International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 9, Issue 01, January-2024, http://ijmec.com/, ISSN: 2456-4265

HOUSE SECURITY THROUGH FACE RECOGNITION USING OPENCV

Basva Ravi Chandra, Mr. N. Thirupathi Rao

[1]B.tech Student, Department Of Electronics and Computer Engineering, J.B Institute of Engineering and Technology

[2]Associate Professor, Department Of Electronics and Computer Engineering, J.B Institute of Engineering and Technology

Abstract: Face Recognition is a widely researched and applied field with numerous practical applications, from security systems to personalized user experiences. This project introduces a Face Recognition system that harnesses the power of machine learning and OpenCV, a versatile computer vision library, to accurately identify individuals from facial images. The project employs a combination of feature extraction, machine learning algorithms, and facial landmark detection techniques to create a robust face recognition model.

Machine learning algorithms like Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs) or k-Nearest Neighbors (k-NN) are trained on the extracted features to create a face recognition model. The model's accuracy is evaluated using various metrics and benchmark datasets, ensuring its effectiveness across different lighting conditions, poses, and facial expressions. The project also supports real-time face recognition from video streams or camera feeds. The implementation addresses privacy concerns by focusing solely on facial features for recognition, rather than storing or sharing personal data.

Keywords: Convolutional Neural Networks, Support Vector Machines, k-Nearest Neighbors.

I. INTRODUCTION

In response to the evolving demand for robust and scalable face recognition systems, this project centers around the creation of a sophisticated prototype. The overarching goal is to integrate Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) to achieve a multi-faceted and accurate face recognition system. The initial phase involves employing CNN for efficient face detection and high-dimensional feature extraction, providing a foundation for subsequent processing. SVM is then implemented to classify individuals based on these embeddings, enhancing recognition precision. Furthermore, the incorporation of KNN introduces adaptability and flexibility to the system, allowing it to accommodate diverse datasets and environmental conditions. This prototype acts as a pioneering solution, not just limited to its immediate applications but laying the groundwork for future developers interested in advancing automated face recognition technology. The emphasis on incorporating multiple algorithms aims to create a comprehensive system that can adapt to various scenarios, making it a valuable asset for security, surveillance, and identity verification applications. The project strives to set a benchmark by merging cutting-edge technologies, paving the way for a new era in face recognition system development.

II. LITERATURE SURVEY

In a literature review, a researcher examines prior work, research, conference papers, books, articles, etc. With this information, one can learn what work has already been done on the issue, summarize it, and identify gaps in the work. After analysis, they might focus on limits and find ways to get around them to get better results.

The proposed project integrates cutting-edge facial recognition technology with deep learning methodologies, employing Convolutional Neural Networks (CNNs), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) algorithms. By leveraging CNNs, it extracts robust facial features, while SVM and KNN facilitate effective face recognition. The prototype aims to create a versatile facial recognition system with applications in security, surveillance, and personal device access. While the project serves as a functional prototype, it allows for further development by interested parties, showcasing the potential for enhancing security and convenience in various domains. The literature survey explores existing works in facial recognition, deep learning, and related fields, providing a comprehensive foundation for the project's technological integration and future advancements.

G. He and Y. Jiang, have previously published a journal titled "Real-time Face Recognition using SVM, MLP, and CNN," *They have used technologies like Convolutional Neural Networks, Support Vector Machines, Multi-Layer Perceptron to increase the accuracy of the predictions. Their fused algorithm gave an approximate accuracy of 60%, 70%, and 80% with SVM, MLP, and CNN respectively. The project was first developed on the Tkinter module.*

K. T. Islam, R. G. Raj, and A. Al-Murad worked on detecting and classifying faces in their paper titled "Performance of SVM, CNN, and ANN with BoW, HOG, and Image Pixels in Face Recognition," in which they used Support Vector Machines, Convolutional Neural Networks, Bag of Words, Artificial Neural Networks, Histogram of Oriented Gradients and Image Pixels for face recognition. In the project, each subject consists of 10 images with different facial expressions, different illumination, and the dimensions of the images.

