AttendanceEye
A Computer Vision Attendance System

PROJECT SUPERVISOR

Dr. Muhammad Waqas - Assistant Professor

PROJECT CO-SUPERVISOR

Dr. Muhammad Rafi - Associate Professor –Head of Department of AI & DS

PROJECT TEAM

Mohsin Ali Mirza	K20-0353
Waleed Gul	K20-0259
Mehwish Sameer	K20-1895

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

May 2024

Project Supervisor	Dr. Muhammad Rafi
Project Team	Mohsin Ali Mirza K20-0353 Mehwish Sameer K20-1895 Waleed Gul K20-0259
Submission Date	May 14, 2024

Mr. Supervisor Name ______________________________

Supervisor

Mr. Co-Supervisor Name ______________________________

Co-Supervisor

Dr. Zulfiqar Ali Memon ______________________________

Head of Department

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

Acknowledgement

We are profoundly thankful to the Almighty Allah (SWT) for granting us the resilience, insight, and determination needed to bring this project to fruition. Our heartfelt gratitude also goes out to our esteemed supervisor, Dr Muhammad Rafi and Dr Muhammad Waqas, whose indispensable mentorship, steadfast encouragement, and profound expertise in the field of Artificial Intelligence and Computer Vision has profoundly influenced our path. Without the collective guidance and expertise of all those involved, this project would not have achieved the gradual progress and ultimate realization of our vision.

In addition, we extend our sincere thanks to the students who participated in the data collection process. Your contributions have been invaluable to the success of this endeavor, and we are truly grateful for your dedication and hard work.

As the renowned physicist Oppenheimer once said, "Theory will take you only so far," highlighting the essential role that practical efforts have played a vital role in advancing knowledge and innovation.

Abstract

In the pursuit of optimizing attendance management processes, particularly in organizational settings, the integration of advanced technologies has become imperative. This project explores the application of generative artificial intelligence (AI) to streamline attendance tracking systems by addressing challenges associated with manual entry errors and time-consuming procedures.

Our project focuses on harnessing generative AI techniques to enhance face recognition capabilities within attendance systems, specifically accommodating variations in facial features such as Aging. By employing cutting-edge generative AI models, our system dynamically adapts to changes in facial attributes, ensuring accurate identification of individuals without manual intervention.

Through the synthesis of facial images with diverse features, our project aims to develop a robust and efficient attendance management solution. By automating the attendance tracking process and minimizing errors associated with manual entry, our system significantly reduces administrative burden and enhances overall efficiency.

In summary, our project contributes to the optimization of attendance management by leveraging generative AI to create adaptive face recognition systems. By eliminating manual entry errors and reducing administrative overhead, our solution offers a streamlined approach to attendance tracking, ultimately improving efficiency and productivity within organizations.

Contents

List of Figures

List of Tables

Introduction

Attendance management is a fundamental aspect of organizational efficiency, yet traditional methods often prove cumbersome and prone to errors. To address these challenges, this project delves into the realm of face attendance systems to revolutionize attendance tracking systems. By harnessing the power of generative AI, our aim is to develop a sophisticated recognition system capable of automating the attendance process while overcoming the limitations posed by facial changes, manual entry errors, and time-consuming procedures.

It is important to acknowledge potential drawbacks associated with alternative biometric approaches. For instance, employing retina scan technology may raise concerns regarding privacy and user acceptance, as it involves the use of sensitive biological information. Similarly, fingerprint attendance systems, while widely used, can encounter issues related to hygiene, as well as technical challenges in accurately capturing fingerprints, especially in environments with high user volumes.

At the heart of our endeavor lies the face recognition capabilities within attendance systems. By automating the attendance tracking process and minimizing the likelihood of errors associated with manual entry, our system alleviates administrative burdens and enhances overall efficiency within organizational settings.

We collected our own dataset to ensure the robustness of our model and used YOLOv8 **[6]** as a face detector along with a Face Recognition library **[12]** for accurate identification. Moreover, to manage the student data efficiently, we integrated MongoDB into our system, ensuring secure and scalable storage of information.

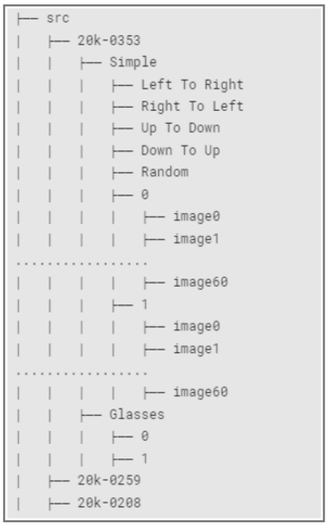
In addition, we integrated GAN models like Attribute Gan and Interface Gan for creating face variations. This integration proved vital, as face recognition systems often fail in scenarios where facial attributes change significantly. By generating diverse facial variations, our system maintains accuracy even in challenging conditions, thus enhancing reliability and usability.

