International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Criminal Tracking Using Deep Learning for Face Recognition in Surveillance Systems

Siyad. S [1], Mr. T. Thiyagarajan [2]

[1] *MCA, Department of Computer Application, Nehru Institute of Information Technology and Management, Coimbatore, Tamilnadu, India*

[2] *Assistant Professor, Department of Computer Application, Nehru Institute of Information Technology and Management, Coimbatore, Tamilnadu, India*

Abstract - There is an abnormal increase in the crime rate and also the number of criminals are increasing, this leads towards a great concern about the security issues. Crime preventions and criminal identification are the primary issues before the police personnel, since property and lives protection are the basic concerns of the police but to combat the crime, the availability of police personnel is limited. With the advent of security technology, cameras especially CCTV have been installed in many public and private areas to provide surveillance activities. The footage of the CCTV can be used to identify suspects on scene. This Real time criminal identification system based on face recognition works with a fully automated facial recognition system. Haar feature-based cascade classifier and OpenCV LBPH (Local Binary Pattern Histograms) Algorithms are used for Face detection and recognition. This system will be able to detect face and recognize face automatically in real time. An accurate location of the face is still a challenging task. This Framework has been widely used by researchers in order to detect the location of faces and objects in a given image. Face detection classifiers are shared by public communities, such as OpenCV. The automatically tagging feature adds a new dimension to sharing pictures among the people who are in the picture and also gives the idea to other people about who the person is in the image. In our project, we have studied and implemented a pretty simple but very effective face detection using CNN algorithm which takes human skin color into account. Our aim, which we believe we have reached, was to develop a system that can be used by police or investigation department to recognize criminal from their faces. The method of face recognition used is fast, robust, reasonably simple and accurate with a relatively simple and easy to understand algorithms and technique.

Key Words: *Face Detection, MTCNN, Deep Learning, Real-Time Surveillance, Criminal Identification, CNN*

1.INTRODUCTION

With the increasing number of criminal activities such as robbery, vandalism, assault, and kidnapping, public safety has become a major concern. Traditional methods of criminal identification like eyewitness reports, fingerprint scanning, and iris recognition are often time-consuming and limited by human error. Facial recognition has emerged as a promising biometric identification method due to its non-intrusive nature and efficiency. The human face plays a central role in social interactions, making facial features ideal for identity verification. This project focuses on developing a real-time facial recognition system that uses pre-stored criminal images to match and identify suspects from surveillance footage. The system employs the Haar cascade classifier and advanced deep learning algorithms, such as custom CNNs, to detect and recognize faces with high accuracy. When a face captured by a CCTV camera matches a record in the database with at least 95% similarity, the system immediately alerts the authorities, thus assisting law enforcement in faster response and investigation.

The proposed system, named "Spot Crime," is designed as a web-based application to automate surveillance and aid the police in detecting criminal activity in real time. It continuously monitors live video feeds using deep learning models trained to classify suspicious human behaviours. By combining behaviour analysis with face recognition technology, Spot Crime reduces the dependency on manual monitoring and enhances the effectiveness of crime prevention. In addition to criminal identification, face recognition can be extended to various domains such as airport security, ATM authentication, and access control. The project lays the foundation for future smart environments, where facial recognition technology can be seamlessly integrated into everyday systems to improve security and convenience.

The growing reliance on surveillance systems in both public and private sectors highlights the urgent need for intelligent monitoring solutions. Spot Crime addresses this by integrating face detection with real-time alert mechanisms, ensuring that potential threats are identified promptly and accurately. Unlike traditional systems that require constant human supervision, this automated approach not only reduces manpower requirements but also minimizes delays in identifying and responding to criminal activities. The system's ability to learn and improve through deep learning models allows for adaptive performance across different environments and lighting conditions. By bridging the gap between technology and public safety, this project contributes to building smarter, safer communities through innovation in artificial intelligence and biometric recognition.

In a time where security threats are rapidly evolving, integrating artificial intelligence with surveillance offers a proactive solution to crime prevention. The Spot Crime system not only enhances the efficiency of law enforcement but also builds a strong technological foundation for future advancements in public safety. By automating facial recognition and behavior analysis, it significantly reduces human error and enables quicker decision-making. This intelligent system demonstrates how deep learning and computer vision can be effectively utilized to support modern policing efforts, ensuring that communities remain safer and better protected through innovative and scalable technology. It also opens new possibilities for smart city development, where automated systems actively contribute to maintaining law and order. The continued refinement of such systems will play a crucial role in reshaping the future of surveillance and public security.

