Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032715
Article

Tech Science Press

Deep Learning Based Face Detection and Identification of Criminal Suspects

S. Sandhya[1], A. Balasundaram[2,*] and Ayesha Shaik[1]

[1]School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai, 600127, India
[2]School of Computer Science and Engineering, Center for Cyber Physical Systems, Vellore Institute of Technology (VIT), Chennai, 600127, India
*Corresponding Author: A. Balasundaram. Email: balasundaram.a@vit.ac.in

Received: 27 May 2022; Accepted: 29 June 2022

Abstract: Occurrence of crimes has been on the constant rise despite the emerging discoveries and advancements in the technological field in the past decade. One of the most tedious tasks is to track a suspect once a crime is committed. As most of the crimes are committed by individuals who have a history of felonies, it is essential for a monitoring system that does not just detect the person's face who has committed the crime, but also their identity. Hence, a smart criminal detection and identification system that makes use of the OpenCV Deep Neural Network (DNN) model which employs a Single Shot Multibox Detector for detection of face and an auto-encoder model in which the encoder part is used for matching the captured facial images with the criminals has been proposed. After detection and extraction of the face in the image by face cropping, the captured face is then compared with the images in the Criminal Database. The comparison is performed by calculating the similarity value between each pair of images that are obtained by using the Cosine Similarity metric. After plotting the values in a graph to find the threshold value, we conclude that the confidence rate of the encoder model is 0.75 and above.

Keywords: Deep learning; opencv; deep neural network; single shot multi-box detector; auto-encoder; cosine similarity

1 Introduction

Crime has always been one of the most challenging and intractable issues that not just affects the individuals in our society, but the whole community. This indirectly results in many consequences that affect the growth of the economy of the specific communities [1] where the crime rates are high. To make a society viable for living with less crime rate, there is a great demand for measures and strategies that would highly assist in lowering the crime rates. According to research [2], community-based measures executed by combining the police forces with support services such as probation officers, health professionals, etc, for the prevention of crime can have a considerably higher benefit in controlling criminal behavior than by just enforcing the law. In order to enhance the cooperation between the police and the citizens, problem-oriented solving such as hot spot policing and community-oriented policing have been introduced and implemented in several nations.

However, due to the ongoing pandemic that has been going on for more than two years, lockdowns have been imposed at times to curb the spread of the disease. Therefore, it has resulted in a shortage of officers, who are either working online or have resigned. In this case, monitoring of areas by the police personnel becomes an issue. So, it is essential to look for other options that would help resolve our dilemma. On the other hand, the rise of digitalization has transformed the lives of the people and contributed a lot to the economic growth of societies around the world. Hence, the combined effort of the police forces paired with impeccable technology and software [3] would aid in resolving crime nowadays.

Time is the important factor that would help in determining the reason a crime has occurred and also in finding the culprit or the suspects who are behind it. With the easy availability of live video streams through cameras, live monitoring is done for security purposes. But live monitoring of these video streams is not possible all the time, hence there is a need for an automated system that fulfills the process. Cities have become increasingly modern and smart as governments aim for robust economic growth by employing Artificial Intelligence in everything [4]. This has resulted in a more connected infrastructure where real-time data is made available in no time. Real-time data combined with artificial intelligence can aid societies in detecting crime immediately once it occurs [5].

The proposed system makes use of the deep learning neural network approach by training an auto-encoder, which is a feed-forward neural network whose input and output values are the same. One of the most famous features of auto-encoders is their ability to reconstruct the input image. Hence, instead of comparing the whole face of the criminal for identity verification, we make use of the output of the encoder that is usually used for reconstructing the input image by the decoder. This simple, yet efficient method is a lightweight model that is used for finding the similarity value between the input image and the images in the criminals' database. For training the auto-encoder model, we make use of the "Labelled Faces in the Wild" dataset. For face detection, we make use of the OpenCV DNN model, which is a Caffe model. For comparing the images to find the similarity value, we make use of Cosine Similarity metrics. All the values are plotted in the graph to find the confidence value and it is found to be more than 0.75.

2 Literature Survey

An instantaneous, automated surveillance system has been proposed by Apoorva et al. [6] for the automatic identification system of criminals where the system executes in real-time. The images are initially obtained by converting the video stream data into frames. All the images in all three databases that are considered are preprocessed to reduce the noise and redundant data that might affect the results we obtain. For the process of face identification in images, Haar classifiers and OpenCV classifiers that has many XML files which are predefined for feature extraction are used for extracting facial features that contribute to finding the coordinates of the position of the face in an image. With the feature extraction part complete, template matching is performed to identify the identity of the captured faces in the video stream. Furthermore, this system which is capable of identifying more than one face in the frame, puts forward another feature of identifying even international criminals too.

