DE-42 (EE) AYESHA, ASFAND, SALMAN, UMAIR

Deep Learning based Smart Attendance System using Facial Recognition

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY RAWALPINDI

2024

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DE-42 EE
PROJECT REPORT

Deep Learning-based Smart Attendance System using Facial Recognition

Submitted to the Department of Electrical Engineering

in partial fulfillment of the requirements

for the degree of

Bachelor of Engineering

in

Electrical

2024

Sponsoring DS:

Supervisor: Asst. Prof Kamran Aziz Bhatti

Co-supervisor: Dr. Usman Ali

Submitted by:

Ayesha Fouad

Asfandyar Khan

M. Salman

M. Umair

ACKNOWLEDGMENTS

Everything that we have achieved in this endeavor and everything that we will go on to achieve, we owe it to Allah Almighty. We couldn't be more humbled and grateful as the Almighty made it easier for us throughout this journey, from brainstorming to ideation to finally execution, it was all Him who guided and enabled us to remain steadfast every step of the way. Without His assistance, nothing would even remotely be conceivable. **Alhamdulillah** for everything.

The FYP process wasn't straightforward and easy. It was accompanied with lots of ups and downs and lots of hurdles as well. The fact that we even got it completed in this time, a lot of credit goes to both our supervisors, **Assistant Professor Kamran Aziz Bhatti** and **Doctor Usman Ali.** Both proved to be a beacon of light for us and enabled us to think outside the box. We extend our deepest gratitude to both our FYP supervisors for always showing us the right way, always being readily available whenever we hit a brick wall and for remaining calm and composed throughout this journey. It would not have been possible had it not been for your support, mentorship, and valuable contributions.

We would also like to express our gratitude to our parents, who have enabled us to get here by working tirelessly to provide us with the right resources and creating the means to us getting here. We would forever be indebted to them.

Here's to all the peers and colleagues who helped us throughout this journey, your invaluable contribution was detrimental to our success, and we will forever be indebted to everyone involved.

ABSTRACT

The implementation of an attendance management system is a crucial practice that any institution or organization adopts to record the attendance of their employees or members. The manual process of recording attendance using a paper-based or file-based system is fraught with deficiencies, including the potential for data loss, tampering, or catastrophic events. The current standard specifies the implementation of a smart attendance system that utilizes RFID tags, fingerprints, iris scans, speech recognition, and other technologies. Currently, advancements in technology are promoting the practical application of facial recognition methods for a more effective attendance management system. The facial recognition-based attendance system offers convenience and additional benefits by eliminating the need for human intervention, helping to mitigate the transmission of infections. This study proposes the use of a facial recognition technology in a real-time attendance management system to identify persons.

Deep learning-based face recognition systems have demonstrated superior accuracy and performance in comparison to other face identification approaches, such as eigenfaces. Contemporary face recognition systems have distinct stages, including face detection, face alignment, feature extraction, face representation, and face recognition. This paper presents a deep learning methodology for creating a class attendance system that is based on face recognition. The Multitask Convolutional Neural Network (MTCNN) is employed for the purpose of detecting and aligning faces, while a lightweight hybrid high performance model is utilized.

Ultimately, the project successfully achieved its goals and objectives, which involved creating a facial recognition system specifically built for tracking attendance and completing a comprehensive evaluation of existing methodologies and outcomes in facial recognition algorithms. In addition, the web application built by Streamlit offers key functionalities such as database creation, facial recognition, and attendance monitoring. Its purpose is to simplify attendance monitoring for end-users. An analysis was conducted to evaluate the system's performance and usability, with the aim of identifying areas for future improvements.

SUSTAINABLE DEVELOPMENT GOALS

GOAL 4: QUALITY EDUCATION

Quality Education is essential for personal development, societal growth & economic development of any society around the world. This SDG not only focuses on inclusive but equitable education for all. The *Smart Attendance System* directly contributes to the above by automating attendance tracking. Accurate and timely attendance tracking allows all the educational institutes to allocate resources efficiently. Not only this, but The *Smart Attendance System* can also be used for identification of a certain pattern of absenteeism within students and hence can be used for personalized assistance to individual students and helping them to improve their academic shortcomings.

GOAL 8: DECENT WORK AND ECONOMIC GROWTH

The SDG 8 aims to promote sustained, inclusive, and positive economic growth. Our system ensures better and equitable resource allocation through efficient attendance management. Educational institutes can optimize staff schedules and allocate resources effectively. In the real-world scenarios, particularly in the scenario of the country we are living in, we know how time-consuming manual attendance taking is, killing up time which could be better utilized in something productive. Automation of the attendance system through our system frees up administrative restrictions, opening time for more productive tasks for both the students and the faculty members. As a result of increased productive time, the culture of educated workforce comes through and hence contributes to economic growth and a thriving economy.

GOAL 9: INDUSTRY, INNOVATION & INFRASTRUCTURE

SDG 9 encapsulates the essence of building robust and reliable infrastructure systems, promotes sustainable industrialization and fosters innovation. Our attendance system relies on infrastructure components such as the camera, raspberry pi, and a database. Since our system is robust, it can withstand many technical challenges and ensures consistent attendance tracking. Optimum placement of the camera within the educational institute also enhances system reliability. Since our system is fair for all students and staff and does not perpetuate inequalities, it aligns perfectly with SDG 9. Not only that, but our system also fosters innovation by leveraging cutting edge technology and advanced programming language utilization for efficient and robust project.

TABLE OF CONTENTS

CONTENTS:

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS & ACRONYMS

- **3D**: Three-Dimensional
- **ANN**: Artificial Neural Network
- **CMC**: Cumulative Match Characteristic
- **CNN**: Convolution Neural Network
- **COVID**: Corona-virus Disease
- **CPU**: Central Processing Unit
- **DET**: Detection Error Trade-off
- **DNN**: Deep Neural Network
- **DOM**: Document Object Model
- **EER**: Equal Error Rate
- **FAR**: False Accept Rate
- **FCNN**: Fully Convolutional Neural Network
- **FN**: False Negative
- **FP**: False Positive
- **FPR**: False Positive Rate
- **FPS**: Frames Per Second
- **FRR**: False Reject Rate
- **GLM**: Generalized Linear Model
- **GUI**: Graphic User Interface
- **HOG**: Histogram of Oriented Gradients
- **IoT**: Internet of Things
- **KPCA**: Kernel Principal Component Analysis
- **LBP**: Local Binary Patterns
- **LDA**: Local Binary Patterns
- **LFW**: Labelled Faces in the Wild
- **MAE**: Mean Absolute Error
- **MAP**: Mean Average Precision
- **MSE**: Mean Squared Error
- **MTCNN**: Multi-Task Cascaded Convolutional Neural Networks
- **OpenCV**: Open-Source Computer Vision
- **PCA**: Principal Component Analysis
- **RAM**: Random Access Memory
- **RFID**: Radio-frequency Identification
- **RMS**: Root Mean Square
- **ROC**: Receiver Operating Characteristic
- **ROI**: Regions of Interest
- **SIFT**: Scale-Invariant Feature Transform
- **SSD**: Single-shot Multibook Detector
- **SVM**: Support Vector Machine
- **TAR**: True Acceptance Rate
- **TN**: True Negative
- **TP**: True Positive

CHAPTER 1: INTRODUCTION

1.1. THEORETICAL FRAMEWORK

Attendance is defined as the consistent presence at a certain event or gathering. The system is essential for numerous organizations and institutions, including schools, universities, and enterprises. The government establishes workplaces to effectively monitor the performance and assess the competence of its employees. In educational institutions all around the globe, it is obligatory for lecturers to record the attendance of students. Conversely, in the industrial sector, companies utilize an attendance management system to assess daily monitoring, maintain leave records, and calculate overtime for employees, thereby facilitating payroll systems.

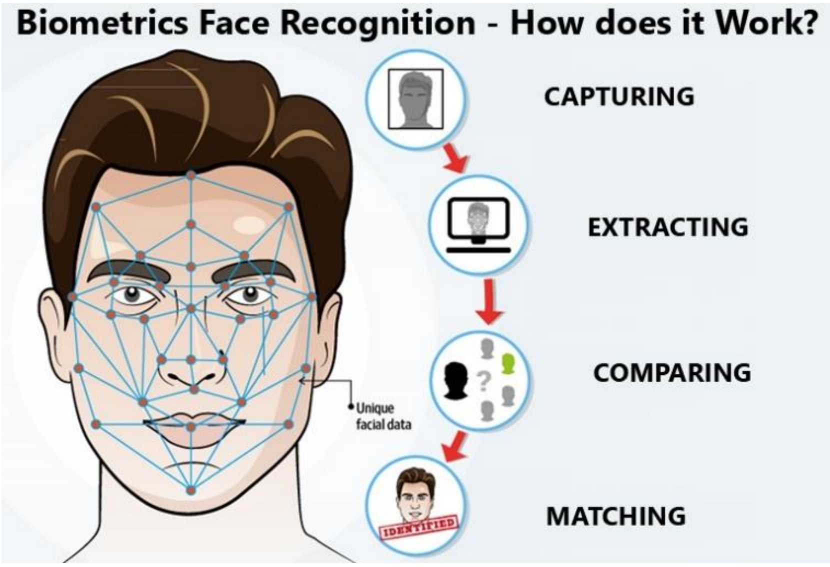


Figure 1: Breakdown of face recognition system's working.

Many institutions continue to employ paper-based or file-based methods for managing attendance, which are prone to several shortcomings including the potential for data loss, lack of efficiency, manipulation, and vulnerability to calamities such as floods or fires. Simultaneously, certain organizations have adopted automated attendance system tactics utilizing biometric techniques such as fingerprints, iris, voice recognition, or facial identification. Nevertheless, face recognition continues to be a prominent field of study because of its noninvasive characteristics and its role as the predominant method of human identification.

Figure 2: Commonly used Attendance Tracking System

Face recognition is a biometric system that can identify or recognize individuals based on images or continuous frames from a video (Sabeenian et al., 2020). Face recognition was initially developed in the 1960s using a somewhat automated technique. The recognition device will automatically identify a person's face traits. This technique includes the identification of faces, determination of their positions, preparation of images, improvement of image quality, and recognition of faces. The algorithm entails scanning the entire image to identify the coordinate system of all faces present. This is followed by using the face recognition algorithm, which outputs the shapes of the detected faces, such as rectangles or squares.

Facial recognition technologies have been adopted by several businesses in Southeast Asia. In Singapore, citizens utilize face recognition technology to gain access to government services, public transit, and even participate in election voting. The Singaporean government is renowned all around the world as a leader in this technology, enabling individuals to access government services conveniently and efficiently without physical touch.

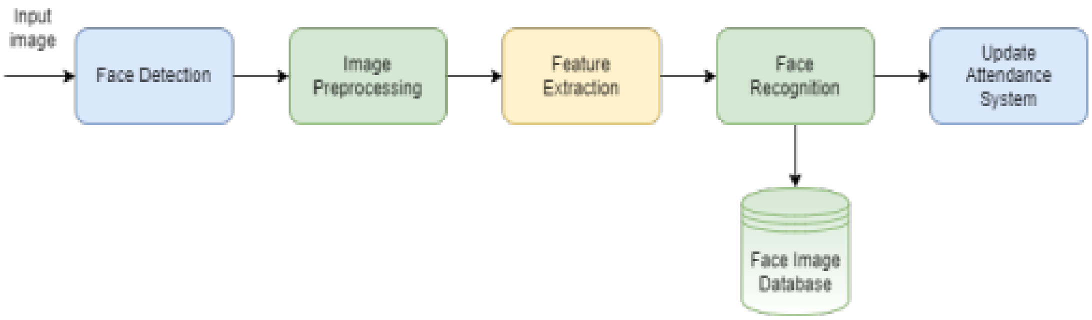


Figure 3: Facial Recognition processing flow diagram.

Face recognition systems have been effectively integrated into several application domains due to their need for high levels of acceptance, data collection, and universality. Multiple businesses have incorporated facial recognition into numerous applications. Apple has implemented a cutting-edge facial verification system that utilizes an advanced TrueDepth camera. This camera precisely captures and analyzes thousands of invisible dots to generate a detailed depth map of an individual's face. This depth map is then used to unlock customers' devices.

Deep learning can be combined with face recognition to demonstrate its robustness and precision in managing large volumes of data. Currently, deep learning is the primary framework for the image identification process, utilizing neural networks that employ several layers to extract features.

Due to technological advancements and the application of deep learning, the efficiency of facial recognition algorithms has greatly improved, enabling their widespread usage in various commercial applications in modern society. This project presents a smart attendance system that utilizes face recognition through a deep learning methodology. The aim is to address the limitations of traditional approaches, which are both time-consuming and intricate.

1.2. IMPORTANCE OF STUDY

Biometrics, cards, and iris systems are all part of the traditional attendance system. Biometric approaches have become a valuable method of identifying people in recent times, as opposed to depending on tangible or virtual domains like smart cards, tokens, passwords, and keys. For example, passwords can be easily guessed or hacked by third parties, but the magnetic cards can become unreadable or taken by others. The card system takes into account the possibility of attendance records being misplaced, which could lead to third parties taking fraudulent attendance. The possibility of these methods being missing, forgotten, or replicated comes with them, which is why the attendance system is ineffective.

However, it is difficult or impossible for someone to replicate a person's biological characteristics. Except for the face recognition technique, which allows the user to complete the biological traits—face, fingerprints, iris, palm, and voice—passively from a distance using a camera or video recorder. This situation involves the use of the facial recognition method for identification, wherein the picture of an unknown person is processed by comparing it to a database of photographs of known people. Face recognition technology is gaining popularity as a solution, due, to its user friendly nature compared to other biometric methods. Many industries have already benefited from its deployment in the present markets.

Face recognition is not just for use in attendance tracking; there are a few other uses for this technology. First of all, it is frequently utilized for network security, building access, border crossings, airports, seaports, and electronic device security. The installation of CCTVs can be used for surveillance, which is its second important function, to monitor criminals or offenders. Face recognition augmentation for security can be implemented on existing deep learning and computer vision-enabled security and surveillance cameras.

These days, face recognition technology has made identification verification for birth certificates, national identities, passports, and driver's licenses commonplace. The technology also helps with situations involving immigrants, missing persons, forensics, and a host of other circumstances. It is indisputable that the public has benefited from the implementation of facial recognition technology in terms of convenience, efficiency, and security, notwithstanding privacy concerns voiced by some.

