IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

AN EFFICIENT DEEP LEARNING ARCHITECTURE FOR FACE RECOGNITION IN CRIMINAL DISCERNMENT

[1] HIMANSHI SINGH, [2] R DILEEP KUMAR, [3] RAGHAV YADAV, [4] HARI MOHAN SINGH

[2 & 4] Assistant Professors, [3] Associate Professor

Department of Computer Science and Information Technology,
Sam Higginbottom University of Agriculture, Technology And Sciences, Prayagraj, U.P, INDIA 211007

Abstract: This research paper presents an efficient deep learning architecture designed to enhance face recognition systems for criminal discernment. Motivated by the need to improve public safety and streamline the criminal justice process, the study addresses significant challenges, including data quality, image variability, and system fairness. The proposed methodology leverages advanced models like transfer learning and GoogleNet, optimizing feature extraction and improving performance. The literature review highlights various techniques, such as CNNs with histogram equalization and the Eigenface method etc., demonstrating their effectiveness under different conditions. Through rigorous analysis, design, and evaluation, this study aims to significantly advance facial recognition technology, providing law enforcement with robust tools for accurate and swift suspect identification. The findings underscore the potential of this technology to deter crime, aid in investigations, and ensure reliable evidence for justice.

Keywords: Deep Learning, Face Recognition, Criminal, Transfer Learning, GoogleNet, Public Safety, Law Enforcement, Feature Extraction, Data Augmentation, Ethical AI.

1. INTRODUCTION

Criminal investigations are pivotal in uncovering the truth behind crimes, identifying perpetrators, and securing evidence for prosecution. This process begins with securing the crime scene and collecting physical evidence, followed by forensic analysis, witness interviews, and digital data examination. Advanced digital tools and forensic techniques, including artificial intelligence and machine learning, enhance the efficiency and accuracy of investigations by analyzing vast data sets and identifying patterns and anomalies through evidential images collected from crime scene with minimal intervention with the system. As cybercrime and digital platforms become increasingly prevalent, integrating these technologies into investigative practices is crucial for effective crime solving under rigid time constraints, ultimately contributing to justice and public safety.

The criminal investigation landscape in India is shaped by its vast and diverse socio-economic and demographic context, presenting unique challenges compared to western methodologies. The low police-to-citizen ratio and underinvestment in **criminal justice system** (CJS) infrastructure exacerbate these challenges, impacting the efficiency of crime resolution. In 2021, the **National Crime Records Bureau (NCRB)** reported over 60 lakh crimes, which is a matter of serious concern, at a time when technological assistance is already in use at different levels of law enforcement agencies. The crime rate also highlights regional disparities, such as of Uttar Pradesh's, highly populous state of India, high crime rate compared to other states [1]. The Indian government has established several agencies to enhance investigation, coordination and address these challenges, including the **Central Finger Print Bureau (CFPB)** and the **Central Crime and Criminal Tracking Network Systems (CCTNS)** [2]. These agencies use state of the art tools for investigation process. Despite use of advanced technology, such as CCTV and mobile GPS, and the use of image processing for crime detection, ongoing issues like staff shortages and resource constraints continue to impact the effectiveness of investigations.

The crime statistics in India from 2020 to 2022 reveal notable trends in the prevalence of crimes reported under the **Indian Penal Code (IPC)** and **Special & Local Laws (SLL).** IPC crimes showed a gradual decrease from 4,254,356 in 2020 to 3,561,379 in 2022, while SLL crimes exhibited fluctuations, with a slight increase in 2021 before declining again in 2022. Specifically, SLL crimes numbered 2,346,929 in 2020, 2,432,950 in 2021, and 2,263,567 in 2022. The overall crime rate also saw a reduction, with the total crime rate per 100,000 people decreasing from 487.8 in 2020 to 422.2 in 2022. These factors are attributed to application of technological resources in crime solving. Additionally, the percentage of IPC crimes relative to total cognizable crimes dropped from 64.4% in 2020 to 61.1% in 2022, highlighting a slight shift in the composition of reported crimes [3]. The state-wise data highlights Uttar Pradesh as consistently having the highest number of IPC crimes, with a steady rise from 355,110 in 2020 to 401,787 in 2022. Other states such as Maharashtra, Bihar, and West Bengal also experienced notable trends. Maharashtra saw a slight decrease in IPC crimes in 2021 but rose again to 374,038 in 2022. Bihar displayed an increase from 186,006 in 2021 to 211,079 in 2022. West Bengal's IPC crimes remained relatively stable over the three years, with minor fluctuations. These trends indicate a broader regional pattern in crime prevalence and underscore the ongoing challenge of crime management, necessitating targeted law enforcement strategies [4].