With criminal activities increasing at an alarming rate, the availability of police personnel has decreased significantly as a result of the pandemic. Despite the existence of various technological wonders to handle this issue, the appropriate system is still needed to resolve the problem. Balasundaram et al. [7] have presented a system that makes use of a multi-task cascade neural network to identify human faces and a Siamese neural network approach to retrieve the identity of a person. Initially, this real-time methodology converts the live video into image frames on which face detection is performed. The faces are then compared with the stored images of people using the constructed Siamese Neural Network that works on One-Shot Learning to identify the matched images. The loss that has been employed in this system is the "Contrastive Loss Function". Furthermore, when a matched pair is found, a notification is sent to the appropriate authorities so that necessary actions can be taken.

With criminals being extremely cautious about not leaving behind any biological evidence such as fingerprints at the crime scene, identifying the culprit has become a time-consuming task. Ratnaparkhi et al. [8] presented a methodology that makes use of a deep neural network approach for face detection and identification of criminals. Faces detection for creating the criminals' database is done by using FaceNet, whereas Multi Cascade Convolutional Neural Network (MC-CNN) was developed to detect faces on the input image and store them for comparison. To perform the comparison of these two images, face embeddings are obtained by using FaceNet. Lastly, these two image embeddings are classified using Linear Support Vector Machine and based upon the prediction values, the criminal faces are identified. The model that outputs the embedding vector has achieved an accuracy value of 92.50% for training data, and a value of 90.90% for testing data.

With the rise in crime rate observed all around the world, the need for a smart monitoring system has increased despite the availability of many complex but inefficient systems. Chhoriya [9] has presented an automation system for criminal face identification and recognition. For the detection of faces, the Haar Cascade classifier is implemented. The real-time video stream is converted into frames on which the Haar Cascade Classifier is used. The faces are extracted and then passed on for identity retrieval. The process of retrieving identity is performed by comparing the captured face with the images stored in three different databases. These databases consist of faces of people who are regional criminals, international criminals and lastly the citizens of the country. The initial step performed on the image is the Haar feature extraction followed by the creation of the integral image. Lastly, by using the Cascade classifier, images that are similar are obtained.

Authentication using face recognition has become a critical component of biometric verification amid the pandemic and has been widely used in various applications, such as video monitoring systems, human-computer interfaces, and network security. Srivastava et al. [10] have proposed a smart attendance system that recognizes faces by integrating face recognition technology with open-source computer vision (OpenCV). Using the Haar Cascade classifier from OpenCV, which has the ability to detect faces that are tilted to an extent of 54°, the faces are captured and stored. A database of the student images is already saved in the system. Finally, the eigenvectors are calculated for the captured face and the faces in the database, which are later compared to find the matched face. Finally, the log table is updated accordingly, which is used for updating the student's attendance.

Aanchaladevi et al. [11] have presented a paper that aids in real-time face recognition and automated face surveillance system. The proposed system makes use of a Haar-based classifier for detection which also performs extraction of features like ears, nose, etc. The system's main objective is to compare the captured facial image with the previously trained face images by training the model with both positive and negative images so as it gives the right output with the correct image. The vector comparison of both the images is performed and the results are displayed accordingly. Hence, this forms an automated surveillance system having the goal of recognizing people on a watch list.

Rasanayagam et al. [12] have presented a methodology that helps in identifying a criminal by analyzing characteristics like expressions, gender, age, etc. It works in such a way that unique features of the face like the eyes, nose, and mouth are identified by using the Caffe model that is used for the identification of an individual, and Dlib for basic feature extraction. The facial feature identification is performed by making use of a model that is trained on ML classifiers. Pre-defined facial landmarks by the OpenCV library can be used for obtaining facial features like eyes, nose, etc. Lastly, for age prediction, the Hessian Line Tracking method is implemented followed by the final gender identification process using a CNN model that is trained on different gender images. This system, which can be implemented in public places like airports, malls, etc, has achieved an accuracy of more than 80%.

The methodology proposed by Suwannakhun et al. [13] is one of the criminal identification methods which minimizes the error in retrieving a person's details. This face detection and retrieval system checks for the identity of the person's face that has been captured and compares it with the image on the identification card. Lastly, it is sent to match with the images which have a resolution of 60 × 60 pixels in the existing database. Face detection is performed using a geometry-based method that identifies the different features of a face like a mouth, eyes and nose. In terms of accuracy rate, the system used around 150 images in the experimental phase and the results show 146 correct matchings resulting in the system's accuracy rate of 97.33%.