P. R. K. Reddy and G. Uganya, in their study titled "Comparative Analysis of Accuracy of Face Recognition System Using CNN and SVM" have concluded that Convolutional Neural Network achieves image accuracy rate of 95% and Support Vector Machine achieves image accuracy rate of 83% where the study was conducted on 2 Groups of subjects, wherein, Group 1 with 20 samples of CNN and group 2 with 20 samples of SVM were collected to analyze

III. ANALYSIS

USER REQUIREMENTS

This project addresses essential user requirements by providing efficient face recognition from images and live video streams. Users benefit from a streamlined process of embedding and recognizing faces, enhancing security measures. The system ensures simplicity in operation, requiring minimal user input, and accommodates real-time recognition demands. Additionally, the project allows flexibility in data handling and model training, catering to diverse user needs in face recognition applications. Overall, it meets user expectations for accuracy, ease of use, and adaptability to varying scenarios.

IV. DESIGN

The design of the project is structured to seamlessly integrate cutting-edge technologies for optimal facial recognition performance. The initial phase involves image data acquisition from diverse sources, enabling a comprehensive dataset. Subsequently, a sophisticated face detection model, based on the OpenCV deep learning framework, is employed. This model efficiently localizes faces within images, ensuring accurate subsequent processing. The core of the design encompasses a facial embedding model, utilizing OpenFace's NN4 small2 v1 architecture. This model quantifies facial features into a 128-dimensional vector, providing a robust representation for recognition. The dataset is then processed, extracting facial embeddings and associating them with respective identities.

The training phase utilizes the Support Vector Machine (SVM) algorithm from the scikit-learn library. This classifier, configured with a linear kernel, refines the recognition model by learning the patterns within the facial embeddings. The resulting recognizer, along with the Label Encoder, is serialized and stored for future use

The system architecture extends to real-time recognition through webcam integration. The video stream undergoes the same face detection and embedding processes, allowing instantaneous identification of individuals. The deployment diagram illustrates the seamless interaction between components, ensuring a cohesive and efficient system. Overall, the design prioritizes accuracy, speed, and adaptability, making it a robust solution for facial recognition applications.

DFD OR UML DIAGRAMS

Use Case Diagram

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously putting effort to make a truly industry standard. UML stands for Unified Modeling Language. UML is a pictorial language used to make software blue prints.

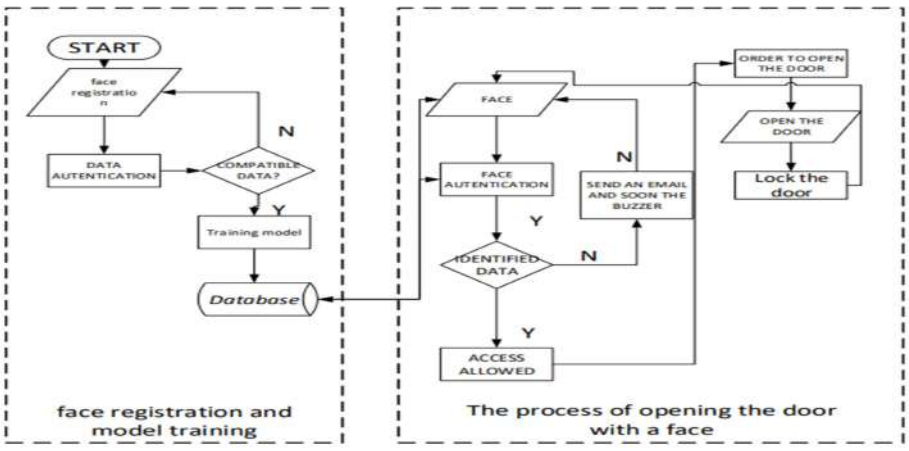

Fig : Use Case Diagram

Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain four compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