In essence, our project represents a significant step forward in the optimization of attendance management practices through the innovative application of generative AI. By eliminating manual entry errors, adapting to facial variations, and reducing administrative overhead, our solution offers a streamlined approach to attendance tracking, ultimately fostering greater efficiency and productivity within organizations.

Dataset

Collecting a diverse and comprehensive dataset is crucial for training and evaluating face recognition systems, particularly in multicultural contexts such as the Indian-Pakistani region. Recognizing the limitations of existing datasets, which often lack representation from diverse ethnicities and cultural backgrounds, our project prioritized the collection of a robust dataset that encompasses a wide range of facial characteristics, poses, and expressions.

The image capture process involved capturing photographs of participants from multiple angles and facial expressions. Each participant's face was photographed in various orientations, including random, up, down, left, and right as shown in **[Fig 1.1 Dataset Hierarchy]** . The experiment was conducted twice to capture a comprehensive range of facial poses. Additionally, participants were instructed to display different facial expressions, such as neutral, smiling, and surprised, to introduce variability in the dataset. To capture real-world variations in lighting, participants were photographed outdoors. A more detailed walkthrough is shown on the next page.


```
├── src
|   ├── 20k-0353
|   |   ├── Simple
|   |   |   ├── Left To Right
|   |   |   ├── Right To Left
|   |   |   ├── Up To Down
|   |   |   ├── Down To Up
|   |   |   ├── Random
|   |   |   ├── 0
|   |   |   |   ├── image0
|   |   |   |   ├── image1
................
|   |   |   |   ├── image60
|   |   |   ├── 1
|   |   |   |   ├── image0
|   |   |   |   ├── image1
................
|   |   |   |   ├── image60
|   |   ├── Glasses
|   |   |   ├── 0
|   |   |   ├── 1
|   ├── 20k-0259
|   ├── 20k-0208
```

[Fig 1.1 Dataset Hierarchy]

Students = 100

5 Actions = Up, Down, Left, Right, Random

Frames in 1 Action = 60

Total Frames For All Actions = (5*Frames in 1 Action)

Experiment Repeated = 2*Total Frames For All Actions

Total Images = Experiment Repeated *100 = 600 images

Requirements

Hardware Interfaces

Camera Systems:

The system interfaces with camera systems installed at entry gates. These cameras capture facial images for attendance tracking. The hardware characteristics include imaging capabilities and the ability to operate under ideal lighting conditions.

Green Light Indicator:

Upon successful face recognition, the system(Raspberry Pie/Arduino) triggers a green light indicator. This interface is crucial for signaling attendees with approved access..

Metal Gates:

The system is integrated with metal gates, each equipped with a dedicated camera. These gates provide the physical structure for the three-lane testbed, facilitating the detection and recognition of individuals.

Software Interface

Web User Interface:

An intuitive user interface allows administrators to register users and manage system settings. The interface is designed to be user-friendly, contributing to efficient system administration.

Facial Recognition Algorithm:

The core software interface involves the Facial Recognition Computer Vision Algorithm. This algorithm is responsible for processing facial input, cross-referencing it with the database, and determining user existence.

Database (MongoDB):

The system interfaces with a robust database (MongoDB) to store and retrieve employee information. The database facilitates the comparison of captured facial images with registered user data.

Client Application:

The client side application will receive the feed from the cameras and utilize the facial recognition algorithm to recognize the user from the registered Database(previously entered by the admin via the Web User Interface).

Implementation

During the Implementation Phase, Agile practices and sprints play a pivotal role in enabling iterative development of the system. In parallel with design insights, the interface is meticulously crafted to encompass user-friendly features, including the capability for administrators to register users.

The implementation phase of the project entails the deployment of a testbed comprising three cameras positioned atop each gate, specifically targeting the zones where students are instructed to stand within the vicinity of each lane. These cameras are configured to conduct face recognition processes utilizing YOLOv8 **[6]**, a sophisticated convolutional neural network architecture renowned for its efficacy in object detection tasks. Upon detecting a human face within its field of view, the camera initiates the process of transmitting the captured facial image to the designated face recognition library for further analysis.

Within the face recognition library, the received facial image undergoes a series of computational procedures to generate facial embeddings, which encapsulate unique features of the individual's facial characteristics. Subsequently, these embeddings are cross-referenced against the comprehensive database of registered students stored within the MongoDB system. Upon identifying a matching entry within the database, indicative of the presence of a registered student, the system proceeds to activate a green LED light, serving as a visual indicator to verify the successful completion of the face recognition process

[Fig 2.4].

