

## An Automated Framework for Patient Identification and Verification Using Deep Learning

Suresh Tommandru\*, Domnic Sandanam

Department of Computer Applications, NIT Tiruchirappalli, Tamilnadu 620015, India

Corresponding Author Email: [405118003@nitt.edu](mailto:405118003@nitt.edu)



<https://doi.org/10.18280/ria.340605>

### ABSTRACT

**Received:** 20 October 2020

**Accepted:** 1 December 2020

#### Keywords:

*deep learning framework, face detection, face recognition, patient identification, patient verification*

Automated patient identification and verification are very important at a medical emergency and when patients are not carrying his/her identity. It is a risk factor that identifying the correct patient identity for doctors to provide medical treatment. The majority of the identification or verification is being done by wristbands, RFID tags, fingerprint, face detection by using handcraft feature-based face recognition systems. A new framework based on robust deep learning model and contrast enhancement is proposed in this paper. In the proposed work, the light illumination problem has been addressed by the contrast enhancement technique for deep learning models to recognize the face. It is proved that the inclusion of contrast enhancement is improving patient identification and verification. To evaluate the deep learning framework, the proposed deep learning models have been trained on our own dataset and have been tested with a real-time medical providing agency. The experimental results show that the proposed framework exhibits more robust test results with accuracy than existing hand-crafted techniques under the live webcam video capture for the real-time patient detection system.

## 1. INTRODUCTION

Patient identification and verification play an important role to provide medical facilities to the patient in the hospital environment at the time of emergency. Over the last few years, it has become popular in the healthcare industry because of its wide range of uses such as providing corresponding doctor's consultation, providing safety to the patients, acquiring healthcare facilities by the patient, storing and retrieving the electronic health records. Till now, the identification of patients is being done by using patient health cards, identification wristbands [1, 2] or radio frequency identification tags [3], fingerprint, iris, the palm of the patient and questioning the patient for their name and date of birth or his/her supporters for their information. According to their identification, hospital staff can get their medical electronic health record to provide corresponding doctor's consultation and to provide medical facilities. But, at the time of patients' visits to the hospital or in an emergency, if the patient is unable to provide their identity as above, then it is a challenging task to identify the patient.

If the hospital staff is doing patient identification manually then wrong patient identification often occurs due to staff lack of knowledge, poor practices, and staff fatigue with the workload. A recent edition of Serious Hazards of Transfusion annual report 2017 [4] contains 115 cases of mistakes including inadequate identification of patients [5]. This type of mistake leads to performing surgery on another patient, error in medication usage or treatment, etc. Some of the examples are removed the patient's wrong eye, replacement of hip to the wrong patient, and breakdown of the wrong limb [6]. The truth is that it causes patients' death because they have been wrongly identified.

There is a strong need in identifying the patient

automatically without questioning or asking the patient's identity when the patient is unable to carry the health cards, tags, or bands. Different solutions have been provided by the existing conventional methods for automatic patient identification. Generally, automatic patient identification and verification can be done in two ways [7], (i) face detection, and (ii) face recognition. The main objective of Face detection is to identify the face presented in the given image and produce facial features with their locations [8]. These facial features are used to recognize the face. The traditional face detection systems have been developed based on feature-based approaches. The feature-based approaches have used Facial features [9, 10], Textures [11], the color of the Skin [12, 13], Eigenface [14] and View-based [15] for face detection. The existing face detection systems have also used Neural network [16], Support Vector Machine [17], Naïve bayes [18], Hidden Markov Model [19], Multi-resolution Rule [20], Predefined Face Templates [21] and Deformable Templates [22] for detecting the face.

Recognition of the face is non-contact. Faces can be captured from a web camera and recognize without any user interaction. Face recognition consists of three stages [23] i.e. (i) capturing the face by using the web camera (ii) detecting the face in the captured image (iii) matching the detected face features with stored face features which are in the database and providing the similarity score [24]. If the score is high, then the corresponding patient identity will be provided to the user. The applications of face recognition are patient identification and verification in medical emergencies [25], patient heart rate estimation [26], accessing out-patient information through electronic medical records [27], video oculo-graphy [28], and embedded security systems for accessing medical facilities [29]. Algorithms for Face recognition can be classified as Template and Geometric Feature-based approaches [30],

Piecemeal and Holistic approaches [31], Statistical Method based approaches such as considering principal components [32], transformation technique [33], Linear combination of features [34], linear projective maps [35], Wavelet Transformation [36], Independent Component Analysis [37], Kernel Principal Component Analysis [38] and Neural Network-Based [39]; and View-Based & Modular Eigenfaces [40] approaches.