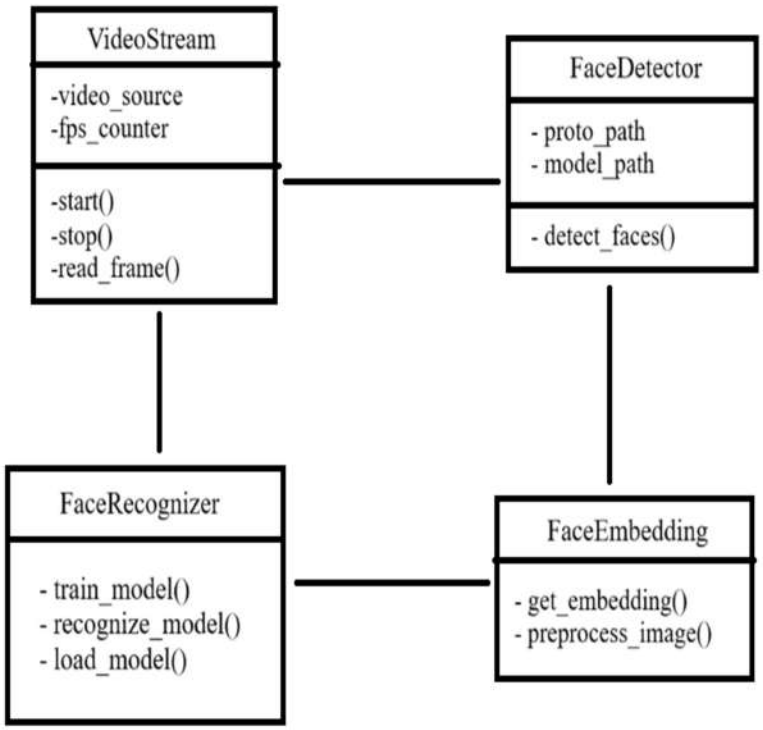

Fig : Class Diagram

V. BLOCK DIAGRAM

A block diagram is a schematic representation of a system or process, illustrating its key components as interconnected blocks. Each block signifies a distinct function or entity, and the connections between blocks

depict relationships or interactions, providing a concise visual overview of the system's structure and functionality.