Face Detection

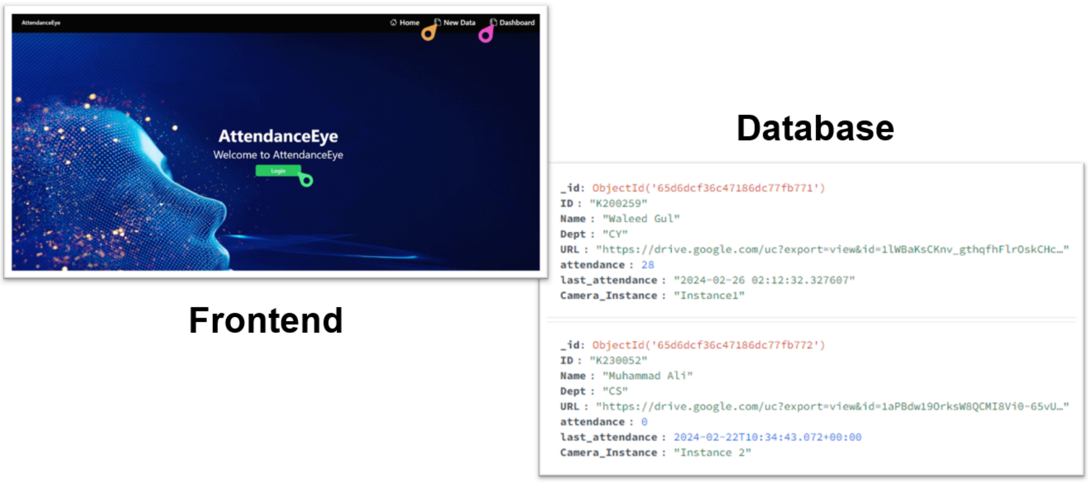
Various libraries were employed for face detection, including OpenCV and Dlib **[7]**. However, YOLOv8 exhibited superior performance compared to its counterparts. The results are delineated below in the Experimental Testing section.

Face Recognition

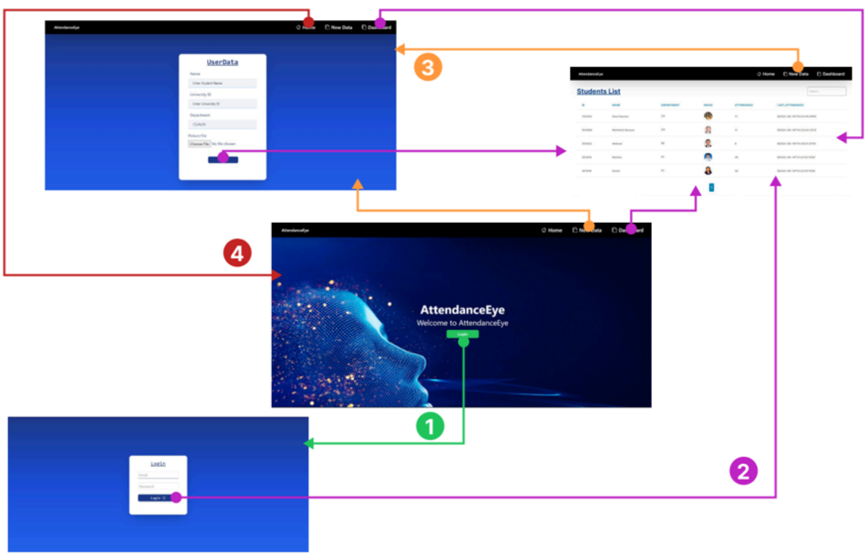
There are three main components to this project.

1. Server Side:

The website is made using the MERN stack (Mongodb, Express, React and NodeJs) shown on **[Fig 2.2 Frontend and Database]**. It contains an admin authentication and a dashboard UI to show all the registered students. It also supports registering, updating and deletion of the students in the database shown on **[Fig 2.1 Guide for Website]**.



[Fig 2.2 Frontend and Database]

