Criminal Recognition System

An Engineering Project in Community Service

Final Report

Submitted by

Team Members List

1. 20BCY10173 Shruti Lakhara

2. 20BCY10174 Ayush Sharma

3. 20BCY10151 Sainyam Agrawal

4. 20BCY10140 Aditya Dadasaheb Lawand

5. 20BCY10220 Navaldeep Singh

6. 20BHI10001 Prakashini Srivastav

7. 20BCE10209 Aswin Shailajan

8. 20MIP10014 Anushka Mishra

in partial fulfillment of the requirements for the degree of

Bachelor of Engineering and Technology

VIT®
BHOPAL
www.vitbhopal.ac.in

VIT Bhopal University
Bhopal
Madhya Pradesh

March, 2022

Bonafide Certificate

Certified that this project report titled **"Criminal Recognition System"** is the bonafide work of "20BCY10173 Shruti Lakhara, 20BCY10174 Ayush Sharma, 20BCY10151 Sainyam Agrawal, 20BCY10140 Aditya Lawand, 20BCY10220 Navaldeep Singh, 20BHI10001 Prakashini Srivastav, 20BCE10278 Ashwin Shailagan, 20MIP10014 Anushka Mishra**"** who carried out the project work under my supervision.

This project report (Phase II) is submitted for the Project Viva-Voce examination held on 10-03-23.

Supervisor
Dr. Maragatharajan

22/05/2023

Reviewer 1
(Dr. Mayank Sharma)

22/05/23

Reviewer 2
(Dr. Suparna Patowary)

Index

1. INTRODUCTION

By integrating a facial recognition system with CCTV cameras, we can significantly enhance security measures and assist law enforcement agencies in maintaining law and order. This integration brings several benefits and capabilities to existing surveillance infrastructure, ensuring a more robust and efficient approach to public safety.Facial recognition technology employs advanced algorithms to analyze facial features and compare them to a database of known faces. It can accurately identify individuals in real-time, even within large crowds or busy environments. When integrated with CCTV cameras, this technology continuously scans the public and live feeds, looking for matches with its database of known individuals. If a match is found, the system promptly displays all relevant details related to the person, such as criminal records, aliases, or outstanding warrants.

The integration of this system with existing state police portals and the centralized "digital police" portal for India would create a unified platform for law enforcement agencies to access and share information seamlessly. By providing real-time alerts and detailed information about potential suspects or individuals of interest, authorities can respond swiftly and effectively to prevent crimes or apprehend perpetrators.One of the primary advantages of integrating facial recognition with CCTV cameras is its ability to act as a force multiplier for law enforcement personnel. It relieves the burden of manually monitoring numerous surveillance feeds by automating the identification process. This frees up human resources, allowing officers to focus on critical tasks, proactive policing, and community engagement.

Moreover, the facial recognition system can assist in criminal investigations by providing valuable leads and evidence. When incidents occur, law enforcement can review CCTV footage, extract facial images of suspects, and compare them against the database. This accelerates the identification process and aids in tracking the movements and activities of individuals involved in criminal activities. Additionally, the system can contribute to the development of comprehensive criminal intelligence. By analyzing patterns, associations, and trends in the collected data, law enforcement agencies can gain insights into criminal networks, modus operandi, and potential threats. This intelligence can facilitate proactive measures, targeted operations, and a more strategic approach to crime prevention.
However, it is important to address potential concerns surrounding privacy, data protection, and ethical use of facial recognition technology. Striking the right balance between security and individual rights is crucial. Robust safeguards and regulations must be in place to ensure responsible and transparent deployment of these systems, including proper consent, data handling protocols, and strict access controls.

By integrating facial recognition technology with CCTV cameras and existing police portals, we can harness the power of AI and automation to improve security, deter criminal activities, and maintain law and order. This proactive approach not only enhances public safety but also instills a sense of confidence and trust in the community, fostering a more secure and harmonious society.

1.1 Motivation

Due to growing crime rates in India, big data for criminal records has increased more thanever. The increased database has resulted in slower detection and identification of pre-existing criminals for related crimes. In this new cyber world, India's security is more important than ever. Numerous organized criminal operations are anticipated, some of which might aim to undermine our law enforcement and defense organizations. Technology becomes of utmost importance in such a situation. Video analytics is one use of technology used in law enforcement, and video analytics contains many different parts. Finding patterns in different videos is one of the elements.

The task here is to analyze a photograph of a particular person and see its resemblance with respect to the related criminal. Now, if the aforementioned technology is applied to thousands of movies compiled from hundreds of intelligence sources, with cross-checks across all recordings, highly important information on trends in any kind of organized crime can be discovered. Therefore, a system where the details of the criminal can be observed on a descriptive basisby facial recognition will be of great use for the investigation.

1.2 Objective

In today's world, with increasing rates of crime, it can be challenging for investigators to manually gather information about a specific suspect and identify them. To streamline the investigation process, facial recognition technology can be integrated into police systems to identify individuals with a criminal record and alert law enforcement officers. Such a system will be highly useful for the investigating officers. Face recognition software to detect criminals in images and videos, noting their time & place of occurrence, and providing a history of their crimes within a specified time frame would greatly aid the investigation process. Besides saving time for other processes and formalities, it will also increase accuracy and help in detecting patterns related to certain crimes.

2 Existing Work / Literature Review

Facial recognition is an algorithm-based technique that identifies and maps the unique facial characteristics of an individual, creating a digital representation of their face. This representation is then compared to a database for potential matches.