Harikrishnan et al. [14] have proposed a smart monitoring system that replaces the traditional attendance system. Among all the numerous face detection algorithms, this method makes use of the common Haar Cascade classifier for detecting or identifying faces during live monitoring followed by preprocessing them and storing them in the database. This acts as the base database with which the captured images are compared. On the other hand, to start with the smart monitoring system, images are captured in real-time, and face detection is performed. These images are preprocessed using Bicubic interpolation before comparing them with the faces present in the base database. Image recognition is performed using the "Local Binary Pattern Histogram," with which numerical predictions are made. These prediction values of all the images are compared, and if any matches are found, then the attendance is updated accordingly. This robust, user-friendly system has achieved a maximum accuracy of 74% while it is being executed in real-time.

A wide range of systems has been developed for automating the attendance system using computer vision techniques for the detection of faces and validation of identities. Balasundaram et al. [15] have developed a system that performs the same process efficiently with accurate results. Rather than the usual attendance systems like the Radio Frequency Identification (RFID) system or the use of biometric authentications might feel like an invasion of privacy, the employment of smart automated systems makes the whole procedure easy going. A Siamese neural network approach is followed for constructing the model that is going to perform image comparison. The initial face detection is performed using Haar Classifiers that make use of certain features for identifying the different features of the face before detecting the whole face. The detected face is captured and passed through the pre-trained Siamese network to obtain the embedding vector. The database of students consisting of the images that are to be compared with the captured images are also passed through the model and the embedding vectors of all the images are compared. Based upon the resulting output of the comparison using various loss functions, the attendance is marked accordingly.

The main objective of the proposed system is to create an automated criminal face detection and identification system. This lightweight system would aid in catching the culprit once a crime occurs at a location. With the availability of camera everywhere, despite just obtaining the facial information from the cameras, we could use it for comparing it with the images in the criminals' database to find their identity. This automated system reduces the physical effort that has to be put forth by the police officials in identifying the identity of the criminals.

3 Proposed Methodology

The architecture of the proposed system is depicted in Fig. 1. The criminal identification system targets two main functionalities that is criminal face detection and face identification. Initially, face detection is performed on the input image. This input image can be any image with people present in it, or it can be the image frames in the video stream. Detection of faces in the image is performed by implementing Single Shot Multi-box Detector with Resnet-10 as the backbone using OpenCV Deep Neural Network [16], which is a Caffe model that performs object detection. Once the face is detected, the image is cropped in such a way that only the face is present. This image can now be used for comparing with the images in the criminal database.

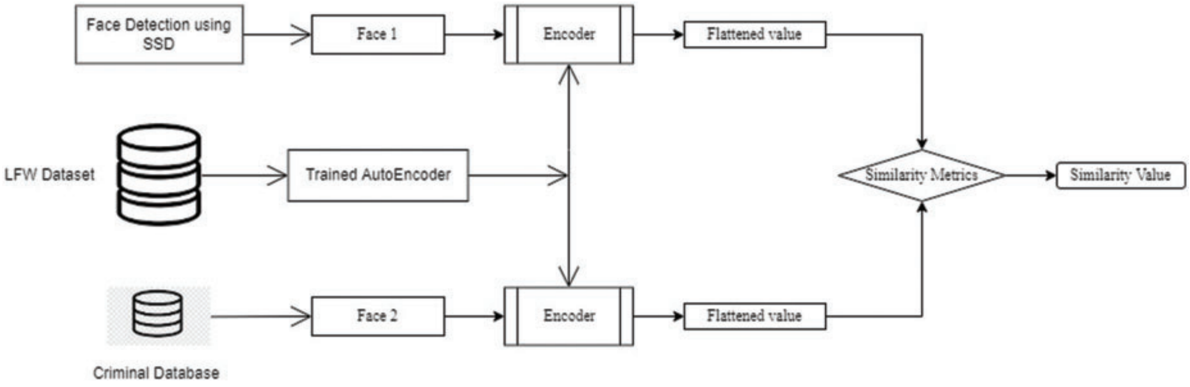

Figure 1: Proposed system architecture

For comparing both the images, instead of comparing the images directly, we construct an auto-encoder where only the encoder part is selected and used in the proposed system. This encoder is capable of converting the input image to an embedding vector format. This embedding vector, when sent as an input to the decoder, can reconstruct the original image. Hence, the input image and also the images in the criminal database are passed to the encoder model for comparison. For comparing both images, we make use of Cosine Similarity [17] to obtain the result. Here similar images obtain a higher similarity value, where the results of dissimilar images are practically lower than the previous value.

