Face Recognition Using Face Embedding Method

Ihab Amer Abdullah[1*] **Jane Jaleel Stephan**[2]

[1]Informatics Institute for Postgraduate Studies, Baghdad, Iraq
[*] Corresponding author's Email: ms201920542@iips.icci.edu.iq
[2]University of Information Technology and Communications, Baghdad, Iraq
[*] Corresponding author's Email: janejaleel@uoitc.edu.iq

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: The recognition of faces is of great significance for real worlds applications like video surveillance, classroom students attendance recording, human-machine interaction, security systems etc. It is still several challenges to detection and recognition for recognising multiple faces because it is not easy for detecting multiple faces in one frame, it is also hard to recognise faces with low resolution. Compared to machine learning traditional approaches, approaches based-on deep learning shown better performance in terms of processing speed and accuracy in face recognition. Our method uses a combination of Viola Jones, Face-net and Support Vector Machine (SVM), and achieved an accuracy of 94% for 100 person through 166s and for real time face recognition, it achieved an accuracy of 100% for 10 frames through 9 seconds.

Keywords: Face-net, Face recognition, Face detection, Triplet-loss, SVM

1. Introduction

In this article, we introduce a system for verifying, recognizing and clustering faces. The methodology of our depend on learning the Euclidean embedding for every picture utilizing a deep convolution network that performs a features extraction process. The network is trained in which the squared distances "L2" in embedding space straightly correspond to the similarity of faces. The same person faces have small distances, and the different persons faces have a large distances "margin". As soon as produce this embedding, the above mentioned tasks becomes ready where simply the facial verification involve thresholding the distance among the two embeddings, and face recognition comes to be a classification issue; thus, it needs algorithms like SVM, k-NN; and the clustering can be achieved by using ready-made techniques like k-mean or agglomerative clustering. Former faces recognition methods that depended upon deep networks utilize a classification layer [1,2] trained onto the group of known faces identities where takes an intermediate (bottle-neck layer) as representation utilized for generalizing the recognition beyond the group of identities utilized in the training. The disadvantages of this method are its inefficiency and indirectness: one has to hopefulness that the representation of bottle-neck will generalize good to new faces, and by utilizing the bottle-neck layer, the representation magnitude for every facial is always very great (thousands of dimensions.). Some recently work [1] decreased this dimensionality by utilizing Principal Component Analysis (PCA), yet this linear transform can be easily learned in a single layer from the network. On the reverse of these approaches, Face-net immediately transforms the output to become a compact (vector) 128D embedding utilizing the triplet-loss. The triplets made of two pictures of the same person and a different picture for other person and the loss is intends to separating the pair of positive from the negative via distance margin. To illustrate the incredible differences that can handle by our method (see Fig. 1) that shows multi pictures that formerly considered very hard for facial verification systems. The summary of the remaining article as follow: in section (2) reviewing related work in this field, in section (3) a description of the model architecture used, in section (4) defines experimental results of face detection and recognition, lastly in section (5) and (6) we present a conclusion about the model and also future directions for young researchers.

Figure. 1 Bollywood dataset example that shows a high degree of variation in expression, illumination, occlusion, poses and makeup etc.