1.3. AREA OF INQUIRY

In many organizations, including government offices, businesses, industries, and universities, managing attendance records is a need. While humans are capable of taking manual attendance, the process is laborious and time-consuming because human memory is not as flexible when it comes to memorizing a large collection of faces. Additionally, there is a chance that human data entry will contain inaccuracies, and cross-departmental or departmental verification of attendance data may present challenges. This causes automatic facial recognition systems to grow quickly and has the potential to change a lot of things, such as accurately tracking attendance in real time.

Users typically come into physical touch with the devices while using the traditional method of clocking in and out, which involves using a card system or a biometric system like a fingerprint. Because it doesn't require human intervention, the facial recognition-based attendance system is safer and more convenient than other biometric systems. The virus that causes the sickness can spread through direct touch with an infected person as well as indirect contact with objects or surfaces in the surroundings. As a result, this is the ideal moment to move from the contact-based attendance system to an indirect contact strategy using face recognition to prevent the spread of infectious diseases in the unfortunate case of an outbreak of the COVID-19 pandemic.

1.4. AIMS AND OBJECTIVES

The main aim of our project is to develop a real time smart attendance system that tracks and marks attendance based on facial recognition by using a deep learning-based approach particularly.

The objectives of this project are:

- To research the current systems of attendance tracking that are already in place and create a system of our own that nullifies the problems faced by the other systems.
- To design, test, improve and implement a facial recognition system that can be utilized for attendance tracking in real time.
- To inculcate an anti-spoofing model in our system that increases its authenticity and credibility.
- To evaluate the performance and usability of the developed system in terms of accuracy and compare it with other systems.

1.5. SCOPE AND LIMITATIONS OF THE STUDY

The primary objective of this project is to develop face recognition systems that can be used to update attendance records by incorporating deep learning techniques. The technology is specifically built for an area with an abundance of light, to the extent that the facial features of individuals are easily and clearly discernible. The suggested technique lacks cloud backup, resulting in potential data storage issues and the inability to restore lost data.

1.6. CONTRIBUTION OF THE STUDY

Studies demonstrate that advancements in deep learning and neural networking could produce favor resulting in diverse domains, including imaging processing and patterns identified. Integrated facings recognition systems with deep leanings architectures will enhance performances by streamlining the learned process and improving the efficiency of current modeling. Even though there are concerns about private life, the advantages of technology are funny. The usage of facials recognizing technology is promising in creating convenient and efficient management systems, including other applications.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

1.7. DEEP LEARNING BASED SMART ATTENDANCE SYSTEM:

Attendance monitoring is of utmost importance in various organizations, such as educational institutions, enterprises, and government entities. In higher education systems, students may be obligated to maintain a specified level of attendance in order to qualify for the final examinations. Implementing the attendance system is essential for maintaining academic integrity and promoting students' performance. Within this particular framework, there continue to exist numerous attendance systems that rely on manual and paper-based processes, which are accompanied by a multitude of problems. As technology has progressed, many authors have suggested automated attendance marking systems as a potential alternative to replace the traditional method of collecting attendance.

In addition, a biometric method utilizing fingerprints has been demonstrated to effectively eliminate fraudulent attendance systems, however it does need physical interaction between people within the organization. In addition, certain researchers conducted a study on an attendance monitoring system that utilized QR codes on the Android platform. The system achieved a 100% success rate throughout functional testing. However, this implementation carries the potential for manipulating attendance records and is also a wasteful expenditure of time and energy.

Thus, in light of the shortcomings of the previously discussed attendance systems, facial recognition algorithms have gained popularity recently, suggesting that they could be a good choice for automated attendance marking systems. Table 1 presents a comparative analysis of various attendance system systems that are frequently utilized by the public.

Method	Hardware	Main Advantages	Main Disadvantages	References
Biometric Attendance	Fingerprint Reader, Retina Scan Machine	Automatic	Cost of machine, Maintenance issue	(Mohamed and Raghu, 2012)
RFID	RFID Reader, RFID Tag	Automatic	Risk of losing RFID Tag, Fake attendance	(Chatrati, Naidu and Prasad, 2013)
QR Code	Smartphone Camera, Barcode Scanner	Sub-automatic, cheap	Fake attendance	(Amirulloh et al., 2020)
Facial Recognition	Camera, Server	Automatic	Maintenance	(Nyein and Oo, 2019)

Table 1:Readily used Attendance Systems Comparison

1.8. FACE DETECTION:

All facial analysis algorithms, including those that recognize gender, align faces, identify faces, and detect facial expressions, depend on face detection as a cornerstone. An image-based facial recognition system was first presented by Paul Viola and Michael Jones in 2001. The robust and quick real-time face detection capabilities of the Viola-Jones object detection framework make it stand out. Through the ability to handle 45 degrees of face rotation in both the vertical and horizontal axes, this detector has demonstrated its effectiveness on frontal images of faces.

Nonetheless, research has demonstrated that the Viola-Jones method has its limits when it comes to face detection in low-lightened environments! Due to inadequate lighting and a high false detection ratio, a study conducted by Chaudhari et al. (2018) using the Viola-Jones algorithm only achieved a success rate of 78.6%. With an accuracy of 94%, Deshpande and Ravishankar (2016) presented a very effective face identification and recognition system that uses Viola-Jones, Principal Component Analysis (PCA), and Artificial Neural Networks (ANN) to handle these current problems. Furthermore, according to DaSan, Alqudah, and Debeir (2015) says that the accuracy of the Viola-Jones algorithm and neural network combination was 90.31%. A resemblance database called Bio ID-Face-Database was used for the implementation of both investigations.

Agarwal, Kumar, and Goyal (2017) used OpenCV and Haar-like characteristics to build a research for the detection of faces in HD video. When determining whether an image is a face or not, the Haar classifier face detector employs a single characteristic. Nonetheless, the author came to the conclusion that because of their shorter time length and lower CPU resource requirements, the Viola and Jones detectors are more effective for real-time detection.

Table 2 illustrates the results of evaluating five photos with four different face detection algorithms. Under ideal lighting conditions, DNN outperformed MTCNN.

Number of faces detected for each condition

	Good-Lighting Conditions (N=5)		**Low-Lighting Conditions (N=5)**		**Occlusion (N=5)**		**Frames per Second (Average)**	
	TRUE	**FALSE**	**TRUE**	**FALSE**	**TRUE**	**FALSE**		No. of faces=20
Dlib	5	0	4	1	3	2	19	Score Threshold: 0.5
OpenCV Harr	5	0	2	3	3	2	30	True: No. of images for which faces are detected
MTCNN	5	0	5	0	4	1	7	False: No. of images for which not all faces are detected
OpenCV DDNN	5	0	4	1	4	1	23	

Table 2: Testing Results on each Face Detection Algorithm

1.9. FACE RECOGNITION:

Verification and identification are the two main uses for face recognition technology that have been created. In this case, the verification is the process of using one-to-one matching to ascertain whether the person presenting with an unknown person's face image is who they say they are or not. With the latter, one-to-many matching is used to compare an individual's identity with a database of photographs or videos of people who are known to be who they are.

With the advent of the historical Eigenface approach in the early 1990s, the field of face recognition research has gained significant attention. Uncontrolled face alterations were not taken into consideration by the many holistic techniques that were introduced. Early in the new millennium, face recognition based on local features was developed as a result. Unfortunately, the compactness and originality of these techniques were lacking. Consequently, the field of face recognition saw the introduction of learning-based descriptors. When AlexNet used convolutional neural networks to win the ImageNet competition in 2012, the deep learning approach underwent a significant change.

Even with the progress made in facial recognition technology, there are still numerous obstacles to be solved. The heterogeneity of photos containing the same face is a primary challenge for face recognition systems. Occasionally, the identified face is too small and lacks focus, which causes the recognition process to fail. When comparing an individual's photo with the database, further concerns include variations in posture, lighting, occlusion, and makeup (Solomon, Meena, and Kaur, 2021). As a result, a wide range of face recognition systems with various algorithms have been put forth recently to address the issues and produce encouraging outcomes while dealing with the difficulties presented by unrestricted contexts.

1.9.1. HIERARCHICAL FACE RECOGNITION ANALYSIS

Face recognition solutions can be efficiently organized and planned with the help of the multi-level face recognition taxonomy. Apart from looking at novel approaches to building more potent facial recognition systems, its goal is to help academics better grasp the state of technology in this area. Component and structural representation, global representation, and component representation are the three main face structures covered by the taxonomy. Feature extraction from the full face region is produced by the global representation face recognition technique, which represents the complete face as a single entity. The selection of a method based on the spatial support region and face structure for feature extraction is then taken into account by feature support. According to Sepas-Moghaddam, Pereira, and Correia (2020), the feature extraction level, on the other hand, offers a wider range of classes and a deeper technological landscape in feature extraction.

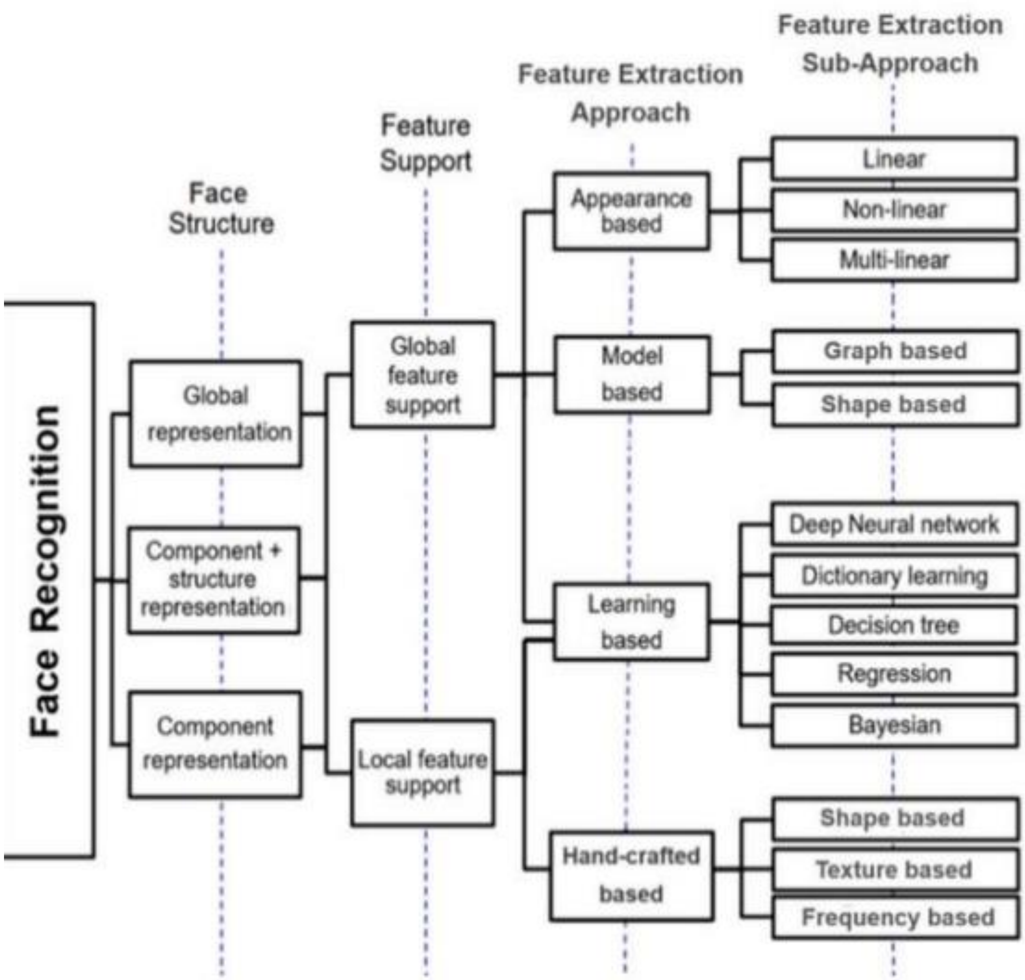


Figure 4: Existing Multi-level Face Recognition Taxonomy.

1.9.2. FACIAL STRUCTURE

Within the realm of face recognition, the term "face structure level" pertains to the manner in which various elements of the face are processed by the facial recognition system. On a global scale, the system prioritizes the face as a unified unit, taking into account its overall structure and characteristics. Facial recognition systems must take into account the structural composition of the face, including its many features such as the eyes, mouth, and nose, and their interrelationships. Under certain circumstances, the identification system may be required to choose a particular face feature while disregarding the rest. This is known as component representation. Precise facial recognition relies heavily on this level of detail, as distinct facial features can offer significant insights into identifying a person.

1.9.3. FACIAL FEATURE EXTRACTION

Data mining and pattern recognition require the crucial step of feature extraction. Through reducing duplication and irrelevance in the image, it seeks to achieve dimension reduction, hence reducing the length of machine learning and space complexity. Utilizing feature extraction methods, the input data in this case is transformed into a set of features comprising the essential information from the original data. Feature extraction keeps an appropriate level of classification accuracy by removing the greatest number of superfluous features.

A team of Google researchers suggested FaceNet, a deep learning network for face identification, to create vector embeddings of 128 values for every individual's face, as seen in Figure 5. This model uses a method called "one-shot training," wherein fresh face classification can be accomplished with just a few tiny arrays or embeddings from previously identified faces. Put differently, training the network with promising outcomes requires less pictures. Several well-known face recognition datasets, including Labeled Faces in the Wild (LFW), YouTube Faces (YTF), and MegaFace, were used to benchmark its remarkable performance.

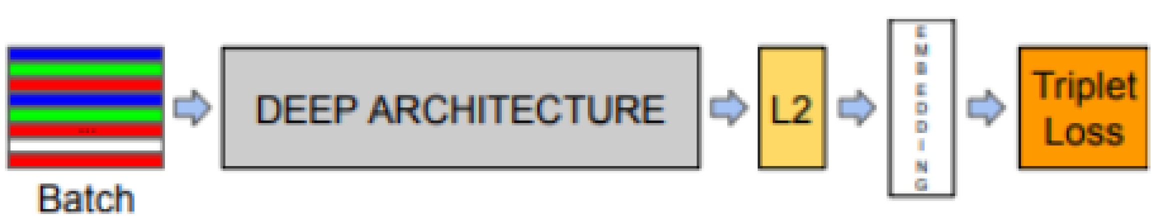


Figure 5: FaceNet Model Structure

Table 3 lists out the comparison and major differences between CNN, SSD and YOLO in terms of accuracy, speed, advantages, and disadvantages.

	CNN	SSD	YOLO
Accuracy	Accuracy is the highest among three models.	Accuracy is lesser as compared to CNN family.	Accuracy is the lowest as compared to CNN family and SSD.
Speed	Slower than SSD and YOLO.	Faster than CNN family but slower than YOLO.	Fastest among three algorithms.
Main Advantages	Reduced chance of overfitting due to fewer parameters to learn.	Can be trained end-to-end to improve accuracy.	Good for real-time processing. Can be trained end-to-end to improve accuracy.
Main Disadvantages	Requires lots of samples to construct a depth model.	Not suitable for small objects.	Difficult to detect objects that are small and close to one another.

