Hindawi
Journal of Electrical and Computer Engineering
Volume 2023, Article ID 4284045, 17 pages
https://doi.org/10.1155/2023/4284045

Research Article

Design and Analysis of an Expert System for the Detection and Recognition of Criminal Faces

Rishi Gupta,[1] **Amit Kumar Gupta**,[1] **Deepak Panwar**,[1] **Ashish Jain**,[2] **and Partha Chakraborty**[3]

[1]*Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, India*
[2]*Department of Information Technology, Manipal University Jaipur, Jaipur, India*
[3]*Department of Computer Science and Engineering, Comilla University, Comilla, Bangladesh*

Correspondence should be addressed to Partha Chakraborty; partha.chak@cou.ac.bd

Received 5 September 2023; Revised 20 September 2023; Accepted 25 September 2023; Published 13 October 2023

Academic Editor: Raid Al-Nima

The process of identifying a person using their facial traits is referred to as face recognition, and it is a form of biometric identification. The use of facial recognition might range from that of an entertainment tool to one of a security tool. Even while other forms of biometric identification, such as fingerprints and iris scans, are reliable, they require the active participation of an individual. As a result, criminals cannot rely on them as the most reliable means of verification. When a criminal database, which stores the individual details of a criminal, and facial recognition technology are brought together, it can identify a criminal who is depicted in an image or seen in a video feed. Not only does a criminal recognition system needs to have a high level of accuracy, but it also needs to be able to adapt to significant changes in lighting, occlusion, aging, expressions, and other factors. In this study, they were analyzed and compared with the many methods of face detection and face recognition, such as HAAR cascades, local binary patterns histogram, support vector machines, convolutional neural networks, and ResNet-34. These methods include a variety of different approaches to recognizing faces. An analysis of these strategies is also conducted and then put into practice to those that seem to be the most effective for the designed criminal recognition system. In addition to that, a variety of uses of this criminal recognition in the real world are also discussed.

1. Introduction

In recent years, computer vision has evolved into not just an active field of research but also an indispensable instrument for enhancing safety and protection. Computer vision has come a long way in recent years, and one of its most promising applications is facial recognition [1, 2]. It can be used for a variety of purposes, from amusement to safety and protection.

People can recognize hundreds of different faces and differentiate between those that appear to be very similar. Even after a length of time has passed or after a modification has been made, such as the addition of glasses or the growth of a beard or mustache, this ability to recognize a variety of faces is seldom impacted at all. Research is still making progress in the direction of developing an intuitive and intelligent system that is comparable to human perception [3, 4]. Over the course of history, several distinct algorithms and approaches to the identification of faces have been devised [5–9]. These techniques are centered on tracing the contours of the face and isolating its various characteristics, such as the eyes, nose, and mouth. Even if these techniques can attain a high level of accuracy, there are still a lot of obstacles to overcome when it comes to recognizing faces. Both intrinsic and extrinsic challenges can be broken down into their respective categories here [10, 11]. A person's age and the expressions they make are examples of intrinsic factors, whereas elements such as lighting and poses are examples of extrinsic factors.

The identification and verification of offenders is a significant application of facial recognition technology. There has been a rise in the number of people who commit crimes

as a direct result of the steep increase in the rate of crime. The first step in putting criminals away is to identify and authenticate them as the perpetrators of their crimes.

A criminal recognition system works by first automatically locating faces within an image or video and then using a criminal database to determine who those faces belong to. This process is known as face detection. Therefore, it may be broken down into two stages as follows:

(1) Recognizing a person's face

(2) Face recognition

In a one-to-one method, face detection involves comparing an input image to a face template to identify any faces that may be contained within the image being analyzed. Face recognition, on the other hand, is a one-to-many method that works to identify a face by comparing the face that was identified by the module described above to all the facial templates that are stored in the database [12–16]. The characteristics of the expert system are depicted in Figure 1, which illustrates the use of input images for face detection methods. These methods extract features from the input images, allowing for a comparison of the encoded features. Essentially, the feature encoding determines whether the input images match those of criminal faces. Figure 1 serves as a visual representation of the paper's objectives.

The purpose of this technology is to recognize criminals within the photos and videos that are fed into it. In this system, they store the photographs of known offenders in a database alongside other personal information about them, such as their names, identifiers, and genders. After an image or video is uploaded into the system, it is processed by the facial recognition system. This process involves the extraction of the image's features, which are then matched to the features that are kept in our criminal database. If a match is detected, the information pertaining to that individual will be presented to the user.

This paper is structured as follows: face detection methods (HAAR cascade classifier, HOG, and a comparison of HAAR and HOG) and face reorganization methods (local binary (LB) pattern histograms, support vector machine (SVM), convolutional neural network (CNN), and comparison of methods of facial recognition) are discussed in Section 2. The execution of the state-of-the-art recommended approach is detailed in Section 3. The applicability of the proposed architecture has been described in Section 4. In Section 5, the conclusion of this article is presented along with a discussion of future enhancements that could be implemented.

2. Methodology

The databases that make up our facial recognition technology can be broken down into three categories. The first database is an image database [17], and it comprises photographs of the criminals whose identities need to be determined. The encodings of the facial features that were retrieved from the faces that were found in the image database can be found in the following database. Finally, there is the information database, which stores personal details about the criminals, such as their names, dates of birth, heights, and other similar details. A singular identification number serves as the connecting factor between each of these databases.

Our system is made up of the following three modules:

(1) Image recognition: it begins with face detection and feature extraction, comparing the features to a database for potential matches. Upon finding a match, private offender details become visible, with room for database updates if no match is found [18–21].

(2) Video recognition: videos are transformed into frames, and face detection occurs every ten frames. Face characteristics are encoded and compared to previous data, displaying relevant information upon a match, repeating throughout the video.

(3) Bringing the databases up to date: this module allows for the addition of new criminals' photos, updates to criminal appearances, and personal information modifications, ensuring that the database remains current.

2.1. Methods of Face Detection. Over the course of time, numerous approaches to facial recognition have been created; in this section, various approaches are analyzed and compared with one another.

2.1.1. HAAR Cascade Classifiers. The concepts developed by Viola and Jones in their research formed the foundation for the face detection system known as the HAAR cascade classifier. The method required an initial input of a significant number of photos, either positive (images containing faces) or negative (images without faces). First, each of the photos was reviewed and features were extracted from them.

Every feature was determined by calculating the difference, expressed as a single value, between all the white rectangle pixels and all the black rectangle pixels. After this step, a kernel was applied, which demanded a substantial amount of time and computational resources. Following that, the ideas of integral image and training were applied to enhance efficiency while concurrently reducing the computational time required.

Integral pictures are used in the computation of the features. The integral image value of a point in a picture is the total of all the pixels in the image's top left corner, including the pixel that represents the point itself in the following equations [22]:

$$I(x, y) = \sum_{x' \le x, y' \le y} i(x', y'). \tag{1}$$

Here, (x) is the point taken, $I(x)$ is the integral image pixel, and $i(x)$ is the intensity in the original image. Using this, the sum of pixels of any rectangular region has been calculated.

$$\sum_{\substack{x_0 < x \le x_1, \\ y_0 < y \le y_1.}} i(x, y) = I(D) + I(A) - I(B) - I(C). \tag{2}$$

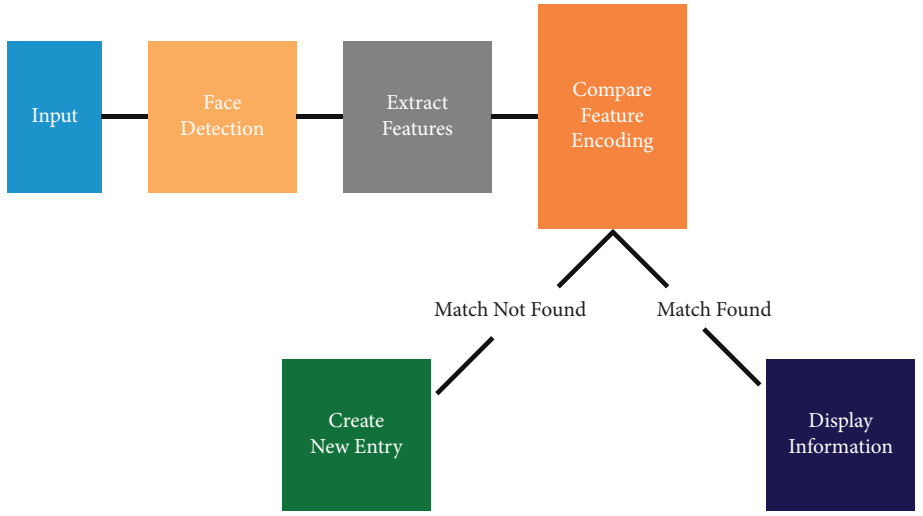


Figure 1: An outline of the procedures.