2. RELATED WORK

Face detection and recognition have been active areas of research for decades, particularly in the domains of computer vision, biometrics, and security systems. Numerous methods have

been proposed to accurately detect human faces in images or videos, ranging from classical machine learning techniques to modern deep learning-based approaches. Early face detection techniques relied on handcrafted features and traditional classifiers. The most notable among them is the Viola-Jones algorithm [1], which utilizes Haar-like features and AdaBoost for real-time object detection. While this method is computationally efficient and widely used in early surveillance systems, its performance degrades under complex conditions such as varying lighting, occlusion, or non-frontal face poses.

With the rise of deep learning, convolutional neural networks (CNNs) have significantly improved face detection accuracy. DeepFace [2] and FaceNet [3] introduced deep architectures capable of learning high-level facial representations, leading to breakthroughs in face recognition tasks. However, these systems typically require high computational power and are not optimized for real-time deployment in low-resource environments. Multi-task Cascaded Convolutional Networks (MTCNN), proposed by Zhang et al. [4], brought a major advancement by combining face detection and alignment in a single framework. MTCNN consists of three cascaded CNNs—P-Net, R-Net, and O-Net—each responsible for refining the face detection process. The framework achieves high accuracy and low latency, making it suitable for real-time surveillance applications.

Several works have attempted to integrate MTCNN with real-time CCTV surveillance for security purposes. For instance, Li et al. [5] applied MTCNN for automatic attendance systems, while others explored its use in smart city monitoring [6]. Despite these applications, limited research has been conducted on leveraging MTCNN specifically for criminal identification using facial databases and real-time video streams. Additionally, criminal identification often involves more than just face detection; it requires accurate classification and matching with stored records. Systems integrating face recognition with law enforcement databases, such as India's Automated Facial Recognition System (AFRS) [7], demonstrate the growing importance of biometrics in public safety. However, these systems often lack real-time processing capabilities or rely heavily on manual intervention.

In contrast to previous methods, the proposed system "Spot Crime" aims to address these limitations by utilizing MTCNN for real-time face detection from surveillance feeds, followed by matching detected faces with a criminal database. The system enhances detection robustness while maintaining low computational overhead, making it well-suited for real-time law enforcement applications.

3. SYSTEM OVERVIEW

The proposed face recognition-based criminal detection system is designed to operate in real-time using live video feeds from surveillance cameras. The system leverages deep learning techniques to detect and recognize faces, generating instant alerts when a match is found against a criminal database. It is composed of three main modules: Face Detection, Face Recognition, and Alert Generation.

1. Face Detection with MTCNN:
The Multi-task Cascaded Convolutional Network (MTCNN) is employed for efficient and accurate detection of faces at various scales. It uses an image pyramid and a cascade of three CNNs—P-Net, R-Net, and O-Net—to progressively refine face localization. Non-Maximum Suppression (NMS) is applied at each stage to eliminate redundant detections. MTCNN also provides facial landmarks for accurate face alignment.

2. Criminal Face Recognition using CNN:
Once faces are detected, they are preprocessed through normalization, resizing, and padding before being passed to a trained Convolutional Neural Network (CNN). The recognition model, based on architectures such as ResNet or VGG, classifies the faces by comparing them to a criminal face database. The system supports flexible training and retraining of the CNN model using labeled images of known criminals.

3. Real-Time Alert and Notification System:
When a criminal face is positively identified, the system triggers alerts in real time. Notifications can be delivered through multiple channels such as sound alarms, SMS, emails, and integrated dashboards. Real-time communication protocols like MQTT or WebSockets ensure quick and reliable delivery of alerts. Alerts contain metadata such as the timestamp and location of detection, and are transmitted to appropriate law enforcement or security personnel for prompt action.

Workflow Summary:
- Live video stream is captured from surveillance cameras.
- Each frame is processed by MTCNN for face detection.
- Detected faces are preprocessed and fed into the CNN for recognition.
- If a match is found in the criminal database, alerts are instantly generated and transmitted.