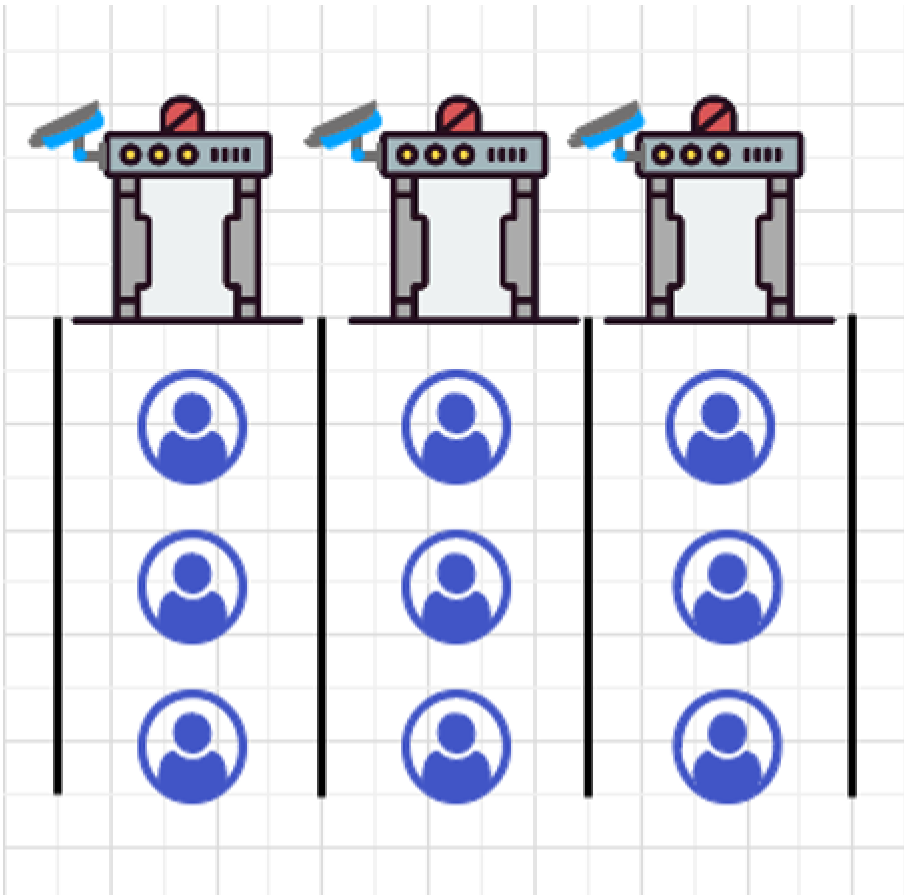


[Figure 2.1 Guide for Website]

2. Client Side - Hardware Implementation:

Three designated lanes will be provided for users to mark their attendance via camera recognition. Each gate will be equipped with a camera positioned overhead, which will facilitate the attendance marking process for a single student at a time. The camera will capture an image/frame of the user, initiate face detection algorithm, Create facial encodings using face recognition, cross-reference these against registered encodings stored in the database, and subsequently retrieve the pertinent information. Upon successful identification, the LED indicator will illuminate green.

The implementation utilizes the pyfirmata library to interface with the LED indicator on the Arduino board.



[Fig 2.3 Test Bed]

3. Client Side - Face Recognition:

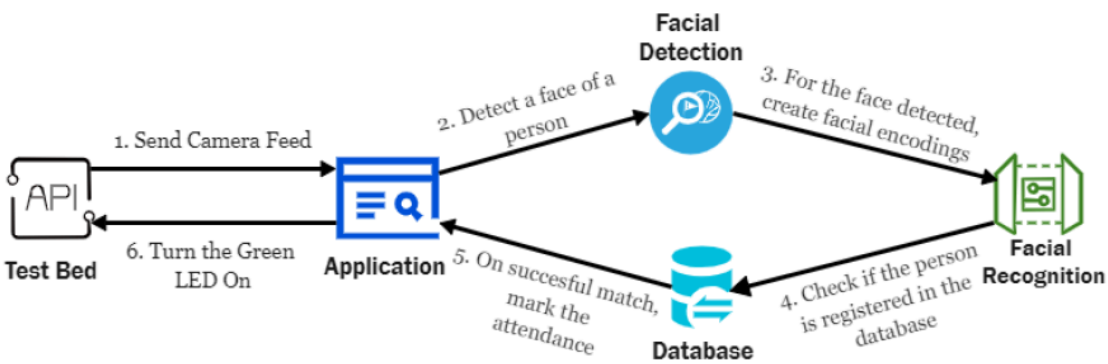
The face recognition functionality underwent rigorous testing utilizing two prominent libraries: DeepFace and FaceRecognition **[12]**. Following thorough evaluation, FaceRecognition emerged as the superior performer, showcasing optimal performance compared to DeepFace.

4. Multiple Face Recognition Using Multiple Cameras

Face recognition will be conducted using multiple cameras, with each camera focusing on a single student or employee within its field of view. This system employs multiprocessing, ensuring that each camera instance operates independently of others. Each instance is executed as a separate process, initiated by running the file multiple times in different terminals, specified by

python main.py (name of Camera Instance).

This approach enhances efficiency and scalability by enabling concurrent processing of multiple camera feeds without interference between instances.



[Fig 2.4 Face Recognition Process]

Experimental Testing

Facial Detection

In our project's thorough evaluations, we compared the performance of YOLOv8 **[6]**, Dlib **[7]**, and OpenCV. The results showed that YOLOv8 outperformed the others significantly, achieving an impressive F1 Score of 0.984 as shown in **[Table 1]**. The superior performance of YOLOv8 underscores its capability to accurately identify facial features with minimal processing time, making it a standout choice for integrating into our attendance management system. These findings highlight the importance of selecting the most suitable detection framework to ensure the reliability and effectiveness of AI-driven facial recognition systems in real-world applications.

