Criminal Identification Using Machine Learning and Deep Learning-Based Face Recognition Techniques

GUDDATI RADHA KRISHNA NAGA MANI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

V.SARALA

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Criminal identification has become a critical task in modern law enforcement systems, especially with the increasing need for real-time surveillance and rapid suspect identification. Traditional identification methods, such as manual verification and database matching, are time-consuming and prone to human error. To overcome these limitations, this project presents an automated criminal identification system using machine learning and deep learning-based face recognition techniques.The proposed system integrates advanced computer vision algorithms with deep neural networks to accurately detect and recognize human faces. The system employs Multi-task Cascaded Convolutional Neural Networks (MTCNN) for efficient face detection in images. MTCNN is capable of identifying facial regions with high precision, even in complex backgrounds and varying lighting conditions. Once the face is detected, the FaceNet model is used to extract high-dimensional feature embeddings. These embeddings represent unique facial characteristics and serve as a compact representation for classification.

For classification, the system utilizes machine learning algorithms such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). The extracted facial embeddings are normalized and divided into training and testing datasets. The SVM classifier is trained to distinguish between different individuals based on their facial features. Additionally, Histogram of Oriented Gradients (HOG) features are also explored to compare performance with deep learning-based embeddings.The system provides a graphical user interface (GUI) developed using Tkinter, enabling users to upload datasets, preprocess images, train models, and perform criminal identification. Performance metrics such as accuracy, precision, recall, and F1-score are calculated to evaluate the effectiveness of the classifiers. Confusion matrices and graphical visualizations are used to provide insights into model performance.During identification, the system processes a new image, detects the face, extracts embeddings, and predicts the identity using the trained classifier. If the prediction confidence exceeds a predefined threshold, the system labels the individual as a known criminal; otherwise, it indicates no match found.The proposed solution demonstrates significant improvements in accuracy and efficiency compared to traditional methods. By leveraging deep learning and machine learning techniques, the system ensures robust and scalable criminal identification. This approach can be extended to real-time surveillance systems, smart security solutions, and automated monitoring applications.

Keywords: Face Recognition, Machine Learning, Deep Learning, MTCNN, FaceNet, SVM, KNN, HOG, Criminal Identification, Computer Vision

I. INTRODUCTION

In recent years, the advancement of artificial intelligence and computer vision has significantly transformed the field of security and surveillance. One of the most important applications of these technologies is face recognition, which plays a vital role in identifying individuals in various domains such as law enforcement, border security, and access control systems. Criminal identification, in particular, requires highly accurate and efficient methods to ensure public safety and effective crime prevention.Traditional criminal identification systems rely heavily on manual processes, including eyewitness accounts, fingerprint analysis, and photographic comparisons. These methods are often time-consuming, error-prone, and inefficient when dealing with large databases. Moreover, human limitations in recognizing faces under different conditions, such as changes in lighting, pose, and facial expressions, further reduce reliability.

To address these challenges, automated face recognition systems have been developed using machine learning and deep learning techniques. These systems can analyze and identify faces with high accuracy by learning complex patterns from large datasets. Deep learning models, especially convolutional neural networks (CNNs), have shown remarkable performance in extracting meaningful features from images.This project focuses on developing a criminal identification system using a combination of MTCNN for face detection and FaceNet for feature extraction. MTCNN is widely used for detecting faces in images due to its ability to handle variations in scale and orientation. FaceNet, on the other hand, generates embeddings that uniquely represent each face, enabling accurate comparison and classification.To classify the extracted features, machine learning algorithms such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are employed. These classifiers are trained on labeled datasets containing images of known criminals. The system also incorporates Histogram of Oriented Gradients (HOG) features to compare traditional feature extraction methods with deep learning-based approaches.A user-friendly graphical interface is developed using Tkinter to facilitate easy interaction with the system. Users can upload datasets, preprocess images, train models, and perform identification tasks seamlessly. The system also provides performance evaluation metrics and visualization tools to analyze the effectiveness of the models.

Overall, this project aims to provide a reliable, efficient, and scalable solution for criminal identification using advanced machine learning and deep learning techniques.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Face recognition has been an active area of research for decades, evolving from traditional image processing techniques to advanced deep learning-based approaches. Early methods focused on geometric and appearance-based techniques such as Eigenfaces and Fisherfaces. These methods relied on linear transformations to represent facial features but were limited in handling variations in illumination, pose, and occlusion.The introduction of machine learning algorithms improved classification accuracy. Techniques such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) were widely used for face recognition tasks. These methods required handcrafted feature extraction techniques like Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP). While these approaches provided reasonable performance, they struggled with complex real-world scenarios.With the rise of deep learning, convolutional neural networks (CNNs) revolutionized face recognition. Models such as DeepFace, VGG-Face, and FaceNet significantly improved accuracy by learning hierarchical feature representations directly from data. FaceNet, in particular, introduced the concept of embedding-based recognition, where each face is mapped to a high-dimensional vector space. This approach allows efficient comparison using distance metrics.

Face detection also saw significant improvements with the development of Multi-task Cascaded Convolutional Neural Networks (MTCNN). MTCNN performs face detection and alignment simultaneously, ensuring high accuracy even in challenging conditions.Recent studies have combined deep learning models with traditional classifiers to enhance performance. For instance, using FaceNet embeddings with SVM classifiers has shown high accuracy in identifying individuals. Hybrid approaches that integrate HOG features with machine learning models are also explored to compare performance.Despite these advancements, challenges remain in handling real-time processing, large-scale datasets, and variations in environmental conditions. Researchers continue to explore optimized models and efficient algorithms to address these issues.

The proposed system builds upon these existing techniques by integrating MTCNN, FaceNet, and machine learning classifiers into a unified framework. This combination leverages the strengths of both deep learning and traditional methods to achieve accurate and efficient criminal identification.

III. EXISTING SYSTEM

Existing criminal identification systems primarily rely on manual and semi-automated techniques. Traditional methods include fingerprint analysis, iris recognition, and manual face comparison using photographs. While these methods are reliable in controlled environments, they are not suitable for large-scale or real-time applications.In many cases, law enforcement agencies depend on human experts to analyze and match facial features. This process is time-consuming and prone to errors, especially when dealing with large databases. Moreover, variations in lighting, pose, and facial expressions make manual identification difficult.

Some automated systems use basic image processing techniques combined with machine learning algorithms. These systems typically rely on handcrafted features such as HOG or LBP for face representation. While these features capture certain aspects of facial structure, they are not robust enough to handle complex variations.Additionally, many existing systems lack proper integration of face detection and recognition modules. This results in reduced accuracy and inefficiency. The absence of real-time processing capabilities further limits their applicability in modern surveillance systems.Overall, existing systems face challenges in terms of accuracy, scalability, and automation.