Technologies and Tools Used:
- MTCNN: for face detection
- CNN (ResNet, VGG): for face recognition
- TensorFlow/PyTorch: for model training and inference
- OpenCV: for video and image processing
- Flask/Django: for backend services and alert management
- MQTT/WebSockets: for real-time alert delivery

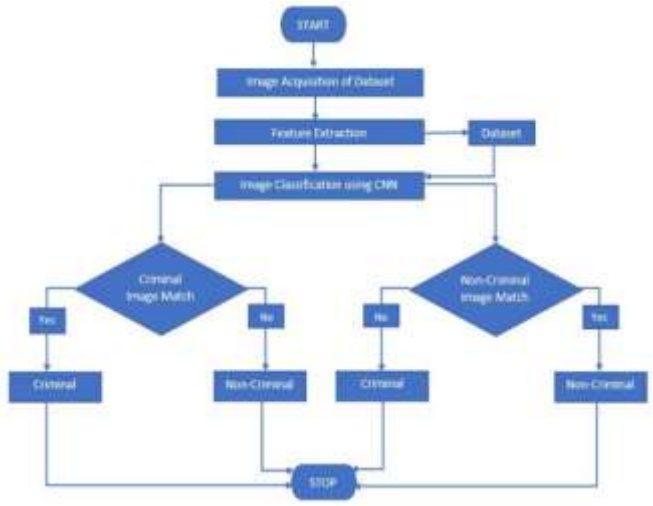

Fig -1: System design

The system enhances public safety by providing a scalable, automated solution for monitoring and identifying potential threats in public spaces. Its modular architecture ensures adaptability to different deployment environments and facilitates integration with existing surveillance infrastructure.

3.1 Module Description

1. Image Preprocessing

In this module aimed at improving face recognition speed, several key features are extracted to optimize the process. Initially, the module crops and resizes face images to smaller pixel values, reducing computational load while preserving essential facial details. However, disturbances within the images, such as noise or lighting variations, pose challenges during model training, leading to inaccurate histograms. To mitigate this issue, the module integrates techniques for noise reduction, illumination normalization, and image enhancement. These techniques help minimize the impact of disturbances, ensuring the accuracy of feature extraction and improving the overall efficiency of the face recognition system. By optimizing preprocessing features, the system can achieve faster recognition speeds without compromising accuracy, even in the presence of image disturbances.

2. Image Segmentation

In this module, the primary focus is on extracting various facial features crucial for identifying the criminal. Grayscale images obtained from this step serve as the foundation for accurate feature extraction. Techniques such as edge detection, region growing, or clustering algorithms are applied to delineate facial regions and isolate key features such as eyes, nose, mouth, and facial contours. These extracted features play a pivotal role in subsequent stages of criminal identification. By segmenting the facial images and extracting relevant features, the system enhances its ability to discriminate between individuals and effectively identify potential criminals, contributing to improved law enforcement outcomes.

3. Criminal Identification

In the criminal identification module, the extracted facial images undergo further processing using the dlib library and deep learning algorithms to precisely identify faces. Leveraging advanced techniques, the system isolates the person's face from the background, ensuring that only the facial features are captured without any interference from the surrounding environment. By employing the dlib library and deep learning algorithms, the system achieves high accuracy in face detection and minimizes the risk of false positives.

4. Database and Dataset

It is a tool for collecting and organizing information. Databases and datasets can store information about people, crime, law, or cases else. Many pictures of criminals along with their identities are stored in the database and dataset.

5. Image Matching

In the image matching module, the resulting images obtained from the facial identification process are compared with existing images stored in the dataset. Utilizing advanced matching algorithms, the system conducts a thorough comparison to determine if a match exists between the extracted facial features and those in the dataset. If a match is found, indicating that the recognized person corresponds to an individual within the dataset, relevant data associated with that image, such as criminal records or personal information, is retrieved and returned. Conversely, if no match is found, the recognized person is deemed not to be a criminal, and further action may be taken accordingly. This module plays a crucial role in the criminal identification process, facilitating the swift and accurate retrieval of pertinent information to aid law enforcement agencies in their investigations

6. Warning And Alert

In this module, upon detection of a criminal, the system triggers an alert message to notify the relevant personnel or authorities. This alert message includes the captured images of the identified individual, along with detailed information extracted from the database regarding the criminal's identity and associated records. By promptly notifying

the appropriate parties, law enforcement agencies or security personnel can take immediate action to address the situation. This proactive approach enhances public safety and facilitates timely intervention in potential criminal activities, leveraging the capabilities of the application to effectively deter and mitigate security threats.