Model	Accuracy	Precision	Recall	F1-Score
Yolov8	0.968604365	0.987970977	0.979616306	0.983775904
Dlib	0.506839378	0.999565311	0.491556220	0.659024145
Open-CV	0.297493667	0.344616351	0.666666666	0.454361894

[Table 1. Face Detection Model Comparison]

Facial Recognition

Our Facial Recognition evaluations found that face recognition (FR) without YOLO outperformed other methods, achieving an F1 score of 0.666. In our comprehensive comparison, we evaluated four approaches. First, we examined face recognition **[12]** with YOLO integration, which yielded an F1 score of 0.66666 **[Table 2]**, showcasing decent performance. However, without YOLOv8, the F1 score dropped to 0.48, indicating a decrease in accuracy. Nonetheless, the time efficiency of face recognition with YOLO was noteworthy, boasting the best time of 0.11 seconds, significantly faster than other methods. On the other hand, face recognition without YOLO took considerably longer, with the best time recorded at 7.4 seconds as shown in **[Table 3]**.

In contrast, Deepface without YOLO integration achieved an F1 score of 0.57, demonstrating competitive performance. Remarkably, Deepface without YOLO excelled in terms of speed, with a best time of 2.05 seconds, showcasing its efficiency in processing facial recognition tasks. However, when Deepface was integrated with YOLO, the best time increased to 9.2 seconds, indicating a significant slowdown in processing speed. Despite this, Deepface's F1 score remained relatively stable, suggesting that while the integration with YOLO might affect speed, it doesn't drastically impact the model's accuracy.

Model	Accuracy	Precision	Recall	F1-Score
FR With Yolo	0.77777	0.6333333	0.699999	0.666666
FR Without Yolo	0.77777	0.55	0.444	0.48
Deepface Without Yolo	0.6	0.59	0.55	0.57
Deepface With Yolo	0.58	0.56	0.38	0.42

[Table 2. Face Recognition Model Performance Comparison]

Model	Best Time	Average Time	Worst Time
FR With Yolo	0.11	0.42122	0.6
FR Without Yolo	7.4	10.24	19.4
Deepface Without Yolo	2.05	2.89	3.31
Deepface With Yolo	9.2	19.0	21.5

[Table 3. Face Recognition Model Inference Time Comparison]

Multiple Camera Face Recognition

The evaluation of multiple camera face recognition systems revealed intriguing insights into their respective strengths and efficiencies. Among the cameras tested, the second camera emerged as the swiftest in terms of face recognition, boasting an impressive time of 0.315 seconds. On the other hand, the first camera demonstrated remarkable proficiency in data retrieval from the database, accomplishing this task in a rapid 2.8 seconds. Furthermore, it excelled in marking student attendance swiftly, clocking in at an impressive 0.07 seconds, a performance closely matched by the third camera. The average face recognition time across all cameras stood at a commendable 0.324 seconds, with the best and worst times recorded at 0.315 and 0.352 seconds, respectively. Database operations consumed an average of 3.24 seconds, with the best time for marking student attendance noted at 0.15 seconds and the worst time at 0.35 seconds as shown in **[Table 4]**. These evaluations underscore the diverse capabilities of each camera in different aspects of the face recognition system, highlighting the nuanced trade-offs between speed and functionality in such systems.

Try	Face Recognition		Database	Marked	
	Best	Worst	Average	Best	Worst
1st	0.34	0.38	2.8	0.07	0.38
2nd	0.315	0.33	3.121	0.31	0.36
3rd	0.317	0.346	3.786	0.07	0.32
Average	0.324	0.352	3.24	0.15	0.35

[Table 4. Face Recognition Model Multiple Camera Comparison]

Facial Batch Recognition

In our analysis of batch recognition methods, we compared face recognition and batch inception models. The face recognition method achieved a commendable F1 score of 0.76 **[Table 5]**, indicating its strong performance in accurately recognizing faces. Additionally, face recognition boasted an impressive best time of 0.0215523 **[Table 6]** seconds, showcasing its efficiency in processing batches of facial data.

On the other hand, batch inception, while still respectable, achieved a slightly lower F1 score of 0.6, indicating a moderate decrease in accuracy compared to face recognition. However, it's worth noting that batch inception also demonstrated efficient processing capabilities, with a best time of 0.139792204 seconds, making it a competitive choice for batch processing tasks despite its slightly lower accuracy.

Furthermore, we delved deeper into the impact of cache utilization on face recognition performance. Initially, our analysis showed that when caching was first integrated, the processing time appeared longer at 0.0342266559 seconds. This initial observation might seem counterintuitive given that caching is generally expected to improve processing speeds. However, this initial longer processing time can be attributed to the system building and populating the cache, which requires some overhead.