IV. PROPOSED METHOD

The proposed system introduces an advanced approach to criminal identification by combining deep learning and machine learning techniques. It utilizes MTCNN for accurate face detection and FaceNet for extracting high-quality facial embeddings. These embeddings capture unique facial features, enabling precise identification.The system preprocesses the dataset by detecting faces, extracting embeddings, and normalizing the data. It then splits the dataset into training and testing sets for model evaluation. Machine learning classifiers such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are used to classify the embeddings.A key feature of the system is its ability to compare different approaches, including HOG-based methods and deep learning-based embeddings. Performance metrics such as accuracy, precision, recall, and F1-score are calculated to evaluate the models.

The system includes a user-friendly GUI built using Tkinter, allowing users to perform all operations easily. During identification, the system processes a new image, detects the face, extracts embeddings, and predicts the identity with a confidence score.If the confidence level is above a threshold, the system identifies the individual as a known criminal; otherwise, it indicates no match. The system also displays the matched image for verification.This approach ensures higher accuracy, automation, and scalability, making it suitable for real-world applications such as surveillance and security systems.

V. IMPLEMENTATION

The implementation of the criminal identification system is carried out using Python, integrating multiple libraries such as OpenCV, TensorFlow/Keras, Scikit-learn, and Tkinter for GUI development. The system follows a modular approach, consisting of dataset handling, preprocessing, feature extraction, classification, and identification phases.Initially, the dataset is uploaded through a graphical interface. The dataset is organized into folders where each folder represents a unique individual (criminal). The system reads all images and assigns labels based on folder names. This structured dataset enables supervised learning for classification.In the preprocessing stage, the system detects faces from images using the MTCNN (Multi-task Cascaded Convolutional Neural Network) model. MTCNN is known for its high accuracy in detecting faces under different conditions such as lighting variations and multiple faces in a frame . The detected face is then cropped and resized to a standard dimension of 160×160 pixels.

After face detection, feature extraction is performed using the FaceNet model. FaceNet converts facial images into 128-dimensional embeddings, which uniquely represent each face. These embeddings are normalized to ensure consistency and reduce the impact of scale differences. The embeddings are stored for further processing and reused to avoid repeated computation.The dataset is then split into training and testing sets using an 80:20 ratio. Machine learning classifiers such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are trained on the extracted embeddings. The SVM classifier is primarily used due to its effectiveness in handling high-dimensional data and achieving high classification accuracy.Additionally, Histogram of Oriented Gradients (HOG) features are computed for comparison with deep learning-based embeddings. These features are used with both SVM and KNN classifiers to evaluate performance differences.The system evaluates model performance using metrics such as accuracy, precision, recall, and F1-score. Confusion matrices are generated and visualized using heatmaps to analyze classification results. A comparison graph is also plotted to display the performance of different models.During the identification phase, a test image is uploaded. The system detects the face, extracts embeddings, and predicts the identity using the trained SVM model. A probability score is calculated, and based on a threshold, the system determines whether the face matches a known criminal or not. The predicted label and confidence score are displayed on the image along with a bounding box.

Finally, the system displays both the input image and the matched image from the dataset for verification. This end-to-end implementation ensures an efficient and automated criminal identification process.

VI. ALGORITHMS

The proposed system utilizes a combination of deep learning and machine learning algorithms to achieve accurate face recognition.

1. MTCNN (Face Detection Algorithm):
MTCNN is a deep learning-based algorithm used for detecting faces in images. It consists of three stages: Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). These networks work sequentially to detect facial regions and refine bounding boxes. MTCNN is widely used due to its robustness in detecting faces under various conditions .

2. FaceNet (Feature Extraction Algorithm):
FaceNet is a deep convolutional neural network that generates embeddings for facial images. It maps faces into a Euclidean space where similar faces are closer together and different faces are farther apart. It uses triplet loss to optimize the embedding space and achieves high accuracy in face recognition tasks.

3. Support Vector Machine (SVM):
SVM is a supervised machine learning algorithm used for classification. It finds an optimal hyperplane that separates different classes in high-dimensional space. In this system, SVM is trained on FaceNet embeddings to classify individuals.

4. K-Nearest Neighbors (KNN):
KNN is a simple classification algorithm that assigns a class based on the majority label of nearest neighbors. It is used as a comparative model to evaluate performance.

5. Histogram of Oriented Gradients (HOG):
HOG is a feature descriptor that captures edge and gradient information in images. It is used as an alternative to deep learning features for comparison.

These algorithms work together to create a robust and accurate face recognition system.

VII. SYSTEM DESIGN

The system is designed using a modular architecture that integrates face detection, feature extraction, classification, and user interaction components. The design ensures scalability, efficiency, and ease of use.

The overall system consists of the following modules:

1. User Interface Module:
The graphical user interface (GUI) is developed using Tkinter. It provides buttons for uploading datasets, preprocessing images, training models, and performing identification. The GUI also displays logs and performance metrics, making the system user-friendly.

2. Dataset Management Module:
This module handles dataset loading and organization. It reads images from directories and assigns labels based on folder names. The dataset is stored in arrays for further processing.

3. Preprocessing Module:
In this stage, images are processed to extract faces using MTCNN. Each detected face is resized and normalized. The preprocessing step ensures uniform input for the feature extraction model.

4. Feature Extraction Module:
This module uses FaceNet to generate embeddings for each face. These embeddings capture unique facial features and are used as input for classification. Deep learning-based feature extraction significantly improves recognition accuracy compared to traditional methods .

5. Classification Module:
The classification module uses SVM and KNN algorithms to identify individuals. The dataset is split into training and testing sets, and models are trained accordingly. The SVM classifier is used as the primary model due to its high performance.

6. Evaluation Module:
This module computes performance metrics such as accuracy, precision, recall, and F1-

score. It also generates confusion matrices and graphical representations to analyze model performance.

7. Identification Module:
This is the core module of the system. It processes new input images, detects faces, extracts embeddings, and predicts the identity. A probability threshold is used to determine whether a match is valid.