4. METHODOLOGY

This section outlines the architecture and workflow of the Multi-task Cascaded Convolutional Network (MTCNN) for real-time face detection. The model comprises three stages, each utilizing a specific convolutional network: P-Net, R-Net, and O-Net. Each stage progressively refines the detection of faces, ultimately returning accurate bounding boxes and facial landmarks.

4.1: Proposal Network (P-Net)

The process begins by passing an input image into the system. To ensure the detection of faces at various scales, an image pyramid is constructed. This involves creating multiple scaled versions of the input image. A sliding window approach using a 12×12 kernel scans each scaled image with a stride of 2 pixels. The P-Net evaluates each region to detect potential face locations and returns a set of bounding box coordinates along with a confidence score

To optimize computation without compromising detection accuracy, regions with low confidence scores are discarded. The remaining bounding boxes are then converted to the coordinate system of the original image. As multiple overlapping detections may exist, Non-Maximum Suppression (NMS) is applied twice — first within each scale and then across all scales — to eliminate redundant bounding boxes while preserving those with the highest confidence.

Bounding boxes are then resized to ensure a consistent square shape, which simplifies subsequent processing and helps standardize input for the next stage.

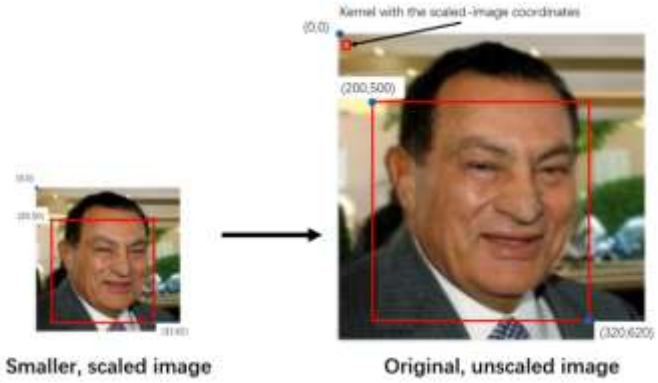

Fig -2: P-net sample

4.2 Stage 2: Refine Network (R-Net)

Each bounding box from Stage 1 is extracted, and if it lies partially outside the image boundaries, padding is applied to fill missing regions with zeroes. The padded image patches are resized to 24×24 pixels and normalized to values between -1 and 1.

These normalized patches are then fed into R-Net, which outputs refined bounding box coordinates and confidence scores. Again, bounding boxes with low confidence are eliminated, and NMS is performed to further refine the results. Bounding boxes are reshaped into squares and converted back to the coordinate space of the original image, ready for Stage 3 processing.

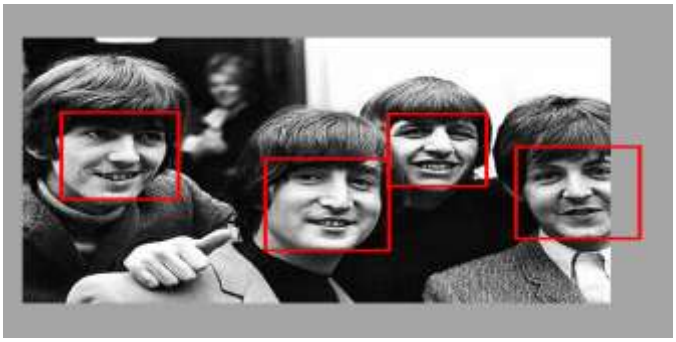

Fig -2: R-net sample

4.3 Stage 3: Output Network (O-Net)

The final stage involves inputting the processed bounding boxes into O-Net. Like R-Net, the boxes are padded if out-of-bounds, resized to 48×48 pixels, and normalized. O-Net outputs three sets of information:

1. Updated bounding box coordinates,
2. Confidence scores, and
3. Five facial landmark coordinates (left eye, right eye, nose, left mouth corner, and right mouth corner).