2. Related Work

R. Meng et al. (2014) [3] proposed a system of realtime face recognition using parallelization based-on Computer Unified Device Architecture (CUDA) to improve recognition speed. ViolaJones is used for facial detection, and PCA is used for recognition. They tested their work on the ORL dataset, and they found that a face recognition system based on CUDA is faster than working with CPU and the recognition rate is 87%. **G. Hu et al. (2015)** [4] proposed a CNN based single shot scale perceptual facial detection "SFDet" model, this model utilizes large scale layers related to the number of various anchors for dealing with different face volumes. For improving the accuracy, each loss function adds the IoU aware weight to the training sample, and the obtained accuracy is 89%. The method of "SFDet" can be recognizing faces in realtime. **IGPS. Wijaya et al. (2016)** [5] created a realtime face recognition engine for electronic keys. They utilized Haar-like features for detecting the face, and they utilized "LBP" feature extraction and "DCT" analysis, Direct Linear Discriminant Analysis "DLDA" for reducing feature vector size, and then test their work on (ITS) and (ORL) datasets and likewise realtime dataset, and in final they obtain an accuracy of 88%. **TS. Gunawan et al. (2017)** [6] proposed a system of face recognition utilizing "Raspberry Pi" that can link with a "smart home system". The camera will snap the front view of the face. The face region will be detected and segmented by the face detection routine. The Eigenface feature vector is extracted, and the PCA algorithm was utilized as a classifier. Face recognition algorithm output is connected with the relay-circuit, where the relay-circuit will lock/unlock the magnetic lock door. Open CV and Python are utilized for implementing feature extraction and classification. The Raspberry Pi limited processing power is discussed, which affects the resolution of picture to be snapped, processing time, memory and power management. When tested with three persons, the recognition rate 90%. **JKJ. Julina and T.S. Sharmila (2017)** [7] utilized the Histogram of Gradients "HOG" and Support Vector Machine "SVM" for face recognition system. They also utilized Viola Jones for facial detection. They achieve 90% of accuracy on ORL dataset. This work did not use realtime video. **S. Saypadith and S. Aramvith (2018)** [8] proposed a method for realtime multi-face recognition utilizing deep learning on an (Embedded GPU) system. The method utilized for facial detection based-on convolution neural network "CNN" with face tracking and state-of-the-art deep "CNN" face recognition algorithm. They achieved a multi-face recognition rate of 5 to 10 fps, with a rate of recognition of about 90%. **H. Ahamed et al. (2018)** [9] proposed a system of face recognition; they utilize: Histogram of Oriented Gradients (HOG) for features extraction and deep Neural Network (CNN) architecture referred as (HOG-CNN) for face recognition. The system achieved an accuracy about 89%. **S.Wu and D.Wang et al. (2019)** [10] studied the effect of gender and age on faces identification. For extracting features of face pictures, they utilize Multi task Cascaded Convolutional Networks "MTCNN" algorithm, and for constructing a model of face recognition, they designed a lighted convolution neural network as a classification method. The average recognition rate obtained is 83.73%.

We observe that most of the previous works did not achieve good accuracy in the case of unconstraint conditions, or they work slowly in realtime face recognition.

So, the major target of this article is to achieve high speed in realtime face recognition and to obtain high accuracy for face recognition in unconstrained conditions like (facial expressions, makeup, illumination, facial pose variation with 45◦, age, weight, occlusion and accessories) by using the Bollywood celebrity faces dataset [11] simulate these conditions.

3. The Proposed Method

Our project details are illustrated in this section; in our proposed system that is shown in Fig. 2,

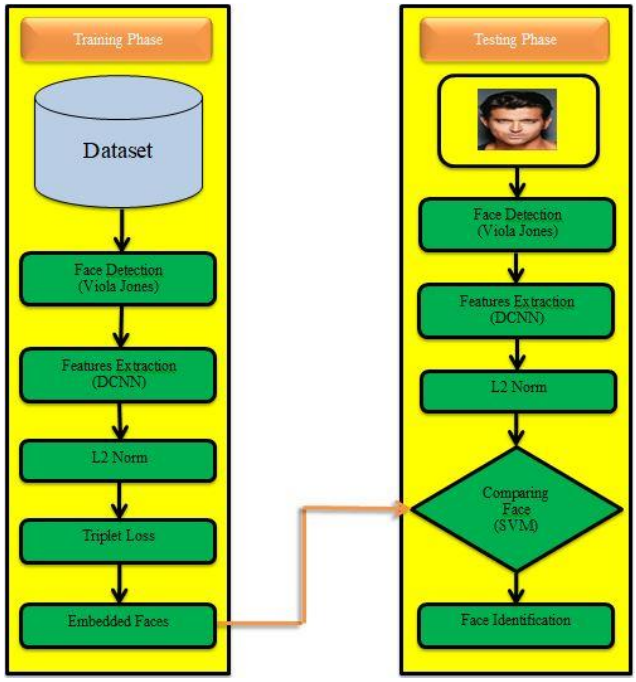

Figure. 2 Face Recognition System Block Diagram.

We use Viola Jones for facial detection and Face-net for feature extraction and embedding and SVM for classifying the face. Face-net utilizes a deep convolution neural network (DCNN) trained for straightly enhance the embedding itself, instead of the intermediate layer of bottle-neck like in former deep learning methods. In looking at the face-net model architecture and look at the details of the model and treats it such as black box (as illustrated in Fig. 3). The more significant portion of our method lies in end-to-end learning to the entire system.