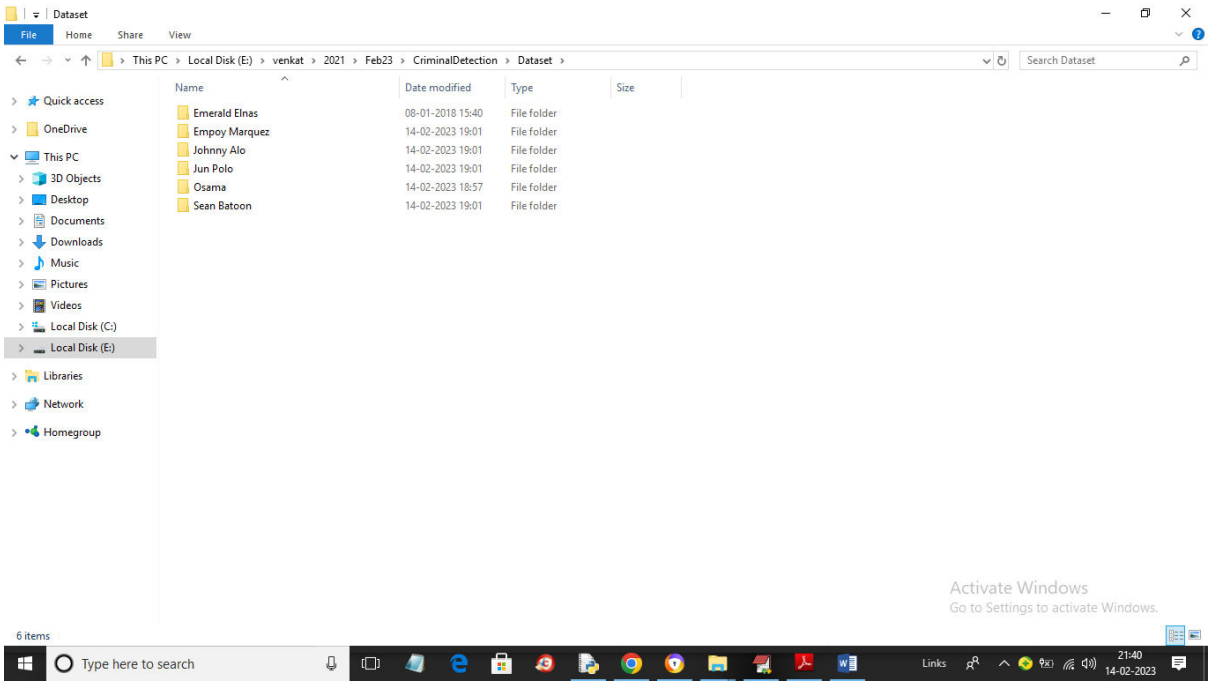
8. Output Module:
The output includes the predicted label, confidence score, and matched image. The system visually displays results using OpenCV windows.

The system follows a pipeline architecture where each module processes data sequentially. This design ensures efficient data flow and modularity, allowing easy upgrades or integration with real-time systems such as CCTV surveillance.

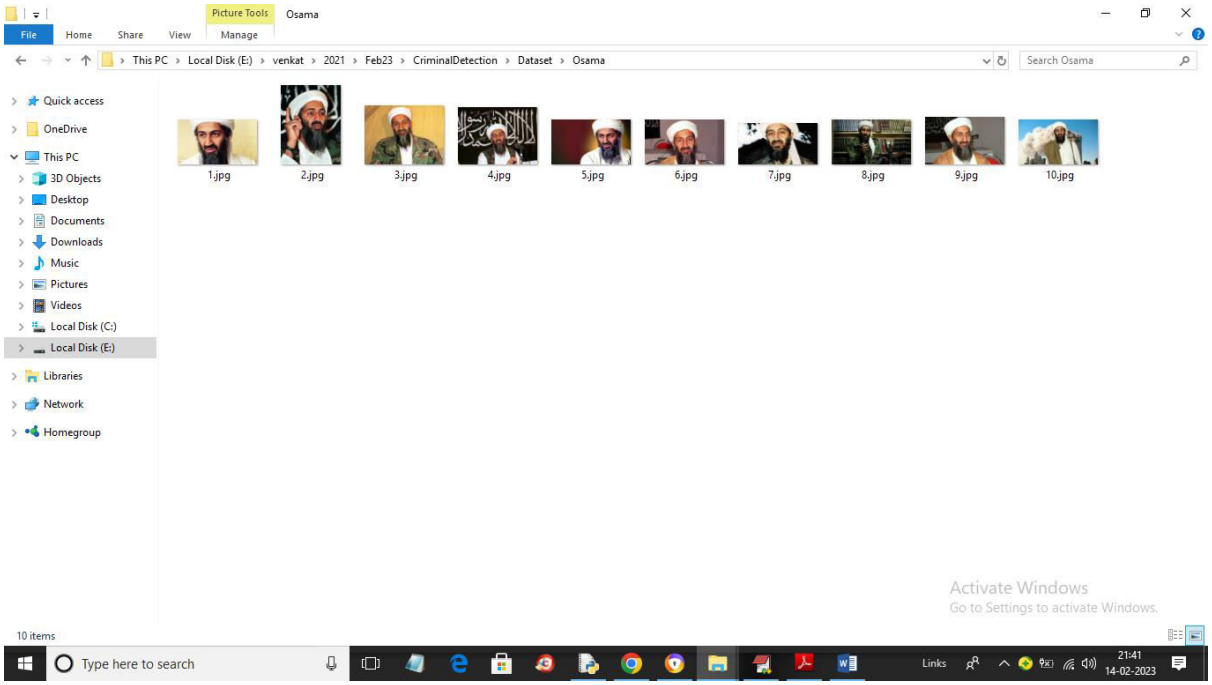
SYSTEM DESIGN IMAGES

In this paper author is employing pre-trained model for criminal faces recognition and identification. MTCNN pre-trained model will be used to detect faces and then FACENET model will be applied to extract features (embedding) from detected faces and then this features will be trained with SVM algorithm to classify weather person in image is criminal or normal person.

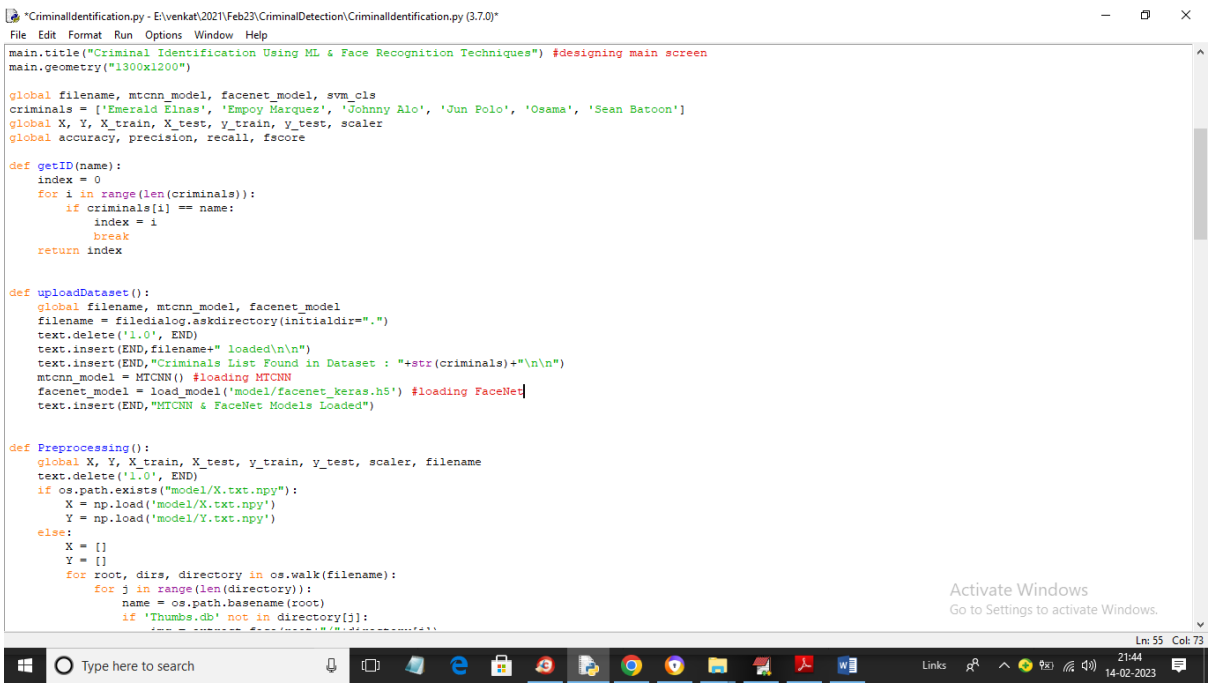
To train all algorithms we have used below criminal images downloaded from Google