Table 3: Comparison Between CNN, SSD and YOLO

1.10.EVALUATION METRICS

A suggested facial recognition system's performance must be assessed using a number of parameters. One particular evaluation matrix that the proposed system has is the phenomenon of getting multiple samples and embeddings whenever a video input is taken for the database. This ensures the collection of multiple samples and facial features collection for the detection of each single person. One particular evaluation matrix that is included in the system is the anti-spoofing mechanism, which ultimately means that features and samples will only be extracted if it's a real person, and if a video and image is shown, it will automatically classify it as fake.

It is crucial to take into account a number of criteria when assessing the effectiveness of a suggested facial recognition system. False Accept Rate (FAR), False Reject Rate (FRR), Receiver Operating Characteristic (ROC), and Equal Error Rate (EER) are four well-known measures that were covered by Harakannanavar et al. (2019). The system's likelihood of mistakenly matching an input pattern to a template that does not match in the database is known as FAR, and its inability to find a match between the input pattern and a matching template in the database is known as FRR. The trade-off between FRR and FAR is shown by the ROC curve, which is a helpful visualization tool. The ROC curve also yields EER. Nguyen-Meidine et al. (2018) suggested using distinct performance criteria in the CNN architecture for face and head identification. Accuracy is measured using a ROC curve, where the false positive rate (FPR) is plotted against the true positive rate (TPR). The ratio of accurately recognized faces within target face regions of interest (ROI) throughout the total region of interest is known as the TPR.

1.11.DATABASE

For the purpose of creating and assessing reliable face recognition systems, choosing an appropriate face database is essential. For the database to encompass controlled variables like illumination, position, expression, occlusion, age, and ethnicity, there must be enough data and variability (Gross, 2005). Having a big training dataset is very important for deep learning-based face recognition algorithms to learn intricate features from photos. For example, using 14 million photos, ImageNet greatly aided in the construction of accurate deep learning object detection models.

One of the earliest face databases is that from the Olivetti Research Laboratory (ORL), which has 400 face photos from April 1992 to April 1994. All of the photos in this database were taken against a uniformly dark background for the face recognition experiment, which resulted in uneven lighting. The YALE dataset, another well-known face collection, has 165 grayscale photos of 15 different people. Nevertheless, there are certain restrictions on this database, including its limited sample size and the lack of information about environmental variables like ambient light (Kriegman, n.d.). The quality of a face database is influenced by various things.

First of all, because there is no standardization for shooting pictures and people may position differently, pose variance could have an impact on the face recognition system's performance. Therefore, because of their rigid imaging settings, position variance may deteriorate the performance of face recognition systems.

Furthermore, occlusion pertains to any obstructions present in an image, such as hands, hair, sunglasses, or other objects. Partial occlusion is defined as the condition where less than 50% of the face is obstructed. In addition, shadows can also be a contributing element to the occurrence of occlusion. These objects can provide challenges for facial recognition systems as they block crucial areas of the face that are necessary for precise identification.

Finally, the lighting is a crucial factor that relies on whether the images were captured in a regulated setting with a uniform background. Lighting can have a significant impact on facial look and result in discrepancies between photographs. The significance of having an ample and varied face database for training and testing a resilient face recognition system is underscored by these aspects.

Choosing an appropriate face database is crucial for the development of a reliable face recognition system. An extensive training dataset that exhibits ample variability has the potential to improve the precision of face recognition systems that rely on deep learning techniques.

Real-Time Attendance System

	Name	Role	facial_features
0	Faisal Iqbal	Student	0.2481621503829956 1.3083226680755615 0.5148047804832458 -0.483881
1	Salman	Student	0.05599689111113548 -0.10464556515216827 0.054543159902095795 0.46
2	Hammad Ur Rehman	Student	-0.8348372578620911 1.1636402606964111 -0.49378567934036255 0.17564
3	Ahtasham Afzal	Student	1.0329076051712036 0.9136879444122314 -1.3930598497390747 -0.854701
4	Abdur Rehman	Student	1.2693639993667603 -1.5193729400634766 -0.6644291877746582 -0.10056
5	Aazain Umrani	Student	1.2390354871749878 1.484392523765564 0.1635521799325943 0.83804225
6	Eni	Student	-1.2357771396636963 0.9158744812011719 -0.010784231126308441 0.0718
7	Hamza Zaib Aleem	Student	0.4786204695701599 -0.05280882865190506 -0.06873483210802078 -0.418
8	Awais Raza	Student	1.2530370950698853 0.681903064250946 -1.6663042306900024 0.2261772
9	Asfandyar	Student	1.288852572441101 -0.07105053216218948 -1.0497329235076904 1.559923

Figure 6: Dataset Collection through Redis

1.12.BEYOND THE RULES: DEEP LEARNING VS. TRADITIONAL METHODS

The facial recognition domain has undergone major modifications throughout the seasons. Initially, the old-fashioned techniques utilized edge and texture descriptors as well as contraption learning techniques embracing Support Vector Machines (SVM), Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA). However, these customary methods appeared to be insufficient in handling shifts in unregulated surroundings. Convolutional neural networks (CNNs) have been efficiently employed for image spatial diminution and recognition in modern periods, owing to their extraordinary precision and operational computation.

Facebook's DeepFace, a high-capacity model, was the first CNN approach used for facial recognition. It obtained an accuracy of 97.35% on the LFW datasets and reduced the error by 27% (Trigueros, Meng, and Hartnett, 2018). Furthermore, a study conducted by Setiowati et al. (2017) using the YALE dataset demonstrated that non-deep learning algorithms can obtain a maximum accuracy of 90.6% for low-high complexity tasks, whereas the deep learning approach can achieve an accuracy of 94.67% for tasks ranging from low to high complexity.

Most of the research that compares conventional and deep learning approaches to face recognition demonstrates that the latter yields superior results. Jayaswal and Dixit (2020) employed the Viola Jones method as the initial strategy for image detection, and subsequently utilized the Local Binary Pattern technique to extract facial features. Nevertheless, the real-time system processed a mere 300 datasets and attained a meager accuracy of 50%.

Afterwards, the deep learning method was implemented by first scaling the photographs using the MTCNN algorithm, and then utilizing the FaceNet model to extract images of superior quality. Consequently, it has been demonstrated that deep learning is well-suited for models that incorporate extensive datasets, as it achieved an accuracy of 96%. In addition, when the YALE dataset was analyzed using different levels of complexity, the CNN strategy achieved a recognition rate of 93.3%, while the LBP approach achieved a recognition rate of 85.75% (Setiowati et al., 2017). Table 2.3 provides a concise overview of the comparison between Support Vector Machine (SVM) and deep learning methods when applied to MNIST datasets.

Approach	**SVM**	**Deep Learning**
Operating time	46.54 minutes	11 hours and 50.41 minutes
Accuracy in training set	94.09 %	100 %
Accuracy in testing set	93.92 %	98.85 %
Means for extracting features	Manually and subjective	Automatically and objective
Means for processing data	Turn images into vector	Directly using images
References	(Lai, 2019)	(Lai, 2019)

Table 4: Performance Comparison for SVM and Deep learning.

In terms of accuracy and resilience, deep learning algorithms outperform classical approaches for face recognition. This is because of its ability to learn complex features straight from unprocessed input, which facilitates its adaptation to many image variants. Moreover, deep learning algorithms are more adept at managing extensive datasets, rendering them a fitting choice for the advancement of facial recognition technology.

In addition to its strong performance in terms of accuracy across several studies, deep learning is also connected with certain limitations. The primary challenge lies in the tremendous complexity of the task, which offers difficulties for deep learning algorithms to handle situations involving diverse facial positions, lighting conditions, facial expressions, and partial occlusions. Furthermore, the implementation of a deep learning methodology requires a significant investment of training time and computational resources to get superior generalization capabilities and prediction accuracy.

An effective solution can be achieved by combining traditional and deep learning methods in the system to achieve accurate results with minimal complexity. For example, when the Local Binary Patterns (LBP) method was used on the ORL dataset, it achieved an accuracy of 97%. In another experiment, the LBP methodology was combined with the Convolutional Neural Network (CNN) technology and used for a similar dataset.

The dataset achieved a recognition rate of 100%, the table below presents a comparison between traditional methods like HOG, LBP, or PCA and advanced deep learning methods like CNN, SSD, or YOLO.

Conventional Approach	Deep Learning Approach
Lower Accuracy	Higher accuracy
Consume lesser training time	Requires larger training datasets with longer training time
Used for simpler application	Solve complex problems

Table 5: Comparison between Conventional and Deep Learning Approach

To summarize, although deep learning methods have demonstrated exceptional performance in face recognition, they still have drawbacks, including their high complexity and resource consumption. An effective option to overcome these constraints is to combine classical methods with deep learning techniques to better enhance the field of face recognition.

CHAPTER 3: METHADOLOGY AND WORKPLAN

1.13.INTRODUCTION

This chapter will provide a detailed explanation of the importance of thorough and effective planning for all project operations. The goal is to monitor progress and ensure the successful execution of the project. In addition, a comprehensive methodology was explained in a concise manner to guarantee that the outcomes align with the project's goals and objectives. The technique will be presented in a step-by-step manner, including project planning, development process, facial recognition methodologies, design of the user interface for the smart attendance system, and remedies to any encountered challenges during the development process.

1.14.PROJECT PLANNING

The entire project was divided into two distinct sections, namely Final Year Project 1 and Final Year Project 2. Each of them lasted approximately 14 weeks. At the outset of the project, the project scope, problem statement, and objectives were explicitly defined to guarantee that the entire project was in line with the desired outcomes.

In addition, a literature analysis was undertaken to examine the background and various methodologies that guide the implementation of the project, while also offering a comprehensive understanding of the project to the audience. Prior to offering the approach or potential solutions, research was undertaken, followed by preliminary testing and analysis of the results.

1.15.PROJECT SCHEDULE

A fully functional prototype was created in Semester 08, along with a performance assessment. As scheduled, the first prototype was completed in Week 7. The performance of the recognition system was then examined in a dataset collection methodology, which helped to improve the prototype even further. The user experience has been improved with the addition of certain helpful features. The entire project was completed in Week 14. Figure below shows the Gantt chart for the project.

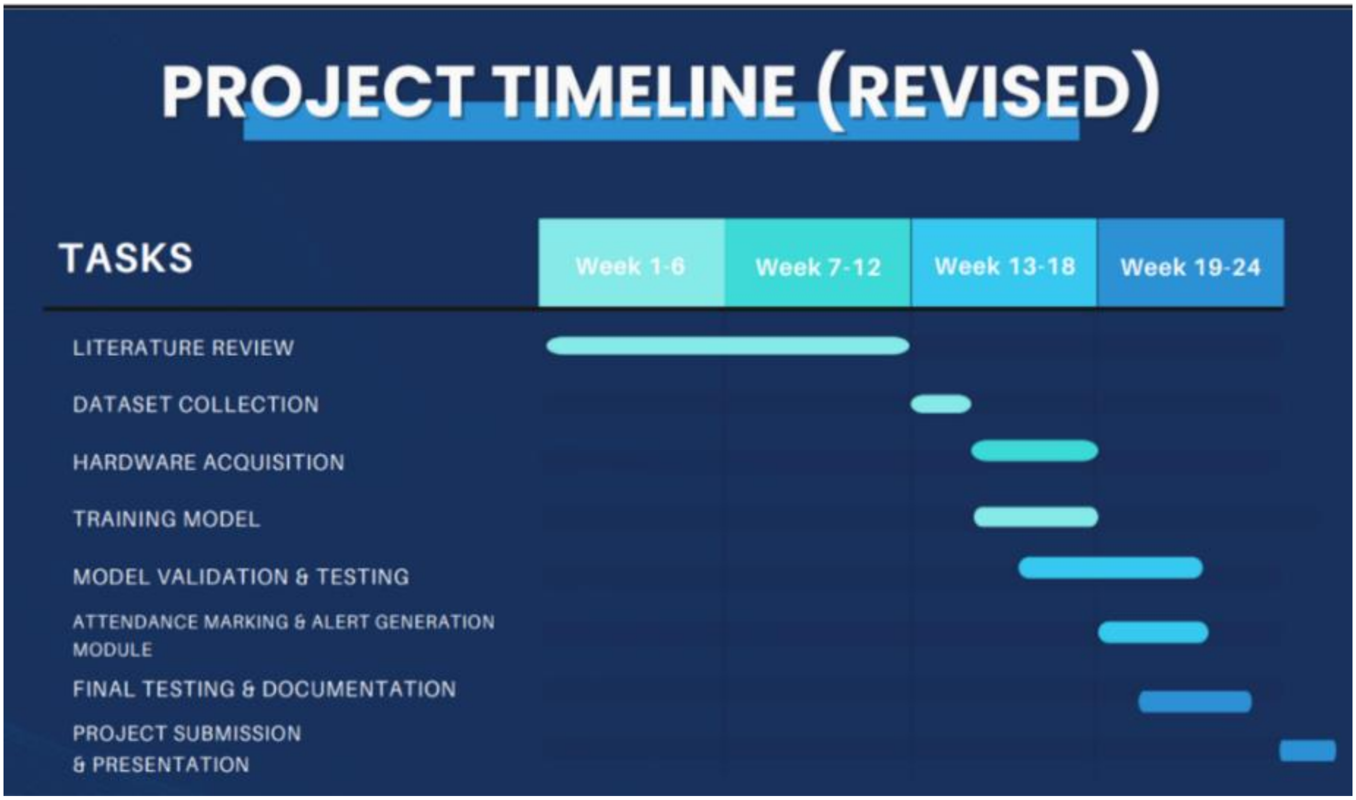


Figure 7: Project Gantt Chart showing timeline of the project.