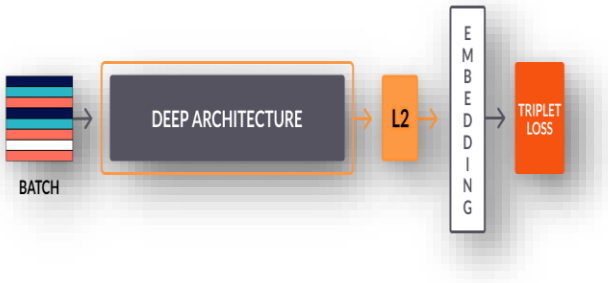

Figure. 3 Face-net architecture.

Then, L2 normalization compute vector length, where vector length is a single non negative value that describes the extent of the vector in Euclidean space. For the training phase, we use triplet-loss, which directly reflect the goals that we want we to attain is clustering the faces and recognize them. That is, we strive to embed f(x) from picture x into the feature space R^d, in which the square distance, regardless of picturing conditions, among all faces with the same identity, are small while the square distance is large for various face identities. In spite of we didn't make a straight comparison with the other losses, such as one that utilizing pairs of the negatives and positives as utilized in [12] Eq. (2). We think that the triple loss is most appropriate for facial verification. The motive is that loss in [12] encourages the projection of all the faces that belong to one identity into the embedding space at a single point. However, the triplet-loss attempts to impose a margin among each pairs face of one person and all the other faces, this allows to faces that belong to one identity to alive in the manifold while continue imposing the distance, therefore, distinguishability to the other identities, And for the testing phase, we utilize SVM where the labels and features are utilized as an input to the SVM classifier for finding the given test data versus relevant that be available subjects in the database.

3.1 Viola Jones Face Detector

We can be implementing facial detection by utilizing Viola Jones [13] algorithm as follow:

1. Haar-like features

In Fig. 4 when regions are similar, the features of every object give high output.

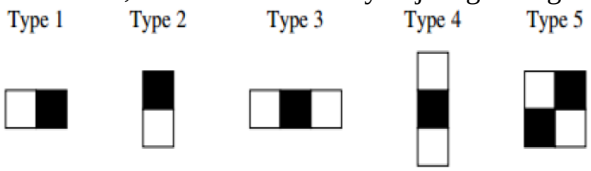

Figure. 4 Variant types of Haar-like features [14].

The generated output is given through formula: output = Σ (black region pixels) - Σ (white region pixels). Issues appear, though, due to the very high numbers of calculations that demand for execution. For example, a sub-window with (24 x 24) can provide more than 160000 features, and the summation of the pixel intensities must computed every time applied any of these features.

2. Sub-Window

This algorithm is inclined to solving the mentioned above issue, and that is done by decreasing the number of computations to be performed. For example, if we want to find sub-window S for a square in the picture below, it is given through Eq. (1).

$$S = \Sigma D + \Sigma A - \Sigma B - C \qquad (1).$$

As seen in Fig. 5, where S has 4 boundaries A, B, C and D, the idea is for converting the intensity of every pixel via summing all intensities of the pixel to the left and above it before Haar features applied, and this decreases the calculations to only 4 numbers per second.

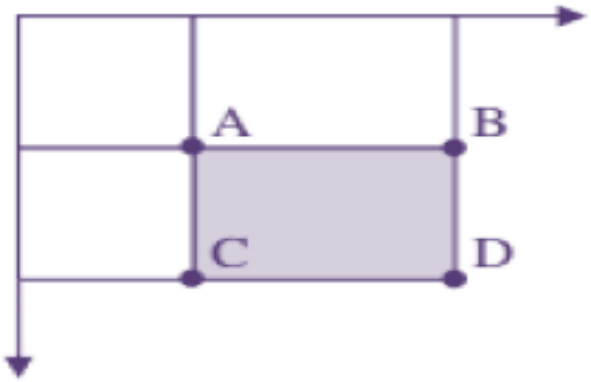

Figure. 5 Finding of the sub-window [14].

