Measurement: Sensors 27 (2023) 100800

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Face recognition using CNN and siamese network

C. Ranjeeth Kumar [a,*], Saranya N [a], M. Priyadharshini [a], Derrick Gilchrist E [a], Kaleel Rahman M [a]

[a] Department of Information Technology, Sri Ramakrishna Engineering College, NGGO Colony, Coimbatore, Tamil Nadu, 641022, India

ARTICLE INFO

Keywords:
Facial key-points
K-nearest neighbor
Siamese network
Facial image verification
Convolutional Neural Network

ABSTRACT

Facial recognition is no longer a cutting-edge technology; it is now a part of everyday life. It has been used for various security and profiling applications around the world. Early face detection models were developed during the 1960s and were used to just classify photos of people. In past decades, the face recognition models were optimized and reengineered to identify all the people in each frame of real-time, high-resolution video input. It still has a wide variety of applications to be implemented and can be further optimized for high precision using different approaches. In this study, we have implemented two different approaches for facial detection. The first is a CNN-based approach that extracts keypoints from an image and classifies it using a KNN algorithm. The next approach uses a Siamese network to classify the input image. The initial part focuses primarily on data collection and training. The following part clearly explains the implementation of both approaches. The performance of these approaches was also evaluated and illustrated optimally.

1. Introduction

In recent days, facial recognition systems have been widely used as biometrics in laptops, mobile phones, attendance monitoring systems, etc. Due to advancements in recent technologies, many organizations use face recognition systems for security access, surveillance systems, home security automation systems, and also in the criminal identification process. Face recognition software basically works by comparing the facial features and geometric features of the person. The traditional method uses a database to store the images of individuals, which will be compared with captured images. Various features like edges, corners, texture descriptors, etc. were extracted and used as facial features. These features were used along with machine learning algorithms for classification. Some of the challenges with the traditional methods include variability in face poses, aging, light illumination, and facial changes with expression [Ref]. One of the proposed systems uses 68 facial key points or landmarks as features for identifying the face of an individual. The different landmarks considered are given in

Fig. 1. The proposed system consists of two different approaches. The first approach utilizes the convolutional neural network (CNN) for extracting landmark features from the given input image. These landmarks are passed to the K-Nearest Neighbor (KNN) algorithm, which identifies a person by comparing the facial keypoints extracted from the person's input image with the facial keypoint values that are captured and stored in the database during the registration of that particular person. The second approach uses the Siamese Network, which identifies a person's face directly by just cropping out the face from the entire image feed. System performance has been analyzed for these 2 approaches, and a conclusion has been drawn from it. Our research tends to concentrate on selecting the best approaches for a certain task, comparing those filtered out methodologies, and then identifying an approach that best performs as expected. Therefore, in this study, we selected two commonly used approaches for facial image detection, trained them with a new dataset, evaluated their performance, and then chose the optimal methodology.

2. Literature review

The automated facial recognition system is widely used to track attendance in businesses, schools, and colleges, among other places. It is faster and more effective than manual approaches, and it eliminates the need for manual work entirely. In a classroom setting, the Vio-Jones methodology is used for face detection, and Linear Discriminant Analysis (LDA) is used for recognition [1]. A deep neural network model is used for disguised face recognition. There are two CNN models in use. One is for predicting the image's facial keypoints. At this point, there are 20 key points plotted around the person's face. Another CNN is used to predict the person based on the calculated ratios and angles [2]. Face

* Corresponding author.
E-mail addresses: ranjeeth.chandran@srec.ac.in (C.R. Kumar), saranya.pravin@srec.ac.in (S. N), priya.murugesan@srec.ac.in (M. Priyadharshini), derrickgilchrist.1805019@srec.ac.in (D.G. E), Kaleelrahman.1805042@Srec.Ac.In (K.R. M).

https://doi.org/10.1016/j.measen.2023.100800
Received 21 March 2023; Received in revised form 8 May 2023; Accepted 20 May 2023
Available online 22 May 2023

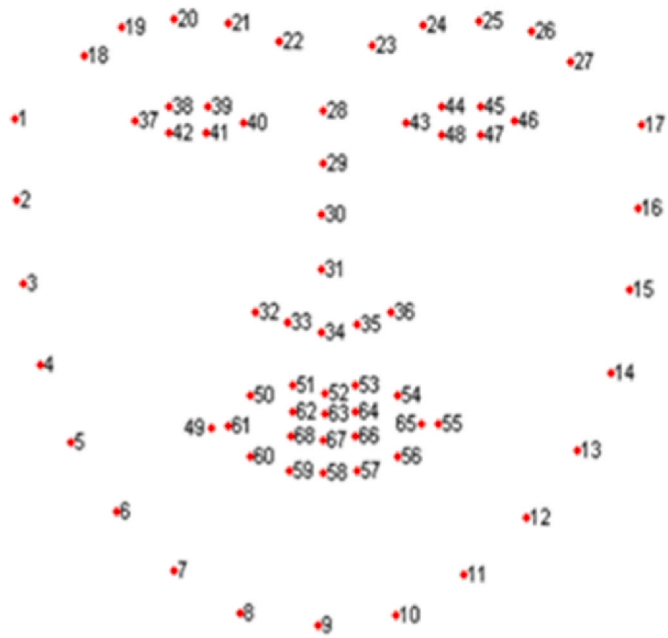

Fig. 1. 68 Facial Key points.

biometrics are used to correctly recognise the person. To accomplish this, the K-nearest neighbor (KNN) method is used to recognise the face. Particle Swarm Optimization (PSO), a metaheuristic optimization algorithm, is used to optimize the approach. When KNN and PSO are combined, they produce an efficient method of predicting faces [3]. For feature extraction from images, the kernel discriminates analysis (KDA) method is used, and machine learning methods such as SVM and KNN are used for classification. The Yale and ORL datasets are used for training and testing in this case. The ORL dataset was slightly more accurate than the Yale dataset [6]. The color information in a facial image is extremely important during the prediction process. The color image is transformed into a multiple color space model, which is then transformed into eigen vectors and eigen values. These are fed into the nearest neighbor classifier, which classifies them. There is a claim that the accuracy of this method outweighs all other Facial Recognition approaches accuracy [7]. e-Voting has been introduced in Indian elections using facial recognition and fingerprint techniques [8].CNN is also important in face recognition, which is common in many implementations with this use case. The CNN model can be simplified further by combining the convolution and sampling layers, resulting in a higher recognition rate [5]. Image matching is an alternative technique that does not require the extraction of keypoints. Using neural network-based feature vectors, the network simply matches the images. The euclidean distance is used to calculate similarity in this case. The feature vector in the siamese network is created using CNN, which is learned with the labeled dataset. The loss function in this case is contrastive loss [4]. A triplet loss approach is proposed for extracting the expressive deep feature by incorporating it into a siamese network. The effectiveness of this approach has been demonstrated through theoretical analysis [9]. The MTCNN algorithm is used to implement a surveillance system that can track a person's face using multiple cameras. Jetson TX2 module is used for this purpose [10]. In the medical industry, the Facial Recognition system has its own set of benefits, such as retrieving a person's medical history, reducing the time spent in the reception area making manual appointments, and avoiding the use of RFID and printed record files. After registering a person's face in a universal medical database, that person can go to any hospital without having to register manually at the reception desk; whenever he enters the hospital, his face is scanned in the medical database and his data is retrieved. If he was already registered [11]. Changing the residual scaling factor in Inception-ResNet to a trainable parameter, as well as changing the activation function from the ReLU to the Leaky ReLU, resulted in a significant improvement in the model's performance [12]. Further more trying different and efficient algorithms will increase the speed and accuracy of the model so that one day no human intervention is required for its applications in major industry such as medical and