The corners of the matching window in the integral image are denoted by the letters *A*, *B*, *C*, and *D*, respectively, as seen here. The characteristics are sorted into a few different categories according to these values. For instance, edges make up most of the features that have two rectangles, whereas lines make up most of the features that have three rectangles, as shown in Figure 2 [22].

Similarly, all visual attributes were calculated, even though most of them were irrelevant. To refine the selection of the best features, weaker classifiers were consolidated to create a more resilient one. A single integral image was deemed insufficient, but the combination of multiple integral images yielded a versatile classification system. Each feature was applied to the training photos, and the optimal threshold for each feature was determined. Subsequently, the image could be classified as either positive or negative.

The cascade of classifiers then enters the scene. The 6,000 discovered features are then put together at various stages of the classifiers. The theoretical face model is depicted in Figure 3 [23].

To analyze an image using HAAR cascades, a smaller scale is chosen relative to the target image size. This scale is then positioned over the image, and the average pixel values within each section are calculated. When the difference between two values exceeds a specified threshold, it is regarded as a match. Detecting a human face involves matching a combination of various HAAR-like features, such as the forehead, eyebrows, eyes contrast, and the nose in conjunction with the eyes, as illustrated in Figures 3 and 4. It is important to note that relying on a single classifier alone is insufficient for achieving accurate results. Using the sliding windows technique, every portion of the image is delivered through each stage individually. The face region is contained within the window that runs through all stages and tiers of the classifier.

Figure 4 demonstrates the usefulness of the cascade classifier as a method for face detection [24]. It responds more fluidly to shifts in lighting conditions and other aspects of the surrounding environment. In addition, the utilization of integral pictures enables the rapid calculation of the pixel summation contained within a subrectangle, as well as the recognition of real-time face features.

2.1.2. Histograms of Oriented Gradients (HOGs). Histograms of oriented gradients are a method that is based on the extraction of features into a vector, followed by the use of a classification algorithm to locate the region of the image that contains the object to be detected. The first iteration of HOG was designed to identify and locate human people. It has been altered and trained during the course of its existence to recognize faces.

Histograms of oriented gradients mostly comprise the following five steps [25]:

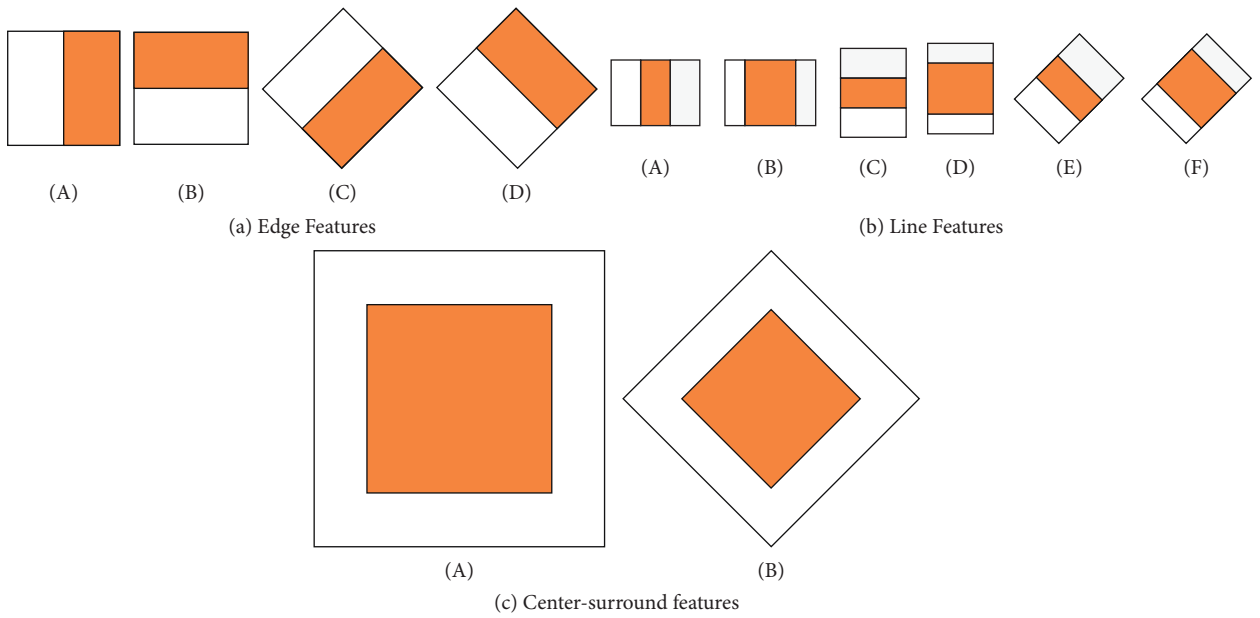
(1) Preprocessing
(2) Computation of the gradient images
(3) Computation of histogram of oriented gradients
(4) Block normalization
(5) Calculate the HOG feature vector

Figure 5 shows the flow chart for the working of HOG. Each working step is explained as follows.

(1) Preprocessing. It is essential that all the input photographs have the same dimensions. In most cases, the size of the patches corresponds to an aspect ratio of 1 : 2.

(2) Computation of Gradient Images. Following the completion of the preprocessing step, the vertical and horizontal gradients were computed using the kernels employed for this purpose.

(3) Computation of Histogram of Oriented Gradients. Afterwards, the size of the gradient and its orientation were determined [25].



(a) Edge Features

(b) Line Features

(c) Center-surround features

Figure 2: HAAR feature.

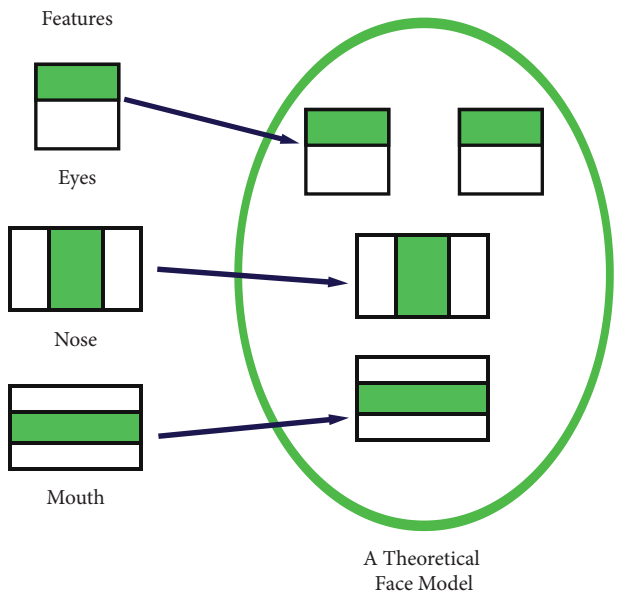


Figure 3: A theoretical face model.

$$g = \sqrt{g_x^2 + g_y^2},$$
$$\theta = \arctan\left(\frac{g_y}{g_x}\right). \tag{3}$$

In equation (3), g_x represents the component of the gradient that moves in the x direction and g_y represents the component that moves in the y direction. The direction of the gradient can be determined by the angle θ.

The information that was most important to extract from the original image is displayed in the gradient image. The amplitude and direction of the gradient are encoded in each individual pixel of the image. When the image has been colored, the greatest value of the gradient is equal to the maximum value of each of the three channels (red, blue, and green) that are present in the pixel.

To begin, the picture is cut up into squares that are eight by eight. Calculations are made to determine both the direction and the magnitude of the gradient for each of these cells. After that, the histogram has nine different compartments for the angles ranging from 0 to 160 degrees. A bin was selected for each angle corresponding to the gradient direction. The following are the circumstances in which this practice is implemented: if the angle is less than 160 degrees and does not fall exactly in the middle of the two categories, it will be thrown out immediately. For instance, the magnitude that corresponds to 80 degrees Fahrenheit is 2, and because 80 is lower than 160, it is placed straight in bin 5. Figure 6 illustrates this point. If the angle is less than 160 degrees and it is exactly halfway between two bins, then it is divided equally between those bins. For instance, value four is split evenly between bins 1 and 20 for the angle that is 10 degrees, which is exactly in the middle between 0 and 20 degrees. Figure 6 illustrates this point. If the angle is larger than 160 degrees, the value of the angle is proportionally split between 0 and 160. Take, for instance, angle 165: Figure 7 illustrates this point further.