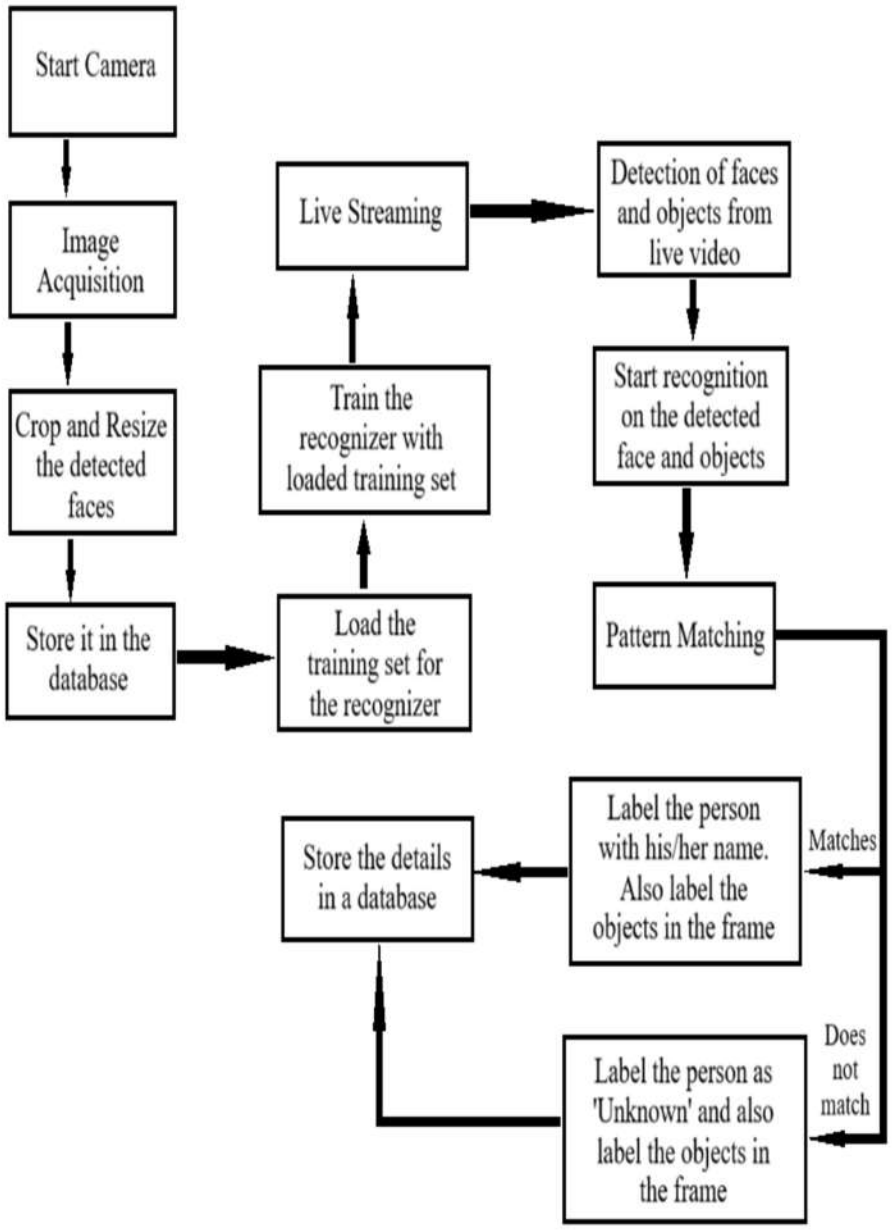

Fig: Block Diagram of Face Recognition using CNN

Sequence Diagram

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the

subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

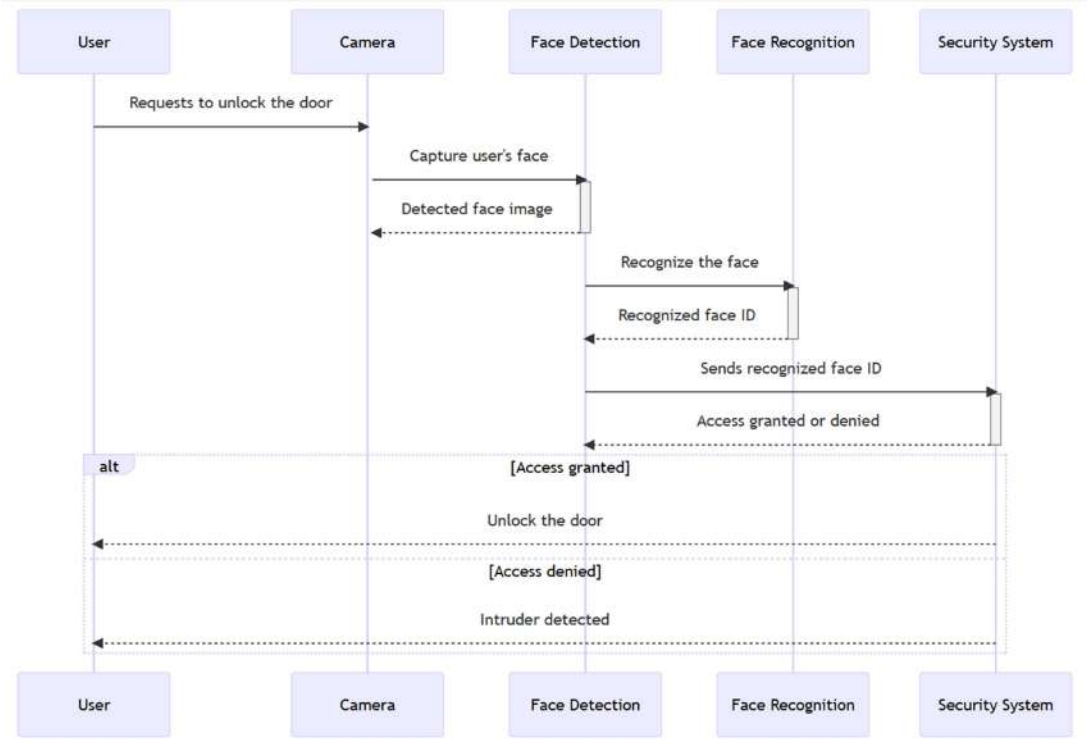

Fig: Sequence Diagram

Deployment Diagram

The deployment diagram illustrates the physical arrangement of software components in a system. It details the distribution of modules across hardware nodes, showcasing how the application is deployed on servers, devices, or platforms. This visual representation aids in understanding the system's topology and interaction between components during execution.

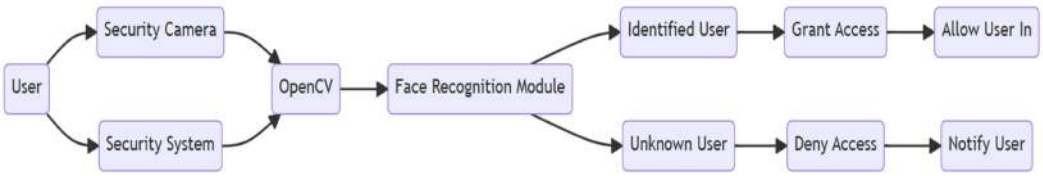

Fig: Deployment Diagram

Activity Diagram

Activity diagrams are graphical representations of the Workflow of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