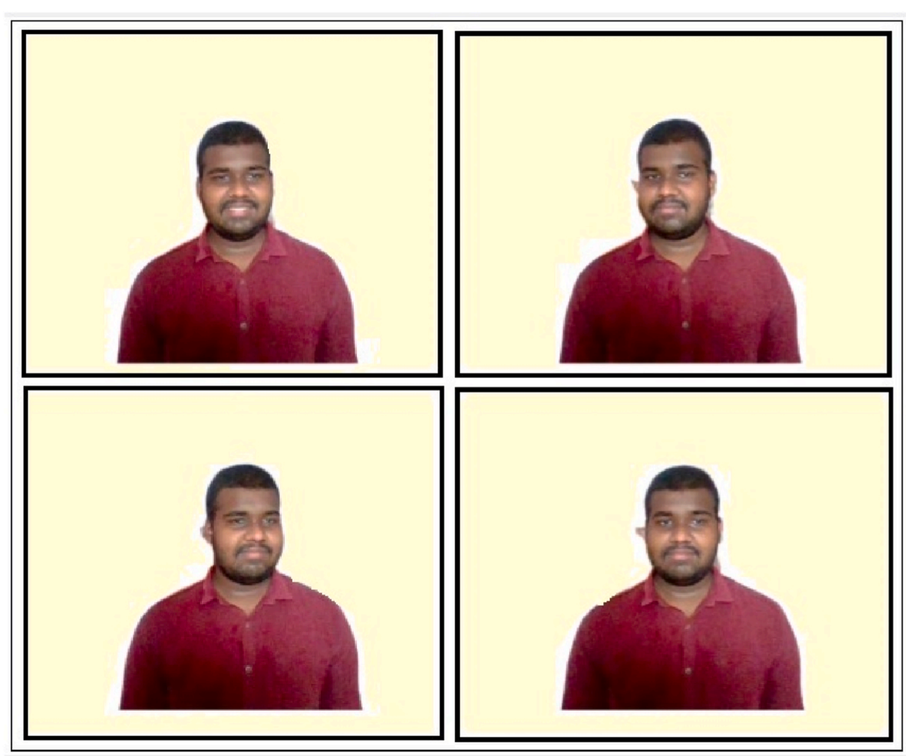

Fig. 2. Sample Facial data.

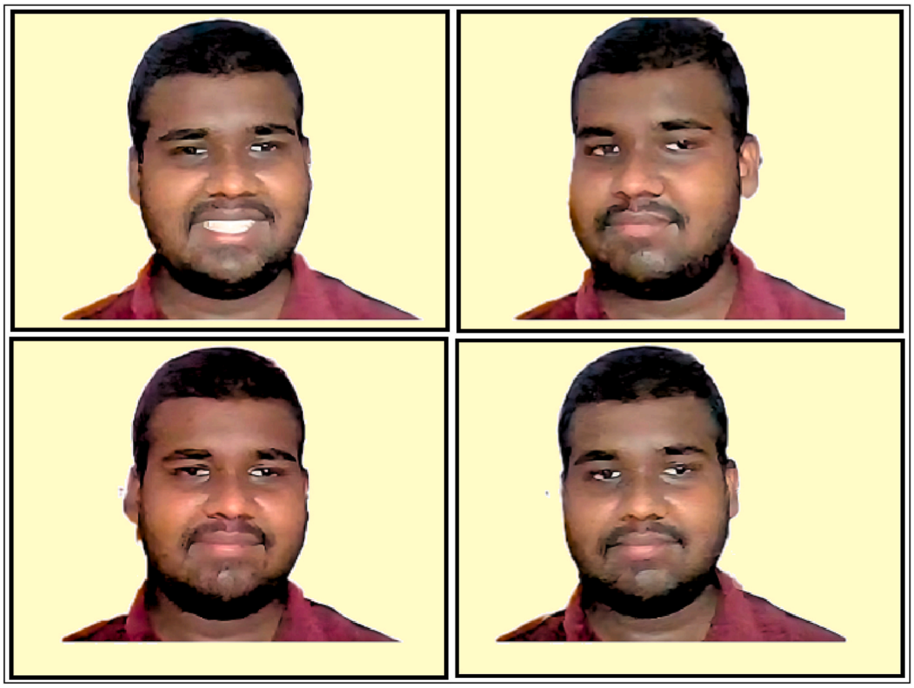

Fig. 3. Sample Cropped Facial data.

defense.

3. Data collection & preprocessing

This project makes use of two distinct datasets. The first is a publicly accessible dataset called 'Youtube Faces Dataset with Facial Keypoints,' which contains celebrity images from YouTube videos. The dataset contains 5770 facial images. The other dataset considered for the project is collected manually and preprocessed. Facial images of people are collected from various angles and postures, as well as in various lighting conditions, to provide the predictor model with a taste of all angles of the person's face for a successful prediction.

We collected facial data images from 25 people ranging in age from 8 to 60. For this project, 200 images of each person are used to train the model to predict that specific person. Instead of taking 200 individual images of a person, which is a time-consuming task, we automated it by recording a 30-s video (11 MB–12 MB) with Python VideoCapture method from OpenCV library and 200 frames (90 KB - 100 KB) are extracted from this video and saved as jpg images. Fig. 2 shows the sample frames extracted from the video captured.

The facial images are now partially prepared for keypoint extraction. The external space excluding the person's face is large in these images, which could lead to misprediction, so the faces are tightly cropped from the images before moving on to training the model. Cascade Classifier is used for this purpose. Cropping is performed by the Cascade classifier using the 'haarcascade frontal face default.xml' file. If an image contains multiple faces, each face will be cropped and saved as a separate image.

Fig. 3 depicts the faces cropped from the image data collected. The cascade classifier does not crop out faces with 100% accuracy. Because when we tried to input 200 images with a single person's face in each image, only 180 to 190 images were successfully cropped and saved. There are factors such as poor lighting and heavy shaking of the person's face, which results in blurring of the face, and so on. The first 2 s of the video which includes the first 10 images out of 200 images are dark because the webcam starts at that point and struggles to capture the target, but after that, the rest of the 28 s video is perfectly captured. So these are the reasons why the cascade missed some image cropping.