These existing conventional methods [9-40] are failed to detect and recognize the face due to light variation, pose variations, occlusion, and background clutter. These methods have used features such as Haarcascade [41], Histogram Oriented Gradient [28], Local Binary Pattern [42], Eigenfaces [40], etc. to extract the facial features from the image. But these features are lacking in face detection accuracy as well as in face recognition accuracy.

Recent studies have been exploited to overcome the drawbacks of feature-based approaches by using deep learning methods. In many areas, deep learning techniques have been rapidly developed and considerable improvements have been accomplished. In the last couple of years, to increase the accuracy of face detection, Convolutional Neural Networks (CNN) have been introduced. Unlike, the hand-generated features, these methods usually take images as input and learn the feature descriptions for different tasks automatically. Due to this, deep learning methods could improve face detection under extreme conditions. Deep learning methods achieve better performance due to the availability of enormous training data and more computational resources such as a large number of CPU cores, GPU cores. There are many deep learning methods to perform face detection and recognition on both images and video such as DDML [43], DeepFace [44], DeepID2+ [45], FaceNet [46], VGGFace [47], CNN-ROI [48],

ASML [49], ADRL [50], SDAE+DBM [51], ABM [52], etc. Unfortunately, these models often contain several parameters and computations that hindrance of efficiency of the face recognition system. Practical implementation of these works has also not been discussed and demonstrated in the recent papers [42-52] on face detection and face recognition. A new framework proposed in this paper can be used for patient identification and recognition in real-time scenarios. The practical application of this work is born out of the growing need in every hospital environment to identify and verify the patient automatically and retrieve the patient health records from the database. This paper contributes to the following:

- This paper investigates an automated framework to identify and verify the patient using the deep learning model in the hospital environment. This work also studies the performance of the methods such as Haarcascade, Local Binary Pattern, Histogram of Gradient, DLlib (library for machine learning) being used in recent years to identify the patient.
- This paper also investigates the need for contrast enhancement for feature-based and deep learning-based methods.
- The paper performs the analysis on the output with and without contrast enhancement of a deep learning-based framework. The results are shown that it is poor without contrast enhancement of the deep learning method for patient identification.
- The proposed framework is implemented and evaluated in real-time using client-server architecture and It is also compared with the output of the feature-based systems.

The architecture of the patient identification and verification framework is given in Figure 1.

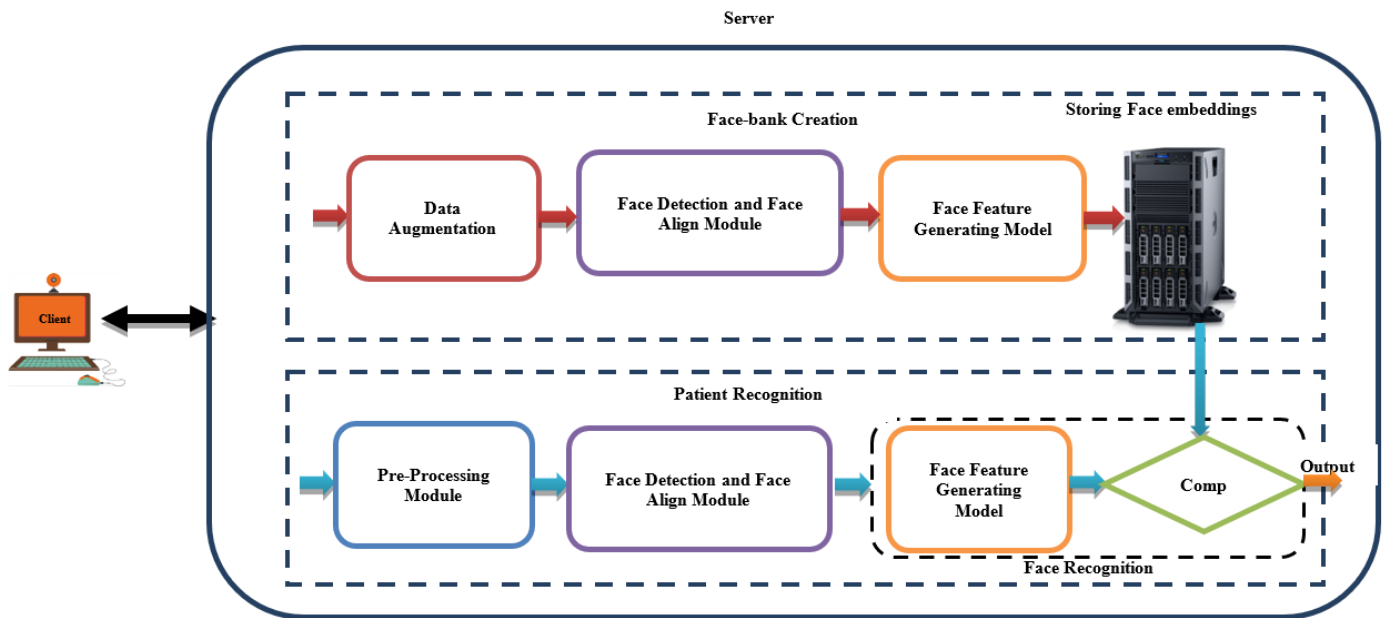


Figure 1. Client-server architecture for automated patient identification and verification