The proposed system can be broadly classified into three modules:

- Detection and extraction of face.
- Construction of encoder.
- Calculating Similarity values for criminal identification

3.1 Detection and Extraction of Face

There are numerous algorithms [18] that are developed for face detection such as the Viola-Jones algorithm [19], Haar Cascade Classifier [20], customized CNNs [21] and OpenCV libraries [22], etc. "Viola-Jones algorithm" is one of the most widely used face recognition algorithms that still outperforms most of the other algorithms. Haar Cascade classifiers is another algorithm proposed by the same creators of the Viola-Jones algorithm and is capable of detecting faces in an image or live video stream. Another one of the widely used CNNs for the detection of faces is the "Multi-Task Cascaded Convolutional Neural Network" which is capable of performing detection of faces and also the alignment of faces. Here, in the proposed system, we make use of OpenCV DNN which makes use of SSD, called "Single Shot Multibox detector" with Resnet-10 as its backbone, which is the fastest object detection model and gives great results in terms of accuracy.

Fig. 2 depicts the flow in which this module works. The input image could be an image, or it can be obtained from the live video stream by converting the video stream into frames and selecting the image that has to be sent as the input image. Secondly, to implement OpenCV DNN for face detection, extraction of the necessary dependencies and the model weights are downloaded from the official GitHub website [23]. Once the Single Shot Detector (SSD) model is implemented, we obtain the coordinates or the position of the face in the image. As face region is the region of interest, with the help of the coordinates obtained, we crop the image up to the face and it is saved. The same process is also performed on the images in the criminal database. The construction of the auto-encoder is described in the next section.

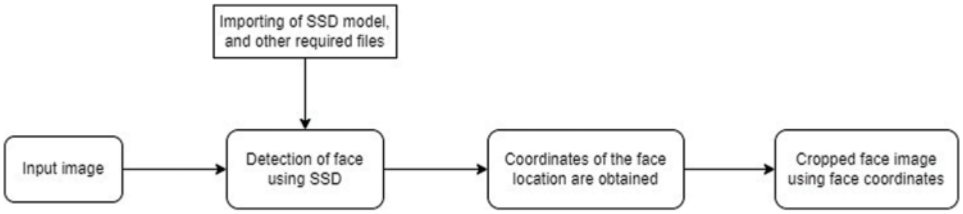
Importing of SSD model, and other required files
Input image
Detection of face using SSD
Coordinates of the face location are obtained
Cropped face image using face coordinates

Figure 2: Workflow of face detection and extraction process

3.2 Construction of Encoder

For the construction of the encoder for the proposed system, the whole auto-encoder has to be constructed. The reason we chose auto-encoder for the proposed system is the fact that auto-encoders follow unsupervised learning and are capable of identifying important features in an image [24]. One of the most well-known applications of auto-encoders is that they can reconstruct the original image that is sent in as input. All the other models that have been chosen for face detection could end up being heavy, whereas we use only the encoding part for the proposed system, and the output of the encoding system which is a flattened vector is only used for comparison.

3.2.1 Dataset Description

There is a wide range of datasets available for different projects from websites like Kaggle, UCI Machine Learning Repository, Google Dataset Search, etc. For projects related to human face identification and verification, the standardized dataset that is used in most cases is the Labelled Faces in the Wild (LFW) Dataset [25]. This consists of more than 13,000 facial images of famous personalities around the world. The original dataset can be downloaded from the LFW Face Database website maintained by the University of Massachusetts, Amherst.

The dataset downloads are available in various formats. The main download consists of all the original 13,233 images zipped in a tar file which is of size 173 MB. These images are also available in two other formats where all the images are aligned with funneling and deep funneling processes. A total of 5,749 people's images are collected under the LFW dataset where around 1680 individuals have more than two images. Hence, there would be 5,749 folders labeled with each person's name and all their images saved in those folders separately. Other kinds of formats are also available where all the images of the person whose name starts with the same letter can be downloaded separately. Likewise, images of the persons whose names start with keywords like 'Ang', 'Bin', 'Che' can also be downloaded together. Different versions of LFW Datasets are also available on other data platforms like Kaggle etc. LFW Dataset is also available as an in-built dataset by TensorFlow and can be accessed by importing the necessary libraries.

For the proposed system, due to a large number of images and the limitations of hardware, we use a subset of the whole dataset. We choose images of the persons whose names start with A and C which is a total of 1956 images.

3.2.2 Model Construction

The auto-encoder model constructed for the proposed system is given in Fig. 3.