Fig. 4 depicts the entire data collection and preprocessing steps undertaken. The extracted frames are then passed to two different models for training and prediction. One approach is K-NN-based, which requires key points to be extracted from the frames for training. Another method is the Siamese Network, which is a convolutional neural network-based technique capable of facial detection and alignment. This approach doesn't require facial keypoints, so the images that have been collected and cropped are literally ready to be fed directly into this model for training.

4. Methodology

The methods used in this research were trained and tested with Anaconda's Jupyter notebook with some basic libraries installed. These algorithms are trained with a normal CPU, so there is no necessity for high-power GPUs.

4.1. KNN

K-Nearest Neighbor method was chosen because training a CNN using the chosen architecture doesn't take very long, and it can precisely plot keypoints on facial images even with minimal training data [3]. The KNN method, which is best suited for this objective of rapidly comparing the facial keypoints, is chosen to support the prediction along with this CNN model. In this method, a Convolutional Neural Network Model that can predict the facial key points is first trained [13]. The K-Nearest Neighbor Model receives these predicted keypoints and compares them to the keypoint values already gathered during registration to identify the individual. The K-Nearest Neighbor algorithm is a popular machine learning technique because of its simple implementation and accurate results [14,15].

4.1.1. CNN based keypoint extraction

When image processing or any kind of prediction using an image comes into play, a convolutional neural network will be the first choice. A standard convolutional neural network will comprise a few basic layers that could be repeated n-times in the network depending upon the subject to be predicted [16,17]. The basic layers include a convolutional layer, which has a filter that will be moved across the input image.

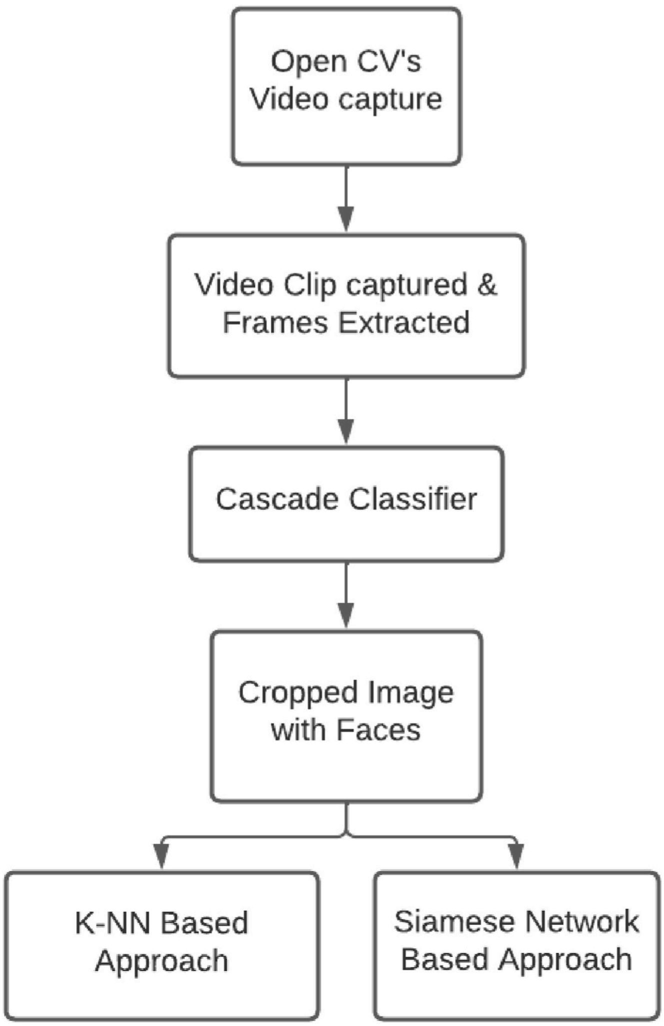

Fig. 4. Data collection & preprocessing flow.

Usually the image will be larger when compared to the filter used to slide upon it. Starting from the top of the image until the last part of it, the filter slides in a horizontal and vertical pattern, calculating the convolutional layer values by using the dot product method. These derived convolutional layer values are then passed to the next layer, which is the pooling layer. This typically reduces the size of the values passed from the previous layer, which is actually the feature extracted from the image fed. This is also attained using a pooling filter, which slides across the previous output. Depending upon the subject that is predicted, the convolutional layer and pooling layers are iterated one after the other to produce the desired output. After the feature is extracted, it passes through a series of condensation and pooling layers, after which it is flattened out. This output is passed to the fully connected layer, where the prediction is done, and then finally at the output layer, the required prediction can be seen. In this case, the 68 key points extracted from the image can be obtained at the output. The common structure of a CNN model can be seen from Fig. 5.

The image goes through additional preparation in the proposed CNN architecture that is not done during the preprocessing stage [18,19]. The RGB-formatted input image is changed to grayscale so that the color space is changed from [0,255] to [0,1]. To ensure the consistency of the original data, which has a size of 224*224 pixels, the converted grayscale image is further shrunk to a standard pixel size [20–22]. The image is fed to the convolution model once these formatting steps are complete. The architecture of the CNN model utilized for keypoint extraction is shown in Fig. 6.

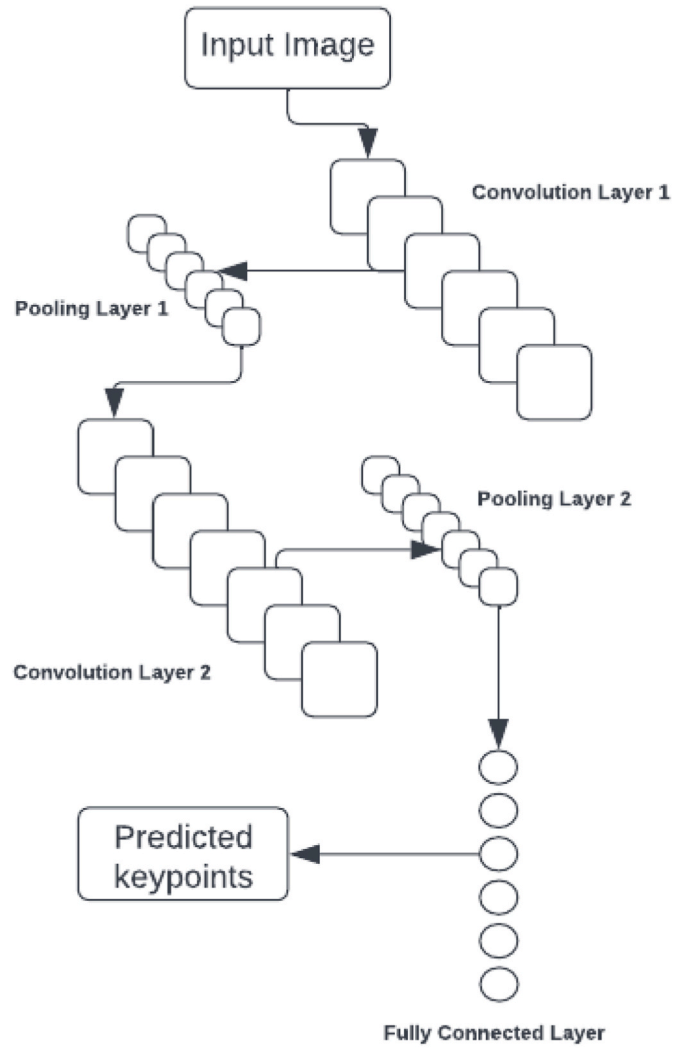

Fig. 5. Common CNN architecture.