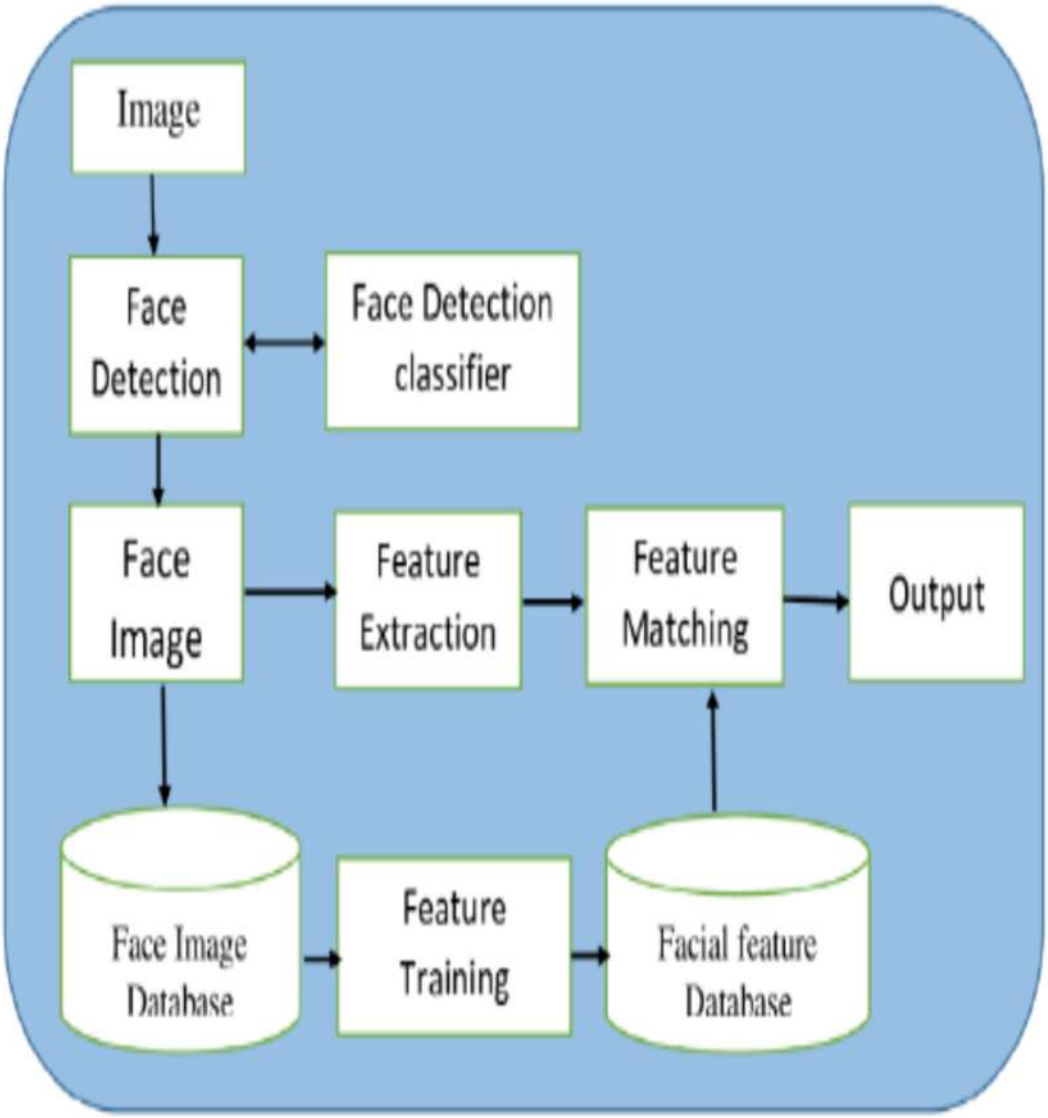

Fig: Activity Diagram

Dataflow Diagram

A Data Flow Diagram (DFD) in a project is a visual representation illustrating the flow of data within a system. It showcases processes, data stores, data sources, and destinations, offering a comprehensive overview of how data moves and transforms throughout the system. DFDs are essential for understanding the system's architecture, identifying data dependencies, and facilitating effective communication among project stakeholders. They help in analyzing, designing, and documenting complex systems, providing a clear and structured representation of data interactions and processes.

