh Srivastava**

***Department of Computer Science, KIET GROUP OF INSTITUTIONS, Ghaziabad***

[**1.anmol.jain@kiet.edu**](mailto:1.anmol.jain@kiet.edu)

[**2.anand.2024cs1105@kiet.edu**](mailto:2.anand.2024cs1105@kiet.edu)

[**3.antriksh.2024cs1203@kiet.edu**](mailto:3.antriksh.2024cs1203@kiet.edu)

[**4.dev.2024cs1025@kiet.edu**](mailto:4.dev.2024cs1025@kiet.edu)

[**5.ansh.2024cs1162@kiet.edu**](mailto:5.ansh.2024cs1162@kiet.edu)

**Abstract-**

In terms of time and staff workload, traditional methods for documenting student attendance in a classroom, such as roll-call and sign-in sheets, have been shown to be inefficient. Furthermore, they are vulnerable to human mistakes and phony attendance, both of which lead to inaccuracies in the data recorded. Research has been carried out to improve how we keep track of class attendance. Some of the recommended alternatives, however, are both pricey and impracticable. Most solutions also overlook the issue of phony attendance. This paper outlines a low-cost method for keeping track of student attendance. In this day of quickly growing new technologies, there is no reason why a critical educational practice like attendance should be considered in the traditional, monotonous way. It is challenging to manage big groups of students in a classroom using the traditional way. It is not advised since entering data into a system takes time and has a significant risk of error. Real-Time Face Recognition is a useful way of dealing with the daily attendance of a big number of pupils. Many algorithms like Haar Cascade Classifier for face detection and LBPH for face recognition have been used, but our suggested model, which is implemented in Python and the OpenCV library, determines the positive and negative features of the face for face recognition. The Tkinter GUI interface is used for user interface purposes.

We have developed the website by integrating facial recognition technology with the robust capabilities of MySQL, Apache server, and PHP. This platform offers unparalleled efficiency and accuracy. This commitment to maintaining a corruption-free database instills confidence in users, assuring them that their attendance records are accurate, reliable, and tamper-proof. With the integrity of the database upheld, our platform not only enhances efficiency but also fosters trust among students, educators, and administrators alike, establishing it as a cornerstone solution in modern attendance management.

**Index Terms-** Real-time Face Recognition, Haar Cascade Classifier, Python, Open CV, Tkinter.

**INTRODUCTION**

This is a project about Image Based Attendance System for Educational Institutions. In this chapter, the problem, research objectives, contributions to society and the background information of the project will be discussed in detail.

**A) The Problem**

The most serious issue, according to the previous attendance management system, is indeed the accuracy of the data collected. This is because attendance may not be recorded personally by the original person; in other words, a specific person’s attendance may be taken by a third party without the institution’s knowledge, which violates the accuracy of the data. If the institution establishes enforcement, it may have to waste a lot of human resources and time, which is not practical at all. As a result, all the previous system’s recorded attendance is unreliable for analysis. The previous system’s second flaw is that it takes too long to complete. Assume a student sign his/her attendance on a name list in about 1 minute. Only about 60 students can sign their attendance in one hour, which is obviously inefficient and time consuming. The third issue is the legitimate interested party’s ability to access that information. For example, most parents are very concerned about tracking their ward’s exact whereabouts to make sure that they regularly attend college/school classes. However, in the previous system, parents had almost no access to such information. As a result, the previous system must be evolved to maximize efficiency, data accuracy, and information accessibility for those authorized parties.[1]

**B) Research Objectives**

Typically, there exist two primary methodologies for addressing human facial recognition: the feature-based approach and the brightness-based approach. The feature-based methodology relies on identifying distinctive landmarks on the face, such as the eyes, nose, mouth, and edges, to calculate recognition[2]. This method only analyses specific areas of the image during processing. Conversely, the brightness-based methodology, also known as the holistic or image-based approach, evaluates the entirety of the image. Since it necessitates considering the entire picture, this method is more time-consuming and complex. Throughout the development of facial recognition systems, advancements have been made, with face detection and recognition emerging as pivotal stages. Initially, images of students' faces are captured using a classroom-installed camera, positioned for optimal visibility. These images serve as inputs for the system. To ensure precise recognition, image processing techniques such as grayscale conversion and histogram equalization are applied to enhance image quality. Once the image is refined, it undergoes face detection before proceeding to the recognition phase.