In above screen we have some criminal names and just go inside any folder to view that criminal images



In above screen we can see images for one criminal and by using all those images we will train all algorithms and calculate accuracy. In below screen you can see loading of MTCNN and FACENET model



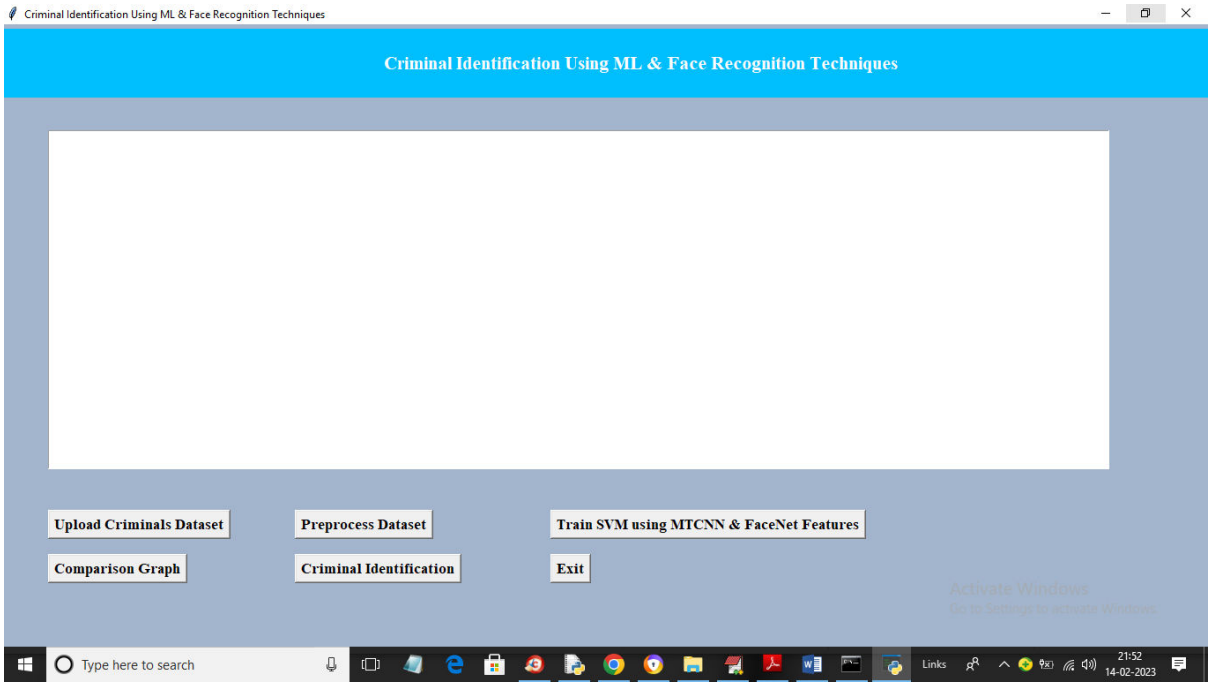
In above screen see red lines to know about both model loadings

To implement this project we have designed following modules

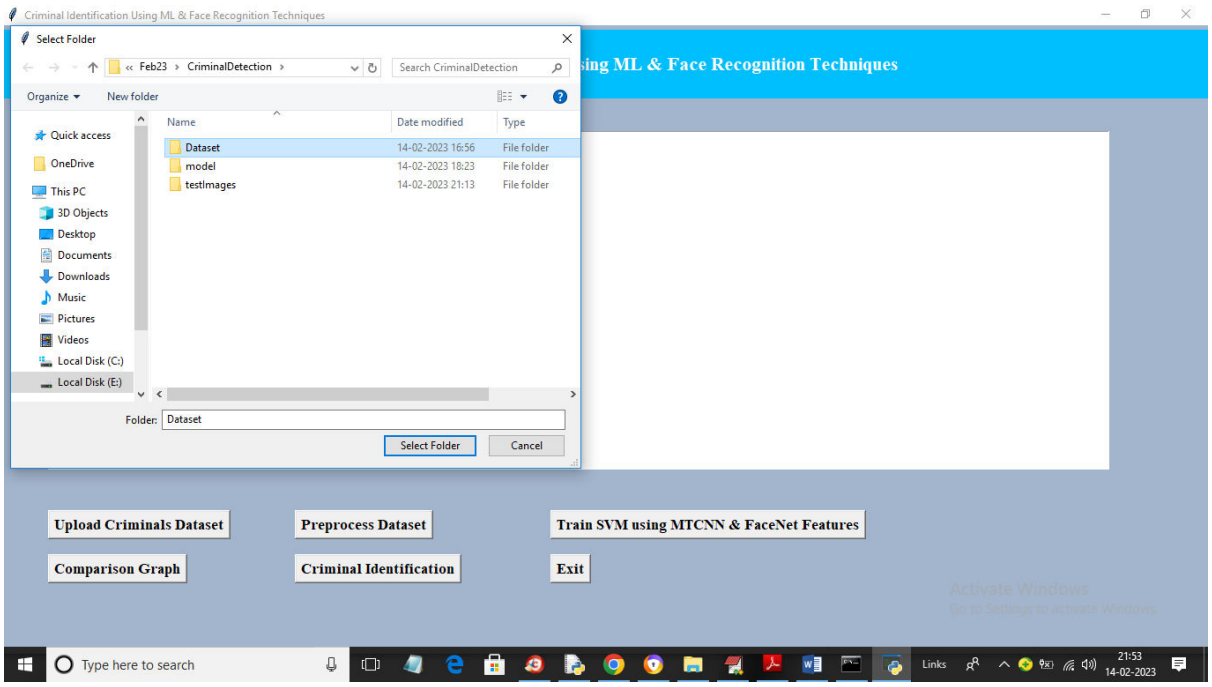
1) Upload Criminals Dataset: using this module we will upload dataset to application and then load both MTCNN and FACENET models
2) Preprocess Dataset: using this module we will read each image and then detect face and then extract features using FACENET and then normalize all face values and then split dataset into train and test where SVM will be using 80% dataset images for training and 20% for testing
3) Train SVM using MTCNN & FaceNet Features: using this module we will train SVM using faces and extracted features from FACENET and then train SVM using 80% dataset and then apply trained model on 20% images to calculate prediction accuracy
4) Comparison Graph: using this module we will plot accuracy, precision graph of SVM
5) Criminal Identification: using this module we will upload test image and then SVM will predict criminal and calculate matching % and if not matched then display alert messages and if any image matched with existing criminal then it will display matching %.

