International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 25 Issue 04, APR, 2025

Criminal Identification Using ML & Face Recognition Techniques

O.Teja Sri[1], D. Venkateswarlu[2]

Student[1], Assistant Professor[2]

Amrita Sai Institute of Science and Technology (Autonomous), Paritala, Andhra Pradesh, India

Abstract –Identifying and tracking criminals at crime scenes has become increasingly challenging, as perpetrators often avoid leaving biological traces or fingerprints. A fast and effective alternative is to leverage modern face-identification systems. With the widespread deployment of CCTV cameras in public and private spaces, continuous video streams can be harnessed to detect suspects, criminals, runaways, and missing persons. This paper presents an automated criminal identification framework that integrates machine learning and deep neural networks. By applying state-of-the-art deep learning algorithms to facial images extracted from surveillance footage, the system recognizes criminal faces with high accuracy. Upon identification, it automatically retrieves the individual's profile from a secure database and dispatches a real-time alert to police personnel, including the suspect's details and current location. This approach offers law enforcement a streamlined, proactive tool for crime detection and prevention.

Keywords -Criminal Identification, Face Recognition, Deep Learning, Machine Learning, CCTV Surveillance, Real-Time Alerts, Database Retrieval.

I. INTRODUCTION

Crime remains one of the most pressing and pervasive challenges faced by society, and its prevention is paramount, especially in the context of rapid urbanization and growing population densities, which make traditional manual surveillance methods increasingly inefficient and difficult to scale. In response to these challenges, extensive networks of closed-circuit television (CCTV) cameras have been installed in both public and private spaces, offering an opportunity for continuous monitoring. However, the sheer volume of video footage generated by these cameras makes it impractical for human operators to monitor in real-time. To address this, significant advancements in computer vision have enabled automated processing of live video streams, making it possible to detect and track criminal activities more effectively. This study proposes a comprehensive criminal-identification system that leverages these Advancements. The system captures real-time images from CCTV feeds and utilizes deep learning-based models, particularly convolutional neural networks (CNNs), to extract and encode crucial facial features from the images.

These facial features are then compared against a pre-existing database of known offenders to identify potential matches. Upon detecting a match, the system provides an immediate alert to law enforcement personnel, displays the offender's name and photograph, and archives the matched image for future reference. By integrating cutting-edge image-processing techniques, including automated face recognition powered by CNNs and support vector machine (SVM) classifiers, this solution significantly improves the speed and accuracy of suspect identification, reducing human error and response time. As a result, it enhances the overall efficiency of crime detection and prevention, contributing to a safer and more secure public environment.

II. LITERATURE REVIEW

The growing demand for automated criminal identification and public safety has led to significant advancements in image processing, computer vision, and facial recognition systems. Several studies have explored innovative methods and frameworks for detecting and identifying individuals from surveillance footage in public places. This literature survey reviews key research contributions that have laid the foundation for developing efficient criminal identification systems. Belhumeur et al. [1] explored two prominent face recognition approaches—Eigenfaces and Fisherfaces. They demonstrated the effectiveness of Fisherfaces in handling variations such aslighting and facial expressions due to its class-specific linear projection method. This work serves as a foundational study for developing robust face recognition systems. Bornet [2] emphasized the practical implementation of face detection using Intel's Open-Source Computer Vision Library (OpenCV).

The research highlighted the efficiency of learning-based techniques in real-world applications, contributing to the evolution of real-time surveillance systems. Brunelli and Poggio [3] compared feature-

based and template-based face recognition approaches. Their findings demonstrated that feature-based methods, which involve extracting distinct facial landmarks, often perform better in varying environmental conditions.

This insight has significantly influenced the development of adaptable recognition algorithms. Viola and Jones [4] introduced a revolutionary framework for real-time object detection using a boosted cascade of simple features. Their work drastically reduced the computational requirements for face detection while maintaining high accuracy, making it a widely adopted standard in security systems. In their follow-up research, Viola and Jones [5] emphasized the importance of robustness and efficiency in real-time detection applications. Their algorithm provided a scalable solution for integrating reliable face detection into surveillance systems. Schroff et al. [6] proposed the FaceNet model, which uses deep convolutional neural networks (CNNs) to directly learn embeddings for face verification and clustering.

This approach offered state-of-the-art accuracy and formed the basis for modern facial recognition systems. Kazemi and Sullivan [7] presented an efficient method for real-time face alignment using ensemble regression trees. Their approach significantly improved the accuracy of facial feature extraction, contributing to better recognition and detection outcomes. Howard et al. [8] developed MobileNets, lightweight deep learning models optimized for mobile and embedded vision applications. Their architecture is particularly useful in surveillance scenarios where computational efficiency is critical. Redmon et al. [9] proposed the You Only Look Once (YOLO) object detection framework, which achieved real-time detection with high accuracy. YOLO's ability to simultaneously detect multiple objects made it a preferred choice for public surveillance systems. Zhao et al. [10] investigated multi-modal approaches for face recognition in complex environments. Their research highlighted the benefits of combining facial, contextual, and behavioural features for enhanced criminal identification accuracy.