4.1.2. Comparison of keypoints using KNN

The KNN algorithm is a well-known supervised algorithm that is widely used for classification. The algorithm predicts the class of the new data by comparing it with previously trained data. It calculates the euclidean distance between the nearest N data points of different classes. These data points are called neighbors. The maximum number of neighbors with the same class will be considered the predicted class of the data point. Fig. 7 illustrates that the new data has more class B neighbors than class A neighbors, and as a result, it is classified as class B.

If there are 30 users registered in our system, there will be 30 separate classes in KNN represents in Fig. 8, which is a Multiclass Multilabel classification. In this case, the class will be the various users that have been registered in our system.

4.2. Siamese network

In this method the siamese network is used for facial recognition. This method does not use facial keypoints like the above method but instead uses euclidean distance calculation between images for recognition. The network uses the Inception Resnet version 1 with pretrained weights for encoding the image and MTCNN (Multi-Task Cascaded Convolutional Network) for face detection. The encoder model is trained by calculating triplet loss for anchor is described in Fig. 9, positive and negative inputs. The model's main aim is to minimize the triplet loss for

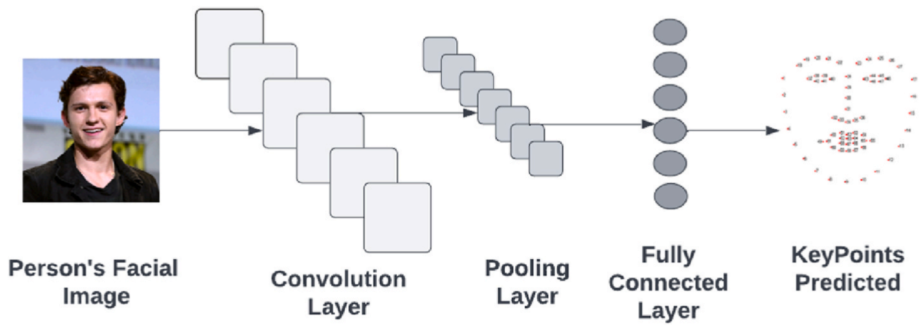

Fig. 6. CNN architecture for Facial Keypoint Prediction.

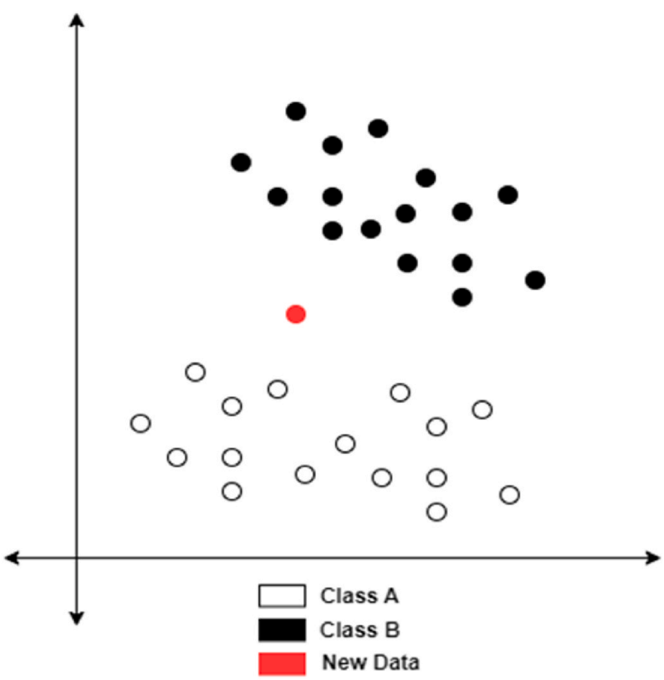

Fig. 7. Before Classification by KNN algorithm.

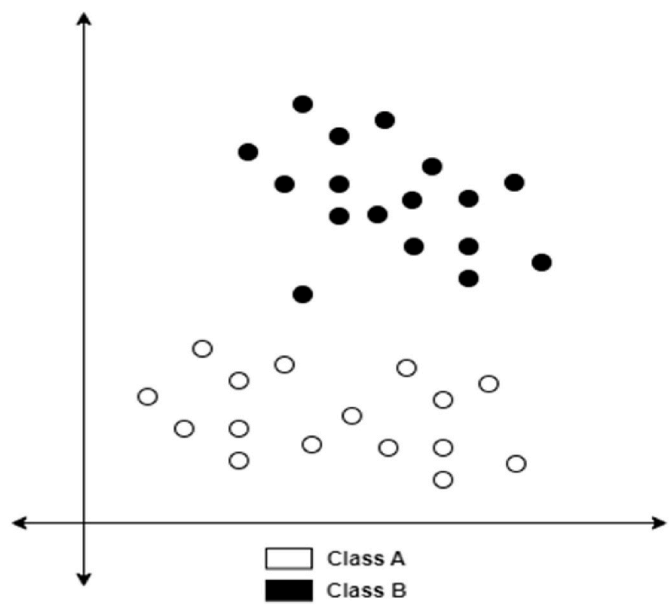

Fig. 8. After Classification by KNN algorithm.

each batch.

4.2.1. *Inception Resnet V1*

Inception Resnet is a combination of inception network and Resnet. There are 2 versions of Inception Resnet models. There are:

- Inception network V1
- Inception network V2

Inception network V2 is more accurate than V1 but it requires more computational resources and processing time when compared to V1. Inception Network V1 is a hybrid network which has Inception V4 and also the performance of Resnet. The main feature of this architecture is that the output from the previous layer is added to the input while maintaining the same dimensions for input and output.

4.2.2. *Triplet loss*

The triplet loss is calculated using following formula:

$$L = \max(\text{distance(anchor, positive)-distance(anchor, negative)}+\alpha, 0)$$

L represents the triplet loss. The euclidean distance between anchor image and positive image is calculated. The euclidean distance between anchor image and negative image is calculated. α represents the threshold by which the two distances must be separated from each other.

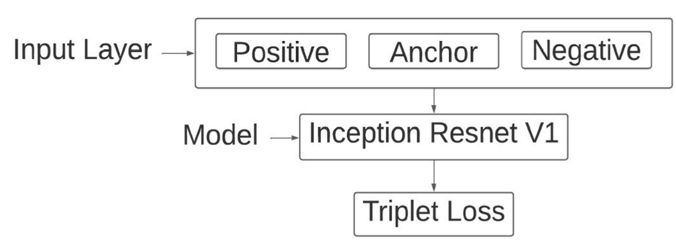

Fig. 9. Triplet loss calculation.