1.16. PROJECT DEVELOPMENT

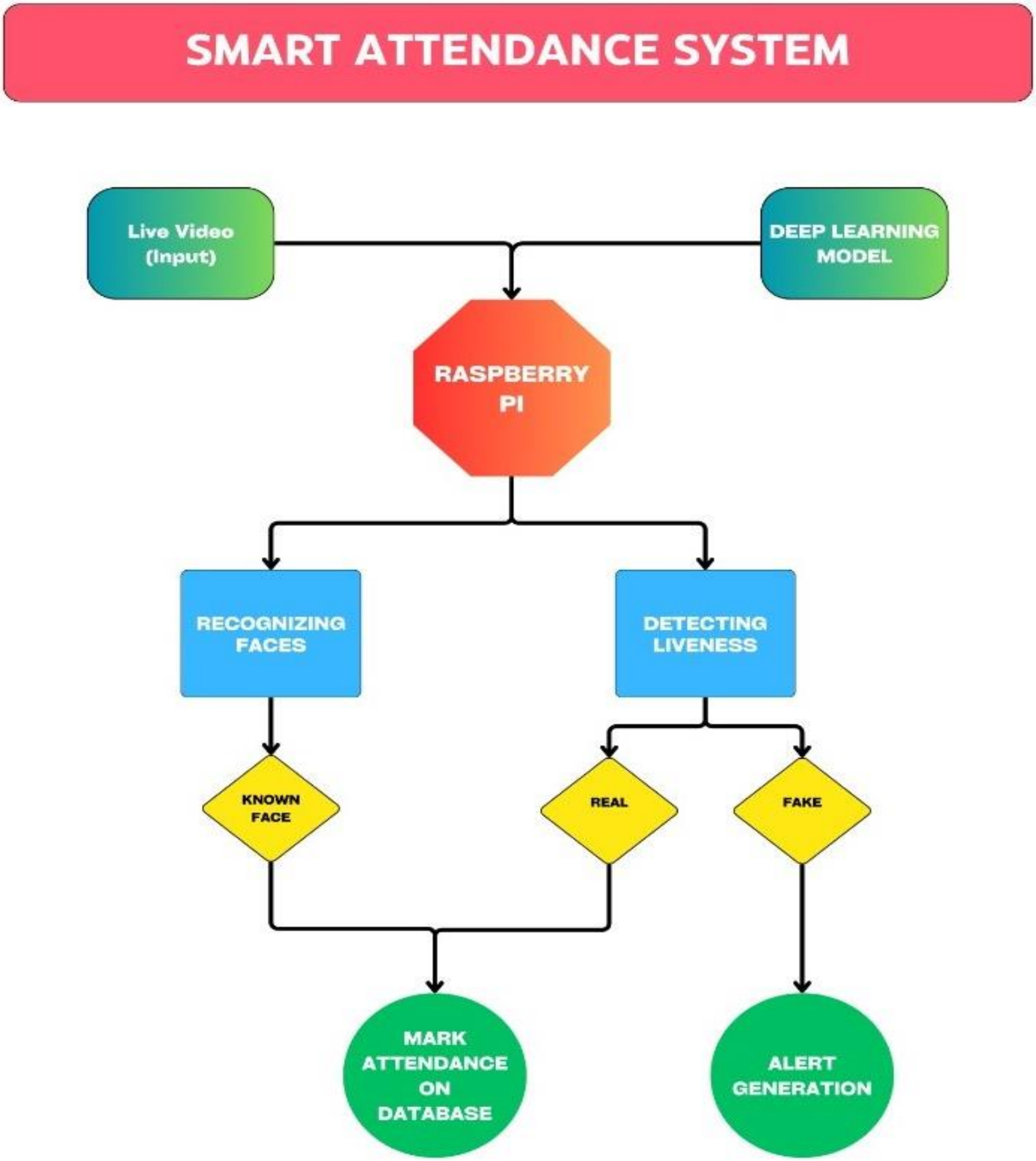


Figure 8: Project Development Chart

1.17.RESOURCES

1.17.1. HARDWARE

The Logitech HD Webcam C525 has been selected for this project's webcam. The C525 Webcams gives a smart and smooth video capturing quality with built-in focus which is ideal for any project that requires database collection through real time video monitoring. Not only that, but it is also very cost efficient and portable as well, which makes it all the more accessible on a wider scale for vase product simulation. All these cameras come in visible light and infrared versions.

Figure 9: Logitech HD Webcam C525

We also used raspberry pi 04 for the integration of our model. Compared to the Pi 3, the Raspberry Pi 4 is around 50% more powerful due to its quad-core 64-bit Broadcom BCM2837 ARM Cortex-A53 SoC processor, which operates at 1.2 GHz. Consequently, office apps and web browsing can be performed on the new Raspberry Pi 4.

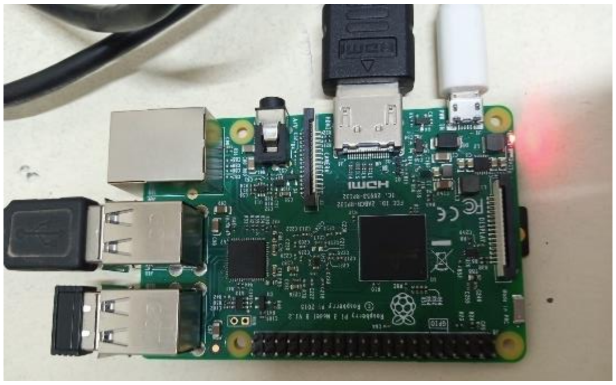


Figure 10: Raspberry Pi 04 connected to the system for model deployment.

1.17.2. SOFTWARE

The smart attendance system has been developed using Python as its primary programming language. Microsoft's open-source VS Code code editor is the development environment that's being utilized.

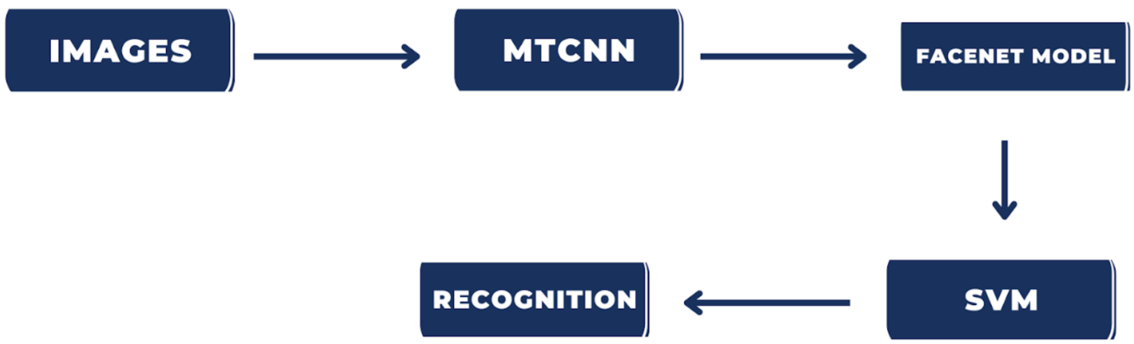


Figure 11: Deep Learning Model of Smart Attendance System

1.18.FACE DETECTION

The initial stage before moving on to the facial recognition portion is face detection. For the face detector, a variety of algorithms and methods can be employed. Multi-Task Cascaded Convolutional Networks (MTCNN), a well-known face detector, were employed in this study and its suitability for the given task was assessed. As a prerequisite for the face detectors to function, OpenCV was first installed. For item identification and detection, it offers sets of practical tools. By installing the MTCNN Library, MTCNN can be used without any further setup. Face and facial landmark detection in photos is accomplished via MTCNN, or Multi-Task Cascaded Convolutional Neural Networks. Zhang and colleagues published it in 2016.Its three convolutional networks (P-Net, R-Net, and O-Net) enable it to achieve real-time performance that surpasses numerous face-detection benchmarks.

MTCNN comprises three cascades of neural networks. P-net, R-net, O-net. Candidate windows are rapidly produced with a flat Convolutional Neural Network (CNN). Next, a more intricate convolutional neural network is employed to fine-tune the window and discard several obscured windows. Ultimately, we employ a more potent Convolutional Neural Network (CNN) to enhance our outcomes and generate the coordinates of the face landmarks.

A summary of the operation of this network is as follows:

- Step 1: To obtain the candidate windows and their bounding box regression vectors, a fully convolutional network known as the Proposal Network (P-Net) was used. The candidates are then calibrated using the computed bounding box regression vectors. Subsequently, candidates that exhibit significant overlap are combined via non-maximum suppression (NMS).
- Step 2: Every candidate is sent into a second CNN known as Refine Network (R-Net), which combines NMS candidates, rejects a significant number of bogus candidates, and calibrates using bounding box regression.
- Step 3: The objective of this step is to provide a more thorough description of the face. Specifically, the network will output the positions of five face landmarks. (O-Net)

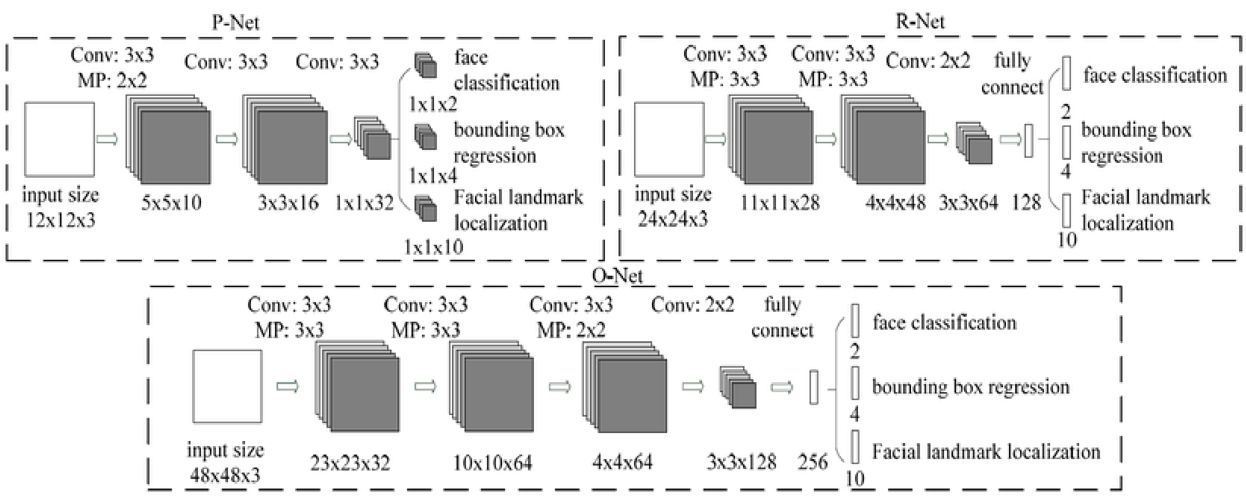


Figure 12: The Architecture of P-Net, R-Net and O-Net

This technique proficiently detects and recognizes faces in both photos and videos, regardless of the faces' diverse sizes or varied orientations. The operation consists of three phases: first, the system identifies probable faces; then, it improves the accuracy of the face's location by using a bounding box; finally, it precisely identifies key facial landmarks such as the eyes and nostrils. This approach offers accurate location information for recognized faces and allows for exceptionally fast processing, making it ideal for real-time applications. The instrument's versatility makes it popular for various facial analysis tasks, such as figure identification, emotion detection, and even estimating a person's age.

The timer is initiated at the start of the video stream and ceases when 15 faces have been caught. Acquiring several images of a person's face improves the precision of the facial recognition system by introducing differences in lighting, perspective, stance, and facial emotions. After doing several test runs, it was determined that 15 frames are the best quantity for constructing a face database. This number strikes a fair compromise between capturing enough variance and avoiding an excessive amount of data that could load the system.

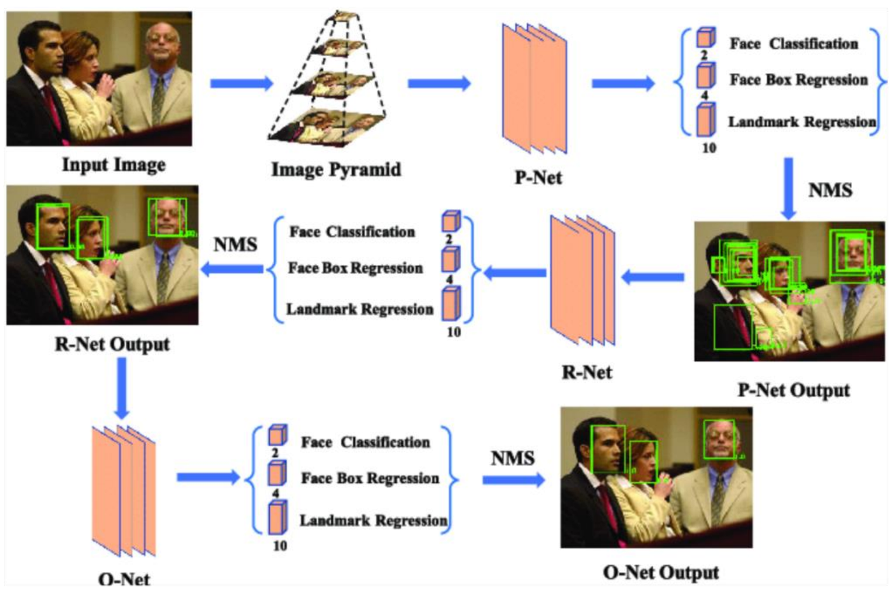


Figure 13: Breakdown of MTCNN model for Smart Attendance System

1.19.FACIAL FEATURE EXTRACTION

FaceNet, a sophisticated deep learning network, was utilised to extract and recognise faces. The system receives a picture of a person's face as input and produces a vector of 128 numerical values that reflect the facial features. The model was trained using the Tensorflow framework on a substantial dataset of facial photos specifically for the purpose of face recognition. The FaceNet model and its weights are first loaded and then compiled using the "compile" method.

Once 20 face photos of the user have been collected, the function "switch_to_main()" stops the video stream, creates a new instance of "EncodingThread" with a batch size of 32, and initiates the thread. The batch size is used in the "encode" method to iterate through all the collected images in batches, rather than processing all the images simultaneously. By utilizing batching, the framework can enhance the computation graph to optimize performance for the particular batch size employed. This can improve memory efficiency and enable speedier processing.

Face encodings will be derived from a collection of training photos that were previously saved during the face detection process. Lists will be established to hold the encodings and their corresponding names. Subsequently, the normalization of the pixel value of the image will be performed. The photographs will undergo resizing to a resolution of 160 × 160 pixels, and their dimensions will be adjusted to fit the model. The loaded FaceNet model will be used to make predictions on a batch of photos. The resulting encodings and names will then be added to the respective lists. Finally, the encoded data will be stored in a pickle file.