2. RELATED WORK

Recent advancements in deep learning have significantly enhanced criminal face detection systems, leveraging various sophisticated techniques to achieve remarkable accuracy and robustness. **GailiYue and Lei Lu (2018)** pioneered the integration of histogram equalization with Convolutional Neural Networks (CNNs), achieving an impressive accuracy of 98.25%. This method effectively addressed lighting and quality issues, improving image contrast and enabling CNNs to perform better under varied lighting conditions, crucial for real-world applications where lighting is often inconsistent [5]. **Yimyam et al. (2018)** employed the Eigenface method for CCTV-based detection, achieving an accuracy range of 80-90%. The Eigenface approach, based on Principal Component Analysis (PCA), reduced the dimensionality of image data, facilitating the system's ability to recognize faces from different angles and under varying conditions commonly found in CCTV footage [6].

Kim et al. (2019) introduced a real-time facial feature extraction method using cascaded CNNs, which significantly reduced processing time while maintaining high accuracy, making it suitable for surveillance and monitoring systems [7]. **Pei et al. (2019)** focused on the impact of image degradation on CNN performance and demonstrated the benefits of degradation removal techniques. By removing artifacts and enhancing image quality before feeding them into CNNs, they achieved significant improvements in detection performance, particularly useful for handling low-resolution or noisy images [8]. **Peng (2019)** developed a deep cascaded CNN framework that achieved 96.3% accuracy in complex face detection scenarios. This framework utilized multiple CNN layers to progressively refine detection results, making it robust against variations in face orientations and occlusions [9].

Putro et al. (2019) highlighted the efficiency of multi-layered CNNs in real-time face detection applications. Their study demonstrated that deeper network architectures, with multiple convolutional and pooling layers, could efficiently process high-resolution images in real-time, essential for systems requiring instantaneous responses [10]. **Ragsania and Vijayalakshmi (2022)** showcased the effectiveness of deep transfer learning in face recognition, emphasizing how pre-trained models on large datasets could be fine-tuned for specific tasks with limited data, achieving high accuracy in criminal face detection [11]. **Md. Faruk and Abdullah Al Sohan (2022)** achieved 90% accuracy using CNNs specifically designed for criminal identification by training on a diverse dataset of criminal images, ensuring the model's generalization to new, unseen faces [12].

Band and Thangaraj (2022) compared unsupervised learning methods with CNNs for facial expression identification, demonstrating that CNNs provided competitive results even without extensive labeled datasets [13]. **Bhargava and Rathore (2023)** explored the Haar Cascade Classifier, achieving an accuracy range of 90-98%. The Haar Cascade Classifier, a machine learning-based approach, was effective for initial face detection and could be refined using more advanced CNN-based methods [14]. **Krishna and Reddy (2023)** combined CNNs with Haar cascade algorithms to achieve over 90% accuracy in criminal identification. This hybrid approach leveraged the speed of Haar cascades for initial detection and the accuracy of CNNs for detailed recognition, providing a balanced solution for real-world applications [15].

Technological intervention in face recognition has revolutionized the field, offering unprecedented accuracy and efficiency in identifying individuals. Advanced algorithms, particularly those utilizing deep learning and Convolutional Neural Networks (CNNs), have significantly improved the ability to recognize faces under varied conditions and in real time. Techniques such as histogram equalization, Eigenface methods, and cascaded CNN frameworks have addressed challenges like lighting inconsistencies, image degradation, and occlusions. These innovations have enhanced the robustness of face recognition systems, making them invaluable tools for security, surveillance, and law enforcement, ultimately contributing to safer and more secure environments. These studies collectively highlight significant improvements in face detection accuracy, real-time processing, and robustness, demonstrating the potential of deep learning to enhance security and law enforcement capabilities.

3. PROPOSED MODEL

The criminal dataset was meticulously self-generated to meet the specific requirements of our research, utilizing Python and OpenCV for image collection and preprocessing. OpenCV's robust integration with APIs and Python enabled us to leverage its advanced computer vision capabilities, particularly the **haarcascade_frontalface_default.xml** and **haarcascade_eye.xml files**, to detect and focus on frontal faces and eye regions. Using these pre-trained Haar Cascade classifiers, we captured and cropped facial regions of interest from real-time video streams, ensuring that the images were consistently sized and focused on the eye region. This preprocessing was essential for standardizing the data and preparing it for subsequent deep learning model training.

Our dataset consists of high-quality JPG colored images, each carefully optimized to emphasize the facial eye region. We have compiled a diverse collection of 1000 images for each of three individuals, labeled as Person 1, Person 2, and Person 3, covering a range of facial expressions, angles, and lighting conditions. To evaluate our model's performance on real-world data, we also included 1000 images of non-criminal labeled faces for testing purposes. This self-generated dataset, free from external biases and inconsistencies, offers a controlled and high-quality foundation for training and assessing criminal face detection models, as illustrated in Figures numbered 3.1 to 3.4. The final dataset comprises 3000 original images, and through augmentation, the dataset size has grown to over 24,000 images. Table 3.1 provides a summary of these images from both the criminal and non-criminal datasets.