5. Implementation

5.1. *KNN*

Facial detection using KNN has two stages which can be seen in Figs. 7 and 8. The first stage is known as the "registration stage," in which a 30-s video of the user is captured, which is then split into frames, and these frames undergo the steps mentioned in the pre-processing phase before being passed to the Convolutional Neural Network model, where key points are extracted for each of the images and then stored in a database along with the corresponding label, in this case the person's name, which is accessible by the K-Nearest Neighbor

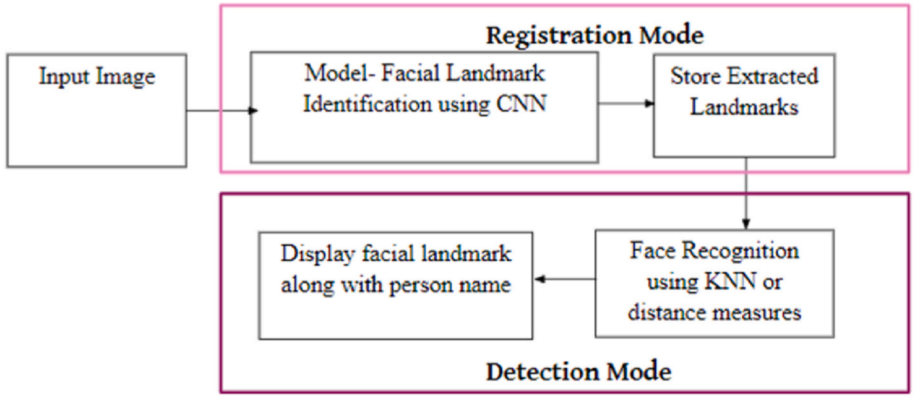

Fig. 10. Phases of KNN approach.

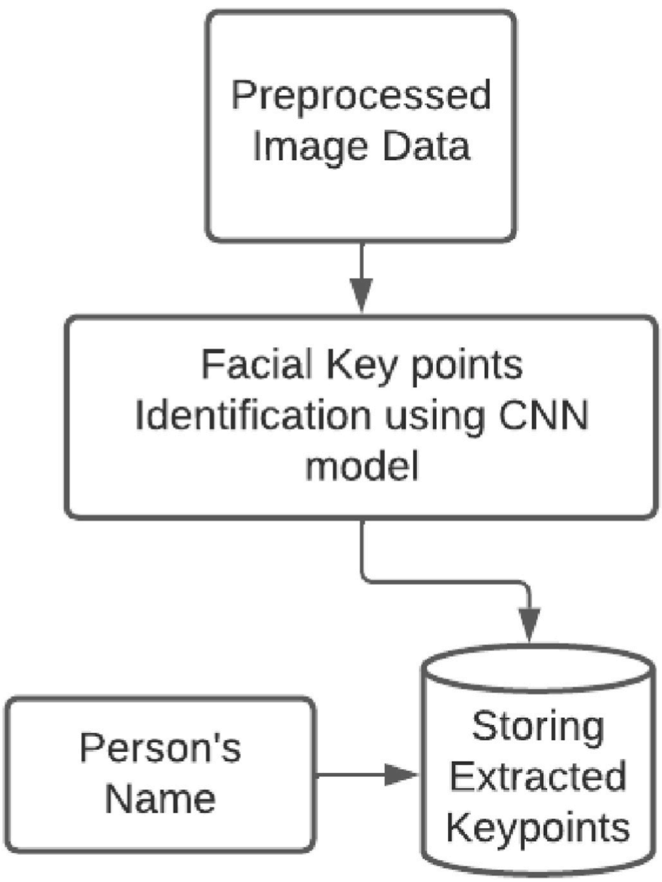

Fig. 11. KNN registration phase.

model. The second stage is known as the "detection stage," where a single image of the user is captured in real time and fed to the preprocessing pipeline, where it gets cropped, and then moved to the CNN model, which extracts the key points from the image, and these key points are compared with the key point values captured during registration using the KNN model, fetching the person's name and displaying it when a close match is found. If the KNN can't find a close match with the data stored in the registration database, then it returns the user as unknown. Fig. 10 represents the phases of KNN approach. The two phases used in this KNN approach are distinct: in phase 1, referred to as registration, a person's facial keypoints from the video collected are extracted using the CNN model and stored in the database along with their name. A person's image is provided to the system during phase 2, which is the detection phase, where keypoints are extracted using the CNN model and then compared with pre-existing keypoints from the registration database using KNN to provide a result.

5.1.1. Registration phase

The user is instructed to maintain proper posture in front of the webcam throughout the registration phase and to move his or her face slightly in all directions so that a 10-s video can be recorded. Another crucial aspect of recording is that there shouldn't be anyone else in the background while the user is registering. The user must make sure that the eyes, nose, and mouth are completely visible while moving the faces into different orientations in order for the algorithm to predict correctly. Additionally, the individual must ensure that there is adequate illumination so that the face can be seen properly. Additionally, the camera quality used for registration must be at least 2 megapixels, which will help to reduce external noise in the video. The preprocessing processes are applied to the video after it has been recorded. The frames extracted from the video usually range from 170 to 190 images, which have clear facial images in them. Fig. 11 represents these images are fed one after the other into the CNN model for extracting the keypoint of that particular image.

In order to do this, a convolutional neural network with six convolutional layers. The first con2d layer's kernel size is a tuple of dimensions (7, 7), whereas the second and third con2d layers' kernel sizes are also tuples (5, 5). The kernel tuples for the other three convolution layers are (3, 3) and (1, 1). In order to apply the filter to the input image, different strides of sizes (3,3) and (1,1) are used. In this case, the stride filter moves 3 steps to the right of the input image's pixel when it is set to (3,3), and the result value is calculated. This process is repeated until the stride moves to the input image's extreme end pixel. Three fully connected layers are configured after the con2d layers, the first two of which had dropouts in them. Fig. 10 depicts the entire architecture of the CNN model used for keypoint extraction. Fig. 12 indicates that the CNN model has been trained with the Youtube dataset and also with the

The system will prompt the user to input their name, which will be recorded with the key points that are being extracted, while it records the person's face. The extracted key points and label, which in this scenario is the name, are saved in a CSV file and will be used later on during the detection phase. The data constructed and entered in the CSV file has 129 columns. In which 128 columns correspond to the keypoints values, since there are 64 keypoints and each keypoint has X and Y coordinate values, it has 128 values, and the last column has the person's name. If there are 50 images extracted from the 30-s registration video, then 50 rows will be entered in the CSV file. Fig. 8 shows the overall steps undergone in the registration process.