## 2. RELATED WORK

Aseem et al. [53] have proposed a strategy called SMART that is incorporating the patient's photograph in the electronic medical record to easily identify the patient that makes it simple for the doctor to provide health care to that patient. This

strategy contains two phases. First, medical identity cards are provided to the patient. Such cards highlighted the benefit of getting a patient photo on a medical card to help the front-end staff to identify patients correctly. Second, the face of the patient is captured at the time of enrolling by arranging a casual face-to-face meeting. This paper shows the importance

of having photographs in the electronic medical record by the hospital staff. The limitation of the strategy is the manual taking of photos of the patient.

Johnson et al. [54] have demonstrated the method for easy and fast retrieval of patient records using face recognition techniques. Capturing the patient face through the web camera, face landmarks identification on the captured image, drawing the bounding box on the image, thumbnailing the bounding box region, making the embeddings as a 128-vector of the thumbnail are the different tasks used for face recognition. Then, the embeddings are sent to the server for retrieving the medical records of the corresponding patient. In the server, a multi-dimensional tree search algorithm takes the embeddings (vector) and retrieve the patient's record from the database by using hashing with vector. The drawback of the approach is that while capturing the image, it fails at the lighting variations.

Jeon et al. [55] has developed an android application with a modified version of the facial recognition engine powered by Oezsoft Inc [56]. After capturing the face of the patient by using the mobile phone camera, the facial landmarks are extracted by using the face recognition method, and the facial template is generated by using three-dimensional facial learning data to accumulate a three-dimensional view from the extracted facial landmarks. Then, the euclidean distances between the facial landmarks from the three-dimensional facial template are calculated, which are used to retrieve the patient record. The drawback of the application is that if the image size is below 120x120 pixels, then the face scan will not be detected. The light sensitivity is also the constraint on the app. It is also consuming time up to 5 minutes when this method is used to recognize the patient in dark places with different light conditions.

Mehra et al. [29] have demonstrated the method of face detection with the DLib [57] and face recognition algorithm [58] by evaluating the distance between the two face embeddings. In this method, faces are captured by the camera. The shape, color, texture features of the faces are extracted and these features are made as to the 128 feature vector or face embeddings. Then, the distance between the extracted face embedding and the target face embedding is stored/trained. The distance value is used to recognize the face. The limitation of the method is that it has not been deployed in a real-time scenario.

Naruniec et al. [28] have proposed an oculo-graphy system that contains four modules (i) face detection (ii) tracking the face after localization (iii) Iris localization (iv) Iris position regression. In this paper, the authors have examined Haar features, Histogram oriented Gradients (HoG) for face detection and simple probabilistic approach, logistic regression approach for classifying the face. But, these features have a more miss-classification rate as per the results reported in the paper.

Pabiania et al. [27] have developed a product with enhanced monitoring and management system for the patient's medical record in the hospital. It is aimed to provide easy identification of the patient and easy access to medical records of the patient to the doctors and nurses. It is composed of two modules, hardware and software modules. The software module used the viola-jones algorithm [41] for detecting the face and the AdaBoost classifier is used for face classification. The limitation of the method is that it produces different detections according to the overlapping windows while extracting the facial features due to its sensitivity to light illumination. It will not generate facial landmarks because it works on holistic

images with a uniform background which may not be satisfied in most natural scenes.

Shagholi et al. [18] have developed a method to predict the heart rate of the patient from a web camera video. Extracted the facial features by using Viola-jones method. The existence of a face in the image is done by using AdaBoost classifier. The method's limitations are sensitive to lighting conditions and face detections.

Feature-based methods are performing face recognition using hand-generated features. But, these are not robust in light conditions, pose variations, occlusions. Hence, To resolve the above issues in a real-time environment, we propose a deep learning framework for face recognition in a client-server architecture.

### 3. PROBLEM STATEMENT

For improving the quality of the health services in hospitals, the identification of patients is being done by using patient health cards, identification bands or tags, fingerprints, iris, the palm of the patient, questioning the patient or his/her supporters for their information to offer better medical treatment. According to their identification, hospital staff can get their electronic health records to provide corresponding doctor consultations and medical facilities. But, at the time of visiting the hospital or in an emergency, the patient is unable to provide their identity as above then it is a challenging task to identify the patient. There is a need in identifying the patient without questioning the patient and a situation that the patient unable to carry the health cards, tags, or bands. Hence, a new deep learning-based framework as given in Figure 1 is proposed in this paper for the face recognition system to identify, verify the patient for the retrieval of the electronic health record of the corresponding patient.