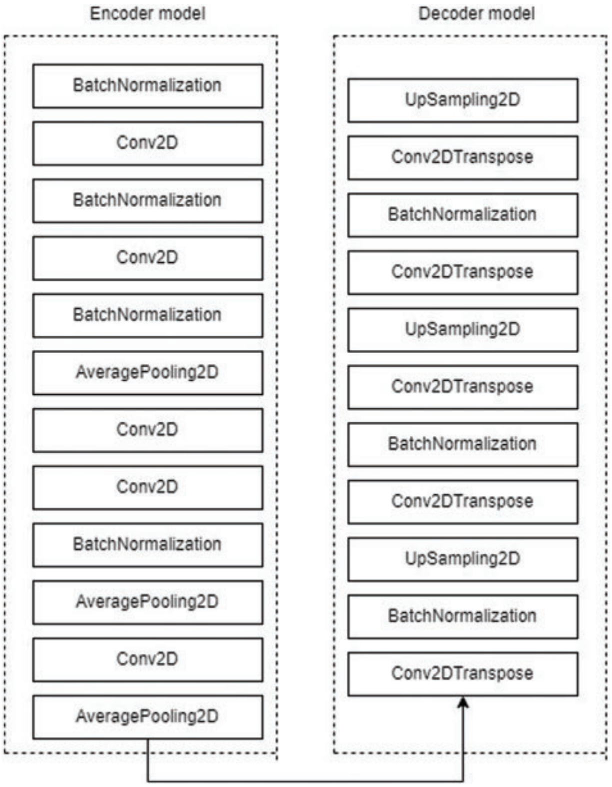

Figure 3: Layers present in the autoencoder model

In the encoder model, we make use of layers like Conv2D, which performs the Convolution 2D operation and the AveragePooling2D layer where we take the average value of the pixel in each kernel. Additionally, the batch normalization [26] layer is inserted at a few places, so that the model is able to give stable and accurate results. Moreover, with the help of this layer, the inputs from the previous layer are normalized and then sent to the next layers, resulting in a more accurate result. In the decoder model, we use layers like Conv2DTranspose, UpSampling and AveragePooling2D layers. Generally, in the auto-encoder model, the output of the encoder model will be sent as input to the decoder model. Hence, those values are the same. Additionally, the decoder's output would be of the same shape as the encoder's input shape. Hence, the auto-encoder is constructed by carefully selecting the filter, kernel and other parameter values. For the proposed system, we only use the encoder part for image comparison. The image comparison by calculating the similarity value is elucidated in the next section.

3.3 Calculating Similarity Values for Criminal Identification

Once the face is detected, the identity of the person has to be verified by checking whether he is a criminal or not. So hence, as per the objective, it is enough to compare the captured input image with the images of the criminals. When the similarity between the images is less, we can conclude that the face does not belong to any criminal. For calculating the similarity value [27] for the pair of images, we make use of Cosine Similarity metrics. There are several similarity measures namely Euclidian distance, Manhattan distance, bilinear distance, cosine similarity etc. The Euclidian distance between any two points b and c is given in (1)

$$d(b,c) = \sqrt{\sum\nolimits_{i=0}^{n} (bi - ci)\hat{}\,2} \tag{1}$$

The Manhattan distance between two points is given in (2) below.

$$d(b,c) = \left|\sum\nolimits_{i=1}^{n} (bi - ci)\right| \tag{2}$$

Despite the use of any similarity metrics, if the model performs well, the resultant similarity value for every image pair comparison would be the same. Hence, we choose the most well-performing similarity metrics which is the Cosine Similarity metric given in Eq. (3).

$$similarity\,(x,y) = \cos(\theta) = \frac{x \cdot y}{|x|\;|y|} \tag{3}$$

Cosine similarity is one of the widely used similarity metrics which calculates the cosine angle between any two given vectors. The cosine similarity metric is generally preferred over any other metric, as it does not calculate the distance, but the orientation or the angle between the vectors [28]. If two images are similar, then both the vectors would not be closer in terms of distance but would be closer in terms of the angle between the vectors of the image. Fig. 4 depicts the process of calculation of the image similarity value.

4 Results and Discussion

4.1 Model Performance

The performance of the auto-encoder model is denoted using the training and validation loss graph. Fig. 5 depicts the graph of the model's performance.

From the graph, it can be seen that initially during the early epochs, both the training and the validation losses were high, and gradually started reducing. When both the training and the validation loss lines intersect each other, it denotes that the model has reached a point that would state that the model is performing well. The model was run for 70 epochs, whereas both the training and validation loss intersected each other at epoch number starting from 8. Despite a fluctuation observed around epoch number 60, the model continued to perform well even after that too till 70. Though the graph validates the performance of the auto-encoder, we can still conclude that the encoder model performs well. We can further validate the encoder by finding image similarity values under different test cases.