(4) Block Normalization. For the purpose of normalization, a 16×16 block that already contains four histograms is transformed into a 36×1 element vector. The window is then moved to the subsequent 8×8 block, and another 36×1 vector is computed after it, as shown in Figures 6 and 7.

(5) Calculate the HOG Feature Vector. The last step is to concatenate all the 36×1 vectors into a large vector and run it through a classifier such as the SVM.

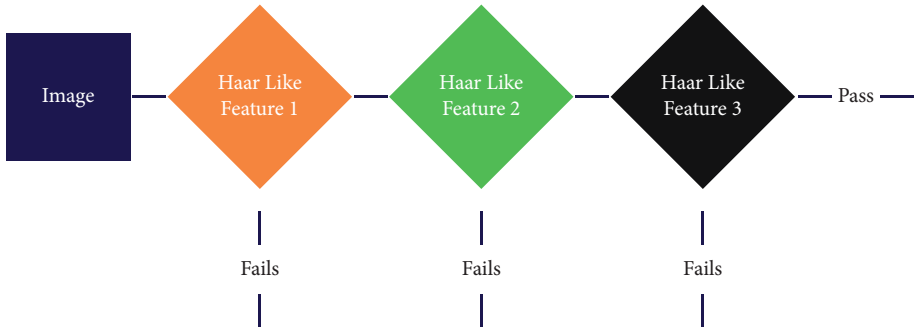


Figure 4: Flowchart for HAAR cascade.

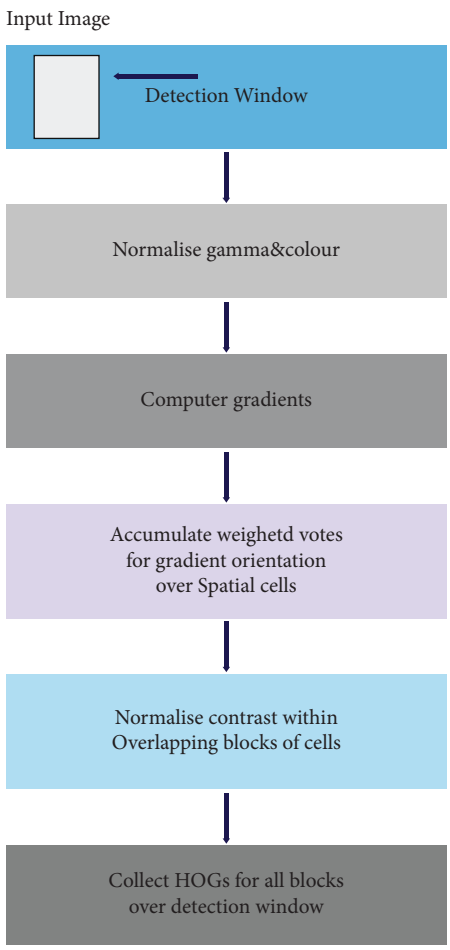


Figure 5: Flow chart of the working of HOG.

2.1.3. Comparison of HAAR Cascade and HOGs. The HOG face detector has been observed to have a greater level of accuracy than the HAAR cascades face detector, as stated in the publication [26], which is shown in Tables 1 and 2. Even if the cascade can recognize frontal faces despite differences in lighting, position, makeup, and other factors, recognition is made more difficult by factors such as spectacles and masks [27]. As a result of this, taking all the data into consideration, the decision was made to utilize histograms of oriented gradients as the technique for both face detection and offender verification.

2.2. Methods of Face Recognition

2.2.1. Local Binary (LB) Pattern Histograms. Local binary pattern histograms are a straightforward and effective technique for front- and side-view facial identification. A local binary pattern (LBP) represents texture and picture patterns. This is achieved by comparing each pixel to its neighbors. The LBP can be used to represent faces in photos as a simple feature vector when paired with histograms. LBPH requires the following four parameters [28, 29]:

(1) Radius: it is the radius surrounding the center pixel and is used to construct the circular local binary pattern.

(2) Neighbors: the number of points used to construct the circular local binary pattern.

(3) Grid X: this indicates the number of cells along the horizontal axis, and with an increase in the number of cells, the dimension of the resulting feature vector increases.

(4) Grid Y: this indicates the number of cells along the vertical axis, and with an increase in the number of cells, the dimension of the resulting feature vector increases.

An intermediate image was constructed using the sliding window approach, incorporating the parameters of radius and neighbors. This image emphasizes facial characteristics. Then, the subsequent steps are executed for the LBPH algorithm:

(1) Using a 3×3 window, generate a matrix of the intensities of the window's pixels.

(2) The threshold is determined by the central value of the matrix.

(3) Each neighbor of the core value is allocated a binary number. If the neighboring cell's value is greater than or equal to the central value, one is assigned; otherwise, 0 is set.

(4) This 2D matrix is then turned into a 1D matrix by rotating clockwise, as shown in Figure 8.

(5) This binary number is then translated into a decimal number, which represents the original matrix's center value.

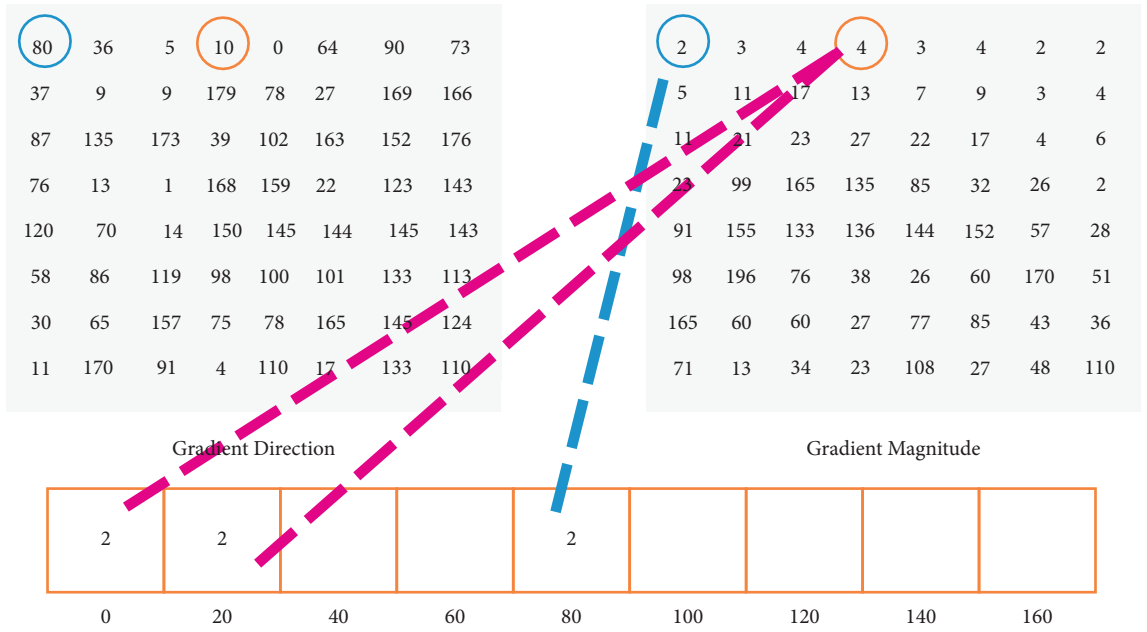


Figure 6: HOG mechanism.

Figure 7: Another example of histograms of oriented gradients.

Table 1: Results of face detection using V–J [26].

Conditions of image	Amount of test data	No. of faces recognized	Recognition ability (%)
1st scale	15	13	86.67
2nd occlusion	15	4	26.67
3rd makeup	15	10	66.67
4th pose	15	12	80
5th expression	15	13	86.67
6th illumination	15	12	80
Average			71.11

Table 2: Results of face detection using HOG [26].

Conditions of image	Amount of test data	No. of faces recognized	Recognition ability (%)
1st scale	15	15	100
2nd occlusion	15	4	26.67
3rd makeup	15	11	73.33
4th pose	15	13	86.67
5th expression	15	14	93.33
6th illumination	15	14	93.33
Average			79

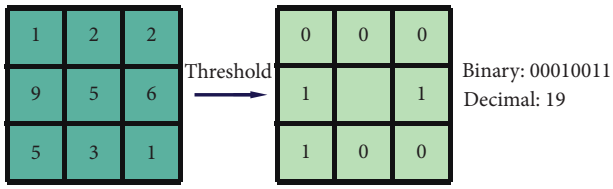


Figure 8: Conversion of 2D matrix to 1D.