### 4. PROPOSED METHODOLOGY

The proposed deep learning-based framework shown in Figure 1 captures the image of the patient through the web camera then the image is passed through the process of face detection and recognition. Our proposed deep learning-based framework contains two parts:

- (1) Face-bank creation
- (2) Patient recognition

Face-bank creation is the pre-requirement for identifying the patient. It will be done at the time of patient enrolment. It is retained in the server. This face-bank is used for patient recognition on every visit of the patient.

#### 4.1 Face-bank creation

The proposed methodology given in Figure 1 shows that there are two major components: face\_detection with alignment, face\_feature generation for face-bank creation. The patient image will be captured through the web camera while the patient is enrolled. Patient images will be collected for all the patients. The model used to detect the face is given in Figure 2. The Face\_detection model receives the patient image as input then generates the pyramid of images with different scales. Those pyramid of images is passed through three convolutional neural networks to generate candidate windows in initial networks and to reject false candidates, merge the

mostly overlapped candidates in later networks. This model will classify face or non-face in the produced candidate window of the image. This process is repeated for every new patient. Then, all the face images produced by the face\_detection model are used to train the face\_feature generation model given in Figure 3 to generate deep facial features. The facial features generated by the trained feature\_extraction model are stored as feature vectors called face-bank. In real-time scenarios, we are capturing the patient's single image through the web camera and hence, it is produced a small amount of data, But, we need a massive amount of data to generate deep facial features and prevent overfitting. In practice, it is expensive to acquire such sets of images, time-consuming and required human interaction. This

problem has been solved by using data augmentation in the proposed framework.

#### 4.1.1 Data augmentation

Data augmentation is the process of synthesizing new data according to the given input data with simple affine and elastic transformations [59]. We are capturing a single image of the patient through the web camera for patient identification. With that patient's single image, the deep model can not be trained well. There is a need for generating more data for the captured patient image. We have generated data by applying horizontal flip, rotation, translation, brightness changes on the original image. These generated data can improve the generalizing capability of deep models.