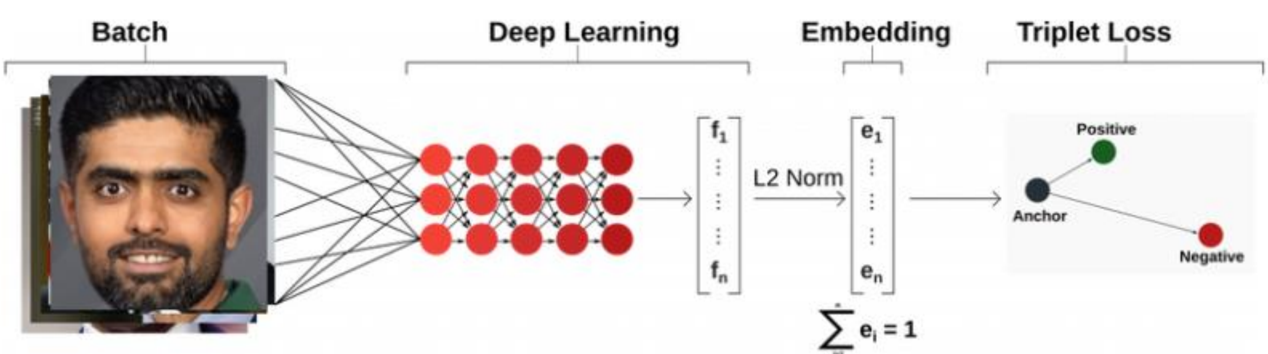


Figure 14: Breakdown of FACENET model for Smart Attendance System

In summary, this technology generates distinct codes for each face, enabling comparison to ascertain similarity. The method involved training a sophisticated neural network using sets of three pictures: a reference face (anchor), a comparable face (positive), and a non-comparable face (negative). Through the analysis of these triplets, the network acquires the ability to encode faces into a high-dimensional space, wherein comparable faces are positioned in close proximity to one another. The Thirss technique provides high levels of accuracy and robustness in tasks such as authenticating someone's identification or identifying them in a crowded environment. Since each face has a distinct embedding, comparing faces becomes efficient by calculating the distance between their embeddings using methods such as Euclidean or cosine distance.

The primary element of FaceNet is the process of end-to-end learning. The triplet loss is employed for the purposes of face verification, identification, and clustering. FaceNet aims to produce an embedding f(x) from an image x into a feature space Rd. The objective is to ensure that the squared distance between all faces of the same identity is small, independent of imaging conditions. On the other hand, the squared distance between two face images from different identities should be large.

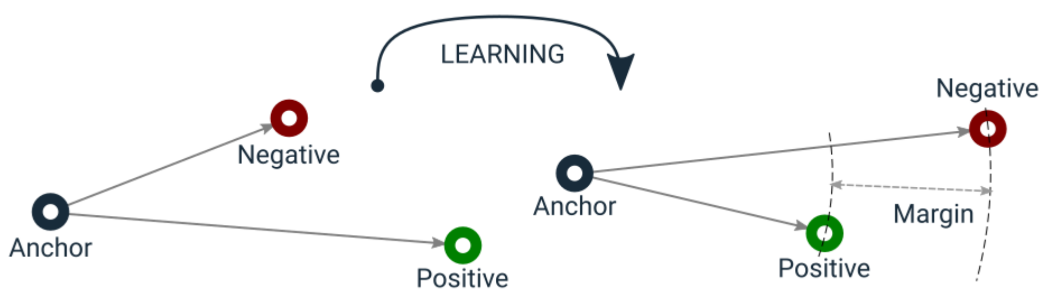


Figure 15: The Triplet Loss

Triplet loss, in contrast, seeks to establish a separation between each pair of a person's faces and all other faces. This enables the coexistence of identity faces on the manifold while preserving a separation, so differentiating them from other identities.

To summarize, triplet loss is a technique that aims to minimize the distance between anchors and positive samples that share the same identity, while simultaneously maximizing the distance between anchors and negative samples that have distinct identities.

1.20.FACE CLASSIFICATION

We have used Cosine Similarity for the classification of the images. Cosine similarity quantifies the similarity between two vectors in an inner product space. The cosine of the angle between two vectors is used to measure their alignment and determine if they are basically pointing in the same direction. It is frequently employed to quantify the similarity between documents in text analysis.

A document can be characterized by numerous characteristics, with each attribute capturing the occurrence rate of a certain word (such as a keyword) or phrase in the document. Each document is represented by a term-frequency vector, which is an object.

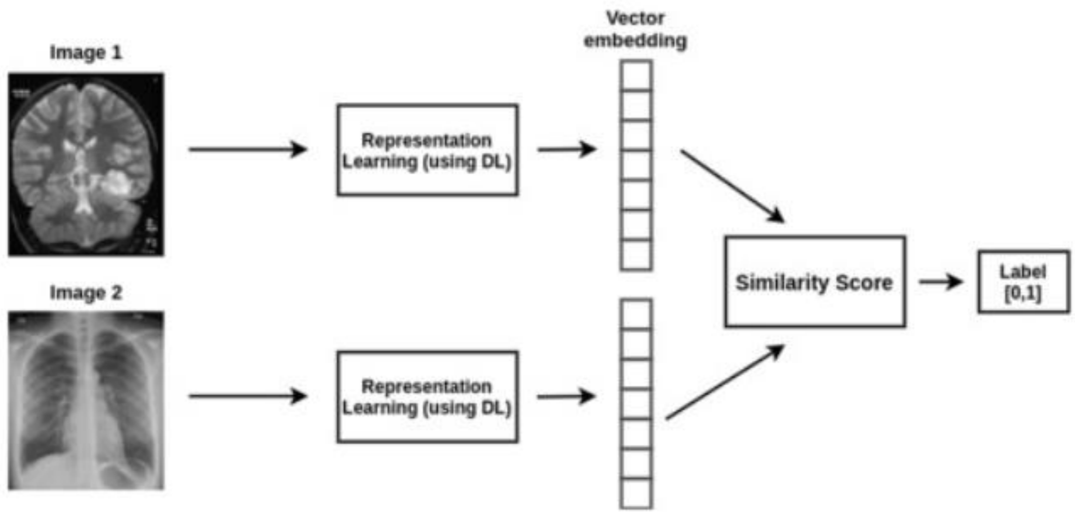


Figure 16: Breakdown of Cosine Similarity Classifier in Deep Learning

Similarity learning focuses on quantifying the similarity between two images or objects. This field has practical applications in tasks like classification and regression. The objective is to acquire knowledge of the similarity function that identifies the most favorable correlation between two comparable or analogous entities in a quantitative manner. Handwritten text recognition, face identification, search engines, and signature verification are among the various applications that utilize similarity measures. Generally, similarity learning entails providing a pair of images as input and determining their level of resemblance. The output is a similarity score ranging from 0 to 1, where 1 indicates total resemblance between the two photos and 0 indicates no similarity.

1.21. ANTI-SPOOFING MECHANISM

The smart attendance system we have implemented includes anti-spoofing coding to prevent unauthorized access and guarantee the accuracy of the data. This code detects and removes any efforts to record attendance using photographs or pre-recorded films, hence reducing inaccurate entries. By implementing this measure, the system enhances overall security by mitigating the risk of attendance fraud and identity theft. The emphasis on strong security measures enhances user confidence and instills trust in the system's precision. Moreover, the inclusion of this anti-spoofing feature ensures compliance with regulatory standards by protecting confidential attendance information.

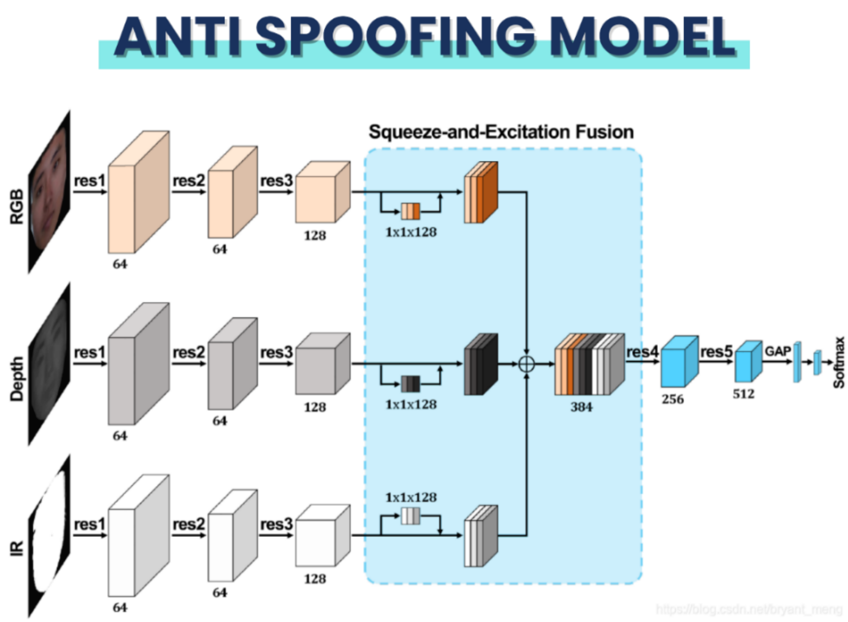


Figure 17: Breakdown of the Anti-Spoofing Model implemented in the Smart Attendance System

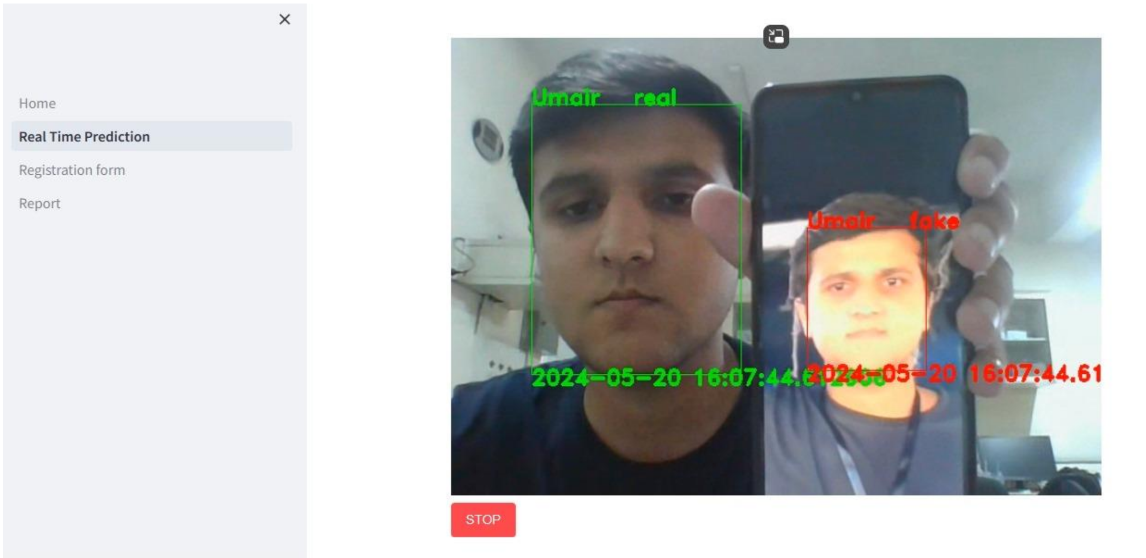
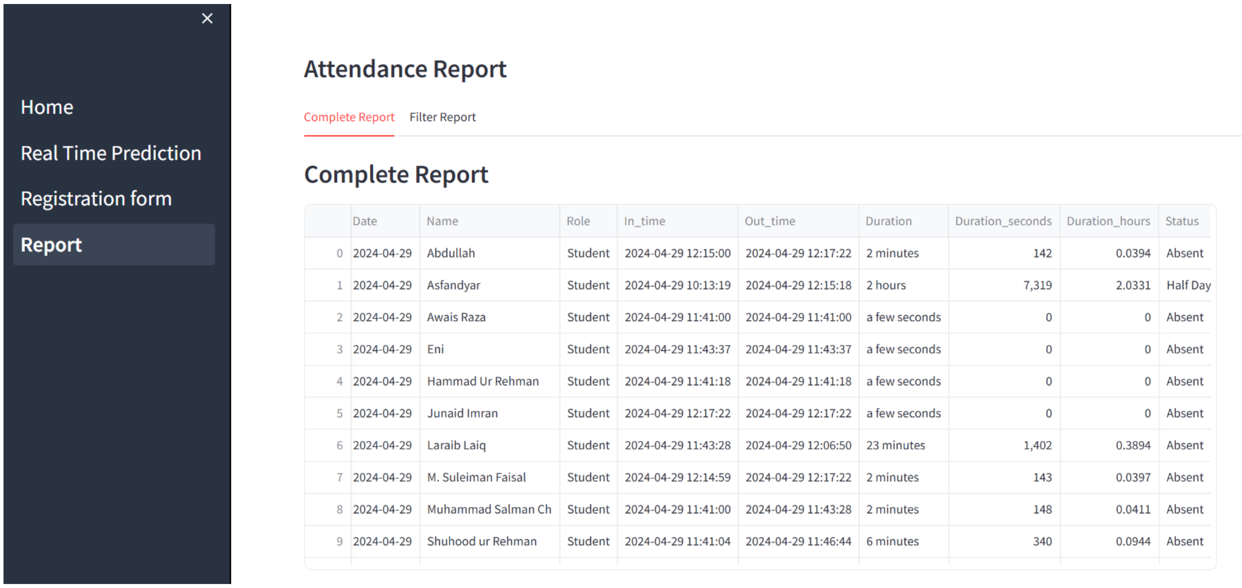


Figure 18: Anti Spoofing model in action.

1.22. ATTENDANCE UPDATE

The system tracks and analyzes employee attendances through the use of a system for attendance management. It calculates the duration of every employee's presence, the in and out-time for a particular date, and retrieves attendance logs from a Redis database. Employees are categorized as "Missing" (less than one hour), "Almost Day" (between one and four hours), or "Exist" (more than 8 hours) by the system depending on this length. Without user involvement needed, the system precisely collects attendance, effectively manages it, and gives insights into employee work behaviors. Then, it will be achievable to get the report as an Excel file, which includes the right name, in and out times, duration, attendance status, and last decision.



	Date	Name	Role	In_time	Out_time	Duration	Duration_seconds	Duration_hours	Status
0	2024-04-29	Abdullah	Student	2024-04-29 12:15:00	2024-04-29 12:17:22	2 minutes	142	0.0394	Absent
1	2024-04-29	Asfandyar	Student	2024-04-29 10:13:19	2024-04-29 12:15:18	2 hours	7,319	2.0331	Half Day
2	2024-04-29	Awais Raza	Student	2024-04-29 11:41:00	2024-04-29 11:41:00	a few seconds	0	0	Absent
3	2024-04-29	Eni	Student	2024-04-29 11:43:37	2024-04-29 11:43:37	a few seconds	0	0	Absent
4	2024-04-29	Hammad Ur Rehman	Student	2024-04-29 11:41:18	2024-04-29 11:41:18	a few seconds	0	0	Absent
5	2024-04-29	Junaid Imran	Student	2024-04-29 12:17:22	2024-04-29 12:17:22	a few seconds	0	0	Absent
6	2024-04-29	Laraib Laiq	Student	2024-04-29 11:43:28	2024-04-29 12:06:50	23 minutes	1,402	0.3894	Absent
7	2024-04-29	M. Suleiman Faisal	Student	2024-04-29 12:14:59	2024-04-29 12:17:22	2 minutes	143	0.0397	Absent
8	2024-04-29	Muhammad Salman Ch	Student	2024-04-29 11:41:00	2024-04-29 11:43:28	2 minutes	148	0.0411	Absent
9	2024-04-29	Shuhood ur Rehman	Student	2024-04-29 11:41:04	2024-04-29 11:46:44	6 minutes	340	0.0944	Absent

Figure 19: Figure showing the Attendance Tracking Report generated on backend.