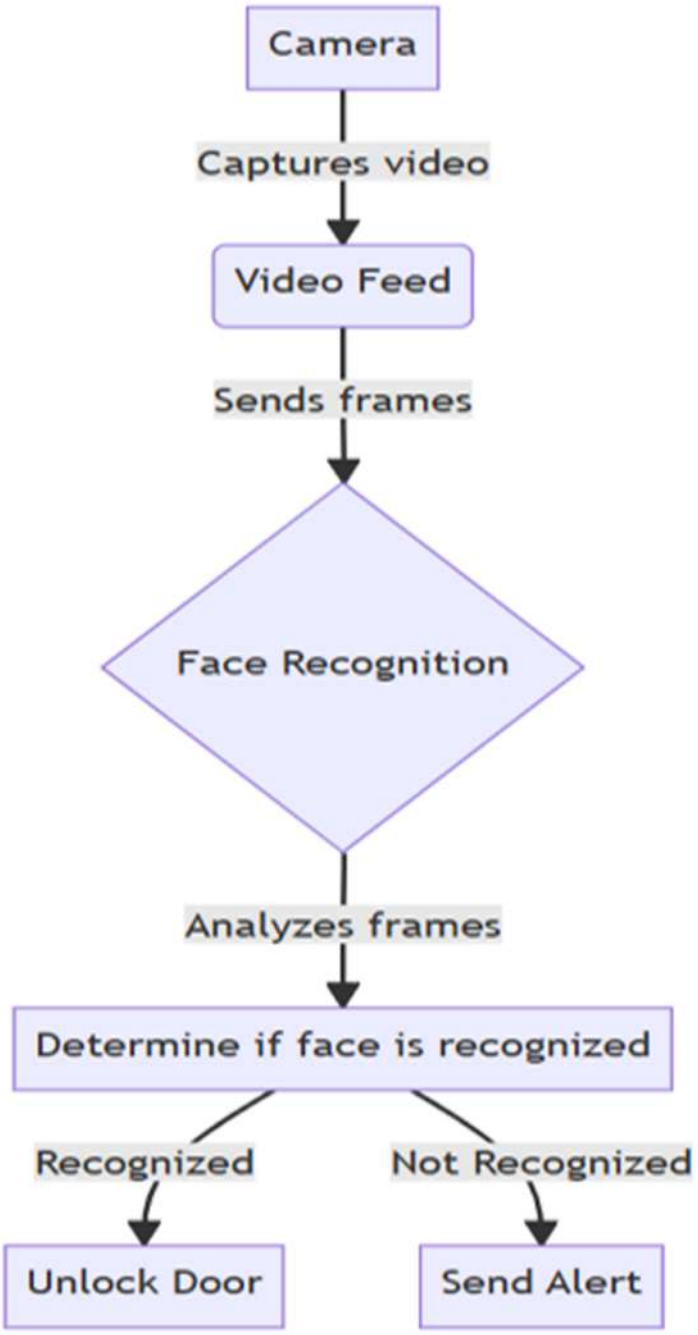

Fig: Dataflow Diagram

System Architecture:

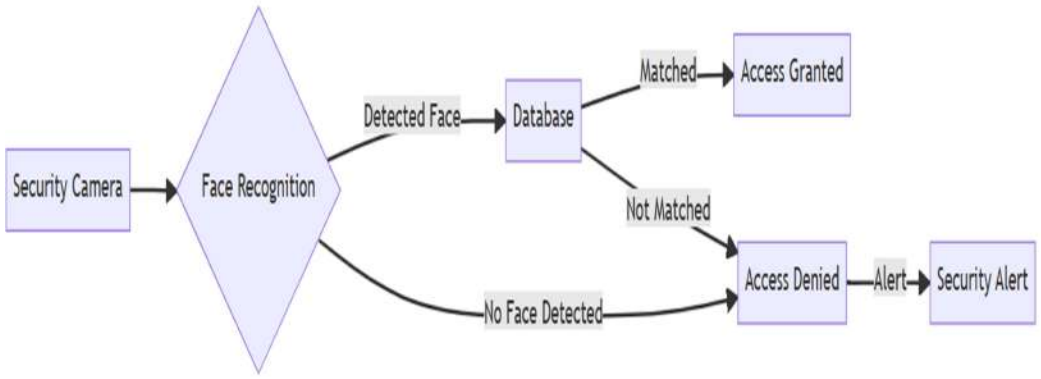

Fig: System Architecture

VI. IMPLEMENTATION

The implementation of the project involves several key steps to enable effective face recognition. Initially, the OpenCV deep learning-based face detector is employed to extract faces from images, which are then resized for consistency. Face embeddings are generated using OpenFace, and a dataset is constructed with corresponding person names. The model training phase utilizes a Support Vector Machine (SVM) classifier from scikit-learn, with labels encoded using LabelEncoder. The trained model, along with the LabelEncoder, is saved for subsequent recognition tasks. The recognition process spans both static images and real-time video streams, with the latter utilizing the imutils and VideoStream modules for efficient processing. The project's deployment is facilitated by packaging it into a deployable format, accompanied by clear deployment instructions. Additionally, the implementation includes performance evaluation metrics, such as Frames Per Second (FPS) monitoring, to assess the accuracy and efficiency of the face recognition model. The implementation provides a solid foundation for further exploration and development, encouraging potential enhancements and contributions from interested individuals.