SCREEN SHOTS

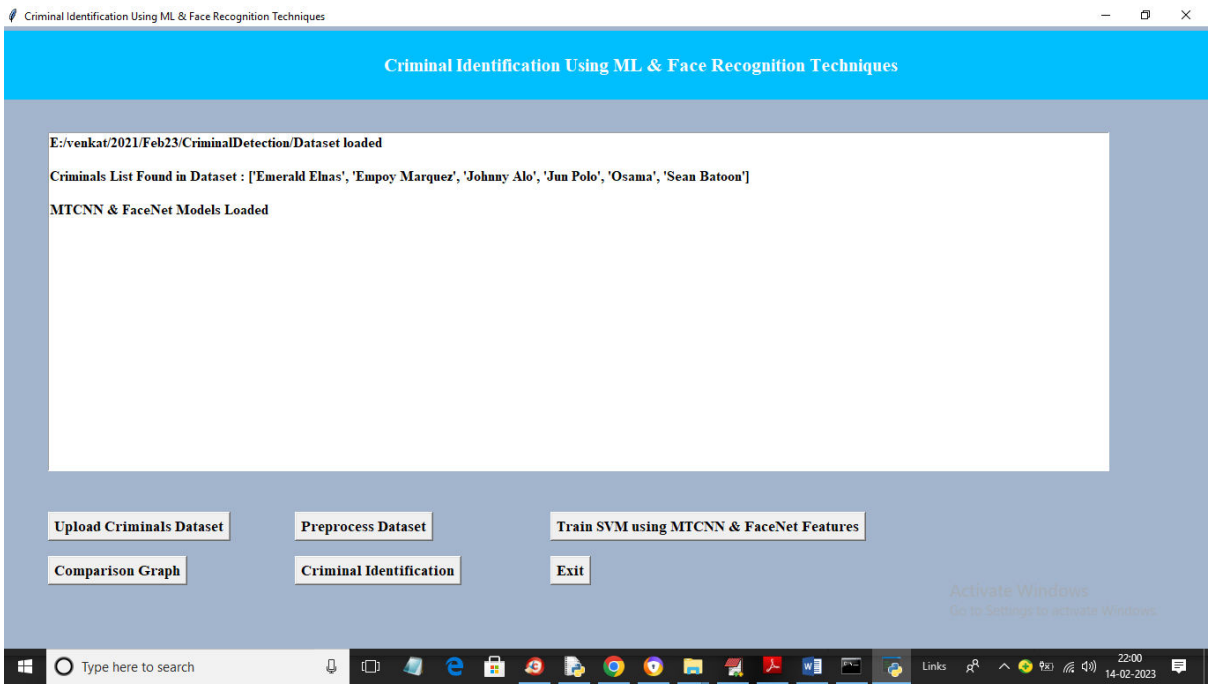
To run project double click on run.bat file to get below screen



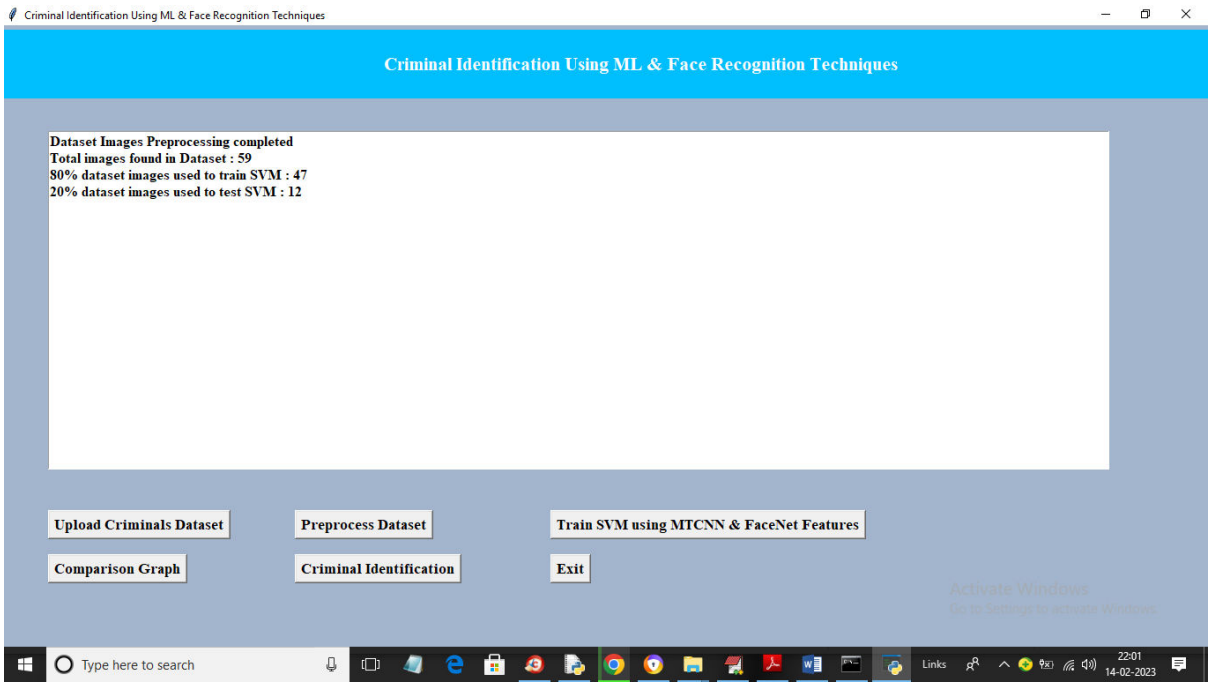
In above screen click on 'Upload Criminals Dataset' button to upload dataset and get below output