III. PROPOSED SYSTEM

The face detection algorithm proposed by Viola and Jones is a widely used method in computer vision for detecting faces in images. It utilizes a series of techniques that allow it to efficiently and accurately locate faces, even in real-time applications. Let’s break down the key components of the algorithm in more detail:

1. Haar Features (Haar-like Features)

The Viola and Jones algorithm relies on Haar features, which are digital image features used to represent characteristics of objects. These features are inspired by the Haar basis functions, a set of simple rectangular features (or patterns) that can be computed very quickly.

These Haar-like features are computed over rectangular regions of the image. For example, the feature might consist of two adjacent rectangles where the pixel values in one rectangle are subtracted from those in the other, capturing contrast or edge-like features, which are common in faces.

The types of Haar features used in face detection typically include edge features, line features, and center-surround features. These features are effective for detecting patterns such as eyes, noses, and the general structure of a face.

2. Integral Image

To speed up the computation of these Haar features, the integral image is used. The integral image allows the sum of pixel values within any rectangular region of the image to be computed in constant time.

This is accomplished by converting the image into a cumulative sum format where each pixel in the integral image contains the sum of all pixels to the left and above it. This allows Haar features to be computed very efficiently, enabling real-time face detection.

3. Cascade Classifier

The core of the Viola-Jones algorithm is the use of a cascade classifier, which is a series of stages, each designed to quickly reject non-face regions while keeping potential face regions for further scrutiny.

Each stage in the cascade consists of a classifier trained to differentiate between face and non-face candidates. The first stages of the cascade are very simple and quickly reject areas of the image that are unlikely to be faces. Only the regions that pass through

these simple stages are passed to more complex stages for further evaluation.

The cascade is trained using AdaBoost, a machine learning technique that selects the most relevant Haar features and trains a classifier to distinguish between face and non-face candidates. This results in a set of classifiers with varying levels of complexity, where early stages are fast but less accurate, and later stages are more accurate but slower.

4. Sub-windows (Face Candidates)

As the algorithm scans the image, it defines sub-windows (or candidate windows) within which the face detection is applied. These sub-windows are typically fixed in size, such as 24×24 pixels, and represent possible face locations.

The algorithm slides the window across the image, examining each potential face location by computing the Haar features and applying the cascade classifier. The goal is to find regions of the image that match the patterns of face features.

5. Multi-Scale Detection

Since faces can appear at different sizes, the algorithm applies multi-scale detection. The fixed-size sub-window is resized at different scales to detect faces of varying sizes. This scaling process helps detect faces that are further away (smaller faces) or closer to the camera (larger faces).

As the sub-window is scaled and moved across the image, the cascade classifier is applied at each scale, which allows the system to detect faces at various sizes in the image.

6. Final Detection

Once a face candidate passes through all stages of the cascade, it is classified as a face, and its bounding box is drawn around the detected face.W

The detected face can then be processed further for other tasks such as facial recognition, tracking, or image analysis.

IV. METHODOLOGY

Fig. 1 illustrates a fully automated, real-time criminal identification pipeline built around CCTV footage and facial recognition. Continuous video streams from public-space cameras are first broken down into individual frames, which are then pre-processed—through resizing, normalization, and color adjustment—to enhance image quality and reduce noise. Each cleaned frame is passed to a feature-extraction module that encodes distinctive facial traits (eye, nose, and mouth shape, etc.) into multi-dimensional vectors, forming a unique "face signature."

These signatures are compared against a secured criminal database of known offenders; when a match is detected, the system immediately displays the suspect's name and photograph, issues an alert notification, and saves the captured image to a dedicated desktop folder for evidence. If no match is found, the individual is classified as "Innocent," avoiding false alarms. By integrating continuous surveillance, preprocessing, deep feature extraction, and rapid database matching, this workflow provides law enforcement with an efficient, proactive tool for identifying and apprehending criminals in public areas.