An RTI filed in 2022 revealed that the Delhi Police is utilizing facial recognition technology (FRT). According to the police, matches with a similarity level of 80% or higher are considered positive identifications, while matches with a similarity level of 80% or lower are deemed false positives and require additional corroborative evidence. The Criminal Procedure (Identification) Act of 2022 has replaced the Identification of Prisoners Act of 1920, and the Delhi Police is currently cross-referencing photos and videos with images collected under Sections 3 and 4 of the Act.

Initially, the Delhi Police employed facial recognition for locating missing children. However, it has also been used for investigative purposes in events such as the 2020 Northeast Delhi Riots, the 2021 Red Fort Violence, and the 2022 Jahangir Puri Riots.

There are certain limitations in the existing system used by the Delhi Police:

- The selection of an 80% similarity threshold for distinguishing true and false positives lacks clarity.

- Classifying results below 80% as false positives rather than negatives suggests that the Delhi Police may further investigate matches below the threshold.

- There is a risk of targeting individuals who share family traits or belong to the same extended families or communities.

In addition, the misuse and technical imprecision of facial recognition pose challenges. The potential for "false recognition" due to technical inaccuracies and the concerns related to "Mass Surveillance" through abusive use of technology are significant issues.

Another current application of facial recognition is the "Digital Yatra" initiative, which aims to facilitate contactless passenger processing at airports. This initiative helps reduce operating costs, digitizes manual processes, enhances efficiency, improves security standards, and enhances overall system performance. The project envisions a paperless and contactless process for travelers to navigate various airport checkpoints using their facial features linked to their boarding passes. The first phase of the Digi Yatra initiative was deployed at airports in Varanasi and Bangalore in August 2022, and it is planned to be implemented in five additional airports in Pune, Vijayawada, Kolkata, Delhi, and Hyderabad by the following March.

However, Digi Yatra also has limitations, including the need to comply with data protection norms and concerns regarding the thresholds established in the Right to Privacy decision. Furthermore, while the deployment of robots or facial recognition technologies can be beneficial for marketing strategies in various industries, they do not always address the operational concerns specific to airports. There are several face recognition libraries available in Python that offer robust tools and algorithms for face detection, recognition, and analysis. Some popular libraries include OpenCV, Dlib, face_recognition, PyTorch, TensorFlow, InsightFace, and MTCNN. Each library has its own features, strengths, and limitations, so it is important to explore their documentation and choose the one that aligns with specific requirements.

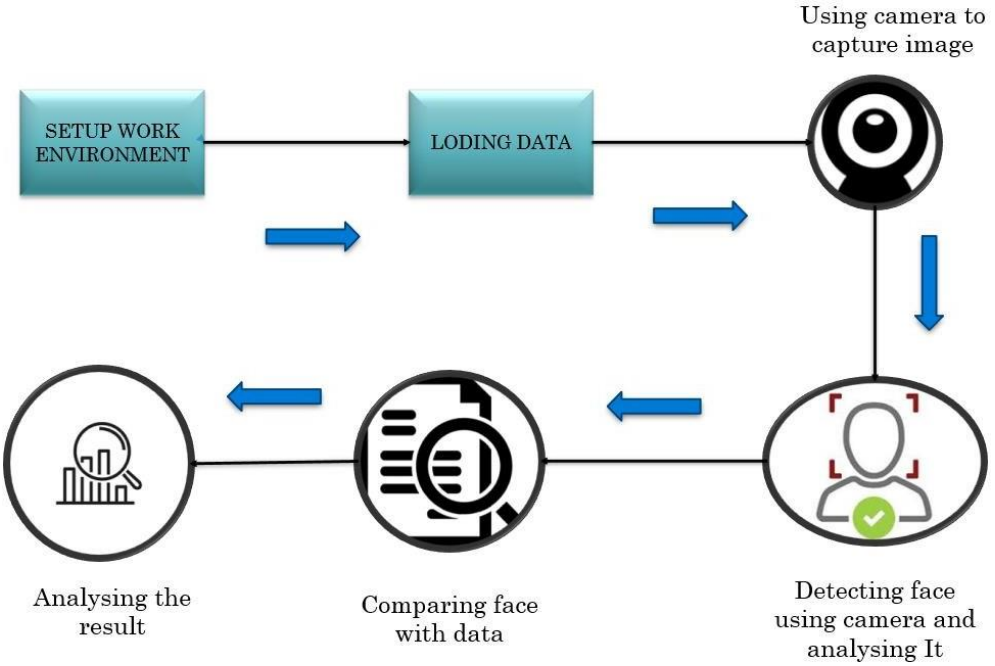
3 TOPIC OF THE WORK

3.1 System Design

The system design of the code is as follows:

Load the necessary libraries and modules: The required libraries and modules are loaded at the very start of the code.
Loading the images of known people: The images of known people (in this case, criminals) are loaded using the face_recognition library. The facial features of each person are extracted and encoded to create a universal representation of their facial features that can be used for comparison. Create arrays of known face encodings and names: The known face encodings and names are stored in separate arrays. The face encodings array contains the universal representations of the facial features of the known people, while the face names array contains their names.
Open the webcam and start video capture: The webcam is initialized and the video capture starts. The captured video frame is resized to 1/4th of its original size for faster face recognition processing.
Convert the image to RGB: The captured video frame is converted from BGR color (which is the format used by OpenCV) to RGB color (which is the format used by the face_recognition library).
Detect faces in the video frame: The face_recognition library is used to detect faces in the video frame. The locations and encodings of the detected faces are stored in separate arrays.
Comparing the detected faces with known faces: The detected faces are compared with the known faces using the compare_faces() function