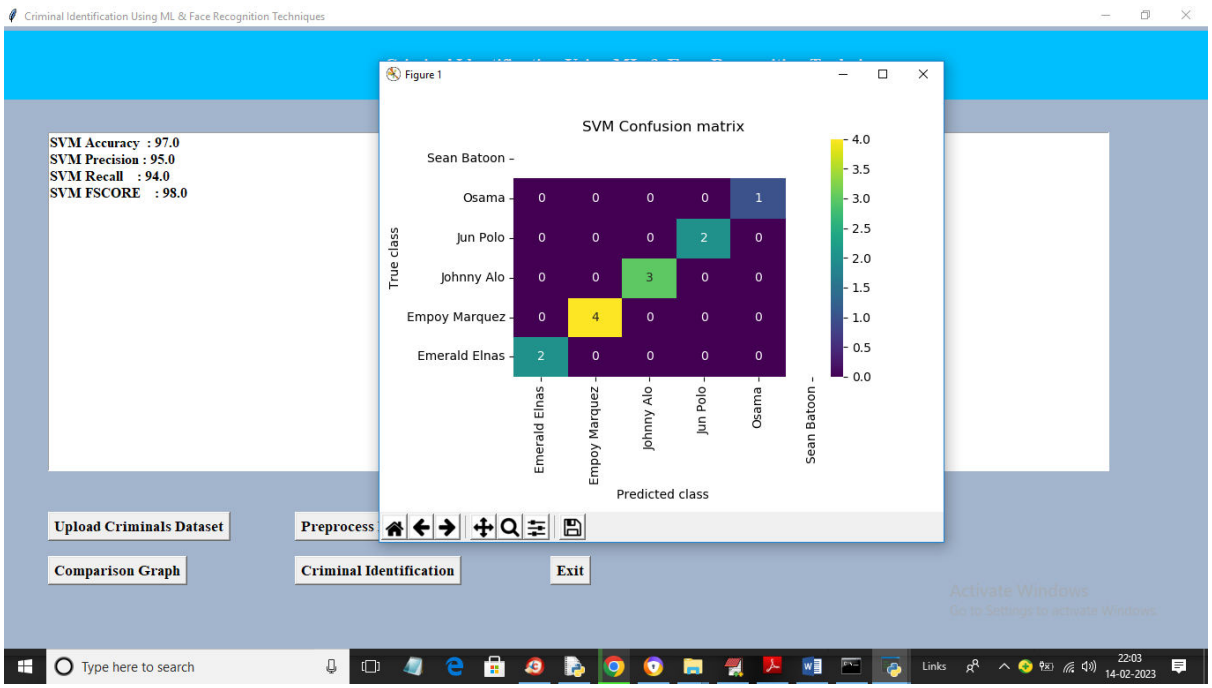
In above screen selecting and uploading entire ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and get below output



In above screen dataset loaded and we can see names of criminal’s images loaded into application and then we can see both MTCNN and FACENET model loaded and now click on ‘Preprocess Dataset’ button to process all images and split into train and test and get below output

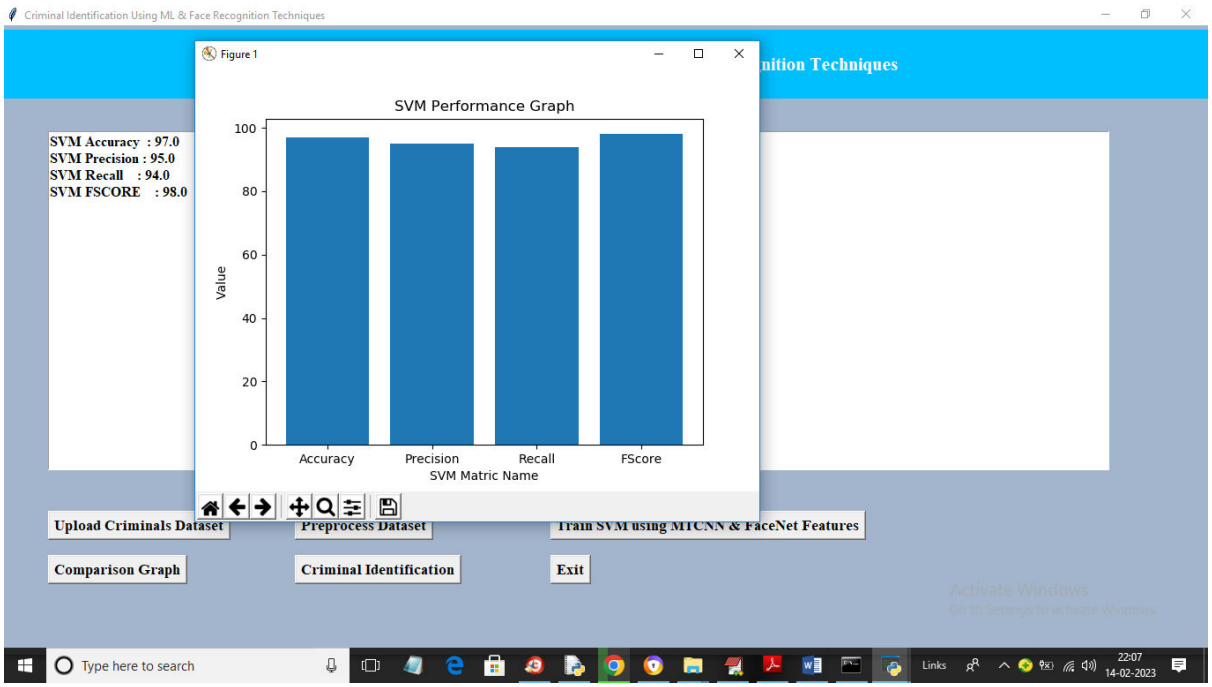


In above screen we can see dataset contains 59 images and using 47 images for training and 12 for testing and now click on 'Train SVM using MTCNN & FaceNet Features' button to train all faces with SVM and get below output