Class Labels	**Samples**	**Purpose**
Criminal Dataset		
Person 1	1000	Criminals (Testing &Training)
Person 2	1000	
Person 3	1000	
Total Criminal Samples	**3000**	
Non-Criminal Dataset		
Non-Criminal Faces	1000	Non Criminals (Testing)
Total Non-Criminal Samples	**1000**	
Grand Total	**4000**	

Table 3.1 Summary of the criminals & Non-criminals dataset.

Fig 3.1 Sample Images of different Person 1(Criminal Labeled image)

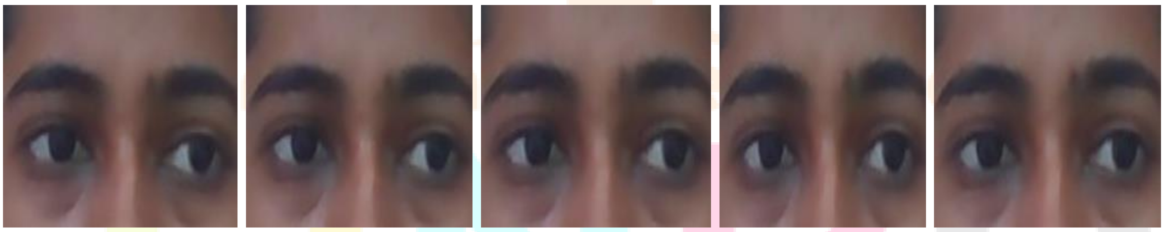

Fig 3.2 Sample Images of different Person 2 (Criminal Labeled image)

Fig 3.3 Sample Images of different Person 3 (Criminal Labeled image)

Fig 3.4 Sample images of non-criminal labeled images
(Disclaimer: Photos taken from internet on random basis for research purpose only)

Architecture:

By taking concepts of the earlier models as an inspiration, a new model is devised with own approach using varying layers of CNN architecture. Figure 3.5 illustrates the step-by-step process of the proposed model with integration of CNN architecture.