As the system continued to operate and the cache became populated, we observed a significant improvement in processing time. Subsequent tests revealed that with cache fully functional, the processing time dramatically reduced to an astonishingly low 0.0000001 seconds **[Table 8]**. This remarkable improvement underscores the effectiveness of caching mechanisms in optimizing face recognition tasks once the initial setup and caching overhead are accounted for.

On the contrary, when face recognition was performed without cache integration, the processing time increased notably to 0.0285738 seconds. This stark contrast highlights the substantial impact of caching on streamlining face recognition processes and achieving optimal performance.

Model	Accuracy	Precision	Recall	F1-Score
Face Recognition	0.64	0.71	0.84	0.76
Batch Inception	0.6	0.5	0. 75	0.6

[Table 5. Batch Recognition Model Performance Comparison]

Model	Best Time	Average Time	Worst Time
Face Recognition	0.0215523	0.01052474	0.0332157611
Batch Inception	0.139792204	0.179710484	0.272111654

[Table 6. Batch Recognition Model Inference Time Comparison]

Face Recognition	Best Time	Average Time	Worst Time
With Video UI	0.0215523	0.01052474	0.0332157611
Without Video UI	0.0271295	0.1515293	0.0366699695

[Table 7. Batch Recognition Speed Comparison]

Face Recognition	First Time	Next Time
With Cache	0.0342266559	0.0000001
Without Cache	0.025386484	0.0285738

[Table 8. Batch Recognition Cache Comparison]

In our overall assessment of face recognition, incorporating caching alongside a user interface (UI) and simultaneous recognition of two people yielded impressive results. Specifically, the best processing time achieved under these conditions was an impressive 0.870 seconds **[Table 9]**. This notable efficiency demonstrates the effectiveness of combining caching mechanisms with a user-friendly interface to streamline the recognition process, even when dealing with multiple individuals simultaneously.

This optimized performance not only enhances speed but also ensures reliable and accurate face recognition outcomes. By leveraging caching and intuitive UI design, the system can efficiently handle real-time scenarios involving multiple individuals, making it a robust solution for various face recognition applications

Overall Model	Best Time	Average Time	Worst Time
Face Recognition	0.870	1.16	1.376

[Table 9. Overall Face Recognition Performance]

Research

1. Interface Gans

Face recognition systems have become increasingly prevalent in various domains, including security, surveillance, and attendance management. However, despite advancements in technology, many existing face recognition systems encounter challenges related to accuracy and robustness, particularly in scenarios where facial features undergo significant changes over time.

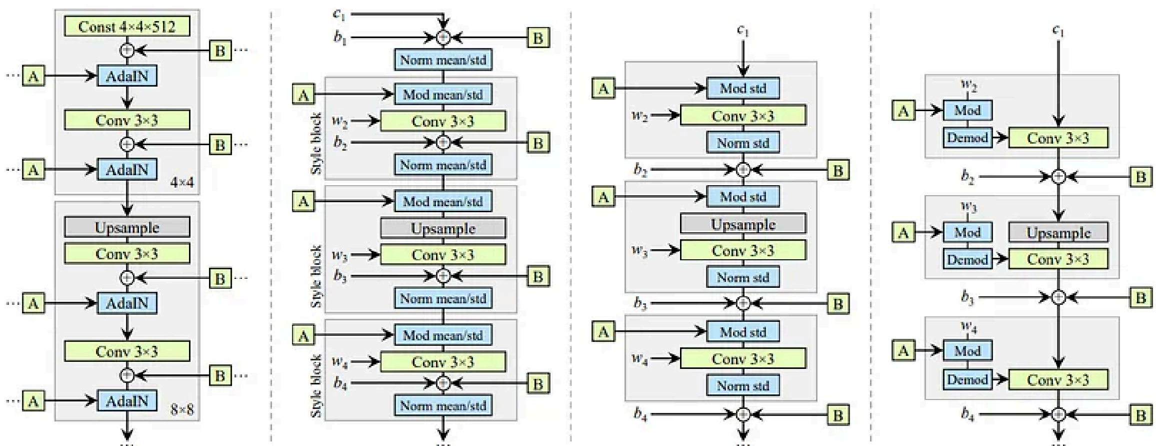
One common issue is the inability of conventional face recognition systems to accurately identify individuals as they age. As people grow older, their facial features undergo subtle yet distinct transformations, such as changes in skin texture, facial contours, and the appearance of wrinkles. These variations can lead to inaccuracies and inconsistencies in face recognition algorithms, resulting in false positives or failures to match stored facial templates with real-time images.

Moreover, facial attributes such as beard, mustaches, and hairstyles further complicate the recognition process. Conventional systems often struggle to adapt to variations in these features, leading to misidentification or rejection of valid matches.