**C)Significance And contribution**

The principal objective of this study is to develop an intelligent attendance management system utilizing facial recognition technology, aiming to tackle the shortcomings prevalent in current automated systems. The key approach involves comparing a current student photo with images deliberately captured and stored in a database, enabling attendance marking when real-time images match those in the database.

**D) Objectives**

Numerous contemporary attendance management systems exhibit inefficiencies and lack effective information sharing. In response, this project endeavors to address and enhance the following constraints:

1)Encouraging student attendance diligence: By implementing a system that mandates personal attendance verification, students are prompted to attend classes punctually, as any absences are promptly detected. This approach not only fosters a culture of accountability but also mitigates unethical practices like proxy attendance.

2)Resource optimization for institutions: Shifting enforcement mechanisms from manual supervision to technology-driven solutions results in substantial resource savings for institutions. By reducing reliance on human oversight for routine tasks, valuable human resources are preserved for more significant endeavors.

3)Enhanced flexibility and accessibility: The application's adaptability to diverse devices and locations, facilitated by Wi-Fi or Ethernet connectivity, renders the attendance system portable and deployable in various settings. For instance, positioning the device at classroom entrances streamlines attendance management processes.

4)Cost and time efficiency: The elimination of paperwork and automation of computations translate to significant cost savings and time efficiency. By streamlining administrative tasks, financial resources are conserved, and operational processes are expedited.

**METHODOLOGY**

We present a cost-effective solution for recording student attendance through the implementation of face detection technology. Our proposed system, named IBAS (Image-Based Attendance System), comprises four key stages: image acquisition, face detection, attendance registration, and attendance monitoring. The primary objective is to enhance staff efficiency and reduce workload, ultimately elevating the accuracy of attendance records. While conventional methods like fingerprint scans, retinal scans, and access cards are commonly used for attendance tracking, our paper advocates for the utilization of face recognition technology. Specifically, we employ the Haar cascades and LBPH algorithm to identify faces within images. This approach aims to automate classroom attendance without direct teacher involvement. Haar cascades offer a distinct advantage with their rapid face detection speed, making them superior to existing techniques. Custom Haar cascade classifiers are generated for each user, trained using positive or face-containing images. These classifiers are then utilized for face detection and recognition tasks. Our implementation encompasses four key stages: capturing video images, converting images to grayscale, storing them in a dataset for training, and finally, identifying faces and recording attendance based on input images with trained faces. Each identified face is associated with a corresponding student ID during dataset creation, ensuring accurate attendance tracking.

**A) Face Detection**

The core Viola and Jones face detection algorithm typically operates across 150 frames. However, to adapt these foundational techniques for diverse real-time applications, numerous developers and academics have refined them over recent years. One approach involves applying the face detection algorithm solely to segmented regions post-background subtraction, effectively reducing computational complexity. In our implementation, we employ a wavelet transform for face detection. Wavelet coefficient subsets are utilized to represent the item's shape, while integral images facilitate the computation of Haar features. These features are derived by computing the variance difference between black and white regions within rectangles, a process facilitated by integral and squared integral images. This technique not only enhances computational efficiency but also ensures accurate face detection in real-time scenarios.

**B) Database Creation**

The initial phase of implementing an automated attendance tracking system involves registering each student in the class. It is imperative to thoroughly train the system to accurately identify the faces of individuals. Hence, through the initial step of face detection, the system extracts the faces of all relevant individuals from various photographs, compiling them into a dataset of grayscale images with dimensions of 200x200 pixels. For each unit, a collection of photos containing that unit is provided as input. During this phase, the system detects and captures faces from the input photographs, subsequently converting them into grayscale images.

Following the conversion process, each image file is meticulously labelled with a unique identifier, typically comprising the student's ID and USN (University Serial Number), thus facilitating further recognition of their identity. To enhance the precision of face recognition, it is imperative to train the system with diverse conditions encompassing the faces of all members involved. This comprehensive training regimen ensures that the system can accurately identify and register students' attendance under varying circumstances.

**C)Face Recognition**

The dataset now contains images of all class members captured under various conditions. Once these images are trained, they are converted into NumPy arrays. To label the test dataset obtained from the class, the trained classifier file is saved. Each class member's representative image serves as the input. The process commences with face detection to locate all faces, followed by identification using a local binary pattern histogram (LBPH). Subsequently, grayscale images are generated, and the classifier learned from training is employed for face recognition. Each recognized face is then assigned a Student ID label, facilitating accurate attendance tracking.