1.23. USER INTERFACE DESIGN

The project uses Streamlit to develop an intuitive user interface for an attendance control system. It has a gradient background, a log, and a unique font. Header highlights the usage of facial recognition and deep learning technologies. The user can get an attendance report, register, and use facial recognition to record their attendance. Design seeks to be efficient and clear, functioning without the need for human involvement.

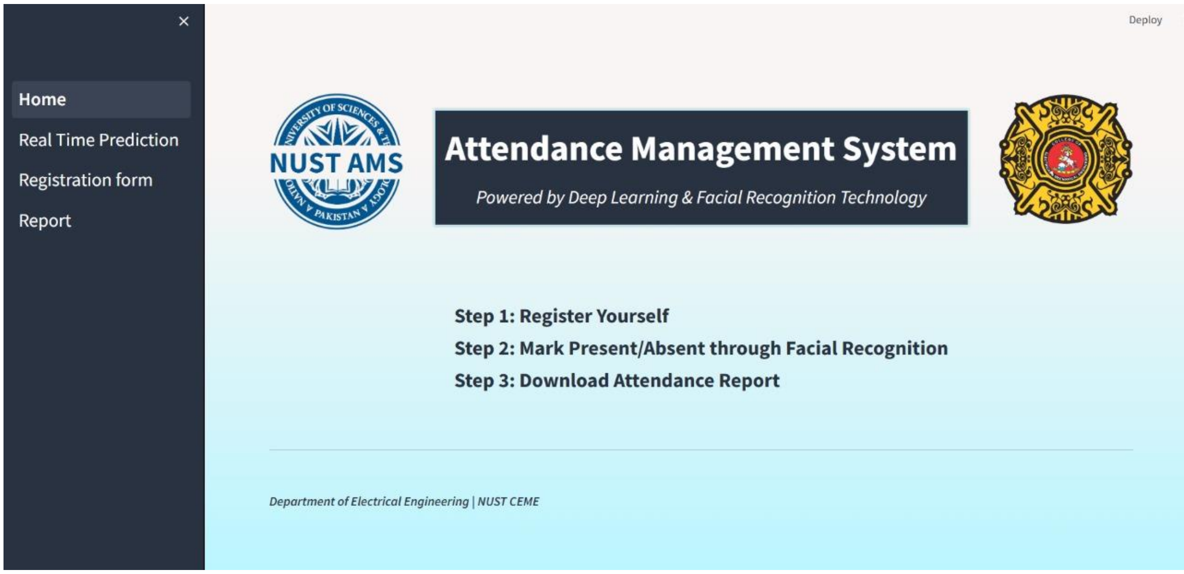


Figure 20:The User Interface of Smart Attendance System is shown.

CHAPTER 4: RESULTS AND DISCUSSIONS

1.24. INTRODUCTION

The performance of the suggested systems' face recognition capabilities was evaluated by testing them on self-collected face datasets. An evaluation was conducted to compare the widely accessible and in-use models in the market in order to determine the most suited one for the attendance system. The face datasets were utilized to optimize the parameters of the face recognition system through a process known as fine-tuning. By testing the recognition system on face datasets, it was possible to evaluate metrics such as accuracy, precision, recall, F1 score, classification report, and confusion matrix. This technique facilitated the identification of any constraints and potentialities in the system's capacity to discern faces for the purpose of attendance records.

1.25. FACE DETECTION EVALUTAION

Facial recognition is a popular computer vision issue that involves identifying and validating human faces in images or films. Several packages for Python offer face recognition functionality. This Python module, called face recognition, wraps around dlib and supports face detection, facial landmark detection, and face recognition. It simplifies the Python face recognition process by providing pre-trained models and straightforward techniques.

The needed libraries, such as dlib, open CV, and face recognition, were first imported by the writers. The database is then used to extract sample photographs, and after that, a video capture window will be launched. If the face captured matches the database, it will indicate that the image has the name visible, and the face may be identified.

Face detection is a crucial component of the entire recognition system since it aids in the identification of face areas inside the video frame, which is necessary for the recognition system to subsequently produce accurate recognition results.

In this situation, well-known face detection methods, such as MTCNN, were tried out to evaluate how well they performed in real-time webcam footage. True positive, false positive, false negative, and true positive are the four possible results. A picture is considered true positive when every face in it is accurately detected, while it is considered true negative when non-facial images are accurately recognized as such. A facial expression is either missing or not recognized in the image or frame. False positives happen when the detection system marks a region or image as having a face when it doesn't.

Making a feature map can significantly improve accuracy in place of the standard method of employing computer vision to capture several user photos. In contrast to a more conventional method, ours solves the issues of illumination, quality, and even the potential for the wearer to become unidentifiable when wearing an item like spectacles.

Figure 21: Smart Attendance System recognizing individual with spectacles Aswell.

True positive, true negative, false positive, and false negative are the four possible outcomes. When every face in an image is accurately recognized, this is called a true positive; when non-facial images are correctly recognized as such, this is called a true negative. A face is missed or not detected in the image or frame. This is known as a false negative. False positives arise when the detection system marks an image or region as having a face when it doesn't.

		Actual Conditions	
		Faces detected	**Faces not detected**
Predicted Conditions	**Predicted faces detected**	True Positive (TP)	False Positive (FP)
	Predicted non-faces detected	False Negative (FN)	True Negative (TN)

Figure 22: Confusion Matrix for the Performance Evaluation of Face Detectors

The accuracy of the results was evaluated with the following formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

where

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

The accuracy and robustness of the system are strengthened by using a compounded model structure with a CNN Structure in the head and MTCNN creating PNet, RNet, and ONet as the base models.

Based on the above implemented strategies, the accuracy of the system comes to be **98%.**

CHAPTER 5 - CONCLUSIONS AND RECOMMENDATIONS

5.1. CONCLUSION

To sum up, the project's goals and objectives—which included creating a facial recognition system especially for tracking attendance—were all successfully met. A thorough analysis of the literature was done, encompassing the most recent developments and outcomes in facial recognition algorithms. Additionally, a user-friendly web application with necessary capabilities was developed to make attendance tracking for end users easier. Lastly, an analysis of the system's usability and performance offered suggestions for future improvements aimed at bolstering the system's resilience.

In order to meet the requirement for automatic classroom evaluation, the Automated Classroom Attendance System helps to increase accuracy and speed ultimately attaining high-precision real-time attendance.

It was discovered from the literature that machine-learning techniques have taken center stage in face identification and will be crucial in advancing future improvements in robustness, accuracy, and speed. All things considered, this research work offers a thorough review of face detection systems and their historical development in addition to a real-time MTCNN identification method. The findings clearly show that, depending on the situation, the suggested approach can achieve recognition accuracy of 96% to 100%. This study also identifies potential research possibilities and provides insights into the advantages and disadvantages of different methodologies. Researchers, practitioners, and anybody else interested in the advancement of facial recognition technology will find the study interesting.

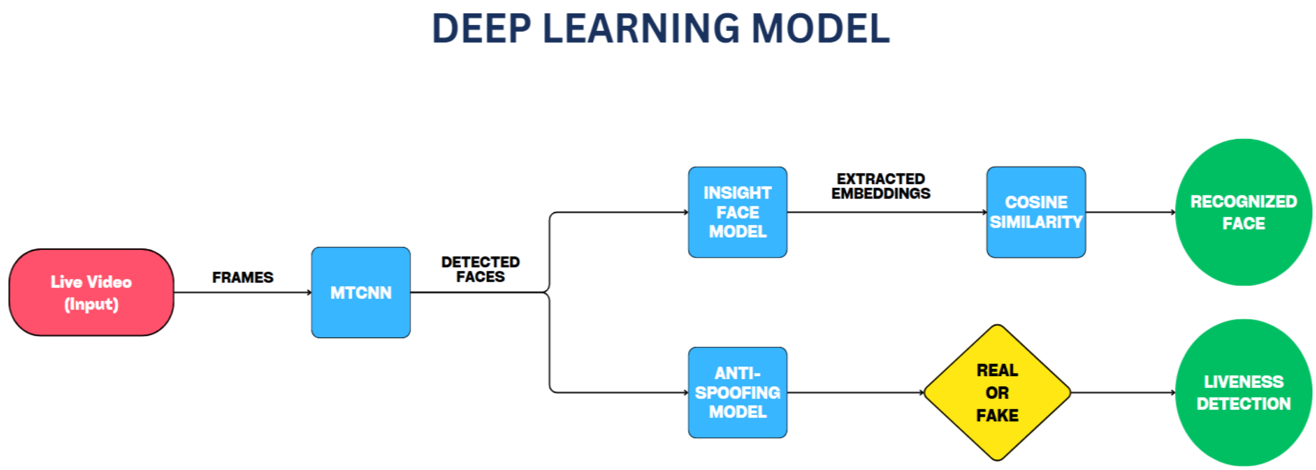


Figure 23: Block diagram of Final Implementable Model

5.2. RECOMMENDATION FOR FUTURE WORK

- While the face recognition system demonstrates a high level of testing accuracy, it is crucial to acknowledge that the dataset utilized for training and testing the system is limited in size. Hence, there is a possibility that the system's performance may not exhibit good generalization to different datasets. In order to overcome this constraint, future research could focus on gathering more varied and extensive training datasets to assess the system's scalability and generalizability. In addition, the face recognition system can be strengthened for future improvements, allowing it to accurately identify faces in many situations, including variable lighting conditions, obstructions, different angles, and faces that are covered with masks.

- The utilization of ensemble learning and adaptive learning approaches can significantly improve the precision and resilience of a face recognition system. Ensemble learning involves the utilization of many models to enhance the system's capacity to learn from the individual strengths and shortcomings of each model. On the other hand, the system can incorporate adaptive learning to guarantee that recognition stays efficient as the user gets older. Moreover, incremental learning can be utilized in deep learning models, allowing them to consistently acquire knowledge from data and adjust to variations in that data as time progresses.

- The present task for the attendance system entails storing the databases of new users into local files. An efficient method to enhance the efficiency and safeguard the data of the system is to transfer the users' databases to a centralized cloud server.

- Examples of cloud computing platforms that support different databases include Amazon Web Services (AWS), which provides support for MySQL and Oracle databases, Microsoft Azure, which supports Microsoft SQL Server, and Firebase, which allows for the storage of real-time data. Each of these cloud server alternatives has robust security measures to restrict database access exclusively to authorized individuals. Moreover, this can streamline data exchange and administration for the supervision of attendance in an organization.

- Retina Identification is a biometric method that identifies an individual based on the distinct patterns on their retina. The tiny nerve on the back of the eyeball that processes light entering via the pupil is captured and analyzed by retina recognition technology. Retinal patterns are very distinguishing characteristics. Even identical twins' eyes differ from one another in terms of their blood vessel patterns, which are unique to each eye. There are efforts underway to build faster, more advanced retina recognition technology.

PROJECT AS A COMPLEX ENGINEERING PROBLEM(CEP)

ABSTRACT:

Any institution or organization that records employee or member attendance must have an attendance management system. Manual attendance recording using paper or files is risky due to data loss, manipulation, and catastrophic occurrences. A smart attendance system using RFID tags, fingerprints, iris scans, voice recognition, and other technologies is required by the current standard. Technology is enabling the use of face recognition for better attendance control. Facial recognition-based attendance systems have several advantages, including removing human interaction and reducing infection risk. This study suggests using face recognition in a real-time attendance tracking system to identify people.

Deep learning-based face recognition algorithms outperform eigenfaces in accuracy and performance. Modern face recognition systems contain detection, alignment, feature extraction, representation, and recognition steps. This research proposes a deep learning facial recognition-based class attendance system. Face detection and alignment use the Multitask Convolutional Neural Network (MTCNN) and a lightweight hybrid high-performance model.

The project completed its aims of establishing a face recognition system for attendance tracking and evaluating facial recognition algorithm methodology and results. The Streamlit online app also includes

database construction, facial recognition, and attendance monitoring. It simplifies end-user attendance tracking. The system's performance and usability were analyzed to suggest opportunities for improvement.

WP1 – DEPTH OF KNOWLEDGE REQUIRED

The implementation of our system involves Python programming, with development carried out in Visual Studio Code. Involving complex deep learning models like MTTNC. We utilize face landmark estimation guidelines to locate and recognize students' faces. By using facial features as bio-metric authentication, we eliminate the possibility of false attendance. The system captures and records attendance information, providing teachers with an efficient and reliable tool for managing student presence. We have integrated two properly functioning Deep Learning Models with extensive libraries and a hardware on raspberry pi for smooth functioning and complex system for attendance.

WP2 – RANGE OF CONFLICTING ENVIRONMENTS

One of the most unique yet incredibly rare feature of our attendance system is the Anti Spoofing model, which detects real faces and distinguishes between real and fake person, so even if a picture of the said person is showed on the mobile phone, the system will pick up on it and generate an alert. This was an incredibly important task for the overall data management and security system of whatever premises the system is used and utilized in.

WP3 – DEPTH OF ANALYSIS

One of the most unique yet incredibly rare feature of our attendance system is the Anti Spoofing model, which detects real faces and distinguishes between real and fake person, so even if a picture of the said person is showed on the mobile phone, the system will pick up on it and generate an alert. This was an incredibly important task for the overall data management and security system of whatever premises the system is used and utilized in.

REFERANCES

[1] Essien, U., & Ansa, G. (2023). *Attendance Based Systems for Face Recognition using a Camera Rail Approach.* Researchers Journal of Science and Technology, 3(1), https://www.rejost.com.ng/index.php/home/article/view/45

[2] Meynet, J., 2003. *Fast face detection using adaboost*. University of Trier, [online]

[3] Wahid, T., 2013. *Face Recognition using Local Binary Patterns (LBP) Face Recognition using Local Binary Patterns LBP*. Journal of Computer Science and Technology Graphics & Vision, [online]].

