AI Facial Recognition System in Healthcare

Adham Hossam Abo El-Magd	211000564
Ahmed Hesham Al Mashad	211000390
Ahmed Khaled	211000699
Youssef Mansy	211001591
Shahd Hamdy Rageb	211001014

ECEN305

Dr. Mohamed Saeed Darwish

December 2023

Contents

Abstract 4
Keywords: Facial Recognition – AI – Machine Learning – OpenCv – SQL – Tkinter – GUI – Healthcare 4
Introduction 5
Methodology 6
1. Machine Learning Model 6
2. GUI Using Tkinter 9
3. Database Using MySQL 10
Results 11
Recommendations and Future Work 13
Conclusion 14
List of References 15

Table of Figures

Figure 1: AI vs ML vs DL 5
Figure 2: Types of Haar-like features........ 6
Figure 3: Haar-like features on human faces........ 6
Figure 4: Integral Image representation........ 7
Figure 5: Cascading Classifier Methodology........ 8
Figure 6: Graphical User Interface (GUI) 9
Figure 7: Database using MySQL........ 10
Figure 8: Face Detection with-and-without face mask........ 11
Figure 9: Face Detection with different type of face mask (N95)........ 11
Figure 10: Face Detection of multiple people in same frame 11
Figure 11: Project overview and lifecycle 12

AI Facial Recognition System in Healthcare

Abstract

Smart biometrics identification systems have become a staple of modern electronic devices, and it is used in an array of security and law enforcement applications as a reliable way to verify peoples' identities. However, our project aims to tackle the weakness of recognition techniques, specifically facial recognition when the face is covered with glasses or face masks. This project aims to examine and apply a system that uses machine learning algorithms using Python to identify different people using their faces. Face masks are highly used by medical care providers in hospitals. The project's main application is a face recognition system in a hospital that can give access to doctors and employees to access restricted areas quickly and prohibit entry to unauthorized people, minimizing physical touch and the dependency on fingerprint biometrics (since doctor's wear gloves most of the time). It can also accurately identify patients (especially for those who don't have their ID or can't remember their names) and allow quick check-in and check-out (streamlining administrative processes, saving more time and saving more lives), which helps avoid medical errors caused by mistaken identity and prevent insurance fraud. The project utilizes several technologies, OpenCv (a real-time optimized computer vision library) for face detection, MySQL (an open-source relational database management system) for storing patients'/doctors' data, and Tkinter (a Python binding to the TK GUI Toolkit) to develop an interactive GUI able to gather patients'/doctors' data, train it, and detect faces.

Keywords: Facial Recognition – AI – Machine Learning – OpenCv – SQL – Tkinter – GUI – Healthcare

AI Facial Recognition System in Healthcare

Introduction

As humans modernize and develop the wide range of different electronic and computer systems that are increasingly spreading into our everyday lives, the problem of security and verification of identity arises. To prevent unauthorized personnel from entering restricted areas, or people trying to gain access to an electronic device with malicious intent, the use of biometrics is becoming a more viable solution as the accuracy and efficiency of those systems is increased.

Most biometrics rely on AI and different machine learning (ML) and deep learning algorithms to achieve the task of accurately and efficiently identifying users. Those terminologies are all reliant on one another with artificial intelligence being an umbrella term for the latter two; it can be defined as any system that can carry out tasks that require human intelligence. A derivative of AI is machine learning as it is considered a system that can learn how to perform tasks through trial and error until it eventually reached extremely high accuracy. A subset to ML is deep learning (as shown in **figure 1**) which can be defined as a system that does not need human intervention almost entirely reaching an accurate conclusion.

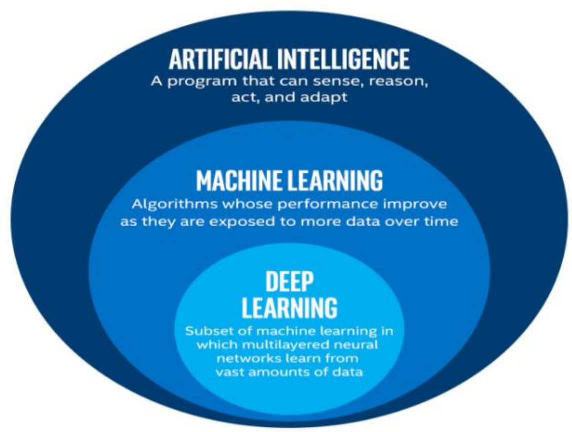

Figure 1: AI vs ML vs DL

Figure 1: AI vs ML vs DL

Face recognition (one of the most powerful security tools) was pioneered by the Viola-Jones object detection framework developed by Paul Viola and Michael Jones in 2001. Facial recognition was even more improved by techniques such as Local Binary Pattern (LBP) as a unique, efficient textural operator, among other techniques such as Convolutional Neural Networks (CNN), to name a few.