3. Adaboost

Even after obtaining the sub-window, the features remaining too many. Adaboost address this issue by decreasing the number of the features. This is attained through Eq. (2):

$$F(x) = \alpha 1 f\,1\,(x) + \alpha 2 f\,2\,(x) + \alpha 3 f\,3\,(x) \ldots + \alpha n f\,n\,(x) \qquad (2).$$

The weak classifiers f n(x), the strong classifier F(x), whenever α larger weight, whenever the feature was more similar. For instance, after the Adaboost, if the total features number is 160000 +, the number of features can be decreased to 6000.

4. Cascading classifier

Lastly, Fig. 6 shows different features that are cascaded into cascading classifiers. Cascaded classifier is a collection of stages that involves a strong classifier. The task of every stage is to check whether the sub-window is definitely might be a facial or non facial; when classifying the sub-window to be a facial, it be passed to the next stage; otherwise, when it classified to non facial, it is discarded.

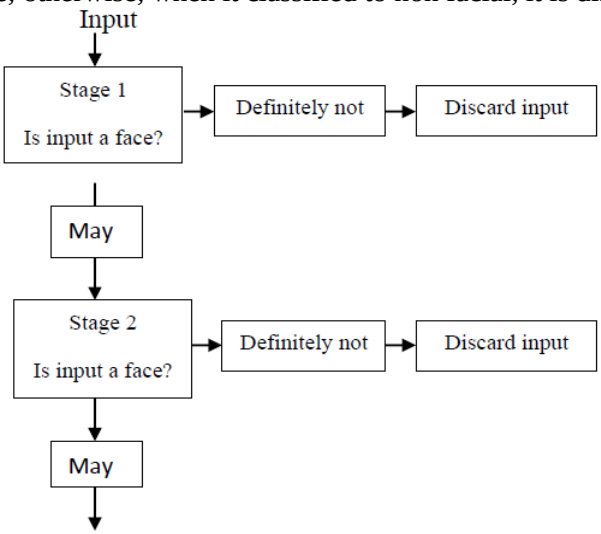

Figure. 6 Cascading classifier [14].

There is the menu of the pretrained models that are available to be used for face detection. For the suggested approach, we will use "haarcascade_frontalface_alt" model by using the Opencv Library.

3. 2 Face-net Facial Feature Extraction

Our method utilizes Face-net [15] to map faces pictures into 128-dimensional vectors and to set a fixed (threshold). This method assumes that 128 dimensional of features are equally important. Face-net (as shown in Fig. 7) was created by Google researchers utilizing Deep Convolution Neural Network (DCNN), which are

training by triplet-loss [15].

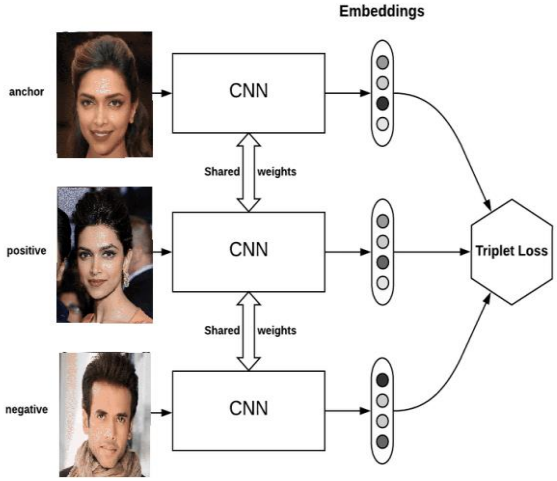

Figure. 7 Face-net feature extraction.