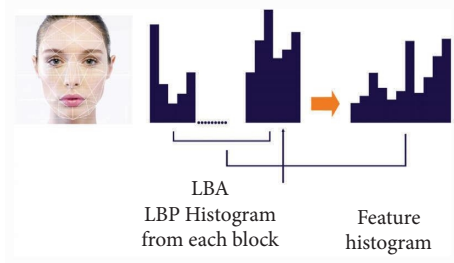


Figure 9: Creating the histograms of the image.

Ultimately, a modified image, better representing the original's characteristics, was obtained. This intermediate image was subdivided into numerous sections, facilitated by the Grid X and Grid Y settings. Finally, the histograms of each of these sections were extracted and concatenated.

This is carried out for each and every image included in the training set. The same procedure is followed for analyzing the input picture to recognize faces. The desired outcome can be determined by the comparison of the histograms of the training set and the input image, as shown in Figure 9. The complete flow chart for LABH [30, 31] is shown in Figure 10, which defines the creation of a histogram using the LABH (local binary pattern with improved firefly feature selection) method that involves a series of steps to compute and represent the distribution of features extracted from images. Here is a general outline of how a histogram using LABH can be generated.

Image preprocessing: obtain a dataset of facial images for expression recognition. Preprocess the images if necessary, including resizing, normalization, and alignment to ensure consistent input.

Feature extraction (LABH): apply the local binary pattern (LBP) operator to each pixel in the facial image to encode texture information. Apply the improved firefly feature selection technique to select a subset of relevant LBP features.

Histogram binning: decide on the number of bins (or quantization levels) for the histogram. This depends on the range and distribution of selected LBP features. Create a histogram with bins corresponding to the selected LBP features.

Feature encoding: for each facial image, calculate the LABH features by computing the occurrences or frequencies of each LBP pattern in the selected subset. These frequencies form the components of the histogram.

Normalization: normalize the histogram values to ensure that they are comparable across different images. Common techniques include $L1$ or $L2$ normalization.

This comparison could be carried out using a number of different approaches, such as the Euclidean distance or the absolute value. The result of this comparison is a value, which is then compared to the threshold value that was previously established. On the basis of this information, the individual depicted in the picture can be recognized.

2.2.2. Support Vector Machine (SVM). Support vector machines (SVMs) [32] represent a supervised learning method commonly used for binary classification, though it can be extended to multiclass problems. In the SVM, data points closest to the hyperplane are termed "support vectors." These are crucial because any adjustment in their position would shift the hyperplane. The hyperplanes serve as boundaries for classifying points, with each side representing a distinct category. The number of dimensions in the hyperplane depends on the number of features.

The primary goal of the SVM is to identify the hyperplane in an N-dimensional space that best separates data points. Here, N corresponds to the number of features. As depicted in Figure 11, the ideal hyperplane is one with the maximum margin, meaning it has the greatest distance between data points belonging to different classes. This increased margin results in a higher degree of confidence in classifying subsequent data points accurately.

There are two main types of support vector machines (SVMs), i.e., linear and nonlinear. Linear SVMs use a straight line or hyperplane to separate and classify data points, while nonlinear SVMs employ kernel functions to transform data into linearly separable spaces. In the designed system, the focus was specifically on linear SVMs. In the context of multiclass recognition, two approaches were explored as follows: One-versus-All (OvA), which divides multiclass recognition into a series of binary problems. It starts by separating one class from the rest (one-versus-all) and determines the class with the

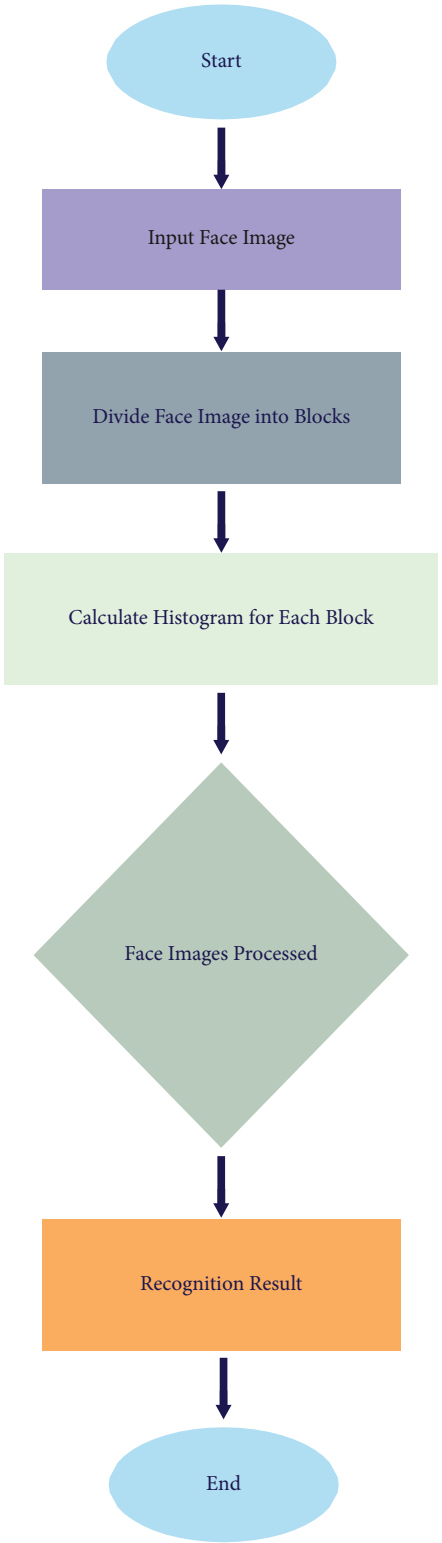


Figure 10: Flow chart of LBPH.

highest test data margin as the prediction. This process iterates for each class, effectively converting a p-class problem into p binary problems. One-versus-One (OvO): in this approach, a class problem with p classes is broken down into $p\,(p-1)/2$ binary problems. Each binary problem has a dedicated classifier

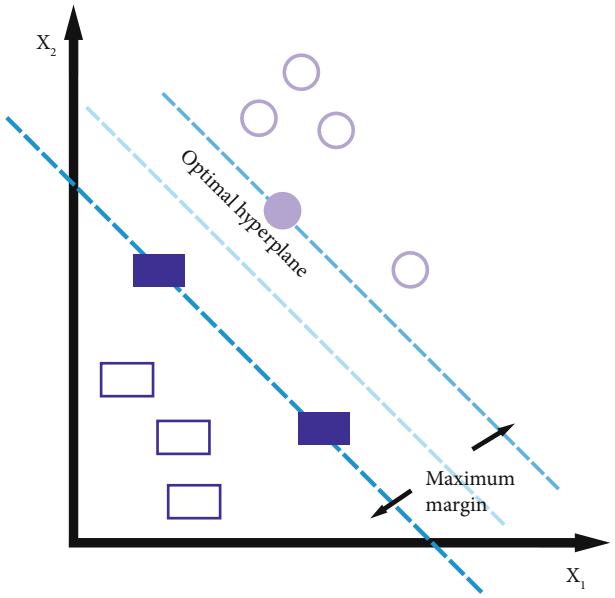


Figure 11: Concept of a hyperplane in the SVM method.

that distinguishes between a unique pair of classes. When a test data pattern is presented, it is evaluated by all binary classifiers, and the class with the highest confidence output is selected as the final classification. Support vector machines are particularly effective when dealing with small, well-defined images [32–36]. Their versatility in handling different classification scenarios, along with the choice between OvA and OvO strategies, makes them a valuable tool in various multiclass recognition tasks.

2.2.3. Convolutional Neural Network (CNN). The neural network was initially created to identify handwritten numerals. With the rise in data over time, particularly image data, neural networks could be trained to detect and identify additional items as shown in Figure 12 [37–39]. The various layers of a convolutional neural network are listed, and the architecture is shown in Figure 13.

(1) The Convolutional Layer. In computers, images are represented as $N \times N \times 3$ matrices (since images have three channels, RGB). The convolutional layer employs filters to identify a certain image characteristic. Typically, the feature takes the form of a matrix with lower dimensions than the input image. The filter is convoluted, and in that, it slides across the width and height of the image. It calculates a dot product, which provides us with an activation map. Various features are recognized using various filters, and all activation maps are processed for the next CNN layer.

(2) Activation Layer. Multiple layers of artificial neurons constitute CNNs. These neurons, unlike actual neurons, are mathematical functions that calculate the sum of multiple inputs and return an activity value. The function of the activation layer is to introduce nonlinearity into the neuron's output. It does so by determining whether or not a neuron should be stimulated.