Methodology

1. Machine Learning Model

Our face recognition system is made up of three main functions: Generate User Data, Train Dataset, and Detect Faces through utilizing OpenCV library. OpenCV, a powerful computer vision library, provides tools for image and video processing, including pre-trained models like the Haar Cascade classifier for effective object detection based on the Viola-Jones Framework. The Haar training module in OpenCV includes tools for customizing classifiers, allowing the development of specific object or pattern detectors using positive and negative samples. Python's OpenCv library comprises of four key stages:

1- Selecting Haar-like features:

Haar-like features operate by analyzing variations in pixel intensities within specific regions of an image, particularly in the context of object recognition, such as detecting human faces. There're 3 types of Haar-like features: Edge features**,** Line features**,** and four-sided features, as shown in **figure 2**.

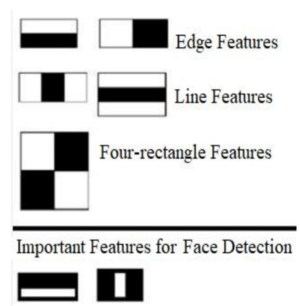

Figure 2: Types of Haar-like features

Thousands of these features exploit the universal properties of facial structures, like the darker nature of the eye region compared to its surroundings and the brighter characteristics of the nose region in contrast to the eyes (as shown in **figure 3**). The calculation of it involves subtracting the sum of pixel values in black area from the sum in white area. This numerical value represents the contrast between the two regions.

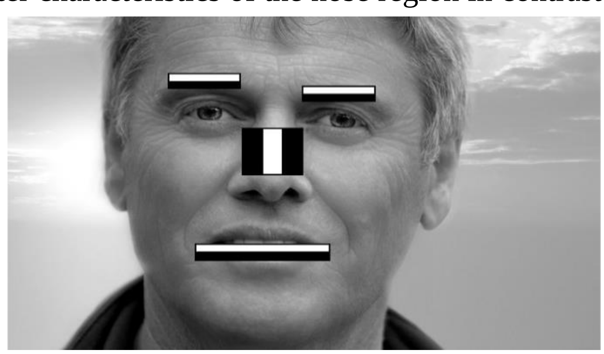
Figure 3: Haar-like features on human faces

2- Creating an Integral image: Converting a normal image to an integral image is achieved by creating a data structure that efficiently calculates cumulative pixel value sums within the image or rectangular regions. This structure, referred to as a summed-area table, markedly improves computational efficiency and reduces time complexity in image processing.

As shown in **figure 4**, in the first row, the pixel in yellow (35) is represented by the sum of all purple pixels before it, the same is done with the pixel 20 in yellow. This accelerated computation plays a pivotal role in the rapid evaluation of Haar-like features and other image-processing tasks, enhancing the overall efficiency of algorithms like those integral to the Viola-Jones framework for face detection.

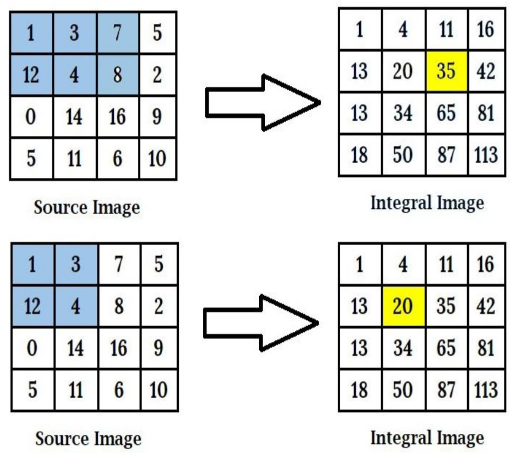

Figure 4: Integral Image representation

3- Adaboost Training: AdaBoost is integral to the Viola-Jones algorithm, homing in on crucial features from around 160,000 in a 24x24 detector window. Each Haar-like feature serves as a weak learner, and AdaBoost evaluates their performance during training. Positive responses in subregions, denoting potential human faces, receive a higher weight. This process yields a robust classifier, effectively discerning significant features. AdaBoost, fed with training data, sets a minimum threshold, enabling the algorithm to predict the usefulness of features in human face detection.