Boxes with low confidence scores are again discarded. The coordinates of both the bounding boxes and facial landmarks are standardized to match the original image's coordinate space. A final round of Non-Maximum Suppression is applied to ensure only the most accurate and non-overlapping detections are retained.

4.4 Result Packaging

The outputs from O-Net are structured into a final dictionary that includes:

1. box: the coordinates of each bounding box,
2. confidence: the confidence level for each detection, and
3. keypoints: the coordinates of five key facial landmarks.

This output can be easily integrated into any downstream application, such as drawing bounding boxes and landmarks on images or triggering security alerts when a face is detected.

5. REQUIREMENT SPECIFICATION

Requirement Specification is a crucial part of this project, detailing the essential hardware and software components required to develop and run the system effectively. This section also outlines the programming tools and environments utilized for implementation.

5.1 Hardware Requirements

Physical computing tools, or hardware, form the backbone of any computing system. The hardware requirements listed below are the minimum specifications needed for the smooth functioning of the proposed system:

Table – 1: Hardware requirements

Component	Specification
Processor	Intel (Minimum 2.4 GHz)
RAM	Minimum 4 GB
Main Memory	8 GB RAM
Processing Speed	Minimum 600 MHz
Hard Disk Drive	1 TB
Keyboard	Standard 104 keys
Camera	Standard HD Resolution

Note: Higher configurations may improve performance, especially in real-time image processing tasks.

5.2 Software Requirements

The software requirements define the platforms, languages, and tools required to design and run the application. The system depends on the integration of multiple tools across frontend, backend, and data processing components.

Table – 2: Software requirements

Component	Specification
Frontend	HTML, CSS, JavaScript
Backend	PHP, Python
Database	MySQL
Server Environment	XAMPP (Apache, MySQL, PHP, phpMyAdmin)
Dataset Format	CSV
Development IDE	Anaconda
Operating System	Windows 11

6. SYSTEM IMPLEMENTATION

System implementation represents the final and most critical phase of the development lifecycle, where theoretical designs are converted into an operational and functional system. It involves the integration of all software and hardware components, system configuration, and deployment into a live environment. In the context of this project—focused on real-time criminal identification through facial recognition—implementation includes configuring servers, deploying machine learning models, establishing a communication system for alerts, and setting up the supporting database infrastructure.

A successful implementation requires thorough planning, verification, and validation to ensure the system performs as intended. Each module must be rigorously tested to guarantee accuracy, responsiveness, and reliability. Equally important is training the end-users and administrators responsible for system monitoring and management.

The primary procedures involved in this system's implementation include:

- Application Development and Integration
- User Interface Deployment
- Server Configuration (Apache, MySQL)
- Model Integration and Testing
- Web-based System Recording and Testing

6.1 Equipment Installation

Anaconda Environment Setup: Anaconda is utilized as the primary environment for machine learning and computer vision tasks. It provides a pre-configured platform supporting Python and essential packages such as NumPy, OpenCV, TensorFlow, and scikit-learn. The Anaconda Prompt allows seamless installation, environment management, and execution of deep learning models for face recognition.
The system was developed using Python within the Anaconda environment. Key models, including MTCNN for face detection and CNNs (like ResNet or VGG) for recognition, were trained and tested using this setup.

Key Benefits:

- Simplified package and environment management
- Integrated support for Jupyter Notebooks and deep learning libraries
- Efficient handling of large datasets for training and recognition tasks

XAMPP for Server and Database Management: XAMPP (Cross-Platform, Apache, MariaDB/MySQL, PHP, Perl) is employed to simulate a web server environment locally. This setup supports the backend of the system, allowing interaction between the facial recognition model and the alert/notification interface.

Installation Steps:

- Download the XAMPP installer from the official site.
- Select desired components (Apache, MySQL, phpMyAdmin).
- Choose the installation directory (default: C:\xampp).
- Use the XAMPP Control Panel to manage Apache and MySQL services.

Benefits:

- Streamlines backend development and testing
- Bundled tools like phpMyAdmin for easy database interaction
- Compatible with all major operating systems

phpMyAdmin for Database Management: As part of the database implementation, phpMyAdmin is integrated to manage and manipulate MySQL databases via a user-friendly interface. It enables structured storage of user profiles, facial data, criminal records, and system logs.