5.1.2. Detection phase

This phase is designed to take an input image that was taken using a

```
Net(
  (conv1): Conv2d(1, 32, kernel_size=(7, 7), stride=(3, 3), padding=(1, 1))
  (conv2): Conv2d(32, 64, kernel_size=(5, 5), stride=(3, 3))
  (conv3): Conv2d(64, 128, kernel_size=(5, 5), stride=(3, 3), padding=(1, 1))
  (conv4): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1))
  (conv5): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1))
  (conv6): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
  (fc1): Linear(in_features=8192, out_features=1024, bias=True)
  (fc1_drop): Dropout(p=0.3)
  (fc2): Linear(in_features=1024, out_features=1024, bias=True)
  (fc2_drop): Dropout(p=0.3)
  (fc3): Linear(in_features=1024, out_features=136, bias=True)
)
```

Fig. 12. CNN model architecture.

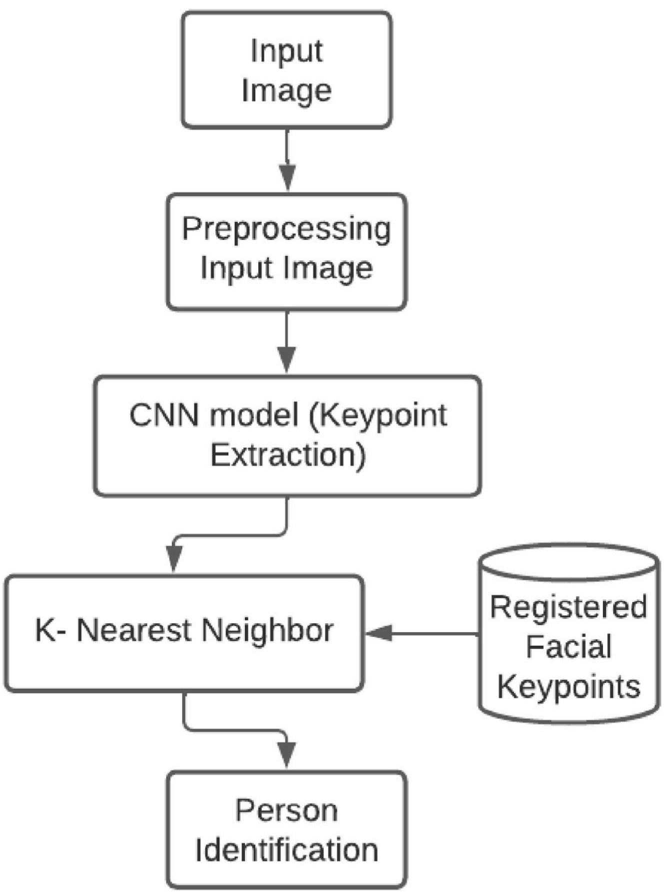

Fig. 13. KNN detection phase.

webcam. Also, this image passes through all the pre-processing procedures before being sent to the CNN model, which will forecast the 68 key points in that face image. The K-Nearest Neighbor algorithm receives this keypoint as input. The KNN model is simultaneously fed with the keypoint values of the already-registered users, which are recorded in a csv file. The retrieved CSV file contains all of the preregistered facial keypoint values as well as the person's name; each row has 129 columns. The first 128 columns contain the X and Y axis values of 64 keypoints, while the last column contains the person's name. The KNN algorithm will compare the key points that are currently being extracted with the key points that have been retrieved from the CSV file, and if it finds a match with an acceptable similarity, it returns that specific person's name, which will be fetched from the registration details CSV that has been retrieved. If no similarity is discovered, "Unknown" will be returned. The KNN measures the separation between the facial keypoint values that are plotted in Euclidean space. The overfitting is minimized in this case since the KNN model is trained with a K value of 10, which results in a smooth curve and also discussed in the Fig. 13. The K value is selected as 10, it a higher value for K even though it is selected because the keypoints data is has several outliers. This K value Leads to low variance and high bias, which eventually provides expected result in this case. The model reported a random person's name from the registered data when it was tested with a new user image after training with a lower K value. It ceased overfitting the data when the value was increased to a high number, like 10. The preregistered keypoint values that are obtained from the CSV file are initially plotted on a feature space on which the newly extracted keypoint value that is to be predicted is plotted. Using the K value of 10, it finds the closest 10 matches, of which 1 is chosen that is highly repeated. The closest match for the person's name linked to the keypoint is returned as the result.

5.2. Siamese network

The model is trained by encoding and calculating the triplet loss for positive, negative and anchor inputs are briefly mentioned in the Fig. 14. The positive and anchor input consists of the images from the same person. The negative input consists of images of different people.

While recognizing, the model crops the face in the input image to avoid the unwanted background using MTCNN and encoded using Inception Resnet v1. The images stored to recognise undergo the same process. The euclidean distance between the input image and images stored is calculated. If the distance calculated is lesser than the threshold value (which is calculated while training) the images are from the same person and steps described in Fig. 15. The name of the person will be displayed. If the distance calculated is greater than the threshold value the images are from different people.

6. Results

The end result of the project is more like a Graphical user interface which has 2 options the first one is 'register' which is used to register the face of the new person in the system. The second one 'Identify' which uses the previously stored facial keypoints values and predicts the user. The black space at the top of the GUI is for entering the name of the

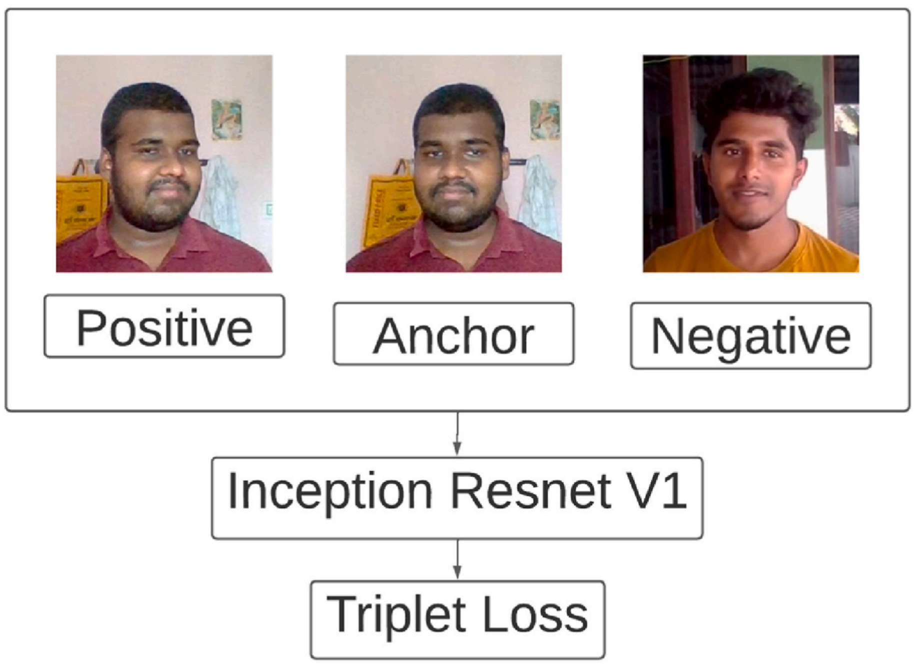

Fig. 14. Triplet loss calculation for real time data.