4- Cascading Classifiers: Cascading Classifiers in the Viola-Jones algorithm streamline face detection by dividing the process into stages: the first stage employs a classifier with the best features, quickly discarding non-faces. Subsequent stages, containing remaining features, progressively filter image subregions. If a subregion receives a negative evaluation at any stage, it is promptly rejected, as illustrated in **figure 5**. The cascading approach optimizes efficiency, swiftly identifying frontal faces by iteratively approving subregions until all classifiers endorse an image as a successful detection.

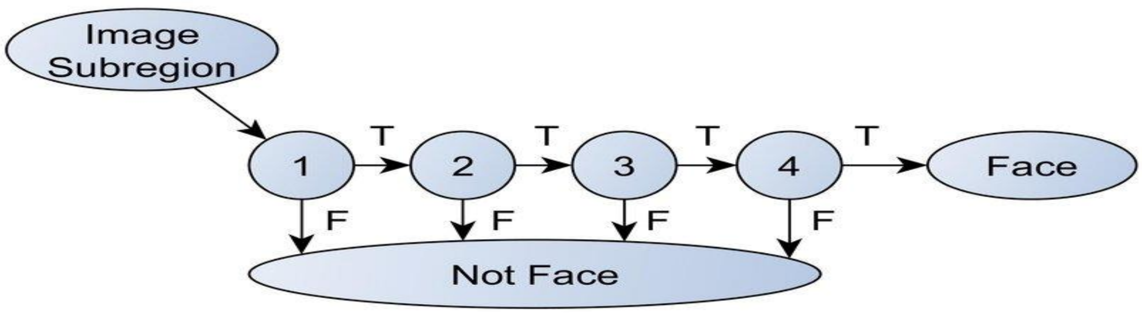

Figure 5: Cascading Classifier Methodology

AI Facial Recognition System in Healthcare

2. GUI Using Tkinter

Upon accessing the GUI, users encounter four primary operations (as shown in **figure 6**):

1- **Entering personal information:** This section enables users, whether medical professionals or patients, to input important personal data, such as name, age, and address. This data is crucial for organizing and associating user-specific datasets and identification purposes.

2- **Generating a new dataset:** Users can generate new datasets through this feature. This step involves capturing facial images or videos, ensuring a diverse representation of faces under various lighting conditions, angles, and backgrounds. This feature is enabled for real-time dataset creation (since the system is connected to MySQL database) so there's no need to manually edit the code and everything is automated.

3- **Training the dataset:** The system uses the collected data to create and fine-tune the recognition model. Training involves feeding the collected images into the algorithm, allowing it to learn and adjust its parameters for effective face detection.

4- **Running the application to detect faces:** The final operation launches the application to detect faces. This function utilizes the trained model to analyze input images or live video streams, identifying faces. Users, whether medical professionals or individuals using the system, can employ this feature for various purposes, such as patient identification or security verification.

Additionally, the GUI's design incorporates visual elements, such as buttons, input fields. This intuitive interface streamlines user interactions, providing distinct functionalities for both medical professionals and patients within the application.

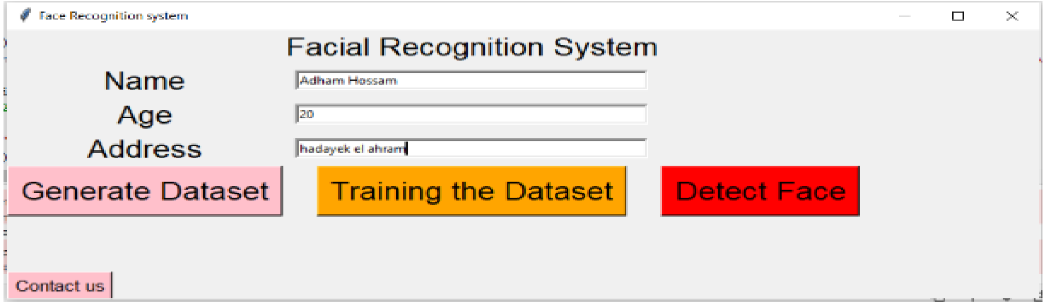

Figure 6: Graphical User Interface (GUI)

3. Database Using MySQL

MySQL served as the backbone for storing crucial patient and doctor information. We organized patient details such as names, ages, and addresses. Additionally, the system efficiently managed doctor information, facilitating the seamless association of patient records with corresponding healthcare professionals. By utilizing MySQL's structured and secure storage, our system ensured reliable data persistence, enabling easy retrieval and management of essential medical data.