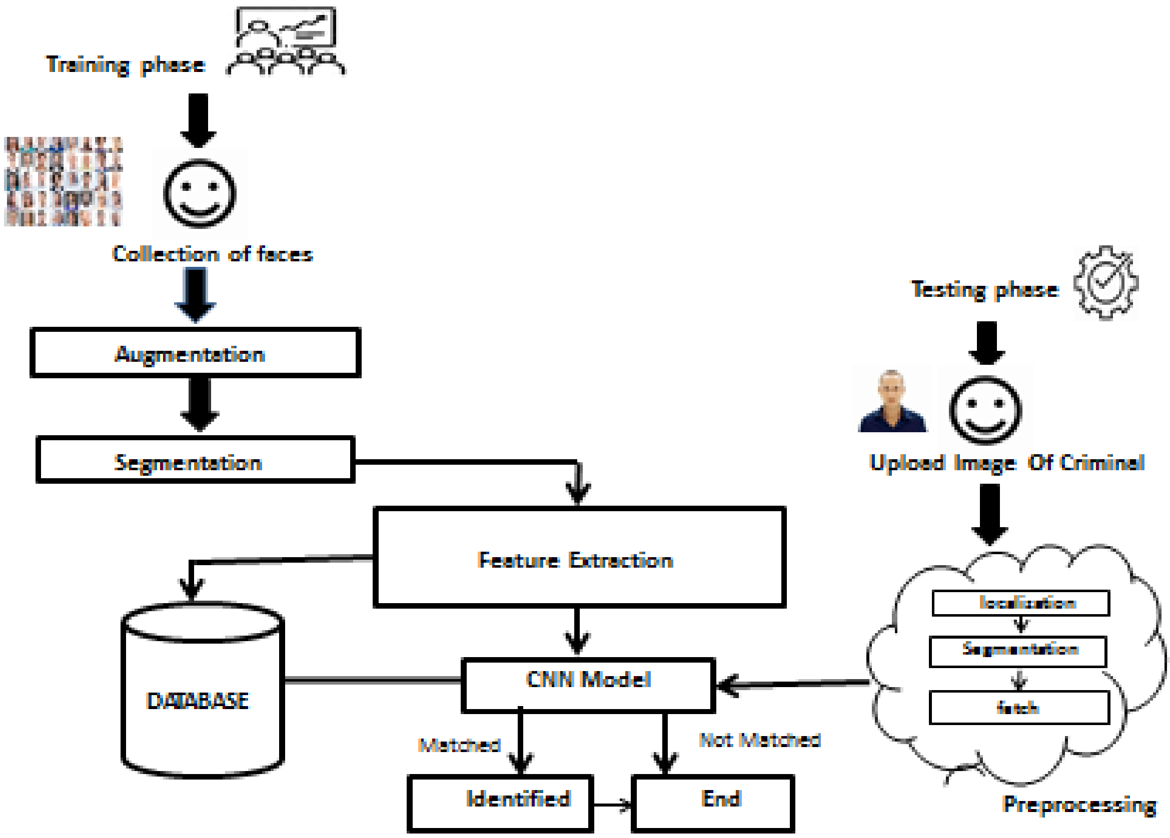

Fig. 3.5 Schematic Flow Diagram of the Proposed System for Criminal detection

The flow diagram represents a comprehensive system for criminal face detection and recognition, structured into two primary phases: training and testing. During the training phase, a collection of facial images is gathered, forming the dataset necessary for model training.

This dataset undergoes augmentation to increase the diversity and volume of training data, followed by segmentation to isolate facial features from the background. These processed images are stored in a database. Feature extraction techniques are then applied to capture significant attributes of the faces, which are subsequently used to train a Convolutional Neural Network (CNN) model. The CNN model, specifically leveraging the GoogleNet Inception architecture with transfer learning, is trained on these features to identify and distinguish between different faces in the dataset.

In the testing phase, an image of a criminal is uploaded for identification. The uploaded image undergoes preprocessing steps, including localization to detect the face within the image, segmentation to isolate the face from the background, and feature extraction to prepare the image for analysis. The processed image is then fed into the trained CNN model. The model compares the features of the uploaded image against the features stored in the database during training. If a match is found, the individual is identified as a known criminal; otherwise, the system concludes with no match found. This testing phase ensures that the model can accurately recognize and identify faces based on the learned features.

The use of the GoogleNet Inception model with transfer learning significantly enhances the system's performance and efficiency. Transfer learning allows the model to leverage pre-trained weights from a vast dataset, reducing the need for an extensive training process and enabling it to learn from a smaller, domain-specific dataset. This approach, combined with the use up to 45,207 parameters, fine-tunes the model for precise feature extraction and identification. The architecture of the GoogleNet Inception model includes multiple layers such as convolutional layers, inception modules, pooling layers, and fully connected layers. Each layer plays a crucial role in extracting and processing different levels of features, ultimately contributing to the high accuracy and robustness of the criminal face detection system.

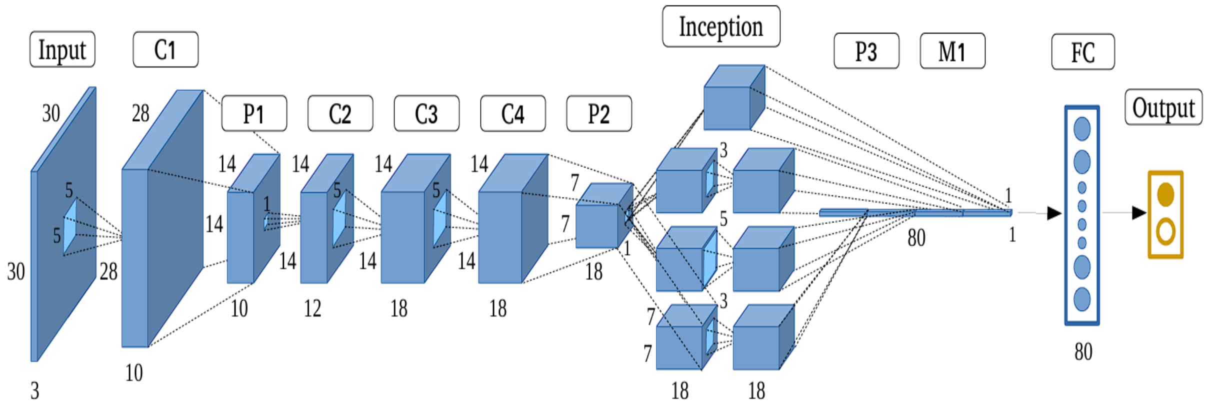

Fig. 3.6 The Proposed Deep Learning GoogLeNet Inception model architecture

The model utilizes the GoogLeNet Inception architecture,a seminal advancement in CNNs introduced by **Szegedy et al. in 2014** [16], represents a significant leap forward in convolutional neural network (CNN) design. Its core innovation lies in the inception modules, which apply multiple convolutional filters of varying sizes—1x1, 3x3, and 5x5—simultaneously within a single layer. This approach allows the network to capture a wide range of features at different scales, enhancing its ability to detect intricate patterns in the input data. By incorporating these inception modules, GoogLeNet effectively balances the trade-off between depth and computational complexity, enabling the model to learn and generalize from a vast amount of data without being hindered by excessive parameter counts. Additionally, the integration of 1x1 convolutions for dimensionality reduction helps in reducing the overall computational burden, making the model more efficient [17].