TESTING

TEST CASES

The primary objective of testing is to detect and rectify errors within a project. Inspection serves as a meticulous process for scrutinizing the functionality of components, sub-assemblies, configurations, and the finalized product. This method is integral to the software development process, aiming to ascertain that the software framework not only fulfills specified requirements but also aligns seamlessly with customer expectations. The diverse types of evaluation mentioned highlight the need for tailored testing approaches. In essence, the paragraph underscores the crucial role of testing in software development, maintaining a neutral and professional tone while elucidating the fundamental principles applicable across projects.

SCREENSHOTS

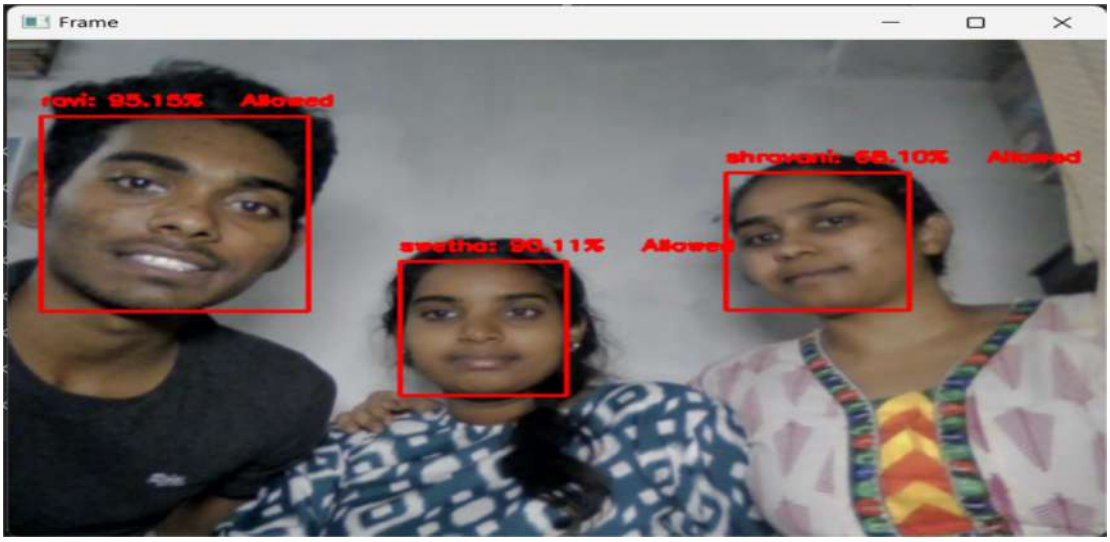

Fig : Output when the faces are recognized by the recognize_video.py module

Here, we can see that the prototype recognizes multiple subjects simultaneously and displays their name, accuracy and their authentication, i.e., if they are allowed or not.

Fig : Output when the faces are not recognized by the prototype

Here, we can the observe the type of output we get when the subjects are not recognized by our face recognition model.