3.2.1. Deep Convolution Neural Network

DCNN add (1×1×d) convolution layers among the standard convolution layers result in a model 22 layers deep. For features extraction, DCNN will transform a face picture into a face features (Vector) in 128d. In our method, this extraction process used 3 channels (RGB) for 3 input of pictures for producing a 128-dimensional vector (as shown in Fig. 8) for each picture.

```
Embeddings = [[ 0.15712556  0.16223869 -0.07350685  0.02304433  0.06626721  0.11
916567
  -0.00147751  0.05570123 -0.09894355  0.00360533 -0.01800766  0.06121255
  -0.08468031  0.13364638  0.01314842  0.05079055  0.19097187 -0.05928673
  -0.00570582  0.06855612 -0.0720257   0.01806271  0.0992079  -0.06257208
   0.05009857  0.03702778 -0.1565432   0.06363221  0.01229365  0.01634614
  -0.06443868  0.01906191 -0.11676791  0.10478934  0.04679768  0.01584635
   0.05732548  0.161399   -0.01598018  0.09414192 -0.02743223  0.17681266
  -0.00460508  0.06940234 -0.09390938 -0.00829129  0.0126388   0.1288662
   0.07493604  0.15105447  0.05540227  0.03488234  0.07892174  0.05620448
   0.03895964 -0.00919146 -0.00779259 -0.02431026  0.08335686  0.03825601
  -0.0792567   0.00191882  0.09594572  0.11329497  0.06353781  0.0285796
  -0.07939383  0.15332824  0.16976792  0.11734817 -0.10451724  0.148393
  -0.06761308  0.06074035 -0.09513466  0.04770609  0.01993667 -0.08713998
  -0.05509707 -0.00522115 -0.08690462 -0.0067193   0.03004554  0.10516798
  -0.01525964  0.0057779   0.1335701  -0.14762495 -0.06814617  0.04127143
   0.06367216 -0.02875072  0.01192278 -0.12360761  0.01796147 -0.0075293
   0.19511141 -0.08859263 -0.08924861  0.04904627 -0.08207577  0.00701277
  -0.2439793   0.14733315 -0.07584107 -0.00099172 -0.09047384 -0.11252569
   0.03913896  0.03627214  0.00087608  0.01879008  0.02023168  0.1374438
   0.04290905 -0.08673028 -0.133026   -0.05342876  0.19645713 -0.16861248
   0.08060341  0.00891395 -0.02053973 -0.12750308 -0.08088595 -0.12381773
   0.07130403 -0.03647793]]
```

Figure. 8 128 dimensional face features.

3.2.2. Triplet-Loss

The triplet-loss function [15] use three pictures during evaluations. The anchor is an arbitrary picture (some person). The picture that is positive belong to the same class (same person). The picture that is negative belong to various class (different person) from the anchor. The triple loss reduces the distance among the anchor and positive picture while increasing distance among anchor and negative picture, using Eq. (3).
We want:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 \; \forall \; (x_i^a, x_i^p, x_i^n) \in T \qquad (3).$$

The loss that is being minimized is then

$$L = \sum_i^N max[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+ \quad (4).$$

Where x refer to the picture, f(x) refers to the face, x_i^a refers to (anchor), x_i^p refers to (positive), x_i^n Refers to (negative), the loss is referred by L, α is a threshold "margin" imposed among negative and positive pairs, T is a set of all the possible triplets in a training set and it has a cardinality N.

When training the DCNN network with function of triplet-loss, the input consists of three pictures, two of which belongs to same class, and the last one belong to various class. So our method process each picture and produce a feature vector. In the end, we use the triplet-loss function in the training to achieve that the distance between two pictures from different classes will be great and the distance between pictures from the same class be small (see Fig. 9).

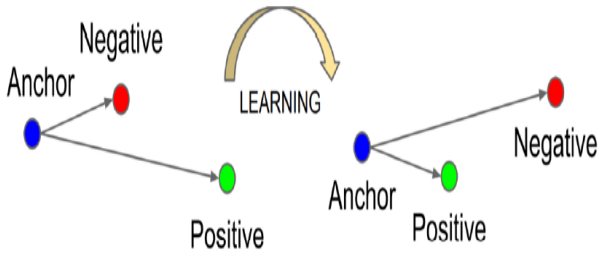

Figure. 9 Triplet-loss Principle.

For ensuring speedy convergence, this meaning that forgiven x_i^a we want to determine an x_i^p "hard positive" where (argmax) x_i^p $\|f(x_i^a) - f(x_i^p)\|_2^2$ and as well x_i^n "hard negative" where (argmin) x_i^n $\|f(x_i^a) - f(x_i^n)\|_2^2$.
It not useful to calculate (argmax) and (argmin) through a full training set. In addition, it may be leading to weakness training, where poorly and mislabelled captured faces will dominating of the hard negatives and positives. Two options are there that avert this problem:

- Generating triplets "off-line" each n steps, employing most modern network check-point and calculating (argmax) and (argmin).
- Generating triplets "on-line" may be done via choosing hard positive and negative samples from inside a (mini batch).

Here we focusing on "on-line" generation and utilize (mini batches) and just compute (argmax) and (argmin) inside a (mini batch).