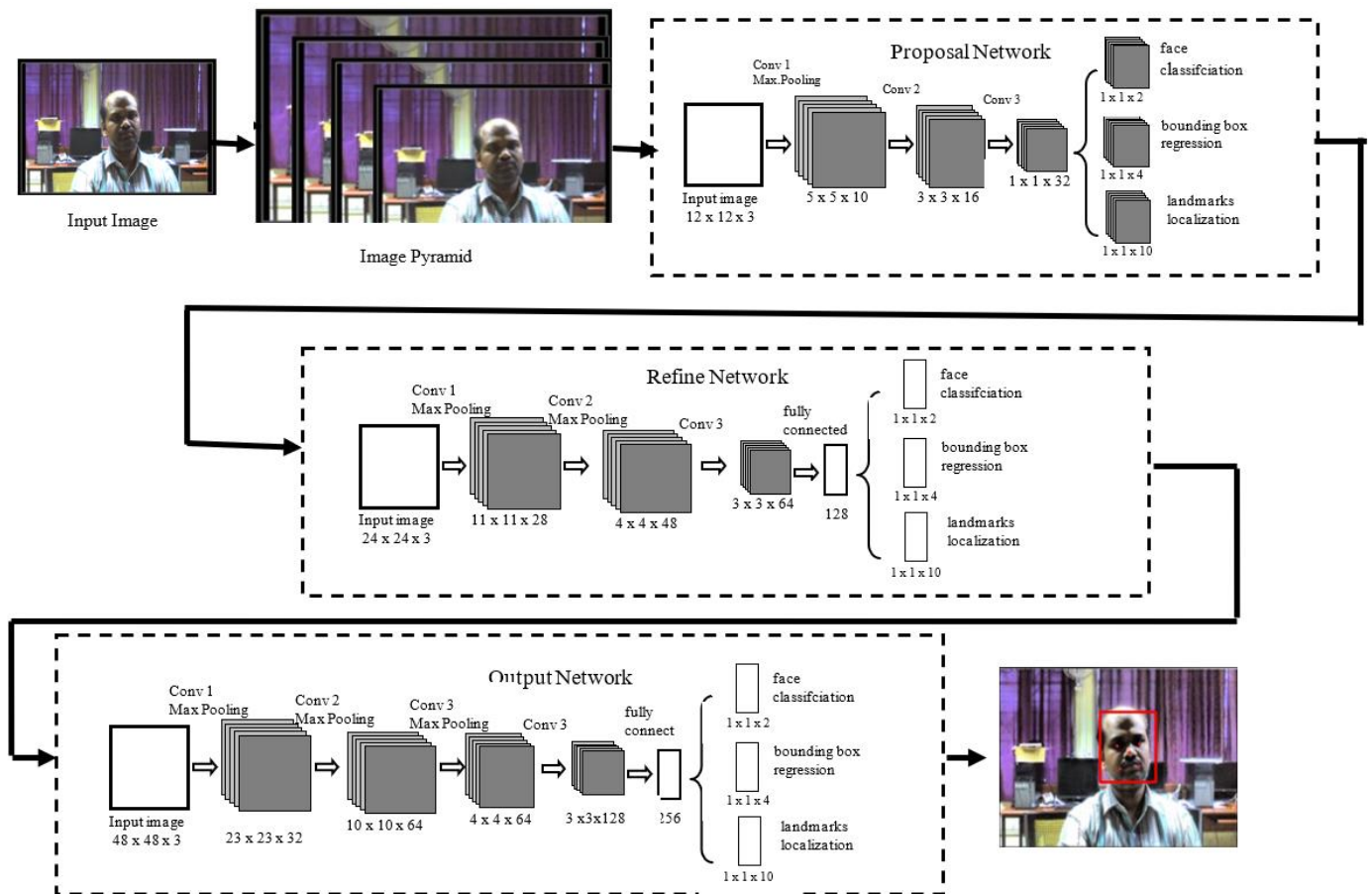


Figure 2. Face detection model

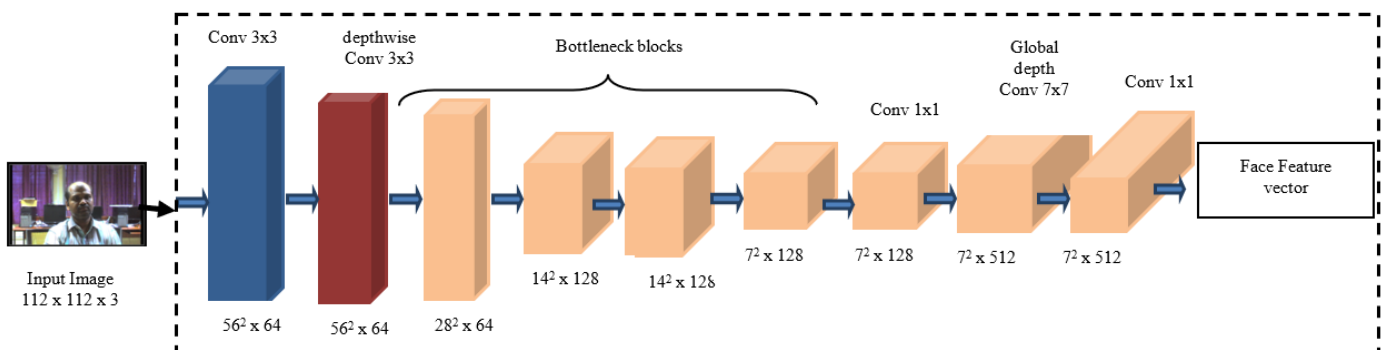


Figure 3. Face feature generation model

#### 4.1.2 Face detection with face alignment

The augmented data is passed through the deep face detection model given in Figure 2 to generate a detected face for each image in the set. In our work, we follow Multi-Task Cascaded Convolutional Network (MTCCN) [60] to detect a face. In this network,  $3 \times 3$  filters are used to reduce the computing time and the nonlinear function PReLU is used after the pooling layer, convolution layer, and the fully connected layer (except output layer) [61]. This network shown in Figure 2 is exploited as a face detection model in our work. The given input image is resized into different scales of images to generate an image pyramid. This pyramid is passed through the cascaded convolutional networks to detect the face. The face detection model contains three stages CNNs. In the first stage, CNN is called a proposal network. It is used to detect potential candidate facial windows and their corresponding regression vectors. The output of the proposal network is represented as

$$\text{CNN1}_{\text{Output}} = \{\text{pbbox}_i, p_i\} \quad (1)$$

where,  $p_i$  is the probability of having the face.  $\text{pbbox}_i$  is the proposed bounding box. After generating the probabilities for each detected window by using the softmax, the new bounding box for the detected window is detected using a bounding box regressor.