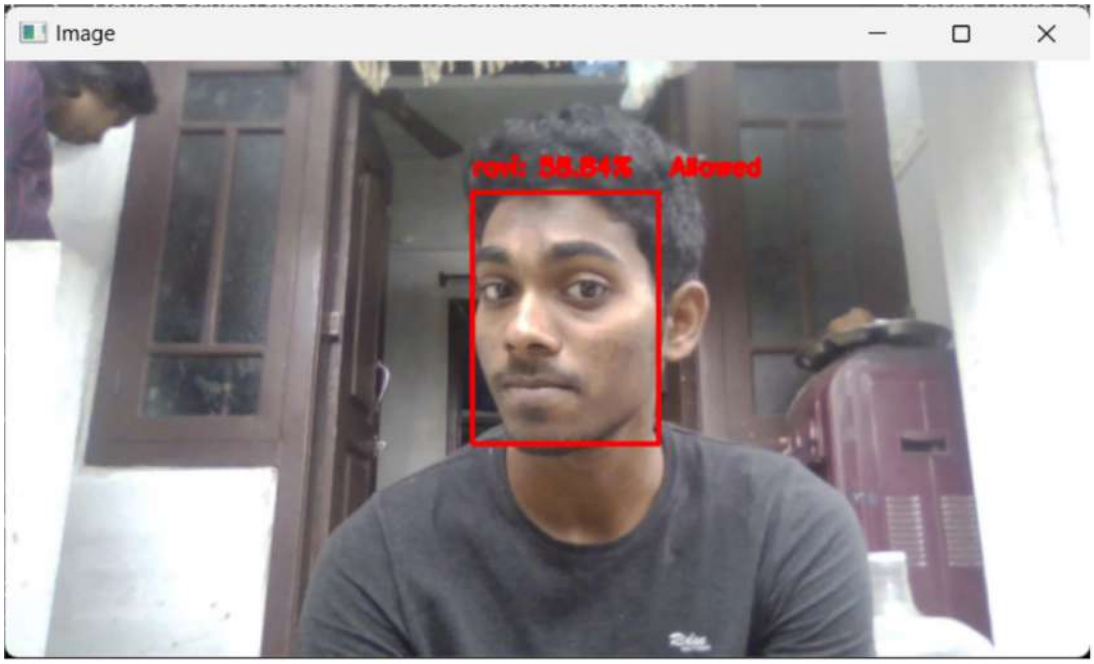

Fig : Output when we use the recognize_image.py

Additionally, we can also perform face authentication on images by using the recognize_image.py module. This is just an addition to the overall project.

CONCLUSION

In conclusion, this project serves as a pivotal step towards an advanced facial recognition system, offering a robust framework for real-world applications. By seamlessly integrating Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and k-Nearest Neighbors (k-NN), the model exhibits a nuanced understanding of facial features, achieving enhanced accuracy and efficiency. The combination of these methodologies ensures a versatile approach, making the system adaptable to diverse scenarios. However, it is crucial to acknowledge certain limitations, such as potential challenges in scalability and the need for a controlled environment for optimal performance. Moreover, this prototype lays the foundation for future developments, providing a modular architecture that encourages collaboration and additional contributions from interested researchers and developers. The integration of CNNs empowers the system to extract intricate facial features, while SVMs and k-NN contribute to robust classification and recognition. The project's deployment diagram envisions seamless integration into existing systems, highlighting its potential for widespread implementation. Despite its current prototype status, this project demonstrates significant strides towards a comprehensive facial recognition solution. As technology evolves, further enhancements and optimizations can refine its performance, ensuring it remains at the forefront of innovation in facial recognition systems. Ultimately, this project not only addresses the limitations of existing models but also establishes a groundwork for continuous improvement and adaptability in the dynamic landscape of facial recognition technology.

REFERENCES

[1] D. B. Desai and S. N. Kavitha, "Face Anti-spoofing Technique Using CNN and SVM," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 37-41, doi: 10.1109/ICCS45141.2019.9065873.

[2] K. T. Islam, R. G. Raj and A. Al-Murad, "Performance of SVM, CNN, and ANN with BoW, HOG, and Image Pixels in Face Recognition," 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE), Rajshahi, Bangladesh, 2017, pp. 1-4, doi: 10.1109/CEEE.2017.8412925.

[3] G. He and Y. Jiang, "Real-time Face Recognition using SVM, MLP and CNN," 2022 International Conference on Big Data, Information and Computer Network (BDICN), Sanya, China, 2022, pp. 762-767, doi: 10.1109/BDICN55575.2022.00149.

[4] P. R. K. Reddy and G. Uganya, "Comparative Analysis of Accuracy of Face Recognition System Using CNN and SVM," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 1-5, doi: 10.1109/ICTACS56270.2022.9988683.

[5] R. C. Damale and B. V. Pathak, "Face Recognition Based Attendance System Using Machine Learning Algorithms," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 414-419, doi: 10.1109/ICCONS.2018.8662938.

[6] S. Guo, S. Chen and Y. Li, "Face recognition based on convolutional neural network and support vector machine," 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, China, 2016, pp. 1787-1792, doi: 10.1109/ICInfA.2016.7832107.

[7] L. Agarwal, M. Mukim, H. Sharma, A. Bhandari and A. Mishra, "Face Recognition Based Smart and Robust Attendance Monitoring using Deep CNN," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2021, pp. 699-704.

[8] S. M. Zahid, T. Nashiya Najesh, S. K, S. R. Ameen and A. Ali, "A Multi Stage Approach for Object and Face Detection using CNN," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 798-803, doi: 10.1109/ICCES57224.2023.10192823.