3.3 Support Vector Machine (SVM)

SVM [16] classifier has a robust ability to learn and generalise and has distinctive benefits in solving issues like small samples and non-linearity. The essential idea of the SVM is to mapping the input vector into a high dimensional space through non-linear transformation and then create the optimum classification surface in the high dimensions space. This non-linear transformation is realized via choosing a suitable inner product kernel function. Converting the vector into a high dimensions space simply change the computation of the inner product, and the complexity of the algorithmic will not get increase with the increasing of dimensions number. In our method, we have the linear separable case illustrated in Fig. 10.

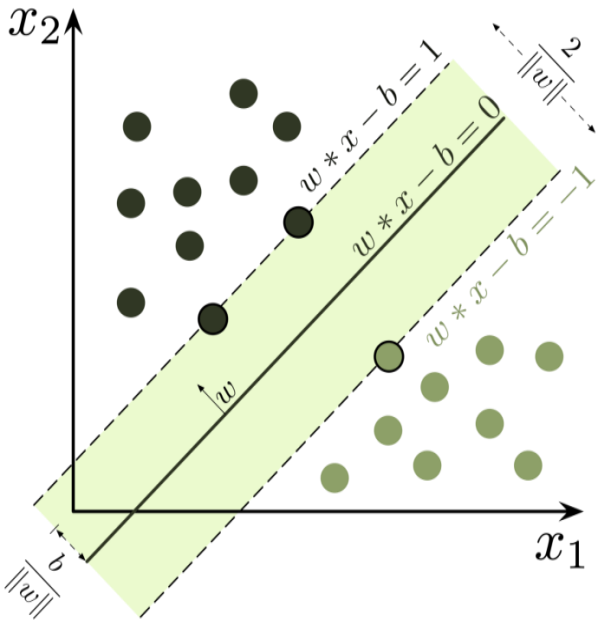

Figure. 10 This figure shows the linearly separable case.

In the linearly separable case for the training set (xi, yi), where xi ϵ R_N, yi ϵ {-1,1}, i=1,2,..…,n, the SVM purpose is that to finding the "hyperplane" that can segregate the two samples types, and maximizing the interval among the two classes. Assume the equation of hyperplane is:

w.x+b=0 (5).

If w.x+b>0, refer to the class 1, and if w.x+b<0, it refer to the class -1.

Maximize the interval among the two classes are equal to minimize the following formula:

$J(W) = \frac{1}{2}||w||^2$ (6).

4. Experimental Result

In this section, the above explained method function is to use a training set. So, the experiments are performed on the Bollywood celebrity faces dataset. In this analysis, the proposed system performance is evaluated based-on the face recognition rate percentage of the test set. This dataset has faces of 100 Bollywood celebrities. Every class of the person has coloured samples between 80 to 150, and samples contain wild conditions such as makeup, different poses, facial expressions, illuminations, age transitions, occlusion etc. In this experiment of face recognition in pictures, for every person, we used 15 pictures for training and another one picture randomly selected for testing, where the obtained accuracy is 94% for the total number of the persons through 166s and for real time face recognition, we gained result 100% for 10 frames through 9 seconds.

We can use a laptop internal webcam or any external webcam. In our method, we used an external webcam Microsoft Lifecam Studio (shown in Fig 11), for realtime experiments.

Figure 11 Microsoft LifeCam Studio.

There is an experimental example in Fig 12 for face recognition in pictures and another experimental example in Fig 13 for real-time face recognition, respectively.

Figure. 12 Picture face recognition examples.

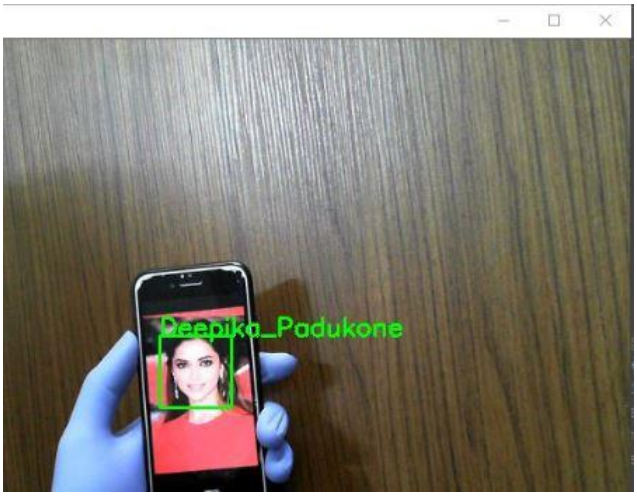

Figure 13 Real-time face recognition.