The regressor takes a set of training pairs  $\{\text{pbbox}_i, \text{gbbox}_i\}$ . Where  $i$  represents the number of training pairs  $i = 1, \dots, N$ . Each set consists of a proposed bounding box  $\text{pbbox}_i = (p_x, p_y, p_w, p_h)$  and a ground-truth bounding box  $\text{gbbox}_i = (g_x, g_y, g_w, g_h)$ . The regressor aims to learn the transformation between the  $\text{pbbox}_i$  and the  $\text{gbbox}_i$ . In the regressor, four transformation function  $\{\Delta_x(\text{pbbox}_i), \Delta_y(\text{pbbox}_i), \Delta_w(\text{pbbox}_i), \Delta_h(\text{pbbox}_i)\}$  are used and called them as offsets <sub>$i$</sub> . Each function in the offsets is modeled as a linear transformation function of the convolution layer features represented as  $\phi(\text{pbbox}_i)$ . The offsets <sub>$i$</sub>  are generated by using:

$$\{\Delta_{\# \in \{x,y,w,h\}}(\text{pbbox}_i)\} = \{W_{\# \in \{x,y,w,h\}}^T \phi(\text{pbbox}_i)\} \quad (2)$$

where,  $W_{\#}$  is the learnable model parameter and it is learned by using Eq. (3).

$$W_{\#} = \underset{\hat{W}_{\#}}{\text{argmin}} \sum_{i=1..N} \left( t_{\#}^i - \hat{W}_{\# \in \{x,y,w,h\}}^T \phi(\text{pbbox}_i) \right)^2 + \lambda \|\hat{W}_{\# \in \{x,y,w,h\}}^T\|^2 \quad (3)$$

where,  $t_{\#}$  are regression targets for the training pair  $\{\text{pbbox}_i, \text{gbbox}_i\}$  and they are calculated using Eqns. (4)-(7).

$$t_x = (g_x - p_x)/p_w \quad (4)$$

$$t_y = (g_y - p_y)/p_h \quad (5)$$

$$t_w = \log(g_w/p_w) \quad (6)$$

$$t_h = \log(g_h/p_h) \quad (7)$$

The offsets or transformation functions given Eq. (2) are used to calibrate the proposed candidates (face region). The calibration will be done by using the below Equations.

$$p_g x_i = p_x_i + \Delta_x(\text{pbbox}_i) * p_w_i \quad (8)$$

$$p_g y_i = p_y_i + \Delta_y(\text{pbbox}_i) * p_h_i \quad (9)$$

$$p_g w_i = p_w_i * \exp(\Delta_w(\text{pbbox}_i)) \quad (10)$$

$$p_g h_i = p_h_i * \exp(\Delta_h(\text{pbbox}_i)) \quad (11)$$

$$\widehat{\text{pbbox}}_i = (p_g x_i, p_g y_i, p_g w_i, p_g h_i) \quad (12)$$

Then the  $\text{CNN1}_{\text{Output}}$  output becomes  $\{\widehat{\text{pbbox}}_i, p_i\}$ . After calibrating the proposed candidates with probabilities, they are given to Non-Maximum Suppression (NMS) due to some of the face regions are mostly overlapping, those regions will be suppressed by using NMS. The output of the network is forwarded to the next convolutional neural network as an input.

In the second cascade network (refine network), false face regions will be removed by applying the threshold on face region probabilities  $r p_i$ . The threshold value is higher than the proposal network threshold value. Bounding box regression is used to calibrate the candidate facial windows and then NMS is employed to suppress the overlapped facial windows. The output of the refine networks is as:

$$\text{CNN2}_{\text{Output}} = \{\text{rbbox}_i, r p_i\} \quad (13)$$

where,  $\text{rbbox}_i$  is the facial region predicted by the second cascade network. The last cascaded network (output network) is similar to a refine network but concentrates on generating the true face region only by applying the threshold value on  $o p_i$ , which is higher than the refine network threshold value. The network will generate an accurate bounding box ( $\text{obbox}_i$ ) and landmarks' position of the face. The facial landmarks are used to face alignment, which could improve the face recognition accuracy.

$$\text{CNN3}_{\text{Output}} = \{\text{obbox}_i, o p_i, \text{landmarks}_i\} \quad (14)$$

where,  $\text{landmarks}_i$  is the  $i^{\text{th}}$  face 5 landmark points (left eye, right eye, nose tip, mouth left corner, mouth right corner). These points are obtained by applying the linear transformation on a fully connected layer of the output network.

#### 4.1.3 Loss function for training

For training the face detection model, various loss functions are used to perform the tasks of face detection, landmark localization, and bounding box regression. The cross-entropy loss function is used for training the face detection task:

$$L_i^{\text{det}} = - \left( g_i^{\text{det}} \log(p_i) + (1 - g_i^{\text{det}}) (1 - \log(p_i)) \right) \quad (15)$$

$$g_i^{\text{det}} \in \{0, 1\} \quad (16)$$

where,  $p_i$  is the probability of having the face is from Eq. (1). Eq. (16) indicates the label of ground-truth.