3.2 Proposed work flow Diagram



3.3 Libraries Used:

Open CV

OpenCV (Open-Source Computer Vision Library) is a powerful and widely-used open-source computer vision and machine learning software library. It provides a vast array of tools and functions that enable developers and researchers to manipulate and analyze visual data, making it a popular choice for various computer vision applications. With OpenCV, developers can perform a wide range of tasks, such as image and video processing, object detection and tracking, facial recognition, augmented reality, and more. It supports multiple programming languages, including C++, Python, Java, and MATLAB, making it accessible to a large community of developers. One of the key strengths of OpenCV is its extensive collection of algorithms and functions. It offers a comprehensive set of image and video processing tools, including filters, edge detection, image segmentation, and feature extraction. These tools enable developers to preprocess images and videos, extract meaningful information, and enhance visual data for further analysis.

OpenCV also provides a range of machine learning algorithms and models, allowing developers to train and deploy computer vision models for tasks such as object recognition, face detection, and gesture recognition. It integrates seamlessly with popular machine learning frameworks like TensorFlow and PyTorch, facilitating the development of complex computer vision applications.

Face Recognition

The face_recognition library is a popular open-source Python package used for face recognition and facial feature extraction tasks. It is built on top of dlib, a robust C++ library known for its excellent performance in computer vision applications. The face recognition library provides a high-level and user-friendly interface to perform face-related tasks with ease.

One of the main features of the face recognition library is its ability to perform face recognition on images and videos. It uses deep learning algorithms to create face embeddings, which are numerical representations of facial features that can be used to compare and identify faces. By comparing these embeddings, the library can determine if two faces belong to the same person or not.

The face recognition library also supports facial feature extraction, allowing developers to detect and locate various facial landmarks such as the eyes, nose, and mouth. This feature is particularly useful for tasks like facial expression analysis, gaze detection, and face morphing.

Moreover, the face recognition library is quite efficient and capable of processing images and videos in real-time. It leverages the performance optimizations of dlib and utilizes GPU acceleration to achieve fast and accurate face recognition results. Overall, the face recognition library is a valuable tool for face-related tasks in computer vision applications. Its ability to perform face recognition and facial feature extraction, along with its simplicity and efficiency, makes it a popular choice among developers and researchers working on face recognition, identity verification, and facial analysis project.

3.4 Working Principle

This is a Python code for a facial recognition system that captures video from a webcam and recognizes the faces in real time using the face_recognition library. The code first imports the required libraries such as face_recognition, OpenCV, numpy, CSV, os, and datetime. It then loads the images of the known people whose faces the system is supposed to recognize. For each person, the system creates a facial encoding using the face_recognition.face_encodings function. The system then creates two arrays, one containing the known face encodings and another containing the names of the known people. The code then opens the video capture device using OpenCV's VideoCapture function. It then starts an infinite loop where it captures a single frame

of the video feed, resizes it to 1/4th of the original size, and converts it from BGR color (used by OpenCV) to RGB color (used by face_recognition). The system then looks for faces in the current frame of the video using face_recognition.face_locations function and extracts the facial encodings using face_recognition.face_encodings function. The system then compares the facial encodings of the faces in the current frame with the known face encodings using face_recognition compare_faces function.
If a match is found, the system records the name of the person whose face matches with the known faces. If the name of the person whose face is recognized is in the list of known criminals, the system removes the person's name from the list and records the name and time in a CSV file. The system displays the video feed on the screen with the name of the person whose face is recognized overlaid on the video. The system runs in real-time until the user presses the 'q' key on the keyboard. Once the user presses the 'q' key, the system releases the video capture device and destroys all OpenCV windows. Finally, the CSV file is close. In summary, the working principle of this code is to use pre-trained facial recognition models to detect and recognize known criminals in real-time video streams, while also keeping a record of the recognized criminals' names and the times they were detected.

About dataset

In the absence of a comprehensive dataset containing images of all known criminals, a temporary solution has been adopted using a celebrity dataset. This approach involves utilizing the face_recognition library to load the images from a designated folder and subsequently encoding them. The encoded data is then stored for reference. However, it should be noted that this dataset will be replaced with the actual criminal dataset once it becomes available.

When a potential criminal is detected by the CCTV system, which utilizes a webcam, their face is captured and processed. The face_recognition library is employed to detect and encode the facial features of the individual in real-time. This encoding process generates a unique numerical representation that encapsulates the distinguishing characteristics of the person's face. The encoded facial data of the potential criminal is then compared with the previously stored encoded data of known individuals. This comparison aims to identify any matches or similarities between the two sets of encoded facial features. If a match is found, the system determines that the person detected by the CCTV system is a known individual from the criminal dataset.

Upon successfully identifying a match, the system proceeds to display the name associated with the encoded facial data. This allows security personnel or relevant authorities to quickly ascertain the identity of the individual and take appropriate action if necessary. While the current implementation utilizes a temporary dataset based on celebrities, the intention is to eventually replace it with a more comprehensive criminal dataset. This upgraded dataset will encompass images and corresponding encoded data of known criminals, facilitating more accurate and reliable identification of potential threats.

By combining the capabilities of the face_recognition library, webcam-based CCTV systems, and a robust criminal dataset, the system aims to enhance security measures by swiftly identifying and flagging known criminals when they appear in monitored areas. This technology has the potential to aid law enforcement agencies in their efforts to maintain public safety and combat crime effectively.