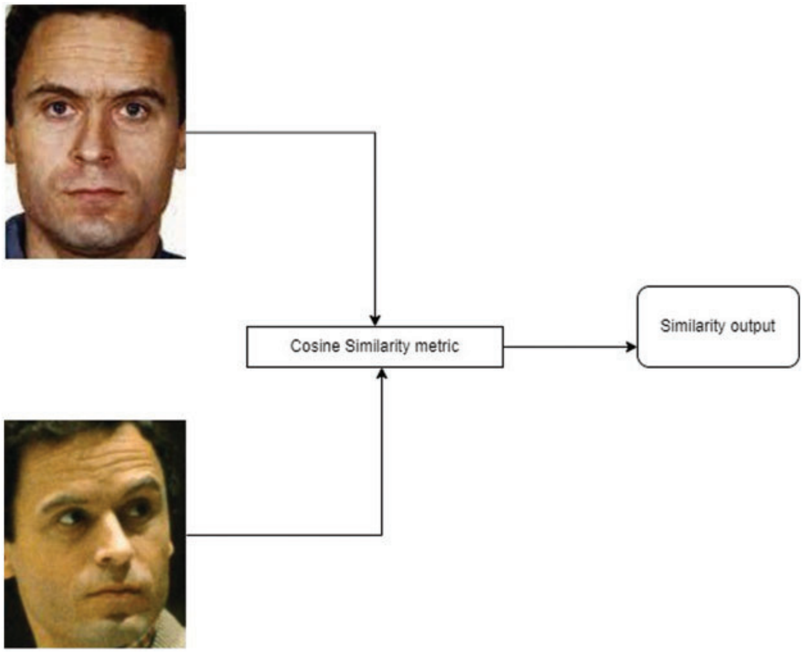

Figure 4: Cosine similarity calculation between two images

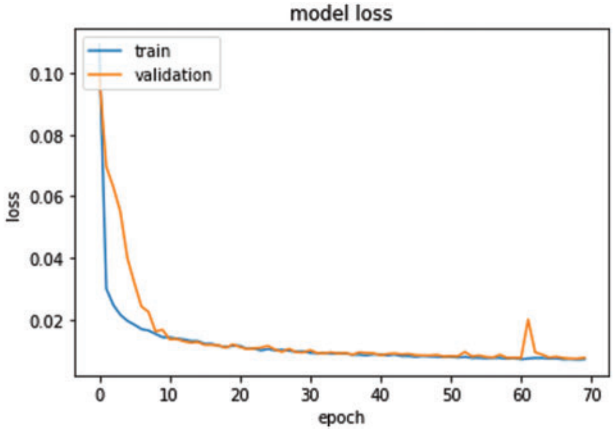

Figure 5: Model performance

4.2 Performing Image Similarity

The encoder model can be validated by running the model through various test cases. Furthermore, we can find the threshold value above which the pair of images can be concluded as similar and below which the pair of images would be considered as dissimilar [28–31]. The comparison process involves the initial process of face detection which is implemented using the OpenCV DNN Caffe model. Once the face is detected and cropped, an image comparison is performed. For obtaining the vector that is used for finding the similarity values between different images, an encoding model is constructed that converts the image into the necessary format, which is the output of the encoder but in the form of a flattened vector.

4.3 Test Cases

From the above-obtained values shown in Tab. 1, we need to find the threshold value so that we can find the confidence value for the proposed system. Hence, all the values are plotted in a chart. Fig. 6 shows the plotted similarity values of the obtained in the above test cases.

Table 1: Test cases under which the model is tested

Test cases	Cosine similarity value
Case 1: Comparison of images belonging to the same criminal ("Ted bundy")	0.783600
Case 2: Comparison of faces belonging to the same criminal ("Richard ramirez")	0.752640
Case 3: Comparison of faces belonging to the same criminal ("Osama bin laden")	0.854505
Case 4: Comparison of faces belonging to the different criminals ("Osama bin laden and ted bundy")	0.636489
Case 5: Comparison of faces belonging to the different criminals ("Richard ramirez and ted bundy")	0.666025
Case 6: Comparison of faces belonging to the different criminal ("Richard ramirez and osama bin laden")	0.631602
Case 7: Comparison of faces belonging to a criminal and non-criminal ("Actor 1 and osama bin laden")	0.653118
Case 8: Comparison of faces belonging to a criminal and non-criminal ("Actor 1 and richard ramirez")	0.594996
Case 9: Comparison of faces belonging to a criminal and non-criminal ("Actor 1 and ted bundy")	0.736666
Case 10: Comparison of faces belonging to a criminal and non-criminal ("Actor 2 and osama bin laden")	0.670823
Case 11: Comparison of faces belonging to a criminal and non-criminal ("Actor 2 and richard ramirez")	0.642838
Case 12: Comparison of faces belonging to a criminal and non-criminal ("Actor 2 and ted bundy")	0.741989

From the above figure, we can see that the values obtained in cases 1, 2 and 3 are the similarity values obtained for the similar images comparison, that is the values obtained for the pair of images belonging to the same criminal. Whereas all the other values lie below the threshold line, which represents the similarity values obtained for the pair of dissimilar images. From the results obtained in the above test cases, we can conclude that dissimilar images have similarity values of 0.74 and below, whereas similar images have similarity values of 0.75 and above. Hence, we can conclude that the confidence rate is 0.75 and above.