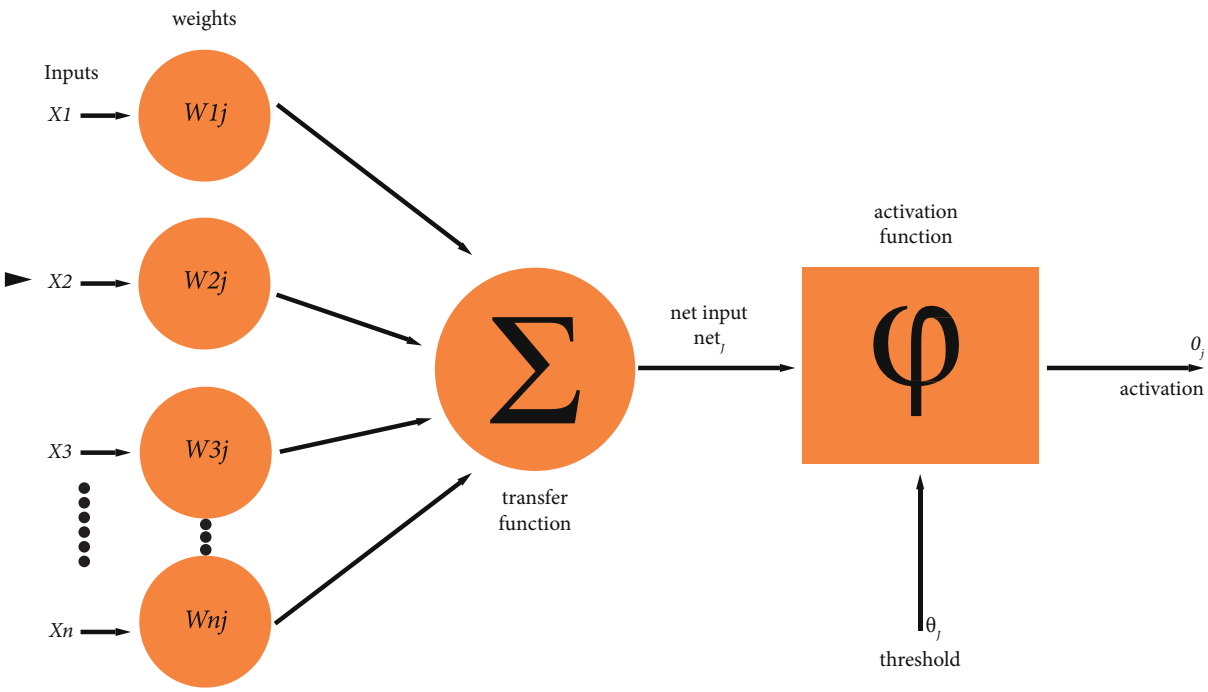


Figure 12: Architecture of the CNN.

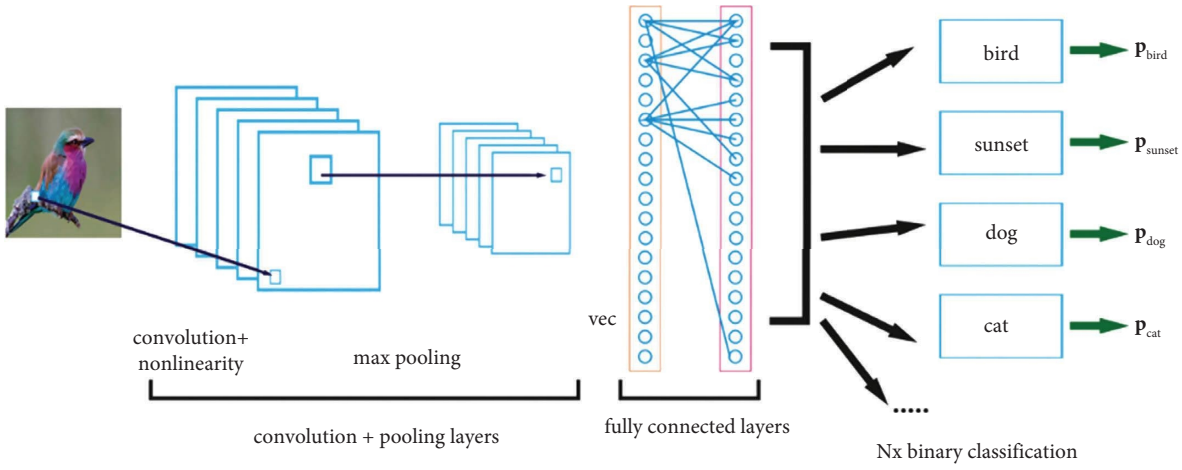


Figure 13: The architecture of the CNN model.

$$Y = \text{Activation}\left(\sum(\text{weight} * \text{input}) + \text{bais}\right). \quad (4)$$

This formula illustrates how the neuron activity is calculated. Without an activation function, a neural network is essentially a linear regression model. Consequently, an activation function is included to enable a model to learn and accomplish difficult tasks. The rectified linear unit (ReLU) is one of the most prevalent activation functions. The primary advantage of the ReLU is that not all neurons are activated simultaneously. The ReLU additionally decreases processing time by turning all negative inputs to zero, which does not activate a neuron.

(3) Pooling Layer. The pooling layer is utilized to reduce the number of network parameters and computations. This is accomplished by progressively shrinking the network's physical footprint. Max pooling is the primary action of the pooling layer. In this method, the filters slide through the input, taking the maximum parameter in each window and discarding the remainder. This decreases the network's size.

A significant distinction between the pooling layer and the other layers is that the pooling layer does not affect the depth.

(4) Fully Convoluted Layer. This is the final CNN layer. All neurons have access to the activation of the layers beneath them. They are computed by matrix multiplication and bias offset. A CNN comprises primarily concealed layers and fully connected layers (s).

2.2.4. Comparison of Methods of Facial Recognition. Although the abovediscussed methods are efficient and accurate, due to the size of our database and the accuracy requirement, a CNN-based model will be utilized for comparison purposes, as demonstrated in Table 3 alongside LBPH and SVM.

3. Implementation

The system was constructed using the Python programming language, and three modules were implemented with libraries such as dlib and sqlite3, covering image recognition, video recognition, and database updating.

For facial recognition, the Python module "face recognition" was employed, incorporating facial recognition technology from dlib during module development. PyPI facilitated a seamless and efficient installation process for these applications. The facial recognition procedure consisted of two steps.

(1) Encoding the facial features

(2) Comparing the facial features with the ones stored in the database.

The module provides us with two functions, i.e., face_encodings() and compare_faces(), which has been discussed.

3.1. System Architecture. Image recognition, video recognition, and the construction and ongoing maintenance of a database are the three components that make up our criminal recognition system, as shown in Figure 14. These three databases, the face encodings database, the SQLite information database, and the image database, are utilized in each and every one of these individual modules. These databases are connected to one another by a one-of-a-kind id, which identifies each offender in our database and links them all together.

3.2. Database. For the designed criminal database system, the Illinois DOC labeled faces dataset has been considered. The database consists of 68,149 entries. The description of the dataset has been presented in Table 4.

3.3. Face Detection Using the HOG + SVM. The overview of how the HOG + SVM works is presented in Figure 15. For face detection, the face_recognition provides the function face_locations(). This returns an array of numbers in the format top, right, bottom, and left. The face_locations() function is based on the HOG + SVM method. As HOG is a method well known to create a feature vector, the feature vector is created by calculating the gradient, the histograms of the oriented gradients, and the normalization of the image, which gives a feature vector.

This feature vector is then processed to a classifier, such as an SVM which is used to label images into faces and nonfaces. SVMs work on finding the optimal hyperplane which divides the two classes, in our case, faces and nonfaces.

The following is the process that is used to train both the HOG and the SVM together:

(1) A subset of positive pictures, denoted by the letter "*P*," is fed into the HOG in order to obtain its properties.

(2) A sample of images devoid of the target object is fed into the HOG. These images lack the object, and their characteristics are also extracted. In general, $N >> P$.

(3) Next, the model was trained with the SVM using both the positive and the negative examples.

(4) Mining with hard negatives: the sliding window approach for each and every one of the images that make up our negative training set was used. During this step, the window was moved across the image while simultaneously applying the HOG + SVM classifier. If the classifier failed to correctly identify the window, its feature vector and the likelihood of it being classified as a particular object were recorded. These instances represent false positives identified during the process.