3.5 Results and Discussion

Due to the increasing workload and rising crime rates, it has become challenging for the police to effectively monitor and respond promptly. However, with the exponential growth of technology, we need to explore remedies for this issue. As people's concerns for their safety have escalated, surveillance cameras have been deployed in numerous locations. These cameras can serve a broader purpose and contribute to addressing this problem. To enhance the accuracy and reliability of the criminal identification system, while remaining cost-effective, we propose utilizing state-of-the-art hardware and software.

Our brainstorming sessions have resulted in a three-step approach to facial recognition:

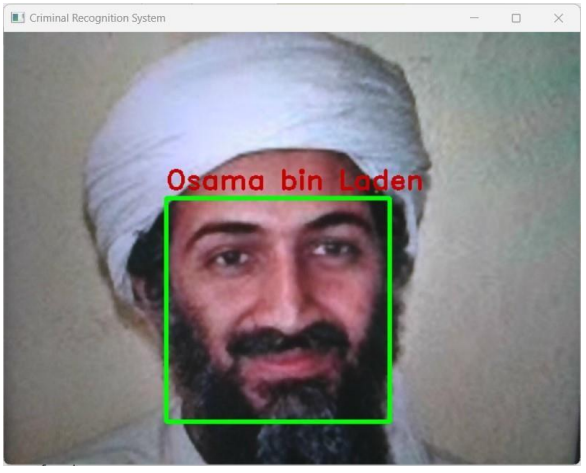
1. Face Alignment and Detection:
The initial stage involves finding faces in the input image. This can be achieved using a machine-learning system known as a Haar-Cascade classifier, which is trained on both positive and negative images. The system needs to identify the presence of faces in pictures or videos. Many cameras today come equipped with face-detection features. Social media platforms like Snapchat and Facebook also employ face identification for applying effects to user-captured images and videos.

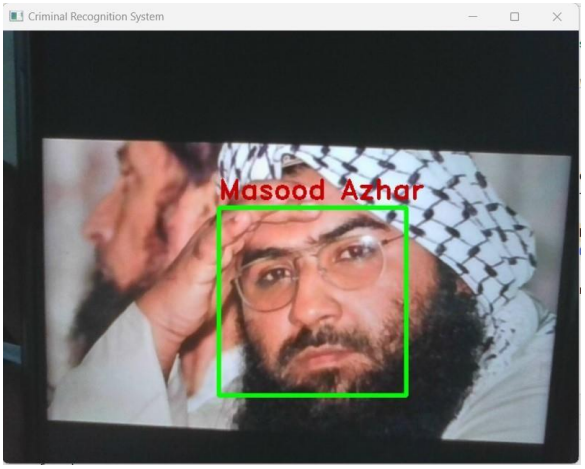
To ensure consistency with faces in the database, the identified face must undergo normalization using an algorithm. One approach is to use generic face landmarks, such as the outer corners of the eyes, the tip of the nose, and various points around the eyes and lips. A machine learning algorithm can be trained to locate these landmarks on any face and align it towards the center.

2. Feature Measurement and Extraction:
Once the faces have been identified and aligned, the next step involves extracting features from them. In this case, a Convolutional Neural Network (CNN) can be utilized. A CNN can extract high-level characteristics from an image, which can then be used to recognize faces stored in a database.

3. Face Recognition:
The final step is to match the extracted features with the faces present in the database. Typically, a Euclidean distance metric is employed to determine the similarity between two feature vectors.

By following this approach, we can significantly improve the accuracy and reliability of the criminal identification system. It combines advanced hardware and software techniques to ensure cost-effectiveness and adaptability for various applications.