For the training the task of bounding box regression, the loss in predicting the bounding box is calculated from Eq. (17).

$$L_i^{bbox} = \|\hat{g}_i^{bbox} - g_i^{bbox}\|_2^2 \quad (17)$$

where,  $\hat{g}_i^{bbox}$  is the predicted regression vector by the network.

For the training the task of landmark localization, the Euclidean loss in Eq. (18) is used in the face detection model:

$$L_i^{landmarks} = \|\hat{g}_i^{landmarks} - g_i^{landmarks}\|_2^2 \quad (18)$$

where,  $\hat{g}_i^{landmarks}$  is the predicted landmarks by the network with five facial landmarks.  $g_i^{landmarks}$  is the ground truth coordinates consists of the left eye, right eye, nose, mouth right corner, and mouth left corner. The total loss for training all the tasks is computed using the equation given in Eq. (19).

$$\min \sum_{i=1}^N \sum_{j \in \{det, box, landmarks\}} \alpha_j \beta_i^j L_i^j \quad (19)$$

where, N is the samples that are considered for training.  $\alpha_j$  represents which task is important in the networks of the face detection model. i.e ( $\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmarks} = 0.5$ ) is used in Proposal Network and Refine Network. ( $\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmarks} = 1$ ) is used in Output Network to achieve a more precise localization of facial landmarks.  $\beta_i^j$  is a sample indicator consists of  $\{0,1\}$ .

#### 4.1.4 Training data

For the training the tasks of face detection, bounding box regression, and landmark localization, we have used randomly cropped part of the images from the dataset to get a positive set of the images which have the intersection-of-union (IoU) between ground truth image and the cropped image is greater than 0.6. The negative set of images that have the IoU less than 0.3 is considered and the rest of the images are ignored.

#### 4.1.5 Facial feature extraction

In our proposed framework, the outputs of the face detection model are used to train the deep network model given in Figure 3 to generate deep facial features. We have used MobileFaceNet [62] to extract the deep facial features, this deep model takes a detected face as the input from the face detection model. It is the pre-trained model on the MS-Celeb-1M dataset. Then, the preprocessing is performed on a detected face to generate an aligned face with the size of 112 x 112 and it is passed through the face generation model given in Figure 3. The global depthwise convolution layer of the model is used to reduce the dimension of the feature vector instead of global average pooling. A deep facial feature vector is generated at the last layer, i.e., a fully connected layer with a dimension of 128. This facial feature vector is different for each patient. This model is trained on our patient data by calling the *FACE\_BANK\_CREATION* procedure to generated deep facial features and to store the corresponding patient names in a file as facial feature vectors or face embeddings called face-bank. It is retained in the server. This face-bank is used as an inference for patient identification and verification.

#### ALGORITHM-1: FACE\_DETECTION\_MODEL()

```

Input:=Collect all positive, Negative and Part_face images
Convert those to 12x12 for PNet, 24x24 for RNet, 48x48 for
ONet
for i in Input do
  Stage 1: Input: Positive, Negative, Part_face images of size
  12x12
    Compute the eq(1) to Predict the probability of having
    face and
    bounding box
    Optimize the loss by using Eq. (15)
    with parameters (  $\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmarks} =$ 
0.5)
    save the weights and bias
  Stage 2: Input Positive, Negative, Part_face images of size
  24x24
    and landmark points
    Compute the Eq. (10) to predict the probability of having
    the face,
    bounding box and landmarks
    Optimize the loss by using Eq. (17)
    with parameters (  $\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmarks} =$ 
0.5)
    save the weights and bias
  Stage 3: Input Positive, Negative, Part_face images of size
  48x48
    and landmarks points
    Compute the Eq. (11) to predict final probability,
    bounding
    box and landmark localization
    Optimize the loss by using Eq. (18)
    with parameters (  $\alpha_{det} = 1, \alpha_{box} =$ 
0.5,  $\alpha_{landmarks} = 1$ )
    save the weights and biases
end for
Save the weights and bias
return weights and bias

```

#### ALGORITHM-2: FACE\_BANK\_CREATION()

```

Collect all patient images PList(input_images)
Initialize face_bank
weights, biases := FACE_DETECTION_MODEL()
for i in PList
  // Detecting the face in image
  Use weights an biases to compute bounding_box and
  landmarks
  of the facei in the imagei
  Use bounding_box and landmarks for face alignment.
  // Generating the facial feature vector of that patient
  Use pre_trained feature generation model to extract facial
  features aligned face as explained in section Facial
  Feature
  extraction
  Add the facial_features and name of the patient to
  face_bank
return face_bank

```

#### 4.2 Patient recognition

Patient identification and verification are performed in the server as given in Figure 1. The client node captures the

patient's image and sends it to the server. The various tasks to be done in the server are explained in the following sections.