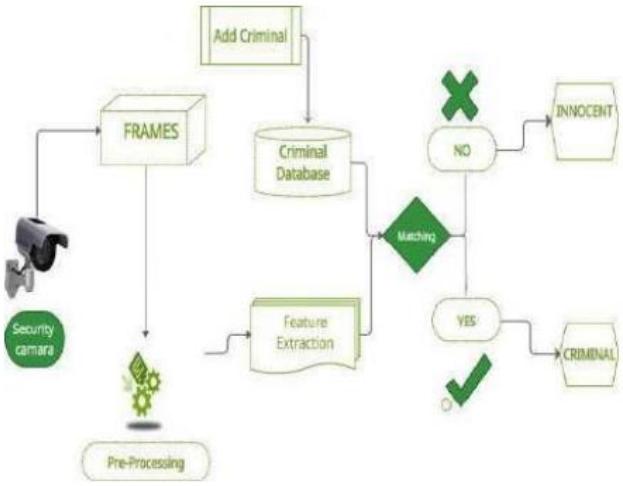

Fig 1: Proposed Model Architecture

V. ALGORITHMS DESCRIPTION

MTCNN (Multi-task Cascaded Convolutional Networks): MTCNN is a deep learning-based algorithm used for detecting faces in images. It works in a cascaded manner, meaning it uses multiple stages to refine its predictions. The first stage quickly filters out non-face regions, while the later stages focus on accurately locating facial features such as the eyes, nose, mouth, and chin. This approach makes MTCNN highly accurate, even when dealing with faces that are slightly turned or tilted. It is commonly used in face

detection as the first step before proceeding to further processing like face recognition.

FaceNet: FaceNet is a powerful deep learning model designed to convert face images into numerical embeddings, or vectors, that represent a person's unique facial features. These embeddings are 128-dimensional vectors that map faces to a Euclidean space, where the distance between two embeddings indicates how similar the faces are. This allows FaceNet to perform both face verification (checking if two faces are the same person) and face identification (identifying a person from a database of known faces). FaceNet is commonly used for high-accuracy face recognition tasks.

SVM (Support Vector Machine): SVM is a supervised machine learning algorithm used for classification tasks. It works by finding a hyperplane that best separates data into different classes. In the context of face recognition, SVM is used to classify faces based on their embeddings from FaceNet. Once a test face is converted into an embedding, SVM compares it with embeddings from a database (e.g., criminal faces) and classifies whether there's a match. SVM is effective in high-dimensional spaces and is commonly used for face matching in security systems.

VI. RESULTS AND DISCUSSION

Fig. 2 presents a graphical user interface (GUI) designed for criminal identification using machine learning (ML) and face recognition techniques. The interface is organized into several components to simplify the process of detecting criminals from a facial image dataset. Initially, the user selects a folder that contains the required files for the system, such as the "Dataset" (images of known criminals), the "Model" (pre-trained recognition models), and "Test Images" (used for evaluating system performance).

After selecting the folder, the user can upload the criminal dataset into the platform for processing. The next step involves preprocessing the dataset, where images are standardized, noise is reduced, and necessary adjustments are made to prepare them for analysis. Once preprocessing is complete, the user initiates the training process by clicking the "Train SVM using MTCNN & FaceNet Features" button.

This step employs Multi-task Cascaded Convolutional Networks (MTCNN) for precise face detection and FaceNet for feature extraction, essential for building an effective recognition model. After training, the user can access a "Comparison Graph" to assess model performance based on metrics like accuracy, precision, and recall. The key feature of the GUI is the "Criminal Identification" button, which allows the system to test a given image against the criminal database; if a match is found, the system identifies the person as a criminal. Finally, the "Exit" button lets the user close the application.

This GUI provides a smooth and organized approach to criminal identification, combining advanced face recognition algorithms and machine learning for better efficiency and accuracy in public safety tasks. Additionally, Fig. 3 displays the performance metrics, and Figs. 5 and 6 illustrate the face recognition results on test images.