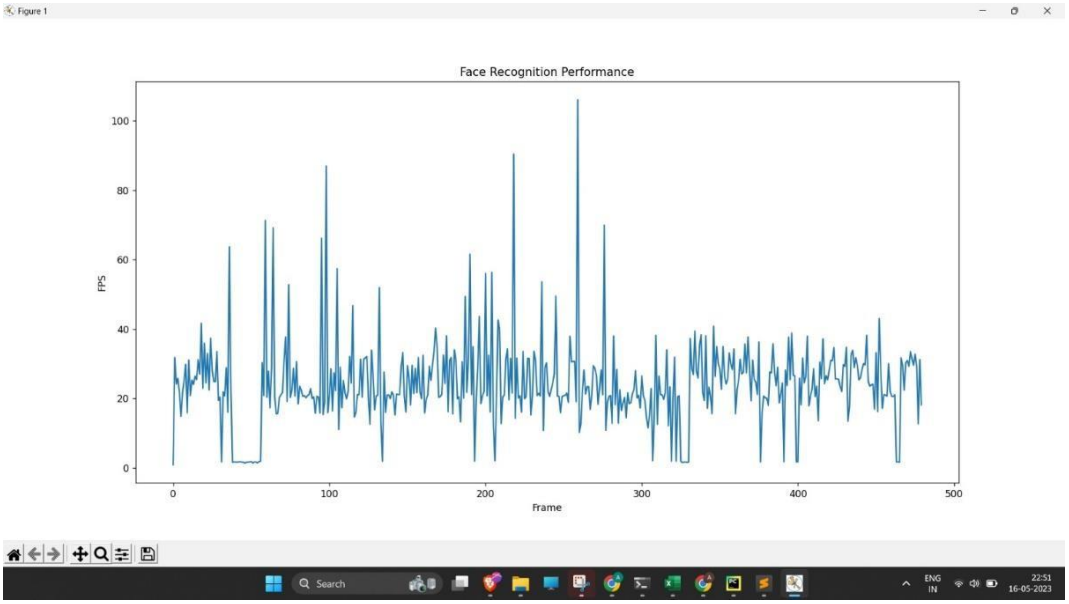




Real time facial recognition system

3.6 Performance graphs

The performance graph signifies the frames per second (FPS) value over time for the face recognition system. Each point on the graph represents the FPS value calculated for a specific frame during the execution of the code.The x-axis of the graph represents the frame number, indicating the progression of frames over time. The y-axis represents the FPS value, which indicates the number of frames processed per second. A higher FPS value indicates better performance and faster processing.By observing the graph, you can analyze the performance of the face recognition system. If the graph shows a consistent and high FPS value, it indicates that the system is processing frames efficiently and in real-time. On the other hand, if the graph shows fluctuations or a low and inconsistent FPS value, it suggests that the system may be struggling to process frames quickly, which could impact the real-time performance.The graph provides a visual representation of the system's performance, allowing you to monitor and evaluate its efficiency and effectiveness over time.



FPS graph (Frames per second)

The performance graph generated in the code signifies the execution time of the face recognition process for each frame of the video stream.

Here's what the graph represents:

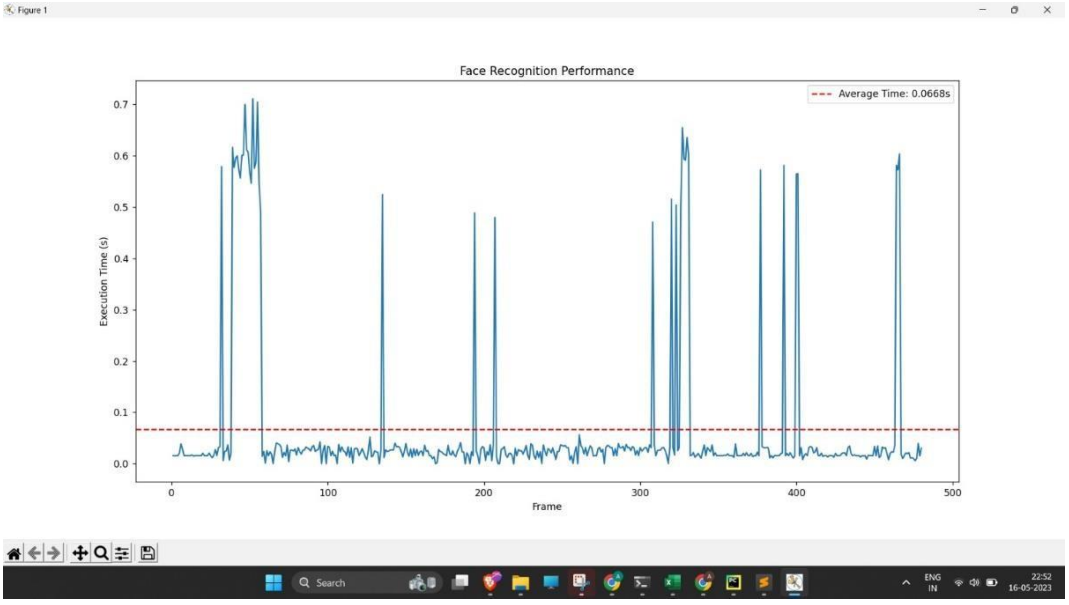
- X-axis: The frame number or index, indicating the sequential order of the frames processed.

- Y-axis: The execution time in seconds for each frame, representing how long it took to perform face recognition on that particular frame.

The graph provides a visual representation of the performance of the face recognition system over time. By analyzing the graph, you can observe the following:

- Execution Time per Frame: Each point on the graph represents the execution time for a specific frame. The higher the point on the graph, the longer it took to process that frame. You can observe variations in execution time, which can indicate fluctuations in the complexity of the frames or processing load.

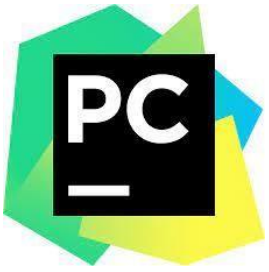
- Average Execution Time: The red dashed line on the graph represents the average execution time per frame across all processed frames. This line helps visualize the overall average performance of the face recognition system. By examining the graph, you can gain insights into the system's performance, identify potential performance bottlenecks, and assess the overall efficiency of the face recognition algorithm on the video stream.