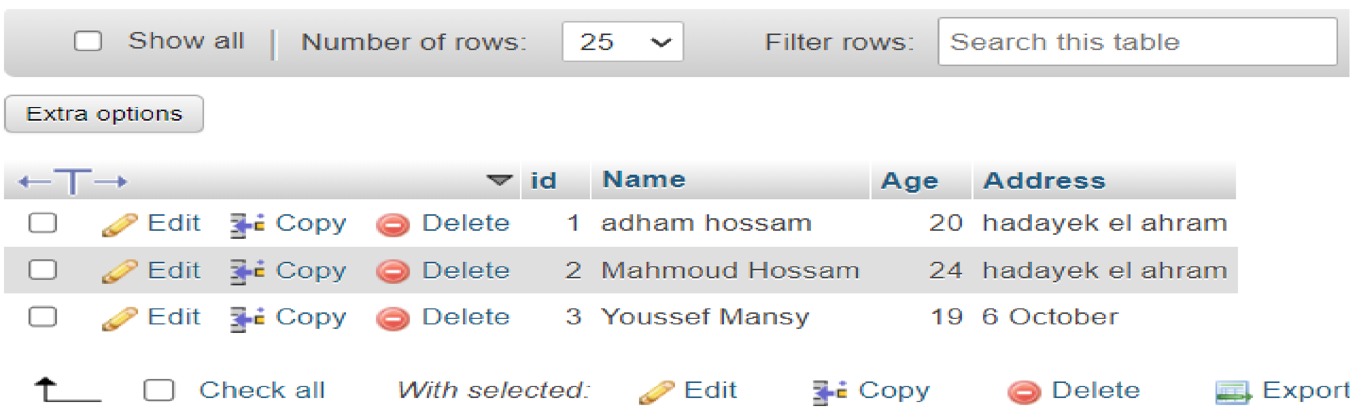

Figure 7: Database using MySQL

Results

After training the model on only 200 pictures, it was successfully able to identify users' faces without masks, with masks, and even identify multiple people in the same frame with different lighting condition – both with and without masks. Figures 8,9, and 10 show the project working under different scenarios/conditions.

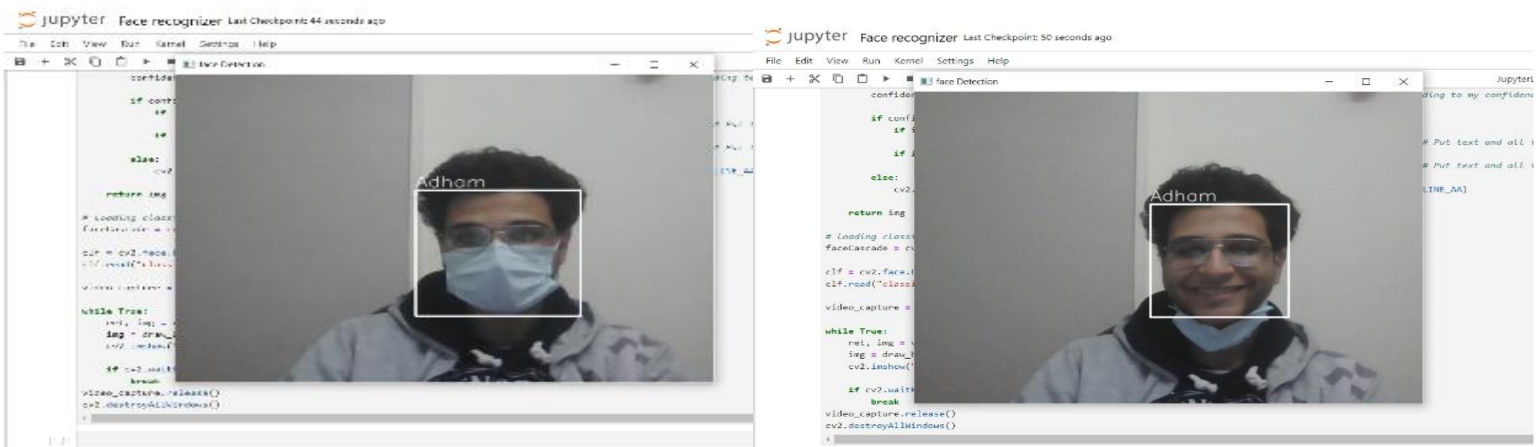

Figure 8: Face Detection with-and-without face mask

Figure 9: Face Detection with different type of face mask (N95)