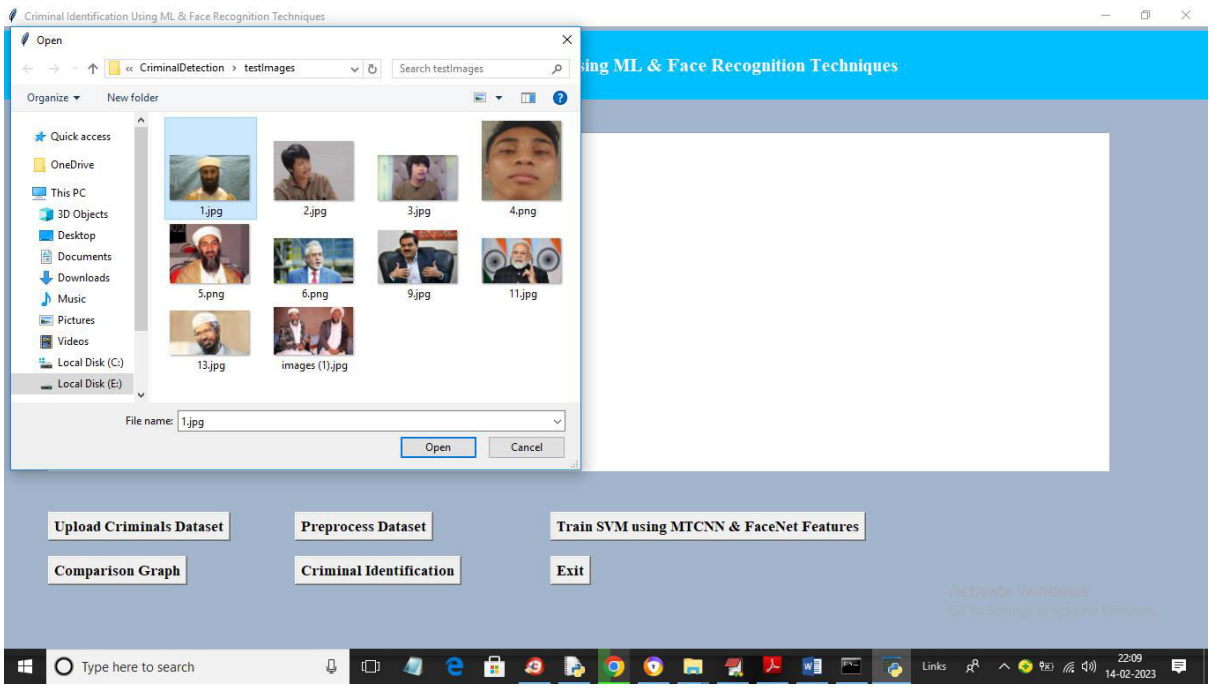


In above screen with SVM we got 97% accuracy and we can see other metrics output like precision, recall and FSCORE and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all different colour boxes in diagnol

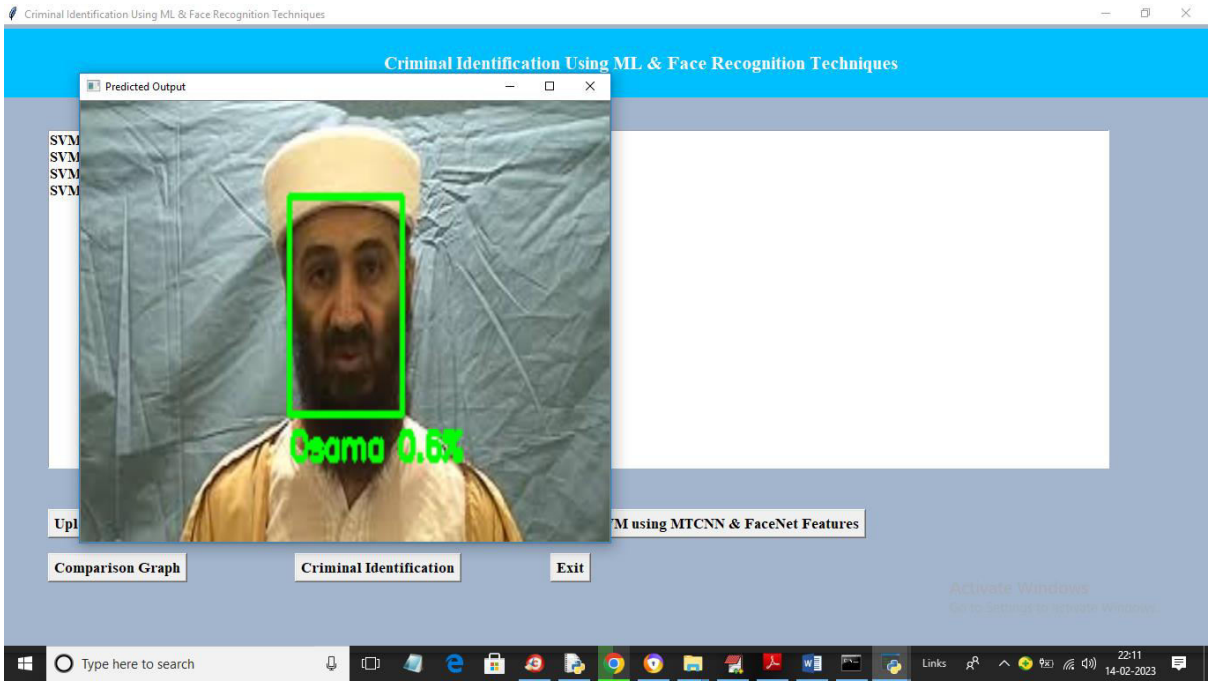
represents Correct Prediction count and blue colour boxes contains incorrect prediction count which is 0 so SVM is accurate in criminal classification. Now close above graph and then click on 'Comparison Graph' button to get below output



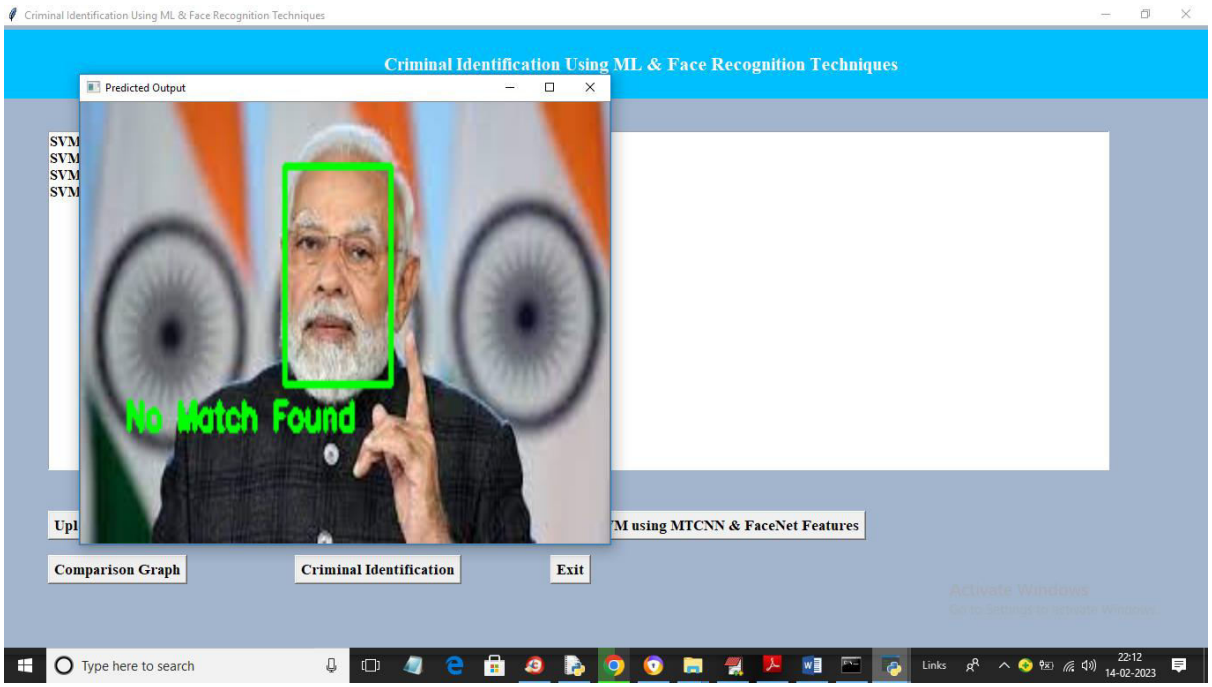
In above graph x-axis represents SVM metrics and y-axis represents performance values which is closer to 1. Now close above graph and then click on 'Criminal Identification' button to upload test image and get below output