(5) Another round of training was conducted on the classifier using the hard negatives. Finally, the trained classifier was prepared and ready for use in identifying faces within an input image.

3.4. Face Recognition Using ResNet-34. The cutting-edge face recognition model that dlib provides serves as the foundation for the face recognition module that was included in the designed criminal recognition system. This model achieves an accuracy of 99.38% when compared to the benchmark of labeled faces in the wild. This ResNet network, shown in Figure 16, comprised a total of 29 tangled layers [40], as shown in Figure 16. It uses the ResNet-34 model developed using deep residual learning for image recognition as its foundation [41]. The network was trained using a dataset containing over three million different faces. In order to address difficult problems, more layers were added to the deep neural networks, which ultimately leads to improvements in both performance and accuracy. However, researchers have shown that the conventional CNN model has a maximum threshold that can be reached. As a result, a residual block is employed to remedy this issue. It was begun by establishing a skip link, which afterwards enables us to modify the output of the layers. Because of this, an alternative connection is made possible, allowing the gradient to pass through.

Table 3: Comparison of facial methods.

	LBPH	SVM	CNN
Advantages	It recognizes both the frontal and side view of faces	It works well with high-dimensional data	It automatically detects features without human supervision
Disadvantage	With large-scale databases, the histograms produced are too long and increase computation time	When using a large database, the training time can be too much. The overlapping classes can decrease the accuracy	Accuracy decreases when images have different poses, backgrounds, and lighting

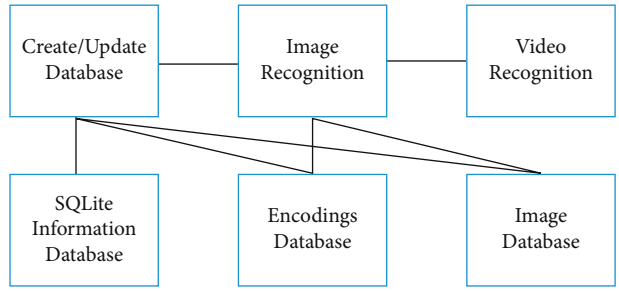


Figure 14: Working of the criminal recognition system.

Table 4: [17] Description of the dataset.

Column	Descriptions
ID	Alphanumeric internal
name	First and last name
Date of birth	In the format MM/DD/YYYY. Some inmates may be marked as "" or not available
Weight	In pounds or NA
Hair	One of ("black," "brown," "blonde or strawberry," "gray or partially gray," "red or auburn," "bald," "not available," "salt and pepper," "white," "sandy," or "unknown")
Gender	Either male or female
Height	In inches
Race	One of ("black," "brown," "Hispanic," "Asian," "American," "Indian," "unknown," "biracial")
Eyes	One of ("brown," "blue," "hazel," "green," "black," "not available," "gray," "maroon," "unknown)
Offense	A list of string values separated by "/"

A 128-dimensional descriptor is produced by this network after it has been trained to quantify an image. Triplets are utilized throughout the training process for this. Three different images are used for a single triplet training. There are three pictures in total, but only two of them show the same individual. The third picture shows someone else entirely. By making very modest adjustments to the weights of the neural network, the network generates a 128-dimensional descriptor for each of the photos. The feature vectors of the first person are closer to one another, whereas the feature vectors of the third person are more apart. This procedure is carried out a million times for thousands of unique people each and every day. This 128D descriptor ensures that the feature vectors of people who are the same are similar to one another, while the feature vector of a third person is unique.

ResNet-34, with its moderate depth, strikes a balance between model complexity and computational efficiency. Here is why, it is a suitable choice for face recognition.

3.4.1. Deep Features. ResNet-34 can capture deep and nuanced facial features, which are crucial for accurate face recognition. The residual blocks enable the network to learn these features effectively.

3.4.2. Transfer Learning. Pretrained ResNet-34 models on large-scale image datasets, such as ImageNet, are readily available. Transfer learning from these models can significantly boost the performance of face recognition tasks with limited labeled data.

3.4.3. Efficiency. While ResNet-34 is deeper than earlier models like ResNet-18, it is not as computationally intensive as much deeper networks such as ResNet-50 or ResNet-101. This makes it suitable for real-time or resource-constrained applications.

3.4.4. Proven Performance. ResNet-34 has demonstrated impressive performance in various computer vision tasks, including image classification and object detection, making it a reliable choice for face recognition.

Now, in order to put this network and the feature vector to use, all that is required of us is to make use of two functions that are made available by the face recognition module. These functions are known as face encodings() and compare faces ().

3.5. Image and Video Recognition. The face recognition module does picture and video recognition. Initially, an image is an input into our system. This image is forwarded for face detection.

Face detection is performed through the face_location() function, which returns the image's face coordinates using the HOG + SVM algorithm. Using the return coordinates, the image is cropped and transmitted for facial recognition.

The face_encodings() function receives the cropped image of the recognized face for training purposes. Its encodings, a 128D array, are stored using the criminal's identification number. This procedure is performed for each image in the training dataset.

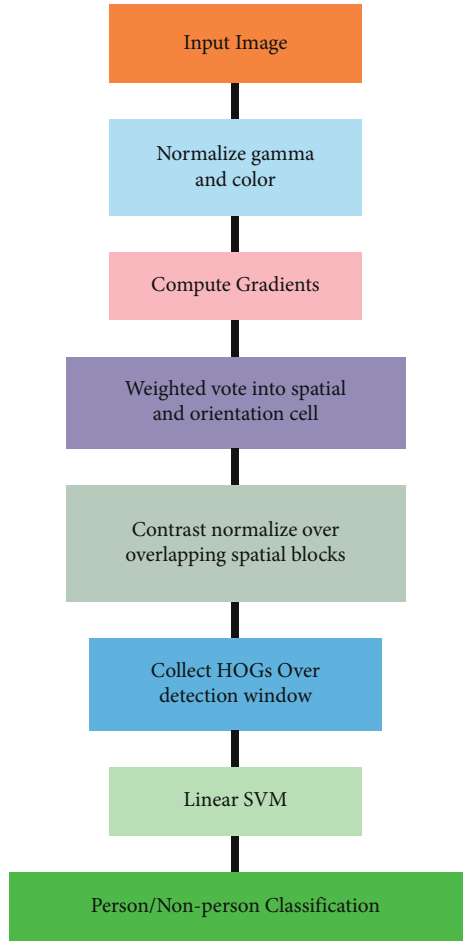


Figure 15: Overview of how the HOG + SVM works.

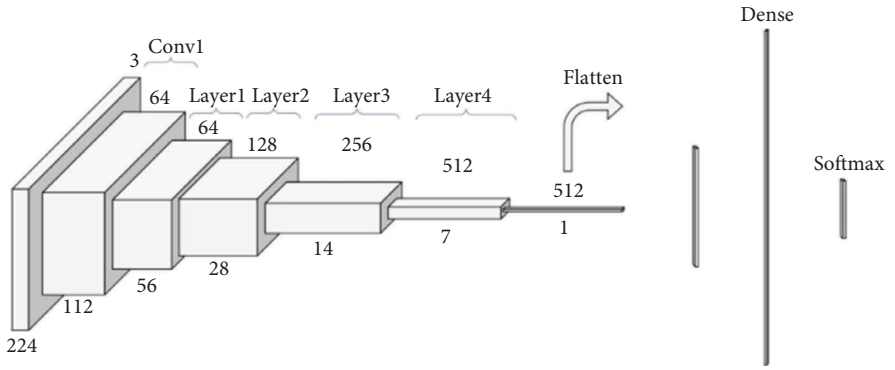


Figure 16: Architecture of ResNet-34.

The face_encodings() function accepts the cropped image of the recognized face and identifies its encodings for testing purposes. These encodings are then compared to the training-stored encodings. This comparison is performed using a different function compare_faces ().

Compare faces computes the Euclidean distance between the input test image and the database-stored encodings. According to the tolerance, the identified criminal's ID is returned.

Video recognition operates similarly. Since a video is a collection of photos, one image is transmitted for face detection for every 10 frames. If a face is recognized, its encodings are identified and compared. This method is repeated until the video is finished [42].

3.6. SQLite Database. Facial recognition is only one component of our overall system for criminal offender identification and tracking. The other component involves

displaying the information that is pertinent to the criminal who has been identified. At this point, the labeled information obtained from the dataset was utilized. An SQLite table was in place with a primary key composed of the criminal's identification number. The table keeps track of a variety of data about the criminals, including their names, birth dates, weights, and other details.