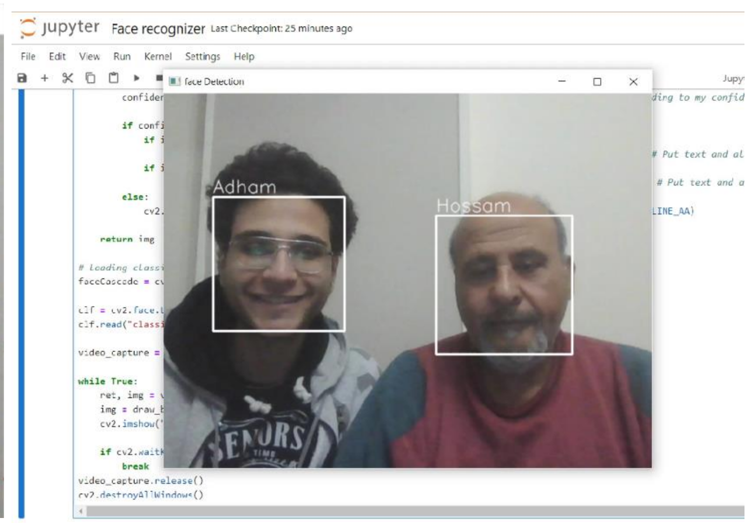

Figure 10: Face Detection of multiple people in same frame

Figure 11 includes a picture that describes the lifecycle/user experience for the application and all of its details.

1- The three main applications for the project are: generating data, training data, and detecting faces.
2- WellScan Facial AI will be implemented in healthcare institutions such as hospitals and small clinics in universities and compounds.
3- The two customer bases are medical staff and patients. For medical staff, they have to undergo data generation twice (with and without mask), while patients are only required to scan only once.