Execution time graph

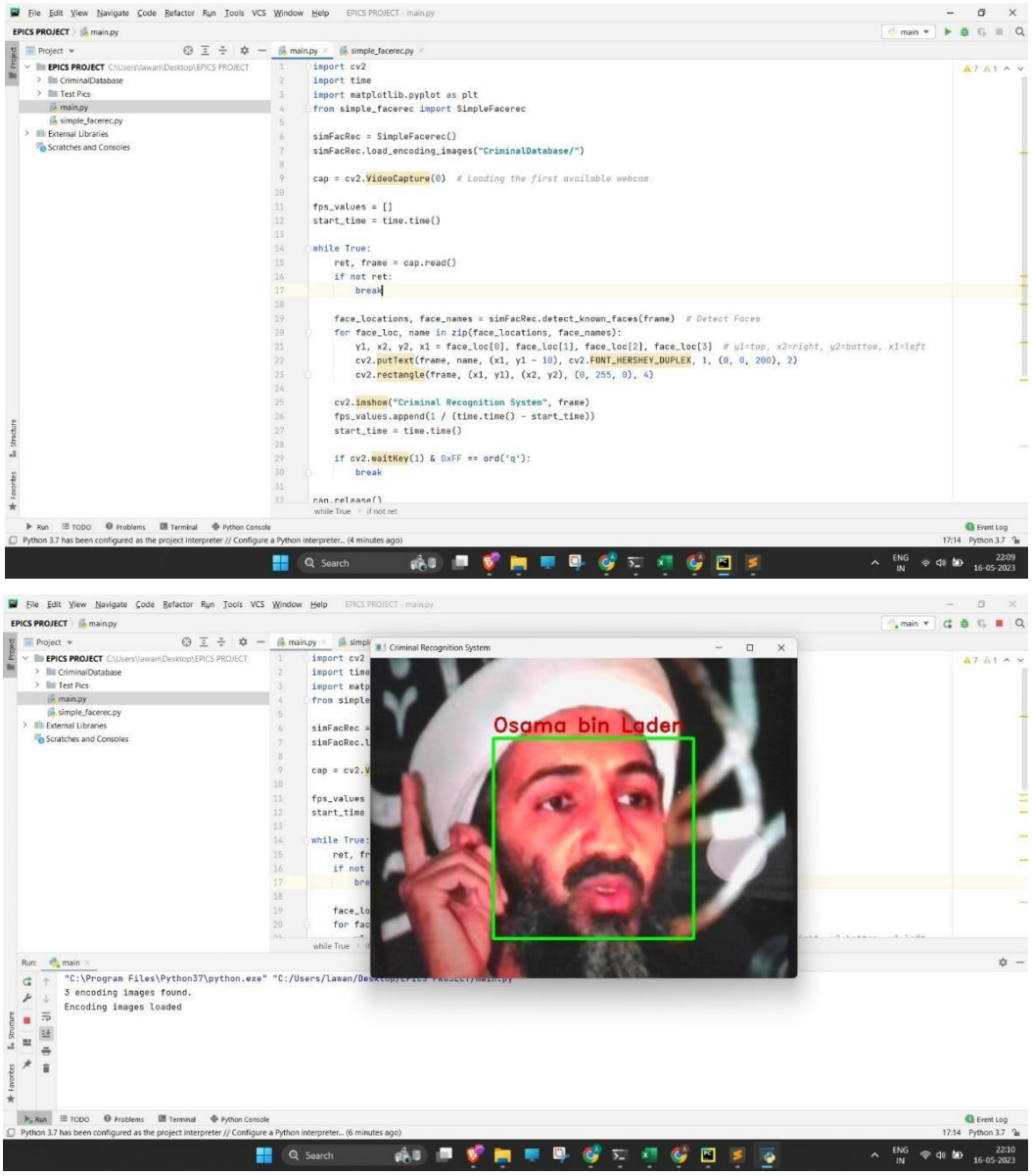
3.7 Simulation Environment

PyCharm is a popular integrated development environment (IDE) for Python programming. While PyCharm does not have a built-in simulation environment specifically designed for running simulations, it provides a rich set of features and tools that can be utilized to create and run simulations effectively. PyCharm's main strength lies in its code editing and debugging capabilities, which are essential for simulation development. The IDE offers advanced code editing features such as code completion, code analysis, and refactoring tools, which enhance productivity and help in writing efficient simulation code. PyCharm also provides support for version control systems like Git, enabling collaborative development of simulation projects. In terms of running simulations, PyCharm integrates seamlessly with Python's extensive scientific computing ecosystem. It supports popular libraries such as NumPy, SciPy, pandas, and Matplotlib, which are widely used for simulation purposes. These libraries offer powerful data manipulation, numerical computation, and visualization capabilities, allowing you to build complex simulation models and analyze their results. PyCharm provides a built-in terminal, which allows you to execute Python scripts and interact with the simulation environment directly from the IDE. You can run your simulation code and observe the results within the terminal, making it convenient to debug and experiment with different parameters or scenarios. Furthermore, PyCharm offers a robust debugger that allows you to step through your simulation code, set breakpoints, inspect variables, and track the flow of execution. This can be invaluable for identifying and fixing issues during simulation development, ensuring the accuracy and reliability of your models.



PiCharm logo

To enhance the simulation workflow, PyCharm supports the creation of virtual environments.Virtual environments allow you to isolate your simulation project's dependencies, ensuring that different projects can use specific library versions without conflicts. This is particularly useful when working on multiple simulation projects simultaneously or collaborating with others on shared codebases. PyCharm also provides comprehensive testing capabilities



Picharm simulation environment

While PyCharm does not have a dedicated simulation environment, its powerful features, seamless integration with scientific libraries, debugging capabilities, and support for virtual environments make it a versatile and efficient tool for simulation development in Python. With PyCharm's comprehensive set of tools, you can create, debug, and analyze simulations effectively, ultimately improving your productivity and the quality of your simulation projects.

3.8 Individual Contribution By Members

Shruti Lakhara (20BCY10173)

Shruti Lakhara played a crucial role in a project focused on criminal facial recognition systems. With her strong research skills and dedication, she conducted in-depth research on various aspects related to these systems. One of Shruti's primary contributions was her extensive investigation into the history of criminal facial recognition systems. She delved into the development of these systems, tracing their evolution from early prototypes to the advanced technologies available today. By understanding the historical context, she gained valuable insights into the progression of facial recognition technology and its applications in the criminal justice field. Furthermore, Shruti thoroughly examined the current status of criminal facial recognition systems. She explored their implementation in law enforcement agencies, analyzing their effectiveness and potential limitations. Her research involved studying real-world case studies, examining the impact of these systems on crime prevention, investigation, and identification of suspects. This analysis provided a comprehensive understanding of the practical implications of using facial recognition technology in criminal justice settings. In addition to her research contributions, Shruti actively participated in the decision-making process for the project's roadmap.