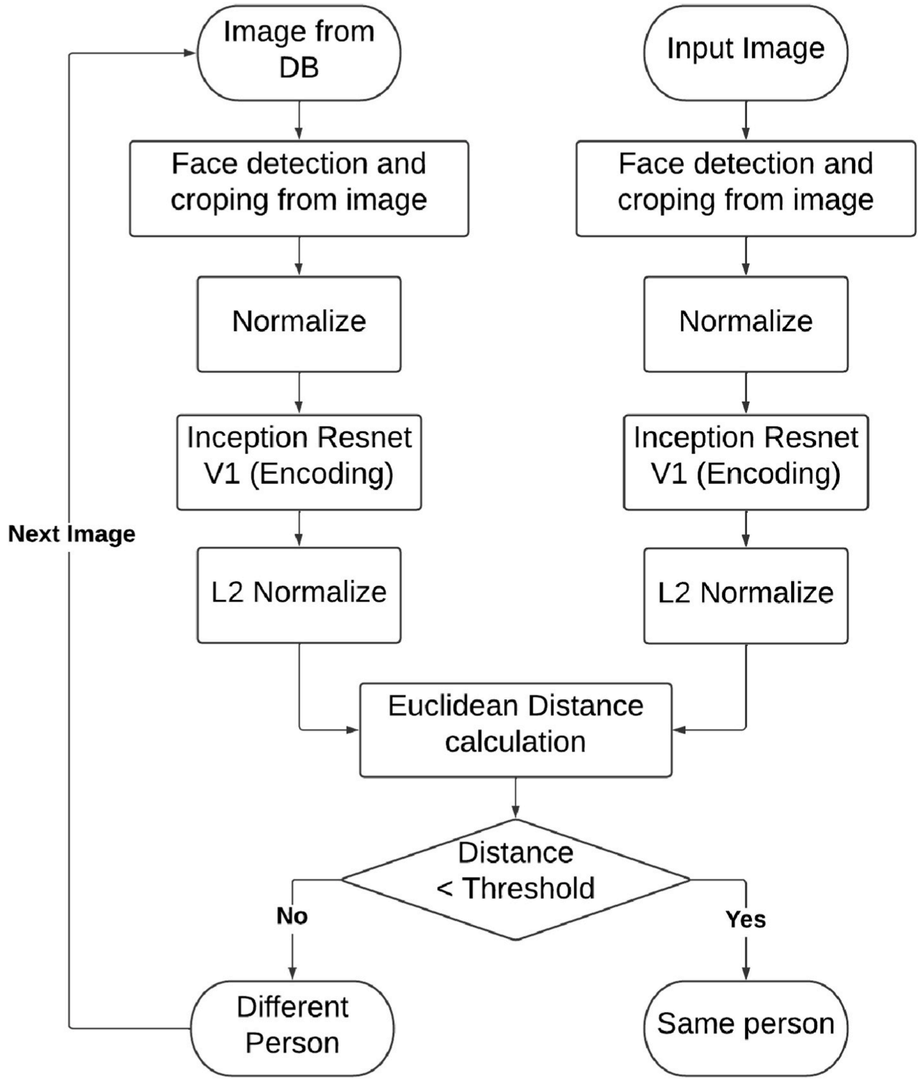

Fig. 15. Implementation of siamese network.

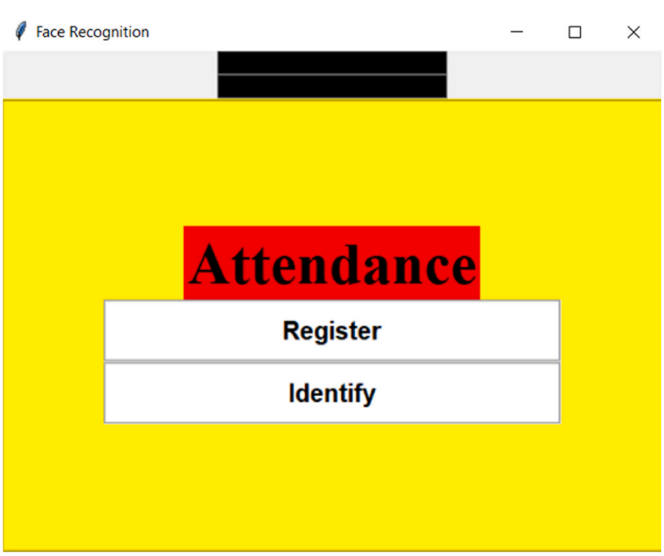

Fig. 16. Python GUI for the Attendance system.

person only during the registration time which can be seen in the Fig. 16.

During the first phase of registration, the user must enter their name and click the register button. A new window will be opened, and the video from the web cam will be recorded for 30 s before being closed. This video will be converted to 200 frames and saved as separate images in a new folder with the person's name as the folder name as well as the image name. For example, if the person's name is Sam, the first frame will be saved as 'sam1.jpg,' the next as 'sam2.jpg,' and so on represented as Fig. 17.

We ultimately decided on the KNN-based approach after achieving 99% prediction accuracy. When we tested our KNN-based model, it accurately predicted the 25 different persons used in training, and we also externally registered 10 new people's faces, which it also accurately predicted. When you click the 'Identify' button, a webcam image is captured and fed to the KNN-based model for testing. If it finds a match with the previously stored image, it will return the person's name as well as the 64 key point plotted image. Otherwise, it will simply display the person's image as is and indicate that they are unknown. Fig. 18 depicts the person's key points plotted image, along with his name. because he had previously registered in the system. In this GUI, method 1, which is a KNN-based approach, is used to make predictions.

Fig. 19 represents the training and validation accuracy of the Siamese Network is 0.9437 and 1. The training and validation loss of the Siamese Network is 0.2138 and 0.075.