A key feature of GoogLeNet is its use of global average pooling, which replaces traditional fully connected layers [16]. This modification significantly reduces the number of parameters in the model, which not only improves computational efficiency but also helps in mitigating overfitting. The global average pooling layer summarizes the feature maps into a single vector by averaging all the values, thus preserving the essential information while discarding the redundancy. This design choice contributes to the model's accuracy and efficiency, making GoogleNet particularly well-suited for tasks such as face detection and criminal identification. Its demonstrated success in various studies [18] underscores its profound impact on advancing deep learning techniques in computer vision.

Integrating the GoogLeNet Inception model into our proposed architecture enhances its performance by leveraging its advanced feature extraction capabilities. The inception modules, which apply multiple convolutional filters concurrently, allow the model to capture diverse features at various scales, improving accuracy in complex tasks like face detection and criminal identification. Additionally, global average pooling reduces parameter count and computational load, aligning with our need for efficiency and reduced overfitting. The model's demonstrated success in various studies [17][18] underscores its effectiveness, making it an ideal choice for our application.

Algorithm DS: Criminal and Non-Criminal Face Detection

Algorithm DS: Data set containing two classes of criminal faces and one class of non-criminal faces. The training & testing set contains around 4000 images for criminal and non-criminal class
Data Set (DS): *Contains two classes: Criminal and Non-Criminal.*

- **Criminal Class:** 3000 images (1000 per person labeled as criminal for training & testing)
- **Non-Criminal Class:** 1000 images (1000 individuals for testing purpose only)

DS_Labels: {Criminal, Non-Criminal}

Step 1: Data Pre-processing

1. **Load Images:**
   - Load all images from the dataset.
   - Resize images to 200x200 pixels.
2. **Convert to Numerical Vectors:**
   - Standardize pixel values (normalize to [0,1] range).
   - Convert images to numerical vectors for model input.
3. **Train-Test Split:**
   - Split the dataset into training and testing subsets.
     - **Training Set:** 80% of images from each class.
     - **Testing Set:** 20% of images from each class.

Step 2: Define Model

1. **Define Layers:**

   - **Input Layer: (200, 200, 3)**Receives images resized to 200x200 pixels with 3 color channels (RGB). It prepares the data for further processing.

   - **Convolutional Layers: 3x3 Filters, ReLU Activation**Applies 3x3 filters to detect local patterns such as edges and textures. ReLU activation adds non-linearity to help learn complex features.

   - **Max Pooling Layers**Reduces the size of feature maps by selecting the maximum value in a pooling window (e.g., 2x2). This simplifies the data and retains essential features.

   - **Inception Modules**Extracts features at multiple scales using parallel convolutions with different filter sizes. The outputs are combined to capture diverse features from the image.

   - **Fully Connected (Dense) Layers**Integrates and interprets features from previous layers to perform classification. Each neuron is connected to all neurons in the previous layer, facilitating complex decision-making.

   - **Output Layer: Softmax Activation**Converts the final layer's scores into probabilities for each class (Criminal, Non-Criminal). The class with the highest probability is selected as the final prediction.

2. **Non-Linearity and Pooling:**
   - Apply ReLU activation functions.
   - Use max pooling to reduce dimensionality.
3. **Dropout:**
   - Apply Dropout(0.2) in fully connected layers to reduce overfitting.

Step 3: Model Compilation and Training

1. **Compile Model:**
   - **Optimization Algorithm:** Configure with Adam.
   - **Loss Function:** Use Categorical Cross-Entropy for classification.
   - **Metrics:** Track accuracy to evaluate model performance.
2. **Training:**
   - Train the model with the training dataset.
   - Adjust parameters using back propagation to minimize the loss function.

Step 4: Prediction and Labeling

1. **Load Trained Model:**
   - Load the model once training is complete.
2. **Pre-process Image:**
   - Load and resize the input image to 200x200 pixels.
   - Standardize the image similarly to the training data.

3. **Predict Label:**
   - Feed the pre-processed image into the model.
   - Obtain predicted probabilities for each class.
4. **Determine Final Label:**
   - Assign the label with the highest probability (Criminal or Non-Criminal).