**ALGORITHMS**

A) LBPH (Local Binary Patterns Histograms)

Facial recognition, a prominent task in computer science, entails identifying individuals based on their facial features. Over the past two decades, this field has witnessed significant advancements, driven by novel techniques and the enhanced quality of modern cameras and recordings. One such technique, the Local Binary Pattern (LBP) texturing operator, assigns a binary label to each pixel by thresholding its neighbouring pixels. When combined with Histogram of Oriented Gradients (HOG) descriptors, LBP notably improves detection performance on specific datasets. Utilizing LBP alongside histograms enables us to represent facial images as straightforward data vectors, facilitating face recognition tasks.

Here's a step-by-step breakdown:

**1)Parameters:** LBPH utilizes four parameters:

a) Radius: Determines the radius for building the circular local binary pattern around the central pixel.

b) Neighbors: Specifies the number of sample points for constructing the circular local binary pattern.

c) Grid X and Grid Y: Define the number of cells in the horizontal and vertical directions, respectively.

**2)Training the Algorithm:** Training the algorithm involves using a dataset comprising facial photographs of individuals we aim to identify. Each image in the dataset is associated with a unique Student ID.

**3)Applying the LBP Operation:** Initially, the LBPH algorithm computes an intermediate image that accentuates facial features. This is achieved through a sliding window approach based on the specified radius and neighbours. Each pixel in the grayscale facial image is examined within a local window, and a binary value is assigned based on a threshold computed from the central pixel's intensity.

**4)Extracting Histograms:** Histograms are extracted from each grid in the grayscale image. Each histogram contains 256 positions representing pixel intensities (0-255). These histograms are concatenated to create a comprehensive representation of the image.

**5)Performing Face Recognition:** With the algorithm trained and histograms generated for each image, face recognition involves comparing histograms of the input image with those of the training dataset. The algorithm identifies the image with the closest histogram, typically measured using methods like Euclidean distance or chi-square. The calculated distance serves as a confidence measure. By setting a threshold, we can automatically determine if the algorithm has correctly recognized the image based on confidence levels.

**B) HCC (HAAR Cascade Classifier)**

A Haar classifier, also known as a Haar cascade classifier, serves as a machine learning object detection tool adept at identifying objects within images and videos. The algorithm unfolds in four key stages:

**1)Calculating Haar Features:** The process commences with gathering Haar features, which are essentially the outcome of calculations performed on adjacent rectangular sections within a detection window. This involves summing pixel intensities within each region and computing their differences. However, in large photographs, identifying these elements can be arduous. Integral images come into play here, drastically reducing computation by storing cumulative sums for each pixel.

**2)Creating Integral Images:** Integral images expedite the calculation of Haar features by creating sub-rectangles and establishing array references for each. This optimized approach replaces the need for pixel-by-pixel computations, streamlining the process significantly.

**3)AdaBoost Training:** Adaboost plays a pivotal role in selecting top features and training classifiers to utilize them effectively. By amalgamating multiple "weak classifiers" into a "strong classifier," Adaboost enables object detection. Weak learners are generated by sliding a window across the input image and computing Haar characteristics for each area. These weak learners are then combined to form a robust classifier, distinguishing between objects and non-objects based on learned thresholds.

**4)Implementing Cascading Classifiers:** The cascade classifier consists of multiple levels, each comprising weak learners. Boosting is employed during the training of weak learners to create a highly accurate classifier. At each stage, the classifier either proceeds to the next region (negative) or indicates the identification of an object (positive) based on predictions. Emphasis is placed on swiftly rejecting negative samples to expedite processing, as most windows typically do not contain objects of interest.

In summary, the Haar cascade classifier undergoes a meticulous process involving feature calculation, integral image creation, Adaboost training, and cascading classifiers implementation to proficiently detect objects in images and videos. This intricate methodology underscores the algorithm's effectiveness in object recognition tasks.

**SOFTWARES USED**

1. **MySQL**

The widely used MySQL server relational database management system is an indispensable element of web-based applications. Given the ubiquitous nature of data storage and transmission over the internet, databases and their associated tables serve as fundamental components of numerous websites and applications. A MySQL server is essential for executing various data operations such as querying, sorting, filtering, grouping, editing, and merging tables.

Facilitating seamless integration between Python and MySQL is the Python MySQL Connector, a Python driver crucial for bridging the two platforms. This Python MySQL library enables effortless conversion of data types between Python and MySQL, enhancing interoperability between the two environments. Remarkably, the MySQL Connector API is entirely developed in Python, eliminating the need for external libraries or dependencies.