5. The Face Recognition System Applications

The face recognition system applications [17] are shown in Fig 14. Where face recognition is utilized in numerous fields, like crime prevention and people verification. For example, as in the following list:

Figure. 14 Face recognition applications [17].

1. Security: the main concern in public areas is security. Around the world, numerous airports utilize face recognition for public safety. Additionally, we can utilize a recognition system in several applications of security, like border checkpoints, ATMs, computer and network security.

2. Biometric Surveillance: is broadly utilized in the field of surveillance such as banks, stores, retail, casinos, and sports arenas.

3. Criminal justice systems: mug-shot/booking systems, forensics and post event analysis.

4. Smart Card Applications: facialprint can be store in the smart card, barcode or magnetic stripe, and recognition occurs via matching the saved template with the targeted picture.

5. Identity Verification: Verifying a claimed identity depending on the facial picture for querying a passport, driver's license etc.

6. Conclusion

The suggested method introduces a type of biometrical system based-on faces recognition that can be used in an intelligent buildings environment. In the experimental section, in the Bollywood Celebrity Faces Dataset, our system realizes new record accuracy of 94% for face recognition in pictures and 100% for 10 frames through 9 seconds for realtime face recognition. The proposed system proved that face recognition technology is the appropriate verifying way in general institutions as it can be used in many implementations like surveillance, classroom attendance, security systems, access control etc. The suggested system is appropriate because it is a method of face detection and recognition with very low consumption, especially it does not need external devices.

7. Future work

The future work for the proposed system will be mainly concentrated on the development of the system. There are several aspects that may develop the system:-

- It can develop the system by adding filters that help to improve the ability of a system to work under more difficult conditions like Contrast-Limited Adaptive Histogram Equalization (CLAHE), face alignment etc.

- Also can replace the SVM classifier with a more robust classifier like (KNN).

References

[1] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2892–2900.

[2] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[3] R. Meng, Z. Shengbing, L. Yi, and Z. Meng, "CUDA-based real-time face recognition system," in *2014 Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, 2014, pp. 237–241.

[4] G. Hu *et al.*, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 142–150.

[5] I. G. P. S. Wijaya, A. Y. Husodo, and A. H. Jatmika, "Real time face recognition engine using compact features for electronics key," in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2016, pp. 151–156.

[6] T. S. Gunawan, M. H. H. Gani, F. D. A. Rahman, and M. Kartiwi, "Development of face recognition on raspberry pi for security enhancement of smart home system," *Indones. J. Electr. Eng. Informatics*, vol. 5, no. 4, pp. 317–325, 2017.

[7] J. K. J. Julina and T. S. Sharmila, "Facial recognition using histogram of gradients and support vector machines," in *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2017, pp. 1–5.

[8] S. Saypadith and S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, pp. 1318–1324.

[9] H. Ahamed, I. Alam, and M. M. Islam, "HOG-CNN Based Real Time Face Recognition," in *2018 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, 2018, pp. 1–4.

[10] S. wu and D. Wang, "Effect of subject's age and gender on face recognition results," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 116–122, 2019.

[11] "100-bollywood-celebrity-faces @ www.kaggle.com." .

[12] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *arXiv Prepr. arXiv1406.4773*, 2014.

[13] K. D. Ismael and S. Irina, "Face recognition using Viola-Jones depending on Python," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1513–1521, 2020.

[14] J. Kaur and A. Sharma, "Performance analysis of face detection by using Viola-Jones algorithm," *Int. J. Comput. Intell. Res.*, vol. 13, no. 5, pp. 707–717, 2017.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[16] H. Chen and C. Haoyu, "Face Recognition Algorithm Based on {VGG} Network Model and {SVM},"

J. Phys. Conf. Ser., vol. 1229, p. 12015, May 2019.

[17] Y. Kortli, M. Jridi, M. Merzougui, A. Alasiry, and M. Atri, "Comparative Study of Face Recognition Approaches," in *2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, 2020, pp. 300–305.