To address these challenges, our research explores the integration of generative AI techniques into face recognition systems. Generative AI, particularly generative adversarial networks (GANs), offers a promising approach to synthesizing facial images with desired attributes. By leveraging GANs, we can generate realistic facial representations that encompass specific features, such as age progression, beard growth, or changes in hairstyle.

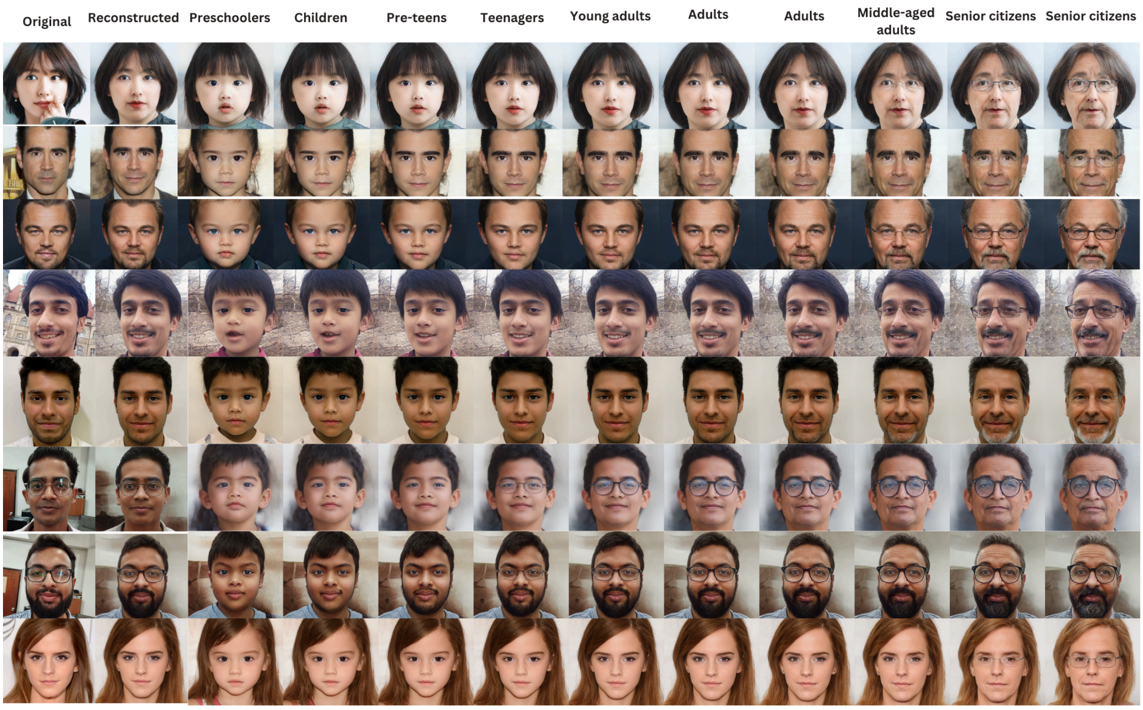
In our approach, we utilize generative AI to augment existing face recognition systems by dynamically generating certain facial features. For instance, when encountering facial images of individuals whose ages have progressed since their initial enrollment in the system, we employ generative AI to simulate age progression and adapt the stored facial templates accordingly. This enables our system to accurately recognize individuals across different stages of life, mitigating the limitations associated with age-related variations.

Furthermore, our research lays the groundwork for future enhancements to incorporate additional facial attributes, such as beard, mustaches, and hairstyles. By extending the capabilities of generative AI, we envision a system that can dynamically generate and incorporate these features into facial templates, thereby improving recognition accuracy and robustness. The InterfaceGan **[11]** uses StyleGan **[10]** whose architecture is as shown in **[Fig 4.1]**. In addition, rigorous testing was conducted on a diverse range of photographs featuring individuals from various demographics. The comprehensive results of these tests are meticulously presented in **[Fig 4.2]** of this report.



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

[Fig 4.1 StyleGan2 Architecture]