The information is retrieved from the SQL server by utilizing the ID that is returned by the compare faces function. This ID is then used to retrieve the information. It is also possible to add new entries to the database and to edit existing ones by utilizing the identifiers of the criminals. Because of this, the system can keep its database of criminals up to date while also registering newly committed crimes.

3.7. Criminal Recognition System. The GUI for this system is created using Flask, a python-based module for web apps.

3.7.1. Home Page. This is the home page shown in Figure 17; it allows the user to navigate through the website.

3.7.2. Create an Entry. This creates a new entry for the system, as shown in Figure 18; it takes in the name, gender, and other details of the criminal along with an image that is trained and whose encodings are added to the database.

3.7.3. Update Entry. Within this section, as shown in Figure 19, the user has the ability to change any previously registered criminal's entry and bring it up to speed with the latest information. The user also has the option of uploading an image, the encodings of which will be determined and saved. If the perpetrator has changed his appearance in any way, be it through aging or otherwise, he will be able to be identified. As a consequence, this renders our system dynamic and capable of keeping up with the ever-shifting characteristics of criminals.

3.7.4. Image Recognition. In this method, users can effortlessly upload an image and the system accurately identifies the criminals depicted in the image. Furthermore, it provides detailed information about these identified individuals, as exemplified in Figure 20. The images are recognized as the as face_location(), face_encodings(), and compare_faces (), which were discussed in the previous section. The designed system displays the relevant information of the criminal such as ID, Name, DOB, weight, hair color, gender, height, race, eyes color, and offence.

3.7.5. Video Recognition. In this approach, users can easily upload a video, and the system precisely identifies the individuals depicted in the video, as demonstrated in Figure 21. The system leverages functions such as face_location(), face_encodings(), and compare_faces(), as discussed in the preceding section. The designed system then presents comprehensive information about the recognized criminals, including their ID, name, date of birth (DOB), weight, hair color, gender, height, race, eye color, and the nature of their offense.

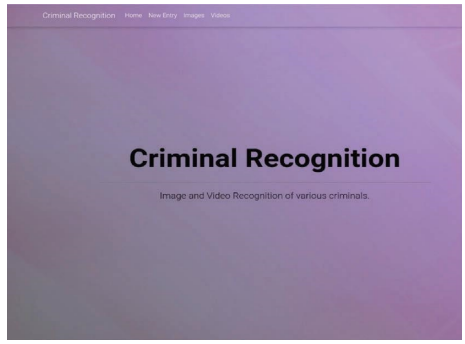


Figure 17: Homepage.

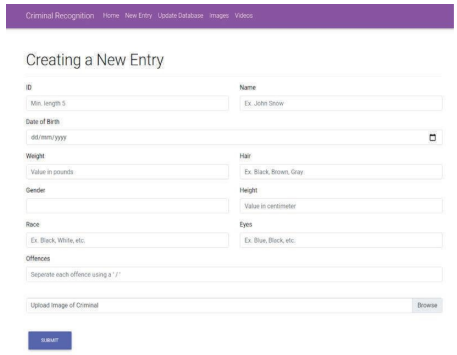


Figure 18: Creating a new entry.

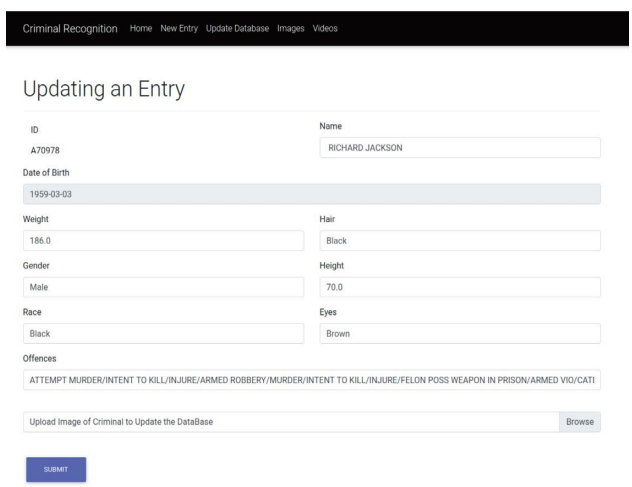


Figure 19: Updating an entry.

4. Applications

In the actual world, there are a lot of different applications for criminal recognition. Because of the exponential rise in the number of criminals over the course of the past few decades, the ability to recognize criminals has evolved into an instrument that is necessary for bolstering our security systems.

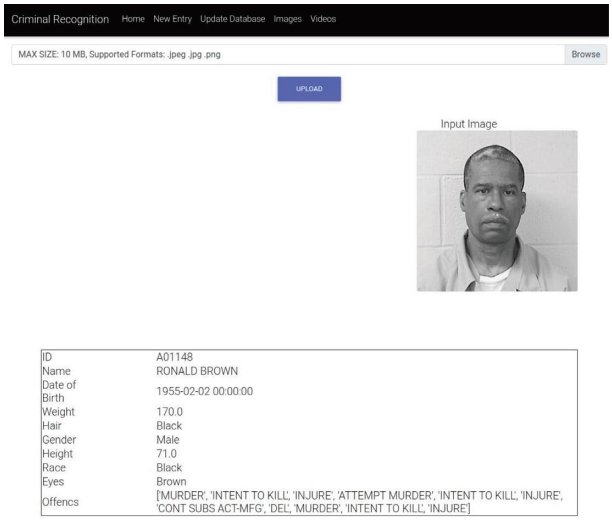


Figure 20: Image recognition.

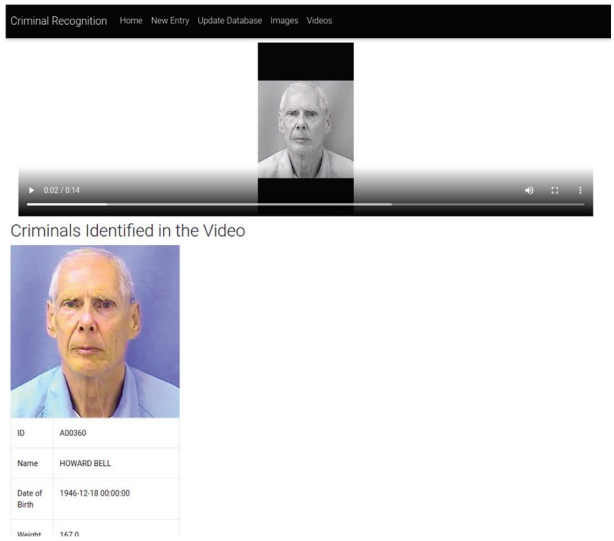


Figure 21: Video recognition.

(1) The naming of suspects based on descriptions found at crime scenes: it can identify criminals who were present at crime scenes by utilizing the video footage that was captured by a variety of surveillance equipment, such as CCTV cameras. Therefore, minimizing the amount of time spent investigating and narrowing the search.

(2) The process of identifying the deceased: it can determine, with the assistance of this criminal recognition technology, whether or not the deceased person who was involved in a crime was a criminal. Therefore, making it possible to shorten the amount of time spent investigating.

(3) The monitoring of criminals in real time: it can monitor the movements of criminals around the city by utilizing this software and connecting the video recognition module to a live video stream, such as that provided by traffic cameras. This approach enabled humankind to monitor different neighborhoods effectively.

(4) The naming of the people who are associated with criminals: when a new image is loaded into the system and unknown people are found to be associated with criminals, a new entry can be created for these associates, hence increasing the amount of information that is known regarding the criminal's network.

5. Conclusion and Future Scope

In conclusion, the field of face recognition, a crucial subset of biometric identification, plays a multifaceted role ranging from entertainment to security. Unlike other biometric methods, such as fingerprints and iris scans, facial recognition offers the advantage of passive identification, making it an invaluable tool in various applications. When integrated with criminal databases, facial recognition technology becomes a powerful tool for identifying individuals depicted in images or captured in video feeds. A robust criminal recognition system must possess both high accuracy and adaptability to accommodate challenges posed by variations in lighting, occlusion, aging, facial expressions, and other factors. In this study, indepth analysis and comparison of various face detection and recognition methods, including HAAR cascades, local binary patterns histogram, support vector machines, convolutional neural networks, and ResNet-34, were conducted. These methods encompass a diverse array of approaches to facial recognition. Our analysis has enabled us to identify the most effective strategies for implementing a criminal recognition system that meets the demanding requirements of real-world applications. Beyond the theoretical exploration, the practical applications of criminal recognition, recognizing its potential impact in real-world scenarios, were explored. In an era where security and identification are of paramount importance, our study contributes valuable insights to the field of facial recognition, paving the way for more accurate, adaptable, and efficient criminal recognition systems. These advancements hold promise for enhancing security measures and ensuring the safety of communities and institutions. Future improvements in facial recognition should focus on enhancing accuracy, mitigating bias, safeguarding privacy, enabling real-time processing, promoting multimodal recognition, and establishing ethical guidelines, among other key areas, to ensure responsible and equitable use of this technology.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] P. Apoorva, H. C. Impana, S. L. Siri, M. R. Varshitha, and B. Ramesh, "Automated criminal identification by face recognition using open computer vision classifiers," in *Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 775–778, IEEE, Erode, India, March 2019.