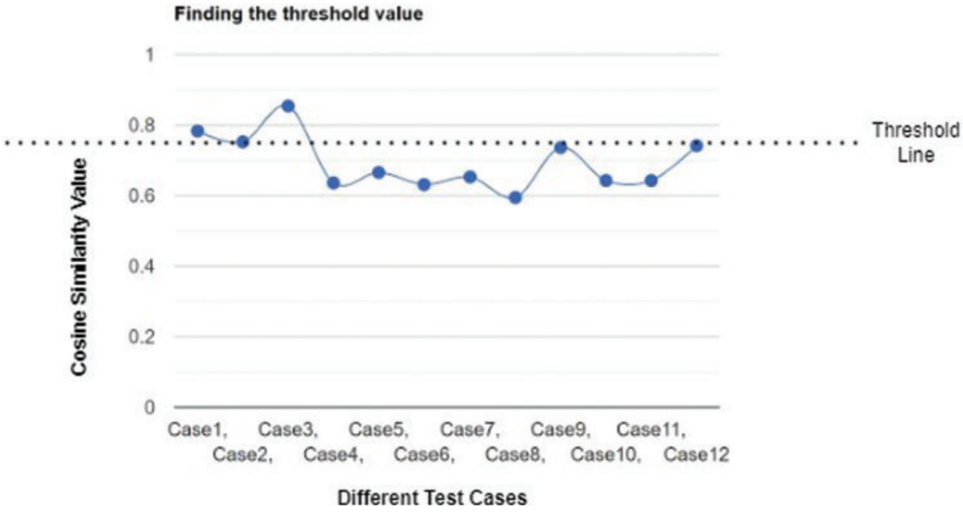

Figure 6: Finding the confidence rate

5 Conclusion

The proposed model not only acts as an efficient criminal identification system but also can be used as an effective, automated monitoring system in places where the environment should be controlled and has to be monitored consistently. This simple, yet efficient methodology can identify and extract the important facial features from an image and further help in identifying criminals by finding the image similarity between the images. Additionally, we found an adequate threshold value that validates the results we obtained. Due to the easy availability of real-time data, the system can be applied in live monitoring systems like CCTV cameras, where the faces are captured in real-time and can be sent for finding the similarity values with images in the Criminals database. There are high chances that the criminals might hide their faces by wearing a mask or any other accessory that might result in occlusion, and the captured image might be of no use. Hence, in this case, we can additionally train the model with images of people wearing a mask, and other accessories. Thus this approach is highly scalable as well and will in turn help with easy verification of identities even if the culprit wears a disguise. Future work will be to upscale this model using advanced ensemble approaches and to make it flexible to not just identifying criminals but also include suspicious activity detection as well.

Acknowledgement: The authors wish to express their thanks to VIT management for their extensive support during this work.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. R. Taylor, “The impact of crime on communities,” *The Annals of the American Academy of Political and Social Science*, vol. 539, no. 1, pp. 28–45, 1995.
[2] T. Hope, “Community crime prevention,” *Crime and Justice*, vol. 19, no. 1, pp. 21–89, 1995.

[3] F. Tombul and C. Bekir, “Police use of technology to fight against crime,” *European Scientific Journal*, vol. 11, no. 10, pp. 11–28, 2015.

[4] Z. Allam and Z. A. Dhunny, “On big data, artificial intelligence and smart cities,” *Cities*, vol. 89, no. 1, pp. 80–91, 2019.

[5] V. Ulagamuthalvi, G. Kulanthaivel, A. Balasundaram and A. K. Sivaraman, “Breast mammogram analysis and classification using deep convolution neural network,” *Computer Systems Science and Engineering*, vol. 43, no. 1, pp. 275–289, 2022.

[6] 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 *2019 3rd Int. Conf. on Computing Methodologies and Communication (ICCMC)*, Erode, India, pp. 775–778, 2019.

[7] A. Balasundaram, S. Ashok Kumar and S. MageshKumar, “Optical flow based object movement tracking,” *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 3913–3916, 2019.

[8] S. T. Ratnaparkhi, T. Aamani and S. Shipra, “Face detection and recognition for criminal identification system,” in *2021 11th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, pp. 773–777, 2021.

[9] P. Chhoriya, “Automated criminal identification system using face detection and recognition,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 1, pp. 910–914, 2019.