Functions within the System:

- Creating tables for storing face encodings, alerts, and identities
- Importing/exporting criminal datasets for model training
- Managing user roles and access permissions
- Monitoring data integrity and usage patterns

System Workflow and Final Integration: Model Training and Deployment: CNN and MTCNN models are trained on criminal face datasets using Python and integrated into the system.

1. Camera Feed Integration: Live video is captured and processed in real-time using OpenCV.
2. Face Detection & Recognition: The MTCNN algorithm identifies faces, and CNN classifies them based on similarity with the criminal database.
3. Alert System Activation: Upon a positive match, alerts are sent via email, SMS, or dashboards using communication protocols such as WebSockets or MQTT.
4. Database Updates: Detected events are recorded in the MySQL database and visualized through the web interface.

7. SAMPLE OUTPUTS

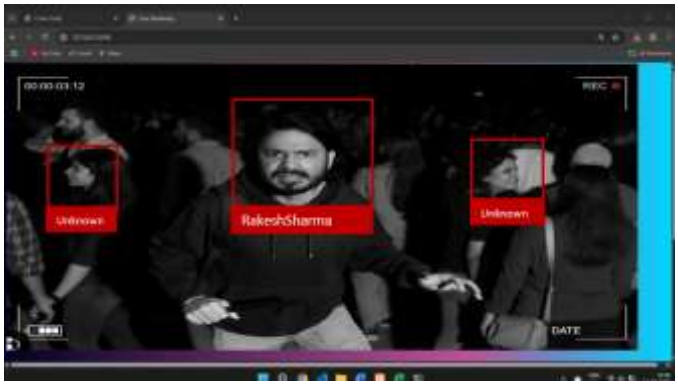

Fig -3: Live camera tracking of criminal face

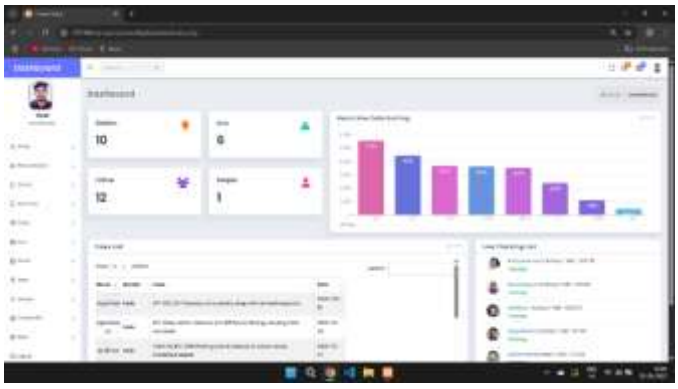
Fig -4: Live tracking admin dashboard

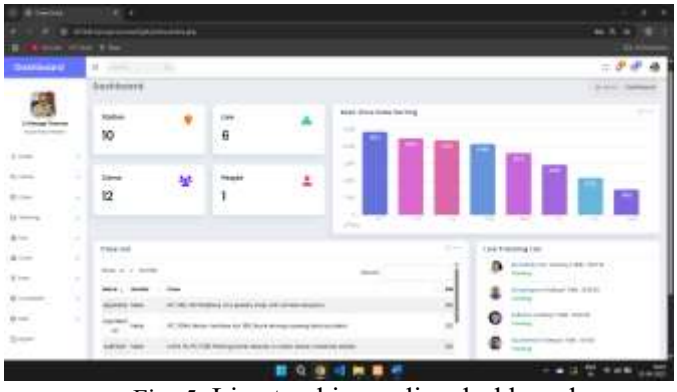
Fig -5: Live tracking police dashboard

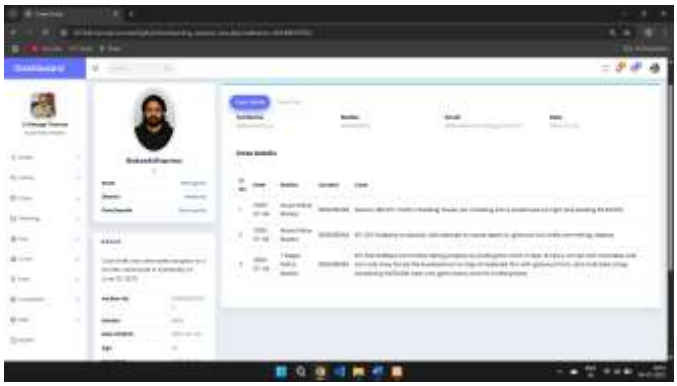

Fig -6: Criminal case records

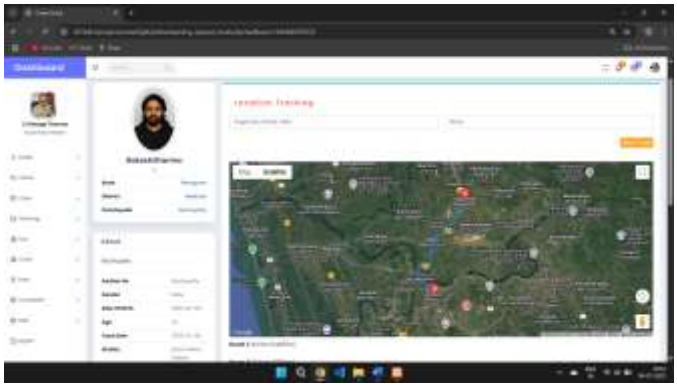

Fig -7: Criminal live location tracking

8. CONCLUSIONS

The real-time criminal identification system will help the police to control the crime rate. This app helps them in many different ways. With the advancement in security technology and the installation of cameras in public areas, it will become easier for police personnel to track, trace and locate criminals from the police control room using this app. In the future, advanced facial recognition techniques can be used to improve the results and a login page must be created so that any police personnel can remotely access this application. Additionally, if a criminal is found in a certain zone, alert messages should be sent to nearby police stations. The developed application is simple and user-friendly; the application interface can be developed more according to the user's requirements. The purpose behind the technology is making criminal identification easier and this project is a step towards achieving that goal. Protective services and authorities often fail to respond to crime incidents efficiently. They mostly follow a reactive approach which relies mostly on witness reports or CCTV footage after the crime takes place. Therefore, in most of the cases, when an event occurs, authorities visit the location