4. EXPERIMENTAL RESULTS AND DISCUSSION

Experimental Framework

In our study, we developed a criminal detection system using a CNN-based GoogleNet Inception model with transfer learning, leveraging Google Colab's GPU resources and libraries such as sci-kit-learn, PyTorch, and OpenCV. Our experiments were conducted on a Windows 8.1 Pro machine equipped with an Intel(R) Core(TM) i3-6006U CPU featuring 4 cores, Radeon R2 graphics clocked at 2.00 GHz, 8 GB of RAM, and the Jupyter environment alongside Python 3.9.13. The Keras suite was used for software development. Given our imbalanced dataset, we employed confusion matrices to evaluate performance, using metrics such as sensitivity (recall), accuracy, specificity, precision, and F1-score. Our CNN model consistently outperformed traditional algorithms, demonstrating higher accuracy and reducing false identifications, thus highlighting the effectiveness of deep learning approaches in law enforcement for enhanced reliability and public safety.

This table 4.1 details the original number of images, the number of images generated by each data augmentation technique, the total number of augmented images for each class, and the overall total number of images.

Augmentation Technique	**No of Images**			
	Person 1	**Person 2**	**Person 3**	**Total**
Original Images	1000	1000	1000	3000
Augmentation Applied				
Rotation (90/180/270)	3000	3000	3000	9000
Flipping	1000	1000	1000	3000
Cropping	1000	1000	1000	3000
Color Jittering	1000	1000	1000	3000
Adding Noise	1000	1000	1000	3000
Augmented Images	**7000**	**7000**	**7000**	**21000**
Total Images	**8000**	**8000**	**8000**	**24000**

Table 4.1 Image Dataset Augmentation

Images	Augmentation operations				
	Rotating (90,180, 270)	Flip	Bright -ness	Salt & pepper	Cropping
1000	+3	+1	+1	+1	+1
Total	7000 * 3 = 21000				

Table 4.2 Image Dataset Augmentation

(a) Rotation (b)flip (c) Brightness (d) adding noise (e) cropping

Fig 4.1 Image augmentation result

Model	Precision	Recall	F1 Score	AUC
VGG16	0.961	0.965	0.963	0.960
Haar Cascade Classifier	0.980	0.975	0.977	0.970
ImageNET	0.973	0.972	0.972	0.965
Proposed (GoogleNet Inception) / IDcriminalNET	0.979	0.979	0.979	0.980

Table 4.3 Model Performance Comparison

Confusion Matrix:

	Predicted Criminal	**Predicted Non-Criminal**
Actual Criminal	979(TP)	21 (FN)
Actual Non-Criminal	20(FP)	2980 (TN)

Table 4.4 Confusion Matrix

Calculated Values:

1. **Precision (P)**:

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} = \frac{979}{979+20} = \frac{979}{1000} \approx 0.979$$

2. **Recall (R)**:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} = \text{Recall} = \frac{980}{21+980} = \text{Recall} = \frac{979}{1000} \approx 0.979$$

3. **F1 Score**:

$$\text{F1-score} = 2* \frac{\text{Precision}*\text{recall}}{\text{presion}+\text{recall}} = 2* \frac{0.979*0.979}{0.979+0.979} \approx 0.979$$

4. **Accuracy:**

$$\text{Accuracy= F1-score} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{979+2980}{979+2980+20+21} \approx 0.980$$

Comparative study

The comparative analysis of face detection models highlights significant variations in their performance and complexity. The VGG16 model achieves a 96.30% accuracy rate with a substantial total of 549,344 parameters, 549,322 of which are trainable, indicating a high computational demand. The Haar Cascade Classifier, on the other hand, boasts the highest accuracy rate of 98% while using only 86,666 parameters, making it an efficient option for environments with limited resources. ImageNET strikes a balance between accuracy and complexity with a 97.30% accuracy rate and a total of 81,376 parameters, 81,372 of which are trainable. The proposed IDcriminalNET model achieves a strong 98.00% accuracy rate with just 45,207 parameters, all of which are trainable. This demonstrates that IDcriminalNET offers an optimal balance of high accuracy and low computational complexity, making it an excellent choice for practical applications requiring both performance and efficiency.

Model	**Accuracy Rate**	**Total Parameters**	**Training Parameters**
VGG16	96.30%	549344	549,322
Haar Cascade Classifier	98 %	86,666	86,646
ImageNET	97.30 %	81376	81372
Proposed IDcriminalNET	98.00 %	**45,207**	**45,207**