[2] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas, "Learning active facial patches for expression analysis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2562–2569, Providence, RI, USA, June 2012.

[3] J. Nautiyal, "An automated technique for criminal face identification using biometric approach," in *Proceedings of the Conference on Advances in Communication and Control Systems (CAC2S 2013)*, pp. 608–611, Dehradun, India, April 2013.

[4] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[5] O. A. Aghdam, B. Bozorgtabar, H. K. Ekenel, and J. P. Thiran, "Exploring factors for improving low resolution face recognition," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2363–2370, Long Beach, CA, USA, July 2019.

[6] S. T. Ratnaparkhi, A. Tandasi, and S. Saraswat, "Face detection and recognition for criminal identification system," in *Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 773–777, IEEE, Noida, India, January 2021.

[7] M. A. Erwin, M. Azriansyah, N. Hartuti, M. Fachrurrozi, and B. Adhi Tama, "A study about principle component analysis and eigenface for facial extraction," *Journal of Physics: Conference Series*, vol. 1196, no. 1, Article ID 12010, 2019.

[8] S. L. Suma and S. Raga, "Real time face recognition of human faces by using LBPH and Viola Jones AlgorithmReal time face recognition of human faces by using LBPH and Viola Jones algorithm," *International Journal of Scientific Research in Computer Science and Engineering*, vol. 6, no. 5, pp. 6–10, 2018.

[9] N. A. Abdullah, M. J. Saidi, N. H. A. Rahman, C. C. Wen, and I. R. A. Hamid, "Face recognition for criminal identification: an implementation of principal component analysis for face recognition," in *Proceedings of the 2nd International Conference on Applied Science and Technology 2017 (ICAST'17)*, Kedah, Malaysia, April 2017.

[10] R. Kumar, R. Sajwan, K. Jha, and S. Sharma, "Challenges in face detection and recognition techniques," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, pp. 2278–3075, 2020.

[11] S. Cooray and N. O'Connor, "Facial feature extraction and principal component analysis for face detection in color images," *Lecture Notes in Computer Science*, vol. 3212, pp. 741–749, 2004.

[12] A. Gurav, A. Chevelwalla, S. Desai, and Prof, "Sumitra sadhukhan, "criminal face recognition system," *International Journal of Engineering Research*, vol. 4, no. 3, pp. 47–50, 2015.

[13] S. Sahni and S. Saxena, "Survey: techniques for aging problems in face recognition," *MIT International Journal of Computer Science and Information Technology*, vol. 4, no. 2, pp. 82–88, 2014.

[14] K. Kim, "Face recognition using principle component analysis," 2009, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.383.6655&rep=rep1&type=pdf.

[15] P. Kakkar and V. Sharma, "Criminal identification system using face detection and recognition," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 7, no. 3, pp. 238–243, 2018.

[16] OpenCV, *OpenCV: Cascade Classifier*, OpenCV, Santa Clara, CA, USA, 2020.

[17] Kaggle, "Illinois DOC labeled faces dataset | Kaggle," 2021, https://www.kaggle.com/davidjfisher/illinois-doc-labeled-faces-dataset.

[18] T. Jebara, *3D Pose Estimation and Normalization for Face Recognition*, Centre for Intelligent Machines McGill University, Ontreal, Canada, 1996.

[19] R. Brunelli and T. Poggio, "Face recognition through geometrical features," in *Proceedings of the Computer Vision—ECCV'92*, Ligure, Italy, May 1992.

[20] S. Dinkar Borse, K. Vijay Purkar, M. Ganesh Patil, and D. S. Shingate, "Real time face detection to identify criminals and missing people," *IJSRD- International Journal for Scientific Research & Development*, 2019.

[21] C. Maas and J. Schmalzl, "Using pattern recognition to automatically localize reflection hyperbolas in data from ground penetrating radar," *Computers & Geosciences*, vol. 58, pp. 116–125, 2013.

[22] K. Kundu, A. Verma, B. Tyagi, P. Vashisth, and A. Professor, "Visualization of midline diastema fixing," *International Journal of Engineering Research & Technology*, vol. 1, 2020.

[23] L. Dinalankara, "Face detection & face recognition using open computer vision classifies," *ResearchGate*, vol. 1, 2017.

[24] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *Proceedings of the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, pp. 130–136, San Juan, PR, USA, June 1997.

[25] C. Rahmad, R. A. Asmara, D. R. H. Putra, I. Dharma, H. Darmono, and I. Muhiqqin, "Comparison of viola-jones haar cascade classifier and histogram of oriented gradients (HOG) for face detection," *IOP Conference Series: Materials Science and Engineering*, vol. 732, no. 1, pp. 12038–8, 2020.

[26] A. Simran, "Study of different approaches to develop an automatic traffic rule violation detection system," *International Journal of Engineering Research and Technology*, vol. 2, pp. 8–11, 2020.

[27] F. Deeba, A. Ahmed, H. Memon, P. Fayaz Ali Dharejo, and A. Ghaffar, "LBPH-Based enhanced real-time face recognition," 2019, https://www.ijacsa.thesai.org/.

[28] A. P. Singh, S. K. S. Manvi, P. Nimbal, and G. K. Shyam, "Face recognition system based on LBPH algorithm," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 5s, pp. 26–30, 2019.

[29] A. Ahmed, J. Guo, F. Ali, F. Deeba, and A. Ahmed, "LBPH based improved face recognition at low resolution," in *Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 144–147, Chengdu, China, May 2018.

[30] A. Elmadhoun and M. J. Nordin, "Facial expression recognition using uniform local binary pattern with improved firefly feature selection," *ARO-The Scientific Journal of Koya University*, vol. 6, no. 1, pp. 23–32, 2018.

[31] S. K. Gupta and D. P. Shukla, "Evaluation of topographic correction methods for LULC preparation based on multisource DEMs and Landsat-8 imagery," *Spatial Information Research*, vol. 28, no. 1, pp. 113–127, 2019.

[32] Y. M. Riyazuddin, S. M. Basha, K. K. Reddy, and S. N. Banu, "Effective usage of support vector machine in face detection," *International Journal of Engineering and Advanced Technology*, vol. 9, pp. 1336–1340, 2020.

[33] J. A. Cadena Moreano, N. B. La Serna Palomino, and A. C. Llano Casa, "Facial recognition techniques using SVM: a comparative analysis," *Enfoque UTE*, vol. 10, no. 3, pp. 98–111, 2019.

[34] N. Prakash and Y. Singh, "Fuzzy support vector machines for face recognition: a review," *International Journal of Computer Application*, vol. 131, no. 3, pp. 24–26, 2015.

[35] D. M. M. Da Costa, S. M. Peres, C. A. M. Lima, and P. Mustaro, "Face recognition using Support Vector Machine and multiscale directional image representation methods: a comparative study," in *Proceedings of the International Joint Conference on Neural Networks*, Killarney, Ireland, July 2015.

[36] X. Zou, J. Kittler, and K. Messer, "Ambient illumination variation removal by active Near-IR imaging," *Advances in Biometrics*, vol. 3832, pp. 19–25, 2005.

[37] A. L. Ramadhani, P. Musa, and E. P. Wibowo, "Human face recognition application using PCA and eigenface approach," in *Proceedings of the 2nd International Conference on Informatics and Computing*, Jayapura, Indonesia, November 2017.

[38] Y. Xia, B. Zhang, and F. Coenen, "Face occlusion detection using deep convolutional neural networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 30, no. 9, Article ID 1660010, 2016.

[39] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 980–993, 2015.

[40] A. Geitgey, "Face-recognition PyPI," 2020, https://pypi.org/project/face-recognition/.

[41] P. Ruiz, *Understanding and Visualizing ResNets | by Pablo Ruiz | towards Data Science*, Towards Data Science, Ontario, Canada, 2018.

[42] A. Kumar and R. Gupta, "Futuristic study of a criminal facial recognition system using open-source images," *Science Talks*, vol. 6, 2023.