[4] Abdalkarim, B.A. and Akgün, D., 2022. *A Literature Review on Smart Attendance Systems. International Conference on Applied Engineering and Natural Sciences A Literature Review on Smart Attendance Systems*.

[5] Qi, D., Tan, W., Yao, Q. and Liu, J., 2021. YOLO5Face: *Why Reinventing a Face Detector*. [online] Available at: [Accessed 3 April 2023].

[6] Wahid, T., 2013. *Face Recognition using Local Binary Patterns (LBP) Face Recognition using Local Binary Patterns LBP.* Journal of Computer Science and Technology Graphics & Vision, [online] Available at: [Accessed 3 April 2023].

[7] Abdalkarim, B.A. and Akgün, D., 2022. *A Literature Review on Smart Attendance Systems. International Conference on Applied Engineering and Natural Sciences A Literature Review on Smart Attendance Systems*. [online] Available at: [Accessed 20 April 2023].

[8] Arora, M., Naithani, S. and Areeckal, A.S., 2022. *A web-based application for face detection in real-time images and videos*. Journal of Physics: Conference Series, [e-journal] 6596/2161/1/012071. 2161(1), pp2-9. https://doi.org/10.1088/1742

[9] Gwyn, T., Roy, K. and Atay, M., 2021. *Face recognition using popular deep net architectures*: A brief comparative study. MDPI, [e-journal] 13(7), pp.1 15. https://doi.org/10.3390/fi13070164.

[10] Lakshmi, D. and Ponnusamy, R., 2021. *Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders*. Association for Computing Machinery, [e-journal] https://doi.org/10.1016/j.micpro.2021.103834. 82, p.103834.

[11] Singhal, N., Ganganwar, V., Yadav, M., Chauhan, A., Jakhar, M. and Sharma, K., 2021. *Comparative study of machine learning and deep learning algorithm for face recognition. Jordanian Journal of Computers and Information Technology,* [e-journal] 7(3), pp.313–325. https://doi.org/10.5455/JJCIT.71 1624859356.

[12] Ghosh, D. (2021). *Real- Time Attendance System Using Face Recognition Technique.* International Journal Of Engineering Applied Sciences And Technology, 5(9). doi: 10.33564/ijeast. 2021.v05i09.043

[13] Serengil, S. I., & Ozpinar, A. (2021). HyperExtended LightFace*: A Facial Attribute Analysis Framework*. 2021 International Conference on Engineering doi:10.1109/ICEET53442.2021.9659697 and Emerging Technologies (ICEET), 1–4.

[14] Gomes, C., Chanchal, S., Desai, T. & Jadhav, D. (2020). *Class Attendance Management System using Facial Recognition*. In ITM Web of Conferences (Vol. 32, p. 02001). EDP Sciences.

[15] Bharat Chandra, Y. and Karthikeya Reddy, G., 2020. *A Comparative Analysis Of Face Recognition Models On Masked Faces*. International Journal of Scientific & Technology Research

[16] Student *Attendance System using Face Recognition*: Samridhi Dev, Tushar Patnaikb(2020).

[17] Duan Q, Zhang L (2020) *Look more into occlusion: realistic face frontalization and recognition with boostgan*. IEEE Trans Neural Netw Learn Syst 32(1):214–228

[18] Madzou, L., 2020. *Facial recognition technology and travel after COVID-19* | World Economic Forum.

[19] Ku, H., & Dong, W. (2020). *Face Recognition Based on MTCNN and Convolutional Neural Network. Frontiers in Signal Processing*, 4(1), 37-42.

[20] Manju, D. & Radha, V. (2020). *A novel approach for pose invariant face recognition in surveillance videos*. Procedia Computer Science, 167, 890-899.

[21] Satpute, R., Sontakke, S., Gondaliya, B., Sonawane, T. & Suryawanshi, K. (2020). *Attendance Management System Using Face Recognition. biometrics*, 7(05).

[22] Amirulloh, I., Iskandar, I.D., Apriyani, Y., Warnilah, A.I., Purnia, D.S. and Surahman, M., 2020. *Teacher Attendance Monitoring System Teaching with QR-Code and Geo Location using Android Platform.* Journal of Physics: Conference Series, [e-journal] 1641(1), pp.1-2. https://doi.org/10.1088/1742 6596/1641/1/012030.

[23] Jayaswal, R. and Dixit, M., 2020. *Comparative analysis of human face recognition by traditional methods and deep learning in real-time environment.* Proceedings - 2020 IEEE 9th International Conference on Communication Systems and Network Technologies, CSNT 2020, [e-journal] pp.66–71. https://doi.org/10.1109/CSNT48778.2020.9115779.

[24] Sepas-Moghaddam, A., Pereira, F.M. and Correia, P.L., 2020. *Face recognition: A novel multi-level taxonomy based survey*. IET Biometrics, [e journal] 9(2), pp.58–67. https://doi.org/10.1049/iet-bmt.2019.0001.

[25] William, I., Rachmawanto, E. H., Santoso, H. A., & Sari, C. A. (2019, October). *Face recognition using facenet (survey, performance test, and comparison*). In 2019 fourth international conference on informatics and computing (ICIC) (pp. 1-6).

[26] Nyein, T. and Oo, A.N., 2019. University *Classroom Attendance System Using FaceNet and Support Vector Machine*. 2019 International Conference on Advanced Information Technologies, ICAIT 2019, [e-journal] pp.171–176. https://doi.org/10.1109/AITC.2019.8921316.

[27] Bhatti, K. L., Mughal, L., Khuhawar, F. Y. & Memon, S. A. (2018). *Smart attendance management system using face recognition*. EAI Endorsed Transactions on Creative Technologies, 5(17).

[28] Jogin, M., Madhulika, M. S., Divya, G. D., Meghana, R. K., & Apoorva, S. (2018, May). F*eature extraction using convolution neural networks (CNN) and deep learning.* In 2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT) (pp. 2319-2323).

[29] Trigueros, D. S., Meng, L., & Hartnett, M. (2018). Face recognition: *From traditional to deep learning methods.* arXiv preprint arXiv:1811.00116.

[30] Arsenovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2017, September). *FaceTime—Deep learning-based face recognition attendance system*. In 2017 IEEE 15th International symposium on intelligent systems and informatics (SISY) (pp. 000053-000058).

APPENDICES

Appendix A – Code of Real Time Prediction

Appendix B – Code of Registration Form

Appendix C – Code of Attendance Report

Appendix D – Code of Face Recording

Appendix E – Code of Homepage

Appendix F – Attendance Generation & UI code

APPENDIX A

```
import streamlit as st
from Home import face_rec
from streamlit_webrtc import webrtc_streamer
import av
import time

# st.set_page_config(page_title='Predictions')
st.subheader('Real-Time Attendance System')


# Retrive the data from Redis Database
with st.spinner('Retriving Data from Redis DB ...'):
    redis_face_db = face_rec.retrive_data(name='academy:register')
    st.dataframe(redis_face_db)

st.success("Data sucessfully retrived from Redis")

# time
waitTime = 30 # time in sec
setTime = time.time()
realtimepred = face_rec.RealTimePred() # real time prediction class

# Real Time Prediction
# streamlit webrtc
# callback function
def video_frame_callback(frame):
    global setTime

    img = frame.to_ndarray(format="bgr24") # 3 dimension numpy array
    # operation that you can perform on the array
    pred_img = realtimepred.face_prediction(img, redis_face_db,
'facial_features',  name_role=['Name', 'Role'], thresh=0.5)



    timenow = time.time()
    difftime = timenow - setTime
    if difftime >= waitTime:
        realtimepred.saveLogs_redis()
        setTime = time.time() # reset time
        print('Save Data to redis database')
```

```
    return av.VideoFrame.from_ndarray(pred_img, format="bgr24")



webrtc_streamer(key="realtimePrediction", video_frame_callback=video_frame_callback,
rtc_configuration={
        "iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}]
    }
)
```

APPENDIX B

```python
import streamlit as st
from Home import face_rec
import cv2
import numpy as np
from streamlit_webrtc import webrtc_streamer
import av


# st.set_page_config(page_title='Registration Form')
st.subheader('Registration Form')

## init registration form
registration_form = face_rec.RegistrationForm()

# Step-1: Collect person name and role
# form
person_name = st.text_input(label='Name',placeholder='First & Last Name')
role = st.selectbox(label='Select your Role',options=('Student',
                                                      'Teacher'))


# step-2: Collect facial embedding of that person
def video_callback_func(frame):
    img = frame.to_ndarray(format='bgr24') # 3d array bgr
    reg_img, embedding = registration_form.get_embedding(img)
    # two step process
    # 1st step save data into local computer txt
    if embedding is not None:
        with open('face_embedding.txt',mode='ab') as f:
            np.savetxt(f,embedding)

    return av.VideoFrame.from_ndarray(reg_img,format='bgr24')

webrtc_streamer(key='registration',video_frame_callback=video_callback_func,
rtc_configuration={
        "iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}]
    }
)


# step-3: save the data in redis database


if st.button('Submit'):
```

```
    return_val = registration_form.save_data_in_redis_db(person_name,role)
    if return_val == True:
        st.success(f"{person_name} registered sucessfully")
    elif return_val == 'name_false':
        st.error('Please enter the name: Name cannot be empty or spaces')

    elif return_val == 'file_false':
        st.error('face_embedding.txt is not found. Please refresh the page and execute
again.')
```

APPENDIX C

```
import streamlit as st
from Home import face_rec
st.set_page_config(page_title='Reporting',layout='wide')
st.subheader('Reporting')


# Retrive logs data and show in Report.py
# extract data from redis list
name = 'attendance:logs'
def load_logs(name,end=-1):
    logs_list = face_rec.r.lrange(name,start=0,end=end) # extract all data from the redis 
database
    return logs_list

# tabs to show the info
tab1, tab2 = st.tabs(['Registered Data','Logs'])

with tab1:
    if st.button('Refresh Data'):
        # Retrive the data from Redis Database
        with st.spinner('Retriving Data from Redis DB ...'):
            redis_face_db = face_rec.retrive_data(name='academy:register')
            st.dataframe(redis_face_db[['Name','Role']])

with tab2:
    if st.button('Refresh Logs'):
        st.write(load_logs(name=name))
```

APPENDIX D

```
import numpy as np
import pandas as pd
import cv2
import tensorflow as tf
import numpy as np
import pickle
import os
import redis

# insight face
from insightface.app import FaceAnalysis
from sklearn.metrics import pairwise
# time
import time
from datetime import datetime

import os




model_path='liveness.model'
le_path='label_encoder.pickle'
encodings='encoded_faces.pickle'
confidence=0.5
args = {'model':model_path, 'le':le_path,
    'encodings':encodings, 'confidence':confidence}

# load the encoded faces and names
print('[INFO] loading encodings...')
with open(args['encodings'], 'rb') as file:
    encoded_data = pickle.loads(file.read())

# load the liveness detector model and label encoder from disk
liveness_model = tf.keras.models.load_model(args['model'])
le = pickle.loads(open(args['le'], 'rb').read())


# Connect to Redis Client
hostname = 'redis-11807.c261.us-east-1-4.ec2.redns.redis-cloud.com'
portnumber = '11807'
password = 'uqO4vRAK2UvaTpOAtpnc1LfI1wRdoVgC'
```

```
r = redis.StrictRedis(host=hostname,
                      port=portnumber,
                      password=password)

# Retrive Data from database
def retrive_data(name):
    retrive_dict= r.hgetall(name)
    retrive_series = pd.Series(retrive_dict)
    retrive_series = retrive_series.apply(lambda x: np.frombuffer(x,dtype=np.float32))
    index = retrive_series.index
    index = list(map(lambda x: x.decode(), index))
    retrive_series.index = index
    retrive_df =  retrive_series.to_frame().reset_index()
    retrive_df.columns = ['name_role','facial_features']
    retrive_df[['Name','Role']] = retrive_df['name_role'].apply(lambda x:
x.split('@')).apply(pd.Series)
    return retrive_df[['Name','Role','facial_features']]


# configure face analysis
faceapp = FaceAnalysis(name='buffalo_l',root='insightface_model', providers =
['CPUExecutionProvider'])
faceapp.prepare(ctx_id = 0, det_size=(640,640), det_thresh = 0.5)

# ML Search Algorithm
def ml_search_algorithm(dataframe,feature_column,test_vector,
                        name_role=['Name','Role'],thresh=0.5):
    """
    cosine similarity base search algorithm
    """
    # step-1: take the dataframe (collection of data)
    dataframe = dataframe.copy()
    # step-2: Index face embeding from the dataframe and convert into array
    X_list = dataframe[feature_column].tolist()
    x = np.asarray(X_list)

    # step-3: Cal. cosine similarity
    similar = pairwise.cosine_similarity(x,test_vector.reshape(1,-1))
    similar_arr = np.array(similar).flatten()
    dataframe['cosine'] = similar_arr

    # step-4: filter the data
    data_filter = dataframe.query(f'cosine >= {thresh}')
    if len(data_filter) > 0:
        # step-5: get the person name
```

```python
        data_filter.reset_index(drop=True,inplace=True)
        argmax = data_filter['cosine'].argmax()
        person_name, person_role = data_filter.loc[argmax][name_role]

    else:
        person_name = 'Unknown'
        person_role = 'Unknown'

    return person_name, person_role



### Real Time Prediction
# we need to save logs for every 1 mins
class RealTimePred:
    def __init__(self):
        self.logs = dict(name=[],role=[],current_time=[])

    def reset_dict(self):
        self.logs = dict(name=[],role=[],current_time=[])

    def saveLogs_redis(self):
        # step-1: create a logs dataframe
        dataframe = pd.DataFrame(self.logs)
        # step-2: drop the duplicate information (distinct name)
        dataframe.drop_duplicates('name',inplace=True)
        # step-3: push data to redis database (list)
        # encode the data
        name_list = dataframe['name'].tolist()
        role_list = dataframe['role'].tolist()
        ctime_list = dataframe['current_time'].tolist()
        encoded_data = []
        for name, role, ctime in zip(name_list, role_list, ctime_list):
            if name != 'Unknown':
                concat_string = f"{name}@{role}@{ctime}"
                encoded_data.append(concat_string)

        if len(encoded_data) >0:
            r.lpush('attendance:logs',*encoded_data)


        self.reset_dict()


    def face_prediction(self,test_image, dataframe,feature_column,
```

```
                              name_role=['Name','Role'],thresh=0.5):
        # step-1: find the time
        current_time = str(datetime.now())


        # step-1: take the test image and apply to insight face
        results = faceapp.get(test_image)
        test_copy = test_image.copy()
        # step-2: use for loop and extract each embedding and pass to ml_search_algorithm


        for res in results:
            x1, y1, x2, y2 = res['bbox'].astype(int)
            embeddings = res['embedding']
            face = test_image[y1-20:y2+20, x1-20:x2+20]
            face = cv2.resize(face, (32, 32))
            face = face.astype('float') / 255.0
            face = tf.keras.preprocessing.image.img_to_array(face)
            face = np.expand_dims(face, axis=0)
            preds = liveness_model.predict(face)[0]
            j = np.argmax(preds)
            label_name = le.classes_[j]

            label = f'{label_name}: {preds[j]:.4f}'
            print(f'[INFO] , {label_name}')

            person_name, person_role = ml_search_algorithm(dataframe,
                                                            feature_column,
                                                            test_vector=embeddings,
                                                            name_role=name_role,
                                                            thresh=thresh)
            if person_name == 'Unknown' or label_name == "fake":
                color =(0,0,255) # bgr
            else:
                color = (0,255,0)

            cv2.rectangle(test_copy,(x1,y1),(x2,y2),color)

            text_gen = person_name
            cv2.putText(test_copy,text_gen,(x1,y1),cv2.FONT_HERSHEY_DUPLEX,0.7,color,2)
            cv2.putText(test_copy,label_name,(x1
+100,y1),cv2.FONT_HERSHEY_DUPLEX,0.7,color,2)
            cv2.putText(test_copy,current_time,(x1,y2+10),cv2.FONT_HERSHEY_DUPLEX,0.7,color,2
)
            # save info in logs dict
```

```
            self.logs['name'].append(person_name)
            self.logs['role'].append(person_role)
            self.logs['current_time'].append(current_time)


        return test_copy

#### Registration Form
class RegistrationForm:
    def __init__(self):
        self.sample = 0
    def reset(self):
        self.sample = 0

    def get_embedding(self,frame):
        # get results from insightface model
        results = faceapp.get(frame,max_num=1)
        embeddings = None
        for res in results:
            self.sample += 1
            x1, y1, x2, y2 = res['bbox'].astype(int)
            cv2.rectangle(frame, (x1,y1),(x2,y2),(0,255,0),1)
            # put text samples info
            text = f"samples = {self.sample}"
            cv2.putText(frame,text,(x1,y1),cv2.FONT_HERSHEY_DUPLEX,0.6,(255,255,0),2)

            # facial features
            embeddings = res['embedding']

        return frame, embeddings

    def save_data_in_redis_db(self,name,role):
        # validation name
        if name is not None:
            if name.strip() != '':
                key = f'{name}@{role}'
            else:
                return 'name_false'
        else:
            return 'name_false'

        # if face_embedding.txt exists
        if 'face_embedding.txt' not in os.listdir():
            return 'file_false'
```

```
        # step-1: load "face_embedding.txt"
        x_array = np.loadtxt('face_embedding.txt',dtype=np.float32) # flatten
array

        # step-2: convert into array (proper shape)
        received_samples = int(x_array.size/512)
        x_array = x_array.reshape(received_samples,512)
        x_array = np.asarray(x_array)

        # step-3: cal. mean embeddings
        x_mean = x_array.mean(axis=0)
        x_mean = x_mean.astype(np.float32)
        x_mean_bytes = x_mean.tobytes()

        # step-4: save this into redis database
        # redis hashes
        r.hset(name='academy:register',key=key,value=x_mean_bytes)

        #
        os.remove('face_embedding.txt')
        self.reset()

        return True
```