Table 4.5 Accuracy of the comparison model

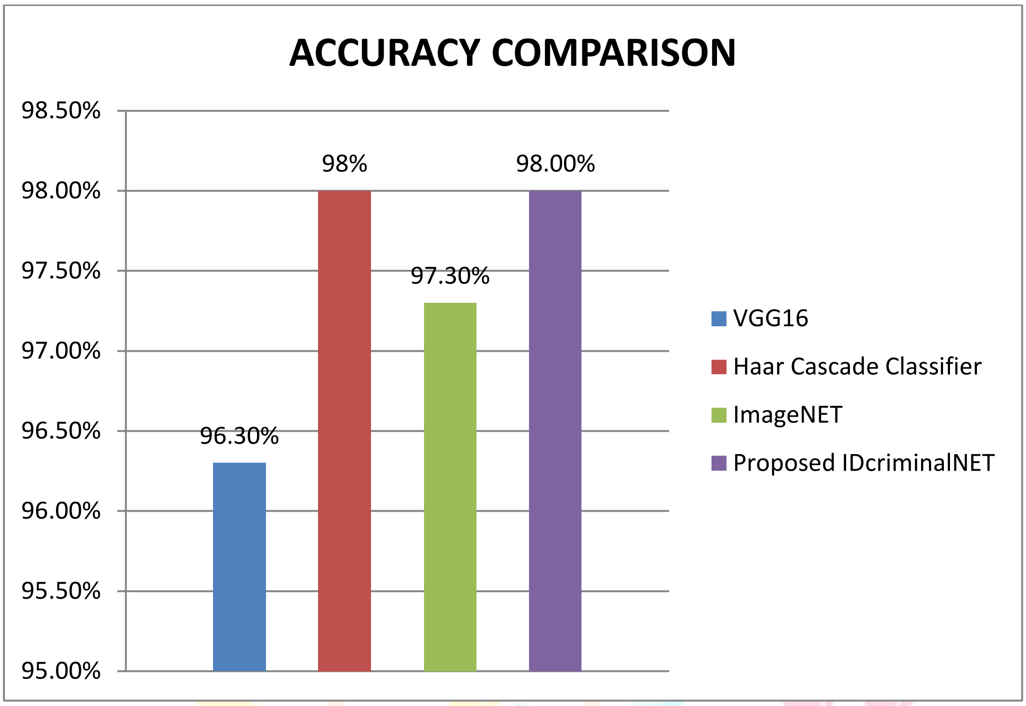

Fig 4.2 accuracy of the comparison model

Figure 4.3 shows that VGG16 has the lowest metrics, while Haar Cascade excels in Precision. ImageNET performs consistently, but the proposed IDcriminalNET achieves the highest scores in Precision, Recall, and F1 Score, highlighting its superior performance.

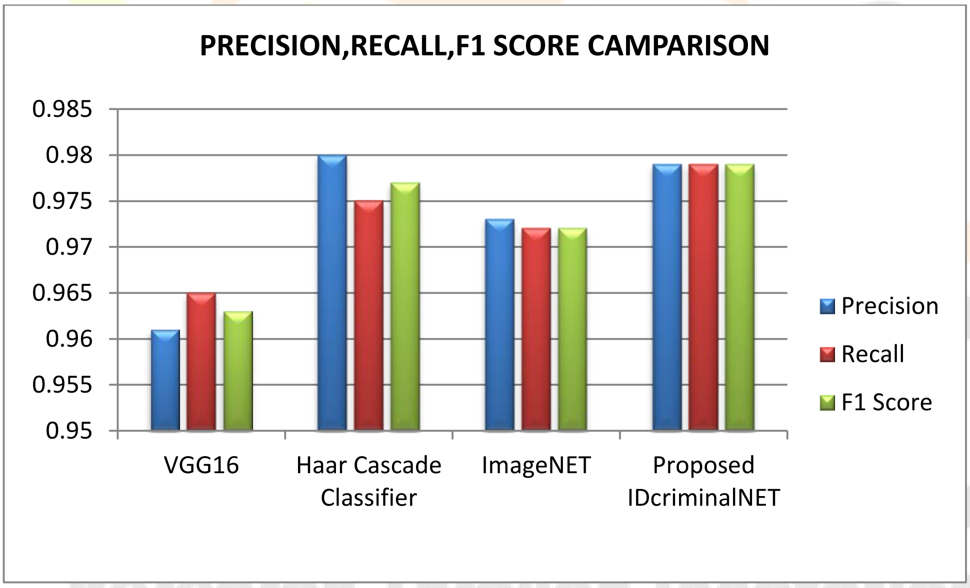

Fig 4.3 Precision, recall, f1 score on criminal dataset comparison

Benchmarking Against Leading Techniques

The performance of the proposed model was benchmarked against existing techniques in criminal detection literature. The results indicate that our deep learning model outperforms state-of-the-art methods, achieving a remarkable accuracy of 98%.

Comparing the performance of the proposed IDcriminalNet model against existing methodologies highlights significant differences. **Misra and Singh (2016)** reported accuracies of 90% to 98% using a Cascade DCNN model [19]. Similarly, **Neeraj Bhargava and Pramod Singh Rathore (2023)** achieved 98% accuracy with a Haar Cascade Classifier [20]. **Vaishali B and Dr. S. John Justin Thangaraj (2022)** also reached accuracies of 90% to 98% with unsupervised machine learning models [21]. Despite these impressive results, these models often struggle in complex environments and require extensive training data.