Ayush Sharma (20BCY10174)

One of Ayush's notable contributions was his extensive knowledge of Python libraries. He leveraged this knowledge to identify and utilize the most suitable libraries for the project, enabling the team to leverage existing functionalities and streamline the development process. Ayush's understanding of library structures allowed him to navigate through complex documentation, extract relevant information, and apply it effectively in the project. Ayush also demonstrated his proficiency in optimizing code. Through his deep understanding of Python, he actively sought ways to enhance the code's performance and reduce its overall complexity. Ayush's expertise in code optimization techniques enabled him to identify areas for improvement, leading to more efficient execution and resource utilization. Furthermore, Ayush's collaboration with Aditya was marked by effective communication and synergy. Together, they shared their knowledge, ideas, and challenges, fostering a productive environment that facilitated creative problem-solving and efficient development. Ayush's willingness to collaborate and his ability to effectively communicate technical concepts significantly contributed to the team's success. In summary, Ayush's Python proficiency, extensive knowledge of libraries, and optimization skills played a pivotal role in the project's development. His contributions not only helped overcome challenges but also led to a more streamlined and efficient codebase. Ayush's collaboration with Aditya brought together a formidable team that combined their expertise to deliver a successful project.

Sainyam Agrawal (20BCY10151)

Sainyam Agrawal (20BCY10151) played a vital role in the project by taking charge of writing the report and contributing to the brainstorming sessions. As a skilled writer, Sainyam was responsible for creating a well-structured and coherent narrative for the project. One of Sainyam's primary responsibilities was to draft the report, beginning with the creation of an outline. This involved identifying the key sections and subsections that would form the structure of the report. By carefully organizing the content, Sainyam ensured that the information flowed logically and smoothly, allowing readers to grasp the project's concepts and findings effectively. Sainyam was also tasked with writing the introduction, conclusion, and other critical sections of the report. The introduction aimed to provide a comprehensive overview of the project, setting the stage for the subsequent sections. It captured the reader's attention, outlined the objectives of the project, and established the context and importance of criminal facial recognition systems.

In the conclusion, Sainyam summarized the key findings and insights derived from the research. This section highlighted the project's contributions, its implications, and potential future directions. Sainyam ensured that the conclusion effectively conveyed the project's significance and provided a concise recapitulation of the main points discussed throughout the report. In addition to these key sections, Sainyam's writing skills were employed throughout the report, ensuring clarity and coherence in the content. This involved crafting informative and engaging paragraphs, synthesizing information from various sources, and maintaining a consistent tone and style. Sainyam's ability to communicate complex concepts in a clear and concise manner greatly contributed to the overall readability and effectiveness of the report.

4. Aditya Dadasaheb Lawand (20BCY10140)

Aditya is an exceptionally talented coder who possesses an extensive knowledge of various Python modules and their underlying principles. From the very beginning, he enthusiastically engaged in the project, leveraging his programming skills to contribute significantly. With the aid of tools like PyCharm, Aditya meticulously constructed the code, ensuring its completion within the specified timeframe and in accordance with the project requirements. Aditya's expertise extended beyond mere coding proficiency. He consistently demonstrated his dedication by actively participating in discussions and providing valuable insights to enhance the efficiency of the code. His suggestions proved invaluable, offering innovative approaches and techniques to optimize the project's overall performance. One notable contribution by Aditya was the integration of a user-defined Python module. This module encompassed functionalities for image scanning and encoding, seamlessly incorporating it into the main Python file housing the project's core functionality. This integration showcased Aditya's ability to design and incorporate modular code structures, enhancing the overall organization and maintainability of the project. Throughout the development process, Aditya's diligence and competence shone through, ensuring the project's success. His commitment to delivering high-quality code and his consistent involvement in refining the project's implementation underscored his invaluable contributions. Aditya's knowledge of Python modules, his proficiency with coding tools, and his proactive approach to problem-solving made him an invaluable asset to the project team.

Navaldeep Singh (20BCY10220)

Navaldeep's insights and expertise helped shape the direction of the project, ensuring that it aligned with the goals and objectives set forth by the team. He provided valuable input based on his research findings, highlighting the key areas that needed to be addressed and identifying potential areas of innovation and improvement. Moreover, Navaldeep played a significant role in researching and analyzing the novelty of the project. He examined existing facial recognition systems and identified gaps and opportunities for innovation. His thorough understanding of the subject matter enabled him to propose novel approaches and techniques that could enhance the accuracy, reliability, and ethical considerations of criminal facial recognition systems. His research findings were instrumental in identifying potential areas of improvement and ensuring that the project incorporated cutting-edge advancements in the field. Navaldeep's dedication to research and his ability to translate his findings into actionable insights greatly contributed to the overall success of the project. In addition to his research contributions, Navaldeep also conducted an extensive analysis of existing works in the field of facial recognition systems. He studied relevant literature, examined previous projects, and explored case studies to gain a comprehensive understanding of the current state of the art. This analysis provided valuable insights into the strengths and limitations of existing solutions, allowing Navaldeep to propose innovative ideas that pushed the boundaries of what had been accomplished thus far. By building upon the existing works, Navaldeep ensured that the project leveraged the best practices and advancements in the field, resulting in a more robust and impactful solution.