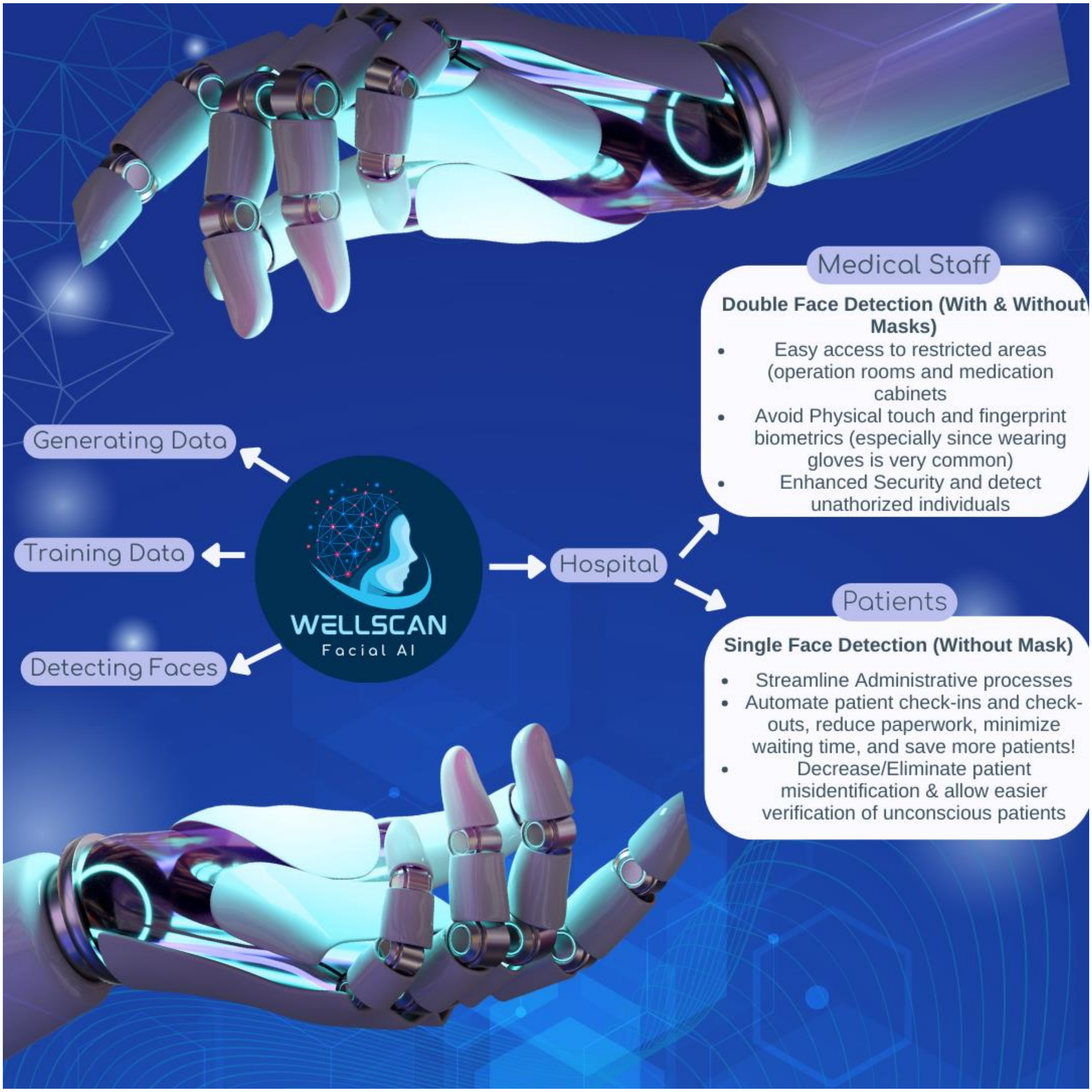

Figure 11: Project overview and lifecycle

Recommendations and Future Work

Eye detection: Detecting facial features above nose level can prevent false negatives from differing facial features due to facial hair or wearing of face masks. This The incorporation of higher facial detail analysis will increase the system's adaptability to diverse facial appearances (such as facial hear and masks) and ensure a more comprehensive recognition process.

Collaboration with hospitals and medical institutions: Collaborative partnerships with healthcare institutions, including hospitals and smaller clinics in universities or compounds, is essential for the successful implementation of our project. The goal of the partnership is to achieve official authorization to deploy our facial recognition system. The dataset collected will include critical biometric details such as facial biometrics, personal identification information (names, ages), medical prescriptions, and historical medical records. The integration of this data into our secure MySQL database infrastructure ensures seamless utilization for improved patient care and management within medical settings, in addition to increased efficiency and timesaving.

Modify the project for different applications: Expanding the scope of the current facial recognition system is essential. Diverse applications include security gates at residential compounds and/or universities to identify and authorize entry, automatic university attendance system, and company employees’ arrival and leaving time.

Conclusion

In conclusion, our Facial Recognition System was able to identify the faces of registered users with a high level of accuracy. The application takes 200 pictures during which the user is encouraged to show different angles of their face. Then, it is trained on the dataset saved on a localhost and immediately shows if it recognizes them or not. The system was also able to identify users wearing a face mask of any color; however, the user needs to input a data set with the mask on for that to happen. Among the future suggestions are adding eye detection to detect facial features above nose level, which can prevent false negatives from differing facial features due to facial hair or wearing of face masks. Collaborating with hospitals (smaller ones in compounds) to officially allow the use of facial recognition on patients and gather data (i.e., face biometrics, name, age, prescriptions, previous medical conditions) to store in our database (using MYSQL).

List of References

[1] Wang, Y. (2014, June 26). An Analysis of the Viola-Jones Face Detection Algorithm. Image Processing online. https://doi.org/10.5201/ipol.2014.104

[2] A. Kumar, A. Kaur, and M. Kumar, “Face detection techniques: A Review,” Artificial Intelligence Review, vol. 52, no. 2, pp. 927–948, 2018. doi:10.1007/s10462-018-9650-2

[3] How to Detect Face Recognition using Viola Jones Algorithm. (2023, June 13). Great Learning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/viola-jones-algorithm/

[4] Python GUI tkinter. (2023, November 9). GeeksforGeeks. https://www.geeksforgeeks.org/python-gui-tkinter/

[5] Link for project visual Mind Map: https://rb.gy/waeh7q