of the incident, retrieve the content manually from the camera, and then proceed to identify relevant footage either by watching the full length of the video or by processing it through specialized video analytics algorithms. In the proposed system, we will be able to detect and recognize faces of the criminals in an image and in a video, stream obtained from a camera in real time and notify the concerned department. We will use Haar feature-based cascade classifiers in Open CV approach for face detection. So, this system will very useful for the police department to identify the criminal through video obtained from camera in real time.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who supported and guided us throughout the course of this project.
We are immensely thankful to our project guide, **Mr. T. Thiyagarajan M.Sc. App. Maths, Asst. Professor, Department of Computer Applications,** for their valuable insights, timely feedback, and constant encouragement, which played a crucial role in the successful completion of this work.
We also wish to thank the **Department of Computer Applications**, **Nehru Institute of Information Technology and Management**, **Coimbatore, Tamilnadu, India** for providing the necessary infrastructure and academic environment.
Finally, we are grateful to our friends and family for their moral support and motivation throughout this journey.

REFERENCES

1. S Samundeswari , Harini M, Dharshini, ”Real-time Crime Detection Using Customized CNN”, 2022 1st International Conference on Computational Science and Technology.
2. P. Sivakumar, and K. S, "Real Time Crime Detection Using Deep Learning Algorithm," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021.
3. F. Majeed, F. Z. Khan, M. J. Iqbal and M. Nazir, "Real-Time Surveillance System based Facial Recognition using YOLOv5," 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC).
4. C. Rajapakshe, S. Balasooriya, H. Dayarathna, N. Ranaweera, N.Walgampaya and N. Pemadasa, "Using CNNs RNNs and Machine Learning Algorithms for Realtime Crime Prediction," 2019 International Conference on Advancements in Computing (ICAC)
5. Nandhini T J and K Thinakaran “Detection of Crime Scene Objects using Deep Learning
https://www.geeksforgeeks.org/python/extract-data-from-database-using-mysql-connector-and-xampp-in-python/
https://medium.com/%40khouloud.haddad/building-a-face-recognition-camera-with-php-and-python-a-step-by-step-guide-70463aae4502
https://arxiv.org/abs/1802.09990?utm
https://mtcnn.readthedocs.io/en/latest/usage
https://www.sitepoint.com/keras-face-detection-recognition