Prakashini Srivastav (20BHI10001)

Prakashini Srivastav (20BHI10001) made significant contributions to the project by focusing on the technological aspects of the facial detection system and participating in content creation and compilation for the presentation. Additionally, she conducted thorough research on the history, development, and current status of criminal facial recognition systems. Prakashini's expertise in technology and Python libraries played a crucial role in the project. She worked on identifying and implementing appropriate technologies and Python libraries for the codebase of the facial detection system. Her knowledge and skills in programming and software development were instrumental in ensuring the efficient and accurate functioning of the system. Prakashini's contributions contributed to the project's technical foundations and enhanced its capabilities in facial detection. Furthermore, Prakashini played a role in content creation and compilation for the project's presentation. She worked closely with the team to gather relevant information and data, ensuring that the presentation effectively communicated the key aspects of the project. Prakashini's attention to detail and ability to synthesize complex information into clear and concise content contributed to the overall cohesiveness and impact of the presentation. In addition to content creation, Prakashini conducted in-depth research on the history, development, and current status of criminal facial recognition systems. Her research involved investigating the evolution of facial recognition technology, including its origins, advancements, and applications in the criminal justice domain. By gaining a comprehensive understanding of the subject, Prakashini contributed valuable insights into the project's context and provided a solid foundation for addressing the system's challenges and potential improvements. Prakashini's research also involved exploring the current status of criminal facial recognition systems. She delved into the implementation of these systems in law enforcement agencies, analyzing their effectiveness, limitations, and ethical considerations. Her research findings helped the team gain a comprehensive understanding of the practical implications and potential ramifications of using facial recognition technology in the criminal justice field.

Aswin Shailajan (20BCE10209)

Aswin's auditing skills were invaluable to the team's development process. He dedicatedly reviewed the complete codebase, meticulously searching for bugs, errors, and potential pitfalls. His attention to detail allowed him to identify and address issues that could have potentially hindered the project's functionality or caused unexpected behavior. Aswin's thoroughness and commitment to quality assurance ensured that the code was robust, reliable, and ready for deployment. Moreover, Aswin's contributions extended beyond code auditing. He actively participated in project discussions and idea generation sessions, providing insightful suggestions and innovative ideas that influenced the project's direction. His broad perspective and creative thinking helped shape the roadmap and ensure that the project aligned with the team's goals and objectives. By actively engaging in code review and contributing to project planning, Aswin demonstrated his dedication to excellence and his commitment to the team's success. His involvement fostered an environment of collaboration and continuous improvement, where his expertise and insights were valued by the entire team. Aswin's multifaceted contributions enhanced the overall quality of the project and significantly contributed to its success.

Anushka Mishra (20MIP10014)

Anushka Mishra (20MIP10014) played a vital role in the project by taking charge of preparing a visually appealing and engaging presentation. Furthermore, Anushka assisted in working on the roadmap of the project. Anushka's expertise in presentation design and her eye for aesthetics enabled her to create a visually appealing and engaging presentation. She collaborated with the team to gather the necessary information and content, organizing it in a structured and impactful manner. Anushka ensured that the presentation effectively communicated the key points of the model, capturing the audience's attention and facilitating their understanding of the project. Her design skills and attention to detail played a crucial role in creating a visually engaging presentation that enhanced the overall effectiveness of the project. Additionally, Anushka played

a role in working on the roadmap of the project. She actively participated in discussions and brainstorming sessions, contributing valuable insights and ideas to shape the direction of the project. Anushka's input helped identify key milestones, prioritise tasks, and allocate resources effectively. Her contributions to the roadmap ensured that the project progressed in a structured and goal-oriented manner.

4. CONCLUSION

In this project, the team successfully developed a Criminal Facial Recognition System that holds immense potential in assisting law enforcement agencies in detecting criminals when they come into view of a camera. The system utilizes advanced algorithms and techniques to analyze facial features and match them against a database of known criminals, enabling law enforcement to track and keep records of criminal activities more effectively. The implementation of this system offers significant advantages, as it can help identify individuals with criminal records in real-time, aiding in the prevention and investigation of crimes. By quickly identifying and tracking potential threats, law enforcement agencies can take prompt action, enhancing public safety and security.

However, it's important to acknowledge that our project's scope has certain limitations in terms of usability. To deploy the Criminal Facial Recognition System in real-world scenarios, additional resources and sophisticated hardware infrastructure would be required. Advanced CCTV cameras capable of capturing high-resolution images, along with powerful computing systems, would ensure optimal performance and accuracy. Incorporating additional variables and data sources, such as incorporating behavioral patterns or integrating with criminal databases, can enhance the system's effectiveness. Exploring alternative algorithms and techniques, such as deep learning or hybrid models, may also lead to better accuracy and robustness. Furthermore, collaboration with law enforcement agencies and experts in the field can provide valuable insights and feedback for refining the system. Conducting rigorous testing and evaluation in real-world scenarios will help identify any limitations or biases and facilitate ongoing improvements. Ultimately, while the Criminal Facial Recognition System developed in this project showcases significant potential, it is important to consider the necessary advancements in hardware and continuous development efforts to maximize its effectiveness and applicability in real-world law enforcement applications.

5. REFERENCE

https://www.researchgate.net/publication/353295394_Criminal_Identification_System_using_Facial_Recognition

wikipedia.org

https://www.drishtiias.com/daily-updates/daily-news-analysis/digiyatra

https://internetfreedom.in/dangers-of-digiyatra/