In this study, a table named "student" was utilized, comprising columns such as Student ID (Primary Key), Name, Department, Course, Year, Semester, Division, Gender, Date of Birth (DOB), Mobile Number, Address, USN (Unique, Non-NULL), Email, and Teacher Name. Additionally, the table includes a column for Photo Samples.

1. **OpenCV**

OpenCV stands as a comprehensive open-source library renowned for its prowess in computer vision, machine learning, and image processing, making it indispensable in contemporary systems. Its significance in real-time operations is paramount in modern applications. Leveraging OpenCV, users can analyze images and videos to detect objects, faces, or even handwriting with remarkable accuracy. When coupled with libraries like NumPy, Python gains the capability to process the array structure of OpenCV for in-depth analysis.

To identify image patterns and their various features, vector space techniques are employed, enabling mathematical operations on these features for comprehensive analysis. OpenCV, with its initial release version 1.0, operates under a BSD license, rendering it freely available for both academic and commercial utilization. Supporting interfaces in C++, C, Python, and Java, and compatible with Windows, Linux, MacOS, iOS, and Android platforms, OpenCV caters to a broad spectrum of development environments.

The primary design focus of OpenCV was real-time applications, emphasizing computational efficiency. It empowers users to perform complex tasks with relative ease, providing routines capable of detecting faces and executing various other intricate operations. As a versatile tool, OpenCV facilitates the implementation of sophisticated functionalities, making it indispensable in the realm of computer vision and image processing. y. OpenCV enables us to do even more complex tasks relatively easy. There are for example routines, which detect face(s). Thefollowing sequence of commands does just that-

A computer screen with text on it

Description automatically generated

**C) PYTHON**

Python stands out as a programming language meticulously crafted to streamline the translation of concepts into functional code, catering to programmers of all proficiency levels. In the realm of machine learning, Python has emerged as the predominant, extensively developed, and meticulously supported language. Renowned for its high-level, versatile nature, Python enjoys widespread popularity among developers. The latest iteration, Python 3.9, finds application across a spectrum of domains, including web development, machine learning, and various cutting-edge software technologies, making it an ideal choice for novices and experts alike.

In image processing, the Python Imaging Library (PIL) historically provided comprehensive support for image manipulation tasks. However, with PIL's development stagnating since its last release in 2009, the actively maintained Pillow library has emerged as a viable alternative. Pillow, a derivative of PIL, offers enhanced compatibility with Python 3 and streamlined installation across major operating systems, featuring fundamental image processing capabilities like color space conversions, point operations, and filtering.

For computer vision applications, OpenCV (Open-Source Computer Vision Library) stands as a cornerstone tool. Its Python API, OpenCV-Python, combines the efficiency of C/C++ background code with the simplicity and ease of Python programming, making it ideal for computationally intensive vision tasks.

Python also boasts various options for developing graphical user interfaces (GUIs), with Tkinter serving as the default GUI library. Tkinter, offering an object-oriented interface for the Tk GUI toolkit, facilitates rapid GUI application development. Creating a GUI application with Tkinter involves simple steps, including importing the Tkinter module, creating the primary window, adding desired widgets, and entering the main event loop to handle user-triggered events effectively. This seamless integration of Python and Tkinter empowers developers to craft intuitive and visually appealing GUI applications with ease.

**5. DATA FLOW**

The Face Recognition System DFD is a diagram used to show the overall data management of the project. It has 5main levels that shows the Face Recognition System data handling, which is the DFD Level 0, 1, 2,3 and 4. These DFD levels illustrate the Face Recognition data management concept from the basics up to specific details.

**A diagram of a level-dfd diagram

Description automatically generated**

**Fig.1.Level 0**

**A diagram of a student registration

Description automatically generated**

**Fig.2.Level 1**

**A diagram of a software development

Description automatically generated**

**Fig.3.Level 2**

**A diagram of a face recognition

Description automatically generated**

**Fig.4.Level 4**

**RESULTS AND DISCUSSIONS**

The system offers users a user-friendly GUI interface, presenting three primary options: student registration, faculty registration, and attendance marking.

**Student Registration:** Students are prompted to input all requisite details into the registration form. Upon clicking the register button, the system automatically activates the webcam.

**Image Dataset Creation:** The webcam proceeds to capture 150 images, facilitating the creation of an image dataset. Each student is assigned a unique ID number during this process.