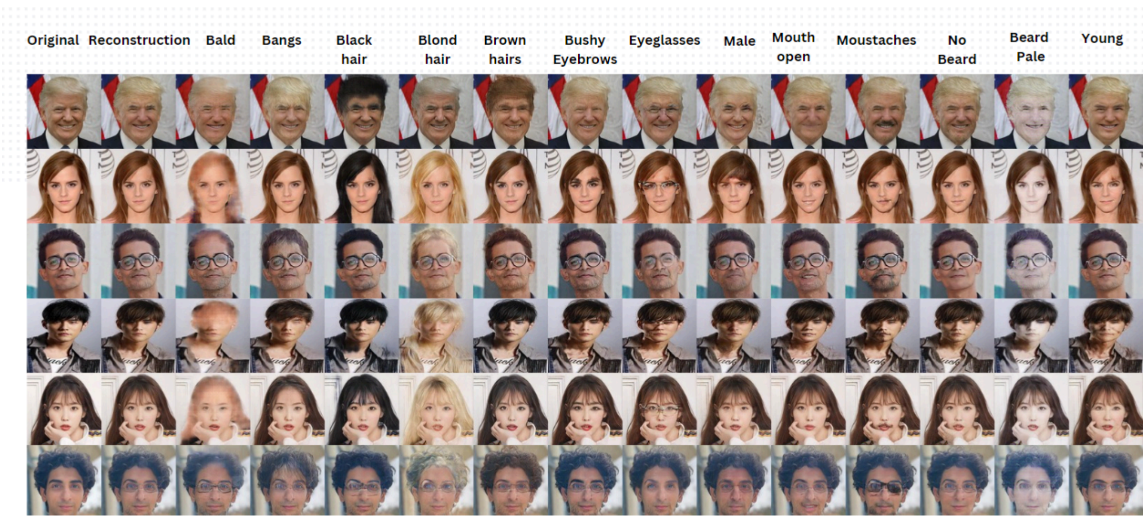
[Fig 4.2 StyleGan2 Results on Images]

2. Attribute Gans

To address the limitations of traditional face recognition systems in capturing variations in facial attributes such as age, hair color, and facial hair, we employed Attribute Generative Adversarial Networks (GANs). These Attribute GANs, also known as attribute-conditioned GANs, enable the generation of realistic facial images with specific attribute variations, thereby enhancing the robustness and accuracy of face recognition algorithms.

Attribute GANs **[9]** were trained to generate synthetic facial images with variations in specific attributes such as age, hair style, hair color, facial hair, and gender. By conditioning the GANs on desired attribute labels, we could control and manipulate the appearance of generated faces according to predefined attribute configurations.

Each synthetic facial image generated by the Attribute GANs was associated with attribute labels specifying the desired variations in facial attributes. For example, attribute labels could include attributes such as Male/Female, Young/Old, Bald/Haired, Beard/No Beard, and Hair Color (e.g., Black, Blond, Brown) as shown in **[Fig 4.3 AttributeGan Results on Images]**



[Fig 4.3 AttributeGan Results on Images]

Conclusion

In conclusion, the integration of generative AI in attendance management systems marks a significant advancement towards efficiency and accuracy in organizational processes. By harnessing the power of AI-driven face recognition, this project offers a promising solution to the challenges posed by manual attendance tracking methods. Through meticulous analysis and strategic implementation, our system demonstrates a commitment to enhancing user experience while ensuring reliable and adaptive recognition capabilities. The systematic approach employed throughout the project, from design to implementation, underscores the dedication to creating a robust and user-centric solution tailored to the needs of modern organizations.

Furthermore, the methodology employed in this project exemplifies a comprehensive and multifaceted approach to developing an advanced attendance management system. The design phase laid a solid foundation by establishing secure facial data capture mechanisms, while the setup configuration emphasized the importance of tailored camera systems for precise detection. The implementation phase further showcased a commitment to accuracy and usability, with a focus on intuitive user interfaces and strategic API integration. By leveraging cutting-edge technologies and a rigorous testing process, this project not only addresses current attendance tracking challenges but also sets the stage for future advancements in AI-driven organizational systems.

References

[1] Paul P. Oroceo,Jeong-In Kim,Ej M. Francisco, Sang-Ho Kim,Sang-Ho Kim,"Optimizing Face Recognition. Inference with a Collaborative Edge–Cloud Network" ,1 November 2022

[2] B.Ali Abdalkarim - "A Literature Review on Smart Attendance Systems"July 2022

[3] W.Chen, H.Huang, S.Peng, C.Zhou , C. Zhang,- "YOLO-face: a real-time face detector",March 2020

[4] S.Khan,A Akram,N. Usman, "Real Time Automatic Attendance System for Face Recognition",March 2020

[5] Dr. A. Babu Karuppiah, M. Jeyalakshmi, L. Johnsilin Shiny, B. Sri Devi,"Online Attendance System",2017

[6] YOLOv8 Face Detection Library (January 2023)

[7] Dlib Face Detection Library(2002) Updated in 2023

[8] YOLO-face: a real-time face detector (March 2020)

[9] Zhenliang He,Wangmeng Zuo, Meina Kan, Shiguang Shan, Xilin Chen ,"AttGAN: Facial Attribute Editing by Only Changing What You Want" (November 2019)

[10] Tero Karras, Samuli Laine, Timo Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks" (Dec 2018)

[11] Yujun Shen, Ceyuan Yang, Xiaoou Tang, Bolei Zhou, "InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs" (18 May 2020)

[12] Face Recognition Library