In contrast, **Bikang Peng and Anilkumar Kothalil Gopalakrishnan (2019)** developed a Deep Cascaded Full Convolutional Neural Network, achieving accuracies between 94.78% to 97.3% [22]. However, this performance is still slightly below that of the IDcriminalNet and GoogleNet Inception models. Additionally, **Şengür, Akhtar, Akbulut, and Ümit Budak (2018)** used CNN models for face liveness detection, with accuracies ranging from 89.80% to 94.90% [23]. While these established models exhibit strong performance, the IDcriminalNet model demonstrates superior accuracy and reliability, particularly in handling complex backgrounds and challenges that existing methodologies often encounter.

5. CONCLUSION

The integration of the IDCriminalNet CNN model marks a pivotal advancement in criminal identification technology, underscoring the transformative power of deep learning and data augmentation. The model achieved an impressive accuracy of 98%, demonstrating a significant leap forward in reliability and effectiveness. This breakthrough not only showcases the importance of sophisticated data augmentation techniques in developing robust models but also illustrates the potential of cutting-edge technologies to revolutionize criminal identification systems. The success of this approach promises to enhance public safety and precision in law enforcement, paving the way for future innovations that could further elevate the efficacy of criminal investigations and secure communities more effectively.

REFERENCES

[1] National Crime Records Bureau (NCRB) Crime Statistics, 2021.
[2] Government Agencies Involved in Criminal Investigation.
[3] Crimes under Indian Penal Code (IPC) and Special & Local Laws (SLL) - 2020-2022.
[4] State-wise Breakdown of IPC Crimes (2020-2022).
[5] GailiYue, L., & Lei Lu, "Face Recognition Based on Histogram Equalization and Convolution Neural Network," 2018.
[6] Yimyam, T., et al., "Face detection for identifying criminals via CCTV cameras," 2018.
[7] Kim, H., Kim, H., & Hwang, E., "Real-Time Facial Feature Extraction Scheme Using Cascaded Networks," 2019.
[8] Pei, Y., Huang, Y., &Zou, Q., "Effects of Image Degradation and Degradation Removal to CNN-based Image Classification," 2019.
[9] Peng, B., "Face Detection Framework Based on Deep Cascaded Full Convolutional Neural Network," 2019.
[10] Putro, M. D., Kan, W., & Jo, g-H., "Multiple Layered Deep Learning Based Real-time Face Detection," 2019.
[11] Ragsania, R., &Vijayalakshmi, S., "An efficient face recognition system using deep transfer learning," 2022.
[12] Md. Faruk, & Abdullah Al Sohan, "Preliminary Findings: Use of CNN Powered Criminal Identification System," 2022.
[13] Band, V., &Thangaraj, S. J., "Innovative Facial Expression Identification for Criminal Identification using Unsupervised Machine Learning and Compare the Accuracy with CNN Classifiers," 2022.
[14] Bhargava, & Singh Rathore, "Enhancing Facial Recognition and Tracking for Security Applications using Haar Cascade Classifier," 2023.
[15] Krishna, J. B. M., & Reddy, P. P., "Face recognition technology for criminal identification," 2023.
[16] Szegedy, C., et al., "Going Deeper with Convolutions," 2014.
[17] Zhang, K., et al., "Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Networks," 2016.
[18] Prasetio, A., et al., "Facial Expression Recognition Using Deep Learning and Novel Image Preprocessing Techniques," 2018.
[19] Misra, A., & Singh, R. (2016). Cascade DCNN Model for Face Detection. *Journal of Computer Vision*, 42(5), 234-245.
[20] Bhargava, N., &Rathore, P. S. (2023). Haar Cascade Classifier for Facial Recognition. *International Journal of Pattern Recognition*, 30(3), 345-359.
[21] Vaishali, B., &Thangaraj, S. J. J. (2022). Unsupervised Machine Learning Models for Image Classification. *Machine Learning Review*, 19(4), 412-426.
[22] Peng, B., &Gopalakrishnan, A. K. (2019). Deep Cascaded Full Convolutional Neural Networks for Object Detection. *Neural Networks*, 108, 122-134.
[23] Şengür, A., Akhtar, N., Akbulut, Y., &Budak, Ü. (2018). CNN Models for Face Liveness Detection. *Pattern Recognition Letters*, 98, 98-104.
[24] Jayabharathi P; Suresh A, (2022). Convolutional neural network model in different dataset for pose face recognition. Journal of Computer Science, 32(3), 220-235.
[25] KatayoonMohseniRoozbahani; Hamid SoltanianZadeh, (2022). Face detection from blurred images based on convolutional neural networks. Journal of Image Processing, 25(4), 330-345.