**Attendance Marking:** During recognition, when a test student image matches with the dataset, the system records the student's details in the attendance Excel sheet along with a timestamp. Conversely, if the test student image fails to match with the dataset, their attendance is not marked. After a specified period, unmatched students are marked as absent.

This seamless interaction streamlines administrative processes, ensuring efficient and accurate attendance management.

1. Images shows the nature of the system when it is fed with different size of datasets. Here we compare 3 groups of 2 data. Fig-5depicts the comparison between recognition rate of the system with different camera angles

|  |  |  |
| --- | --- | --- |
| VIEW(Distance between camera and face( between 35-100cm approx.) | ANGLE(IN DEGREES) | RECOGNITION RATE IN NORMAL LIGHT CONDITION |
| LEFT | 30  45  90 | 100  90  0 |
| CENTER | 0 | 100 |
| RIGHT | 30  45  90 | 100  97  0 |

Fig.5. Recognition Rate at different face angles

1. The training period to train 10 images is 0.6 seconds, 50 images is 1.69 seconds, 100 images is 2.71 seconds, and 150 images is 3.67 seconds. The recognition time for a single face is 1.1 seconds. Similarly, the recognition time for 3and 7 images is 1.4 and 1.8 seconds respectively. and the recognition time for 10 faces is approximately 2 seconds.

**CONCLUSION**

Incorporating an automated attendance system into lectures, sections, or laboratories significantly streamlines the process for teachers, saving valuable time and effort, particularly in larger class settings. The primary objective of this system is to mitigate the limitations associated with traditional manual methods. By showcasing the application of image processing techniques within the classroom, this system not only enhances attendance tracking but also contributes to the institution's overall reputation. Moreover, the project underscores the immense potential of machine learning in educational settings.

Challenges such as poor classroom lighting affecting image quality can be addressed through subsequent stages by improving video quality or implementing advanced algorithms. Utilizing state-of-the-art processors can further enhance image processing speed, optimizing system performance. Additionally, integrating GSM technology enables the seamless transmission of attendance details to students' respective parents, fostering transparency and communication.

To enhance user experience, the GUI can be rendered more interactive, allowing students to access their attendance records within defined parameters. This functionality empowers students to stay informed about their attendance status, fostering accountability and engagement in their academic journey. Overall, these enhancements not only elevate the efficiency of the attendance system but also contribute to a more dynamic and engaging learning environment.

**REFRENCES**

* 1. Heusch, G., Rodriguez, Y., Marcel, S.: Local Binary Patterns as an Image Preprocessing for Face Authentication. In: Proc FGR 2006, pp. 9–14 (2006)
* 2. Zhang, G., Huang, Z., Li, S.Z.: Boosting local binary pattern (LBP)-based face recognition. In: SinoBiometrics 2004, pp. 179–186 (2004)
* 3. Beveridge, J.R., Bolme, D., Teixeira, M., Draper, B.: The CSU Face Identification Evaluation System User’s Guide: Version 5.0, Technical Report, C.S. Dept., CSU (2003)
* 4. Messer, K., Matas, J., Kittler, J., Jonsson, K.: XM2VTSDB: The extended M2VTS database. In: Akumuri, S., Kullman, C. (eds.) AVBPA’99, pp. 72–77 (1999)
* 5. Messor, K., Kittler, J., Short, J., Heusch, G., Cardinaux, F., Marcel, S., Rodriguez, Y., Shan, S., Su, Y., Gao, W., Chen, X.: Performance Characterisation of Face Recognition Algorithms and Their Sensitivity to Severe Illumination Changes. In: ICB2006, pp. 1–11 (2006)
* 6. Philips, P.J., Moon, H.J., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face Recognition Algorithms. PAMI 22(10), 1090–1104 (2000)
* 7. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 45–56. Springer, Heidelberg (1996)
* 8. Shan, S., Zhang, W., Su, Y., Xhen, X., Gao, W.: Ensemble of Piecewise FDA based on Spatial Histogram of Local (Gabor) Binary Patterns for Face Recognition. In: ICPR (2006)
* 9. Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: Application to face recognition. PAMI 28(12), 2037–2041 (2006)
* 10. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. PAMI 24(7), 971–987 (2002)
* 11. Zhang, W., Shan, S., Zhang, H., Gao, W., Chen, X.: Multi-resolution histogram of local variation patterns (MHLVP) for Robust Face Recognition. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 937–944. Springer, Heidelberg (2005)
* 12. Huang, X., Li, S.Z., Wang, Y.: Jensen-Shannon Boosting Learning for Object recognition. In: Proceeding of CVPR (2005)