APPENDIX E

```
import streamlit as st
st.set_page_config(page_title='Attendance System',layout='wide')


with st.spinner("Loading Models and Conneting to Redis db ..."):
    import face_rec


#SETTING BACKGROUND USING CSS
st.markdown('<style>header.st-emotion-cache-18ni7ap.ezrtsby2{ background: rgb(248,245,244);
}</style>',unsafe_allow_html=True)
st.markdown('<style>section.main.st-emotion-cache-bm2z3a.ea3mdgi8{ background: linear-
gradient(0deg, rgba(187,246,255,1) 0%, rgba(248,245,244,1)
72%);}</style>',unsafe_allow_html=True)

#CSS FONT
st.markdown('<style> body { @import
url("https://fonts.googleapis.com/css2?family=Cabin:ital,wght@0,400..700;1,400..700&family=Po
ppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400
;1,500;1,600;1,700;1,800;1,900&family=Radio+Canada+Big:ital,wght@0,400..700;1,400..700&displa
y=swap");</style>', unsafe_allow_html=True)
#CSS FOR NAVBAR
st.markdown('<style>section.st-emotion-cache-1gv3huu.eczjsme11{ border-right: 2px solid
black; background-color:#293241;}</style>',unsafe_allow_html=True)
st.markdown('<style>span.st-emotion-cache-pkbazv.eczjsme5, span.st-emotion-cache-
17lntkn.eczjsme5,span.st-emotion-cache-17lntkn.eczjsme5, span.st-emotion-cache-
17lntkn.eczjsme5{ color: #ffffff; font-size: 23px;  }</style>',unsafe_allow_html=True)
st.markdown('<style>svg.eyeqlp51.st-emotion-cache-1pbsqtx.ex0cdmw0{ color: #ffffff;
}</style>',unsafe_allow_html=True)
#CSS for HEADER
st.markdown('<style>.header { padding-left:10px; padding-right:10px;margin:20px;box-shadow: 0
0 3px rgba(0, 0, 0, 0.3); border: 2px solid #bbf6ff; text-align: center;  background-color:
#293241; color: #ffffff; font-size:smaller; }</style>', unsafe_allow_html=True)
st.markdown('<style>.header h1 {  text-align: center; color: #ffffff; }</style>',
unsafe_allow_html=True)
st.markdown('<style>.header p{font-style: italic; color: #ffffff; font-size:23px; }</style>',
unsafe_allow_html=True)
st.markdown('<style>.steps {  display: flex;justify-content: center; align-items: center;
text-align: center; }</style>', unsafe_allow_html=True)
st.markdown('<style>.steps p {text-align: start; margin-top: 70px;font-weight: bold; color:
#293241;  font-size:25px; }</style>', unsafe_allow_html=True)
st.markdown('<style>div.st-emotion-cache-vdokb0.e1nzilvr5 { font-weight:bold; font-
size:25px;}</style>', unsafe_allow_html=True)
```

```
l_col,c_col, r_col= st.columns([1,4,1])
with l_col:
    st.image('logo1.png',width=200, use_column_width=True)

with c_col:
    st.markdown('<div class="header"><h1>Attendance Management System</h1><p>Powered by Deep
Learning & Facial Recognition Technology</p></div>', unsafe_allow_html=True)
with r_col:
    st.image("logo2.png",width=230, use_column_width=True)

st.markdown('<div class="steps"><p>Step 1: Register Yourself <br> Step 2: Mark Present/Absent
through Facial Recognition <br>Step 3: Download Attendance Report</p></div>',
unsafe_allow_html=True)
st.markdown("---")
st.write("***Department of Electrical Engineering | NUST CEME***")
```

APPENDIX F

```
name = 'attendance:logs'
def load_logs(name,end=-1):
    logs_list = face_rec.r.lrange(name,start=0,end=end) # extract all data from the redis database
    return logs_list

# tabs to show the info
tab1, tab2, tab3 = st.tabs(['Registered Data','Logs','Attendance Report'])

with tab1:
    if st.button('Refresh Data'):
        # Retrive the data from Redis Database
        with st.spinner('Retriving Data from Redis DB ...'):
            redis_face_db = face_rec.retrive_data(name='academy:register')
            st.dataframe(redis_face_db[['Name','Role']])

with tab2:
    if st.button('Refresh Logs'):
        st.write(load_logs(name=name))


with tab3:
    st.subheader('Attendance Report')

    # load logs into attribute logs_list
    logs_list = load_logs(name=name)

    # step -1: convert the logs that in list of bytes into list of string
    convert_byte_to_string = lambda x: x.decode('utf-8')
    logs_list_string = list(map(convert_byte_to_string, logs_list))

    # step -2: split string by @ and create nested list
    split_string = lambda x: x.split('@')
    logs_nested_list = list(map(split_string, logs_list_string))
    # convert nested list info into dataframe

    logs_df = pd.DataFrame(logs_nested_list, columns= ['Name','Role','Timestamp'])

    # Step -3 Time based Analysis or Report
    #logs_df['Timestamp'] = pd.to_datetime(logs_df['Timestamp'])
    logs_df['Timestamp'] = logs_df['Timestamp'].apply(lambda x: x.split('.')[0])
    logs_df['Timestamp'] = pd.to_datetime(logs_df['Timestamp'])
    logs_df['Date'] = logs_df['Timestamp'].dt.date
```

```
    # step -3.1 : Cal. Intime and Outtime
    # In time: At which person is first detected in that day (min Timestamp of the date)
    # Out time: At which person is last detected in that day (Max Timestamp of the date)

    report_df = logs_df.groupby(by=['Date','Name','Role']).agg(
        In_time = pd.NamedAgg('Timestamp','min'), # in time
        Out_time = pd.NamedAgg('Timestamp','max') # out time
    ).reset_index()

    report_df['In_time']  = pd.to_datetime(report_df['In_time'])
    report_df['Out_time']  = pd.to_datetime(report_df['Out_time'])

    report_df['Duration'] = report_df['Out_time'] - report_df['In_time']

    # Step 4: Marking Person is Present or Absent
    all_dates = report_df['Date'].unique()
    name_role = report_df[['Name','Role']].drop_duplicates().values.tolist()

    date_name_rol_zip = []
    for dt in all_dates:
        for name, role in name_role:
            date_name_rol_zip.append([dt, name, role])

    date_name_rol_zip_df = pd.DataFrame(date_name_rol_zip, columns=['Date','Name','Role'])
    # left join with report_df

    date_name_rol_zip_df = pd.merge(date_name_rol_zip_df, report_df,
how='left',on=['Date','Name','Role'])

    # Duration
    # Hours
    date_name_rol_zip_df['Duration_seconds'] = date_name_rol_zip_df['Duration'].dt.seconds
    date_name_rol_zip_df['Duration_hours'] = date_name_rol_zip_df['Duration_seconds'] /
(60*60)

    def status_marker(x):

        if pd.Series(x).isnull().all():
            return 'Absent'

        elif x >= 0 and x < 1:
            return 'Absent'

        elif x >= 1 and x < 4:
```

```
            return 'Half Day (less than 4 hours)'

        elif x >= 4 and x < 6:
            return 'Half Day'

        elif x >= 8:
            return 'Present'

    date_name_rol_zip_df['Status'] =
date_name_rol_zip_df['Duration_hours'].apply(status_marker)

import streamlit as st
st.set_page_config(page_title='Attendance System',layout='wide')


with st.spinner("Loading Models and Conneting to Redis db ..."):
    import face_rec


#Setting background
st.markdown('<style>header.st-emotion-cache-18ni7ap.ezrtsby2{ background: rgb(248,245,244);
}</style>',unsafe_allow_html=True)
st.markdown('<style>section.main.st-emotion-cache-bm2z3a.ea3mdgi8{ background: linear-
gradient(0deg, rgba(187,246,255,1) 0%, rgba(248,245,244,1)
72%);}</style>',unsafe_allow_html=True)

#CSS font
st.markdown('<style> body { @import
url("https://fonts.googleapis.com/css2?family=Cabin:ital,wght@0,400..700;1,400..700&family=Po
ppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400
;1,500;1,600;1,700;1,800;1,900&family=Radio+Canada+Big:ital,wght@0,400..700;1,400..700&displa
y=swap");</style>', unsafe_allow_html=True)
#CSS FOR NAVBAR
st.markdown('<style>section.st-emotion-cache-1gv3huu.eczjsme11{ border-right: 2px solid
black; background-color:#293241;}</style>',unsafe_allow_html=True)
st.markdown('<style>span.st-emotion-cache-pkbazv.eczjsme5, span.st-emotion-cache-
17lntkn.eczjsme5,span.st-emotion-cache-17lntkn.eczjsme5, span.st-emotion-cache-
17lntkn.eczjsme5{ color: #ffffff; font-size: 23px;  }</style>',unsafe_allow_html=True)
st.markdown('<style>svg.eyeqlp51.st-emotion-cache-1pbsqtx.ex0cdmw0{ color: #ffffff;
}</style>',unsafe_allow_html=True)
#CSS for header
st.markdown('<style>.header { padding-left:10px; padding-right:10px;margin:20px;box-shadow: 0
0 3px rgba(0, 0, 0, 0.3); border: 2px solid #bbf6ff; text-align: center;  background-color:
#293241; color: #ffffff; font-size:smaller; }</style>', unsafe_allow_html=True)
```

```
st.markdown('<style>.header h1 {  text-align: center; color: #ffffff; }</style>', unsafe_allow_html=True)
st.markdown('<style>.header p{font-style: italic; color: #ffffff; font-size:23px; }</style>', unsafe_allow_html=True)
st.markdown('<style>.steps {  display: flex;justify-content: center; align-items: center; text-align: center; }</style>', unsafe_allow_html=True)
st.markdown('<style>.steps p {text-align: start; margin-top: 70px;font-weight: bold; color: #293241;  font-size:25px; }</style>', unsafe_allow_html=True)
st.markdown('<style>div.st-emotion-cache-vdokb0.e1nzilvr5 { font-weight:bold; font-size:25px;}</style>', unsafe_allow_html=True)

#-------------------------------------------------------------------------------------------------------------------
l_col,c_col, r_col= st.columns([1,4,1])
with l_col:
    st.image('logo1.png',width=200, use_column_width=True)

with c_col:
    st.markdown('<div class="header"><h1>Attendance Management System</h1><p>Powered by Deep Learning & Facial Recognition Technology</p></div>', unsafe_allow_html=True)
with r_col:
    st.image("logo2.png",width=230, use_column_width=True)

st.markdown('<div class="steps"><p>Step 1: Register Yourself <br> Step 2: Mark Present/Absent through Facial Recognition <br>Step 3: Download Attendance Report</p></div>', unsafe_allow_html=True)
st.markdown("---")
st.write("Department of Electrical Engineering | NUST CEME")
```

MSalman_report

ORIGINALITY REPORT

6%	3%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Sunderland Student Paper	1%
2	Submitted to Jawaharlal Nehru Technological University Student Paper	1%
3	Submitted to UOW Malaysia KDU University College Sdn. Bhd Student Paper	<1%
4	Vagisha Gupta, Shelly Sachdeva, Neha Dohare. "Deep similarity learning for disease prediction", Elsevier BV, 2021 Publication	<1%
5	"Clustering Methods for Big Data Analytics", Springer Science and Business Media LLC, 2019 Publication	<1%
6	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1%