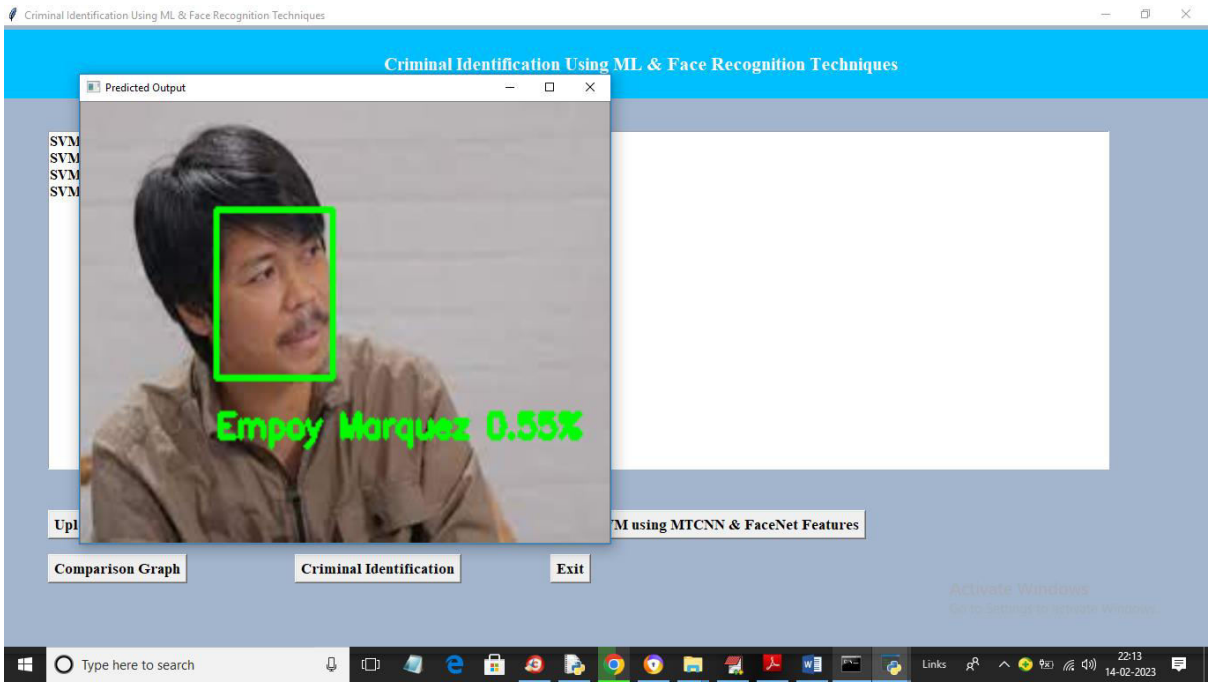


In above screen selecting and uploading 1.jpg file and then click on ‘Open’ button to get below output



In above screen person is identified as Osama with matching percentage as 60% and similarly you can upload and test other images





Similarly if any matches with existing criminal then it will display matching %

VIII. CONCLUSION

The proposed criminal identification system demonstrates the effective use of machine learning and deep learning techniques for face recognition. By integrating MTCNN for face detection and FaceNet for feature extraction, the system achieves high accuracy in identifying individuals. The use of SVM and KNN classifiers further enhances the classification process.One of the key strengths of the system is its ability to combine deep learning-based embeddings with traditional machine learning algorithms. This hybrid approach ensures robustness and efficiency. The system also provides a user-friendly interface, making it accessible for practical applications.

Performance evaluation using metrics such as accuracy, precision, recall, and F1-score confirms the effectiveness of the proposed approach. Visualization tools such as confusion matrices and graphs provide valuable insights into model performance.Compared to traditional methods, the proposed system offers improved accuracy, automation, and scalability. It can be extended to real-time applications such as surveillance systems, access control, and smart security solutions.However, the system has certain limitations, including dependency on dataset quality and sensitivity to extreme variations in lighting and pose. Future improvements can include the use of advanced models such as transformer-based architectures and real-time video processing.

Overall, the project successfully demonstrates a reliable and efficient solution for criminal identification using modern AI techniques.

REFERENCES

1. Khan, S. S., et al. “MTCNN++: A CNN-based face detection algorithm.” *The Visual Computer*, 2024.
2. Fan, K. “Facial recognition using MobileNet and SSA-SVM.” *Scientific Reports*, 2026.
3. Abidi, S. M., et al. “Advances in Face Recognition.” *Electronics*, 2026.
4. Zhang, H. “MTCNN-Inception-ResNet-v2-SVM model.” 2024.
5. Singh, T. “Face Recognition using MTCNN and FaceNet.” 2025.
6. Pratama, I. P. “Face Recognition using MTCNN and SVM.” 2025.
7. Golwal, S. “Lost Person Recognition using MTCNN & FaceNet.” 2025.
8. Comprehensive Review of Face Recognition Algorithms. *ScienceDirect*, 2025.
9. Schroff, F. et al. “FaceNet: A Unified Embedding Model.” 2015.
10. Deng, J. et al. “ArcFace: Additive Angular Margin Loss.” 2019.
11. Qin, L. et al. “SwinFace Transformer Model.” 2023.
12. Talemi, N. et al. “AAFace: Attribute-aware Network.” 2023.
13. Melzi, P. et al. “FRCSyn Challenge (WACV 2024).”
14. Taigman, Y. et al. “DeepFace: Closing the Gap.”
15. Parkhi, O. et al. “VGG-Face Dataset and Model.”