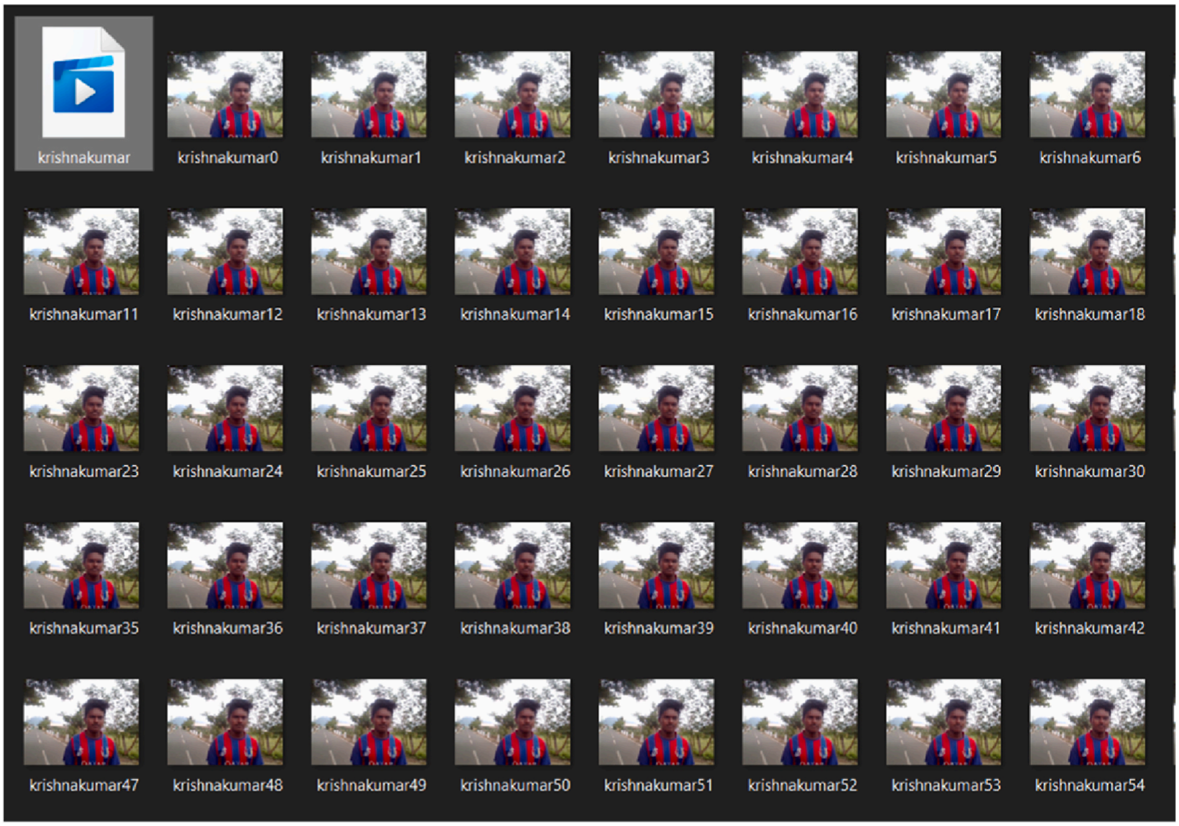

Fig. 17. Registration of a new person.

Fig. 18. Identification of the person.

Fig. 19. Validation loss of the Siamese Network.

7. Conclusions

This study mainly focuses on implementing two different approaches for facial recognition. The two methods were successfully experimented with and have produced optimal results. Both methods extracted the data from the images and used it for prediction extraordinarily. The models were initially trained using 25 people's facial images. When a broad range of facial photos are used, the performance of these approaches may alter appropriately. Therefore, utilising more diverse data for training and studying these models' behaviour will be the future focus of this study.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] V. Patil, A. Narayan, V. Ausekar, A. Dinesh, Automatic students attendance marking system using image processing and machine learning, in: 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 542–546. Link.
[2] S. Kumaar, R.M. Vishwanath, S.N. Omkar, A. Majeedi, A. Dogra, Disguised facial recognition using neural networks, in: 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), 2018, pp. 28–32. Link.
[3] K. Sasirekha, K. Thangavel, Optimization of K-nearest neighbor using particle swarm optimization for face recognition, Neural Comput. Appl. 31 (2019) 7935–7944. Link.
[4] I. Melekhov, J. Kannala, E. Rahtu, Siamese network features for image matching, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 378–383. Link.
[5] Jie Wang, Zihao Li, Research on face recognition based on CNN, IOP Conf. Ser. Earth Environ. Sci. 170 (3) (2018). Link.
[6] Mustafa H. Mustafa Zuhaer Nayef Al-Dabagh, Mohammed Alhabib, Firas H. AL-Mukhtar, Face recognition system based on Kernel discriminant analysis, K-nearest neighbor and support vector machine, Int. J. Relig. Educ. 5 (3) (2018) 335–338. Link.
[7] J.P. Jose, P. Poornima, K.M. Kumar, A novel method for color face recognition using KNN classifier, in: 2012 International Conference on Computing, Communication and Applications, 2012, pp. 1–3. Link.
[8] S. Shinde, M. Shende, J. Shah, H. Shelar, An approach for e-voting using face and fingerprint verification, in: 2020 IEEE Pune Section International Conference (PuneCon), 2020, pp. 59–64. Link.
[9] Xingping Dong, Jianbing Shen, Triplet loss in siamese network for object tracking, Proc. Eur. Confer. Comput. Vis. (2018) 459–474. Link.
[10] E. Jose, G. M, M.T.P. Haridas, M.H. Supriya, Face recognition based surveillance system using FaceNet and MTCNN on Jetson TX2, in: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 608–613. Link.
[11] V.K. Verma, V. Kansal, P. Bhatnagar, Patient identification using facial recognition, in: 2020 International Conference on Futuristic Technologies in Control Systems & Renewable Energy (ICFCR), 2020, pp. 1–7. Link.
[12] Shuai Peng, Hongbo Huang, Weijun Chen, Liang Zhang, Weiwei Fang, More trainable inception-ResNet for face recognition, Neurocomputing 411 (2020) 9–19. Link.
[13] W. Xing, Y. Bei, Medical health big data classification based on KNN classification algorithm, IEEE Access 8 (2019) 28808–28819.
[14] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient kNN classification with different numbers of nearest neighbors, IEEE Transact. Neural Networks Learn. Syst. 29 (5) (2017) 1774–1785.
[15] Z. Chen, L.J. Zhou, X. Da Li, J.N. Zhang, W.J. Huo, The Lao text classification method based on KNN, Proc. Comput. Sci. 166 (2020) 523–528.
[16] S. Kanithan, N.A. Vignesh, E. Karthikeyan, N. Kumareshan, An intelligent energy efficient cooperative MIMO-AF multi-hop and relay based communications for Unmanned Aerial Vehicular networks, Comput. Commun. 154 (2020) 254–261.
[17] Minhajur Rahman, Saimunur Rahman, Mohamed Uvaze Ahamed Ayoobkhan, On the effectiveness of deep transfer learning for Bangladeshi meat based curry image classification, in: 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), IEEE, 2022, pp. 1–6.
[18] Minhajur Rahman, Saimunur Rahman, Mohamed Uvaze Ahamed Ayoobkhan, Fine-tuned convolutional neural networks for Bangladeshi vehicle classification, in: 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), IEEE, 2022, pp. 421–426.
[19] S.T. Suganthi, Mohamed Uvaze Ahamed Ayoobkhan, Nebojsa Bacanin, K. Venkatachalam, Hubálovský Štěpán, and Trojovský Pavel. "Deep learning model for deep fake face recognition and detection.", PeerJ Computer Science 8 (2022), e881.
[20] T. Guo, J. Dong, H. Li, Y. Gao, Simple convolutional neural network on image classification, in: 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), IEEE, 2017, March, pp. 721–724.
[21] F. Sultana, A. Sufian, P. Dutta, Advancements in image classification using convolutional neural network, in: 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), IEEE, 2018, November, pp. 122–129.
[22] Y. Pei, Y. Huang, Q. Zou, X. Zhang, S. Wang, Effects of image degradation and degradation removal to CNN-based image classification, IEEE Trans. Pattern Anal. Mach. Intell. 43 (4) (2019) 1239–1253.