[10] M. Srivastava, A. Kumar and A. Dixit, “Real time attendance system using face recognition technique,” in *2020 Int. Conf. on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, Mathura, India, pp. 370–373, 2020.

[11] S. T. Aanchaladevi, S. G. Shubhangi, S. G. Ashwini, A. M. Mayuri and A. A. Bankar, “Criminal identification by using real time image processing,” *International Journal of Research in Engineering and Science (IJRES)*, vol. 9, no. 6, pp. 37–42, 2021.

[12] K. Rasanayagam, S. D. D. C. Kumarasiri, W. A. D. D. Tharuka, N. T. Samaranayake, P. Samarasinghe *et al.,* “CIS: An automated criminal identification system,” in *2018 IEEE Int. Conf. on Information and Automation for Sustainability (ICIAFS)*, Colombo, Sri Lanka, pp. 1–6, 2018.

[13] S. Suwannakhun, C. Narumol and K. Mahasak, “Identification and retrieval system by using face detection,” in *2018 18th Int. Symp. on Communications and Information Technologies (ISCIT)*, Bangkok, Thailand, pp. 294–298, 2018.

[14] J. Harikrishnan, S. Arya, S. Aravind and A. S. Remya, “Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision,” in *2019 Int. Conf. on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Vellore, India, pp. 1–5, 2019.

[15] A. Balasundaram, M. Naveenkumar, A. K. Sivaraman, R. Vincent and M. Rajesh, “Mask detection in crowded environment using machine learning,” in *2021 Int. Conf. on Smart Electronics and Communication (ICOSEC)*, Trichy, India, pp. 1202–1206, 2021.

[16] S. Karthik, R. S. Bhadoria, J. G. Lee, A. K. Sivaraman, S. Samanta *et al.,* “Prognostic kalman filter based Bayesian learning model for data accuracy prediction,” *Computers, Materials & Continua (CMC)*, vol. 72, no. 1, pp. 243–259, 2022.

[17] E. Hjelmas and B. K. Low, “Face detection: A survey,” *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001.

[18] Y. Q. Wang, “An analysis of the Viola-Jones face detection algorithm,” *Image Processing*, vol. 4, no. 1, pp. 128–148, 2014.

[19] T. Mantoro and A. A. Media, “Multi-faces recognition process using Haar cascades and Eigenface methods,” in *2018 6th Int. Conf. on Multimedia Computing and Systems (ICMCS)*, Rabat, Moroco, pp. 1–5, 2018.

[20] N. Zhang, J. Luo and G. Wuqi, “Research on face detection technology based on MTCNN,” in *2020 Int. Conf. on Computer Network, Electronic and Automation (ICCNEA)*, Xian, China, pp. 154–158, 2020.

[21] K. Kadir, M. K. Kamaruddin, H. Nasir, S. I. Safie and Z. A. K. Bakti, “A comparative study between LBP and haar-like features for face detection using OpenCV,” in *2014 4th Int. Conf. on Engineering Technology and Technopreneuship (ICE2T)*, Kuala Lumpur, Malaysia, IEEE, pp. 335–339, 2014.

[22] A. Balasundaram, S. Ashokkumar, D. Kothandaraman, E. Sudarshan and A. Harshaverdhan, "Computer vision based fatigue detection using facial parameters," *IOP Conference Series: Materials Science and Engineering*, vol. 981, no. 2, pp. 21–37, 2020.

[23] C. C. Tan and C. Eswaran, "Reconstruction and recognition of face and digit images using autoencoders," *Neural Computing and Applications*, vol. 19, no. 7, pp. 1069–1079, 2010.

[24] A. Balasundaram, D. Kothandaraman, P. J. Sathishkumar and S. Ashokkumar, "An approach to secure capacity optimization in cloud computing using cryptographic hash function and data de-duplication," in *2020 3rd Int. Conf. on Intelligent Sustainable Systems (ICISS)*, Chennai, India, pp. 1256–1262, 2020.

[25] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization," *Advances in Neural Information Processing Systems*, vol. 31, no. 1, pp. 1–10, 2018.

[26] Q. Cao, Y. Yiming and L. Peng, "Similarity metric learning for face recognition," in *2013 Proc. of the IEEE Int. Conf. on Computer Vision*, Sydney, NSW, Australia, pp. 2408–2415, 2013.

[27] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, pp. 748–756, 2018.

[28] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.,* "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3560, 2021.

[29] W. Sun, G. C. Zhang, X. R. Zhang, X. Zhang and N. N. Ge, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30803–30816, 2021.

[30] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[31] H. Sun and R. Grishman, "Employing lexicalized dependency paths for active learning of relation extraction," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1415–1423, 2022.