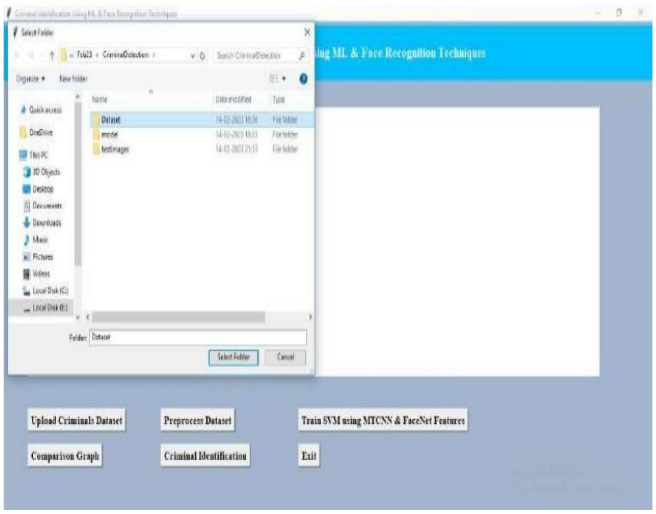

Fig 2:Graphical user interface (GUI) developed for criminal identification

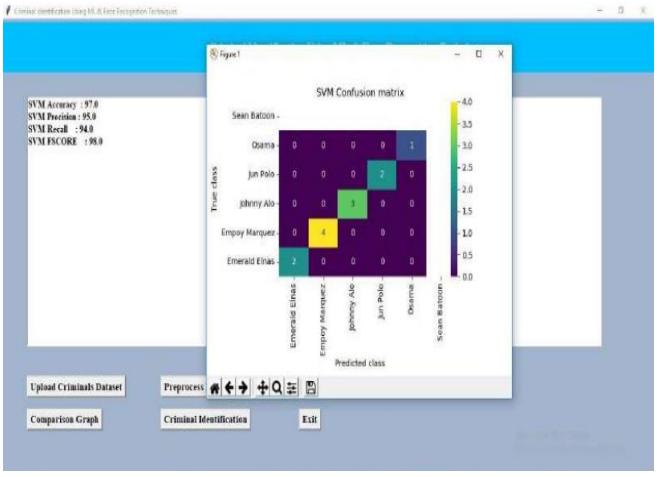

Fig 3: Performance Metrics

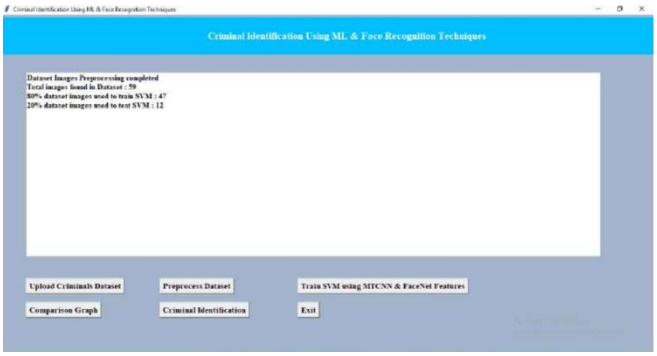

Fig 4: Dataset Loading

Fig 5:Face recognized in test image 1 with confidence score

Fig 6: Face Not recognized in test image 2

VII. CONCLUSION

We achieved real-time criminal face detection and recognition from both still images and live video streams using OpenCV. Face detection relies on Haar feature–based cascade classifiers, which are trained on large sets of positive and negative samples to learn rapid, multi-stage cascade functions. For recognition, we employ the Local Binary Patterns Histograms (LBPH) algorithm, which offers efficient feature selection and scale/location invariance by resizing features rather than the entire image. LBPH can accurately identify faces under varying lighting conditions and even from a single training image per person, and its generic training framework can be extended to detect other objects (e.g., cars, signboards, license plates). However, this approach is optimized for frontal faces and its accuracy diminishes when the face is rotated by about 45° along the vertical or horizontal axis.

VIII. FUTURE WORK

Looking ahead, we plan to replace Haar cascades with modern, CNN-based detectors (e.g., MTCNN, YOLO, or SSD) to boost localization accuracy and robustness to varied poses and occlusions, and upgrade from LBPH to deep-feature embeddings such as FaceNet or ArcFace for stronger discrimination across pose, age, and illumination differences. We will explore data-augmentation strategies and GAN-based face frontalization to handle extreme angles and lighting, while integrating liveness detection modules to defend against spoofing attacks. To further improve recognition confidence, we aim to fuse facial biometrics with complementary modalities like gait or voice, and to port and optimize our models for real-time inference on edge devices via pruning and quantization. Continual and few-shot learning techniques will allow the system to adapt quickly as new profiles are added, and privacy-preserving approaches (e.g., federated learning) will keep sensitive data local. Additionally, we will extend the pipeline to support automated multi-camera tracking and real-time alert dispatch to field units. Finally, we will conduct real-world trials in collaboration with law-enforcement agencies to evaluate performance under operational conditions, address edge-case failures, and ensure full compliance with evolving ethical and data-protection standards.

IX. REFERENCES

[1] Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 19, pp. 711-720. IEEE Computer Society

[2] Bornet, O. (2005, May 19). Learning Based Computer Vision with Intel's Open-Source Computer Vision Library. Retrieved April 2007, 2007, from Intel.com Website: http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/p04_face_dete ction.htm

[3] Brunelli, R., & Poggio, T. (1993). Face Recognition: Features versus templates. IEEE

Transaction on Pattern Analysis and Machine Intelligence, 15 (10), 1042-1052.

[4] Viola, P. and Jones, M. Rapid object detection using boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[5] P. Viola and M. Jones. Robust Real-time Object Detection. International Journal of Computer Vision,

[6] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815-823.

[7] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1867-1874.

[8] Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

[10] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face Recognition: A Literature Survey. ACM Computing Surveys, 35(4), 399-458