#### 4.2.1 Preprocessing - enhancement algorithm

Capturing the patient's face in an uncontrolled environment can affect the detection of the patient in the frame due to light illuminations. To address this problem, different solutions have been provided in the paper [45] such as Histogram equalization, Homomorphic filters, Histogram specification, etc. But, these methods might cause annoying side effects such as artifacts like contour, etc, centered on the difference in the histogram of gray-level distribution. In our work, we used an adaptive histogram equalization based method to produce a good contrast-enhanced image that enables us to detect face from the poor quality image caused due to poor light conditions. In our work, we have investigated the Adaptive Histogram Equalization (AHE) method and Contrast Limited AHE (CLAHE) [63]. AHE amplifies the contrast by adapting the nearest region contrast of the image according to the histogram equalization. Often, the AHE can generate noise which may be amplified in comparison to the nearest regions. It will lead to over noise in the image. To overcome the noise amplification, CLAHE is used as the variant to the AHE to limit the contrast amplification. In CLAHE, the slope of the transformation function is used to amplify the contrast in the distinct of a given pixel value. It limits the amplification by considering the cumulative distribution function after calculating the clipping histogram at a threshold value. The clip limit is evaluated as the value that the histogram is clipped at. It is depending on the normalization of the histogram and the size of the nearest region. In real-time videos, most of the frames affected by light distortions can overcome by using the CLAHE technique. Hence, we have used the CLAHE technique in our proposed method.

#### 4.2.2 Face detection and face alignment

In this module, The enhanced patient image is passed through the face\_detection model as given Algorithm-2 to detect the face and localize the facial landmark of the patient in the image. This deep model is capable of doing multi-task using the cascaded network. Each layer in the cascaded network will generate the pyramid of images with a different scale to detect the patient face in different sizes of images. This is an advantage of a deep learning model to detect the patient at variable distances from the web camera. This face detection module is the same as the module in the face-bank creation.

#### 4.2.3 Facial features extraction and recognition

In this module, a deep learning model is used to extract the deep facial features from the output of the face detection model. That is the face\_feature generation model explained in section *Facial features extraction*. The extracted deep facial features are compared with the all deep facial features from the face-bank. Then Euclidean distance is calculated between the generated facial features and stored facial features to provide the distance score.

$$distances = \sqrt{(Test_{face\_feature} - Trained_{face\_features}^n)^2} \quad (20)$$

$$CScore_{min} = \min(distances) \quad (21)$$

$Test_{face\_feature}$  is the input image face feature which is generated by using a pre-trained feature generation model explained in Section *Facial feature extraction*,  $Trained_{face\_features}^n$  is the all the n number of trained images' face features stored in face\_bank. The minimum distance between the test image face feature and the all trained images' face features are calculated using Eq. (20). The index of the minimum distance score will be the patient's identity with the help of Eq. (21).

## 5. EXPERIMENTAL SETUP AND RESULT

For an experiment, we have used a patient dataset that contains the Indian faces of 1,297 subjects or patients collected from a school. For each subject, we have captured the image through the web camera and it has been augmented to generate five more images, which is bringing the count of images to 6,485 in the dataset. The resolution of each image is 440 pixels by 550 pixels. All images are in color and some are taken in a conditional environment and remaining in an unconditional environment. This dataset contains the frontal face along with variations in brightness and facial expressions. This dataset is used to create a face-bank.

We have also evaluated and tested the proposed deep learning-based framework in a client-server environment. For the implementation and evaluation of the proposed deep learning-based framework, two modules are included in the proposed work: client module and server module. The client module runs on the client system and the server module runs on a server system. The client environment uses windows 64-bit operating system with the REST API Django package of python, web camera, Intel corei7 processor 3.60 GHz (2) processor, and 4 GB RAM. A front-end application is developed with a Tkinter package in python to provide a Graphical User Interface (GUI) to the hospital staff. A sample of the user interfaces shown in Figure 4. The server is in another system with Intel Xeon 3.60 GHz (2) processor, 8 GB RAM, NVIDIA Tesla K20 graphical card, Windows 64-bit operating system, and python with REST API.



**Figure 4.** User Interface for Face Detection and Recognition system

The client and server communication is done by employing the REST API. It provides flexibility and secure communication between client and server. The REST API will securely send the patient image by applying the cryptographic algorithms on the image and convert it into JSON files. The server receives the JSON files, enhances the contrast of the image contained in the JSON file by using the contrast enhancement algorithm: CLAHE and the recognizes patient using the procedure explained in section *Facial feature*

extraction and recognition. We have generated a face-bank that contains deep facial features that are extracted from the images of the patient dataset with 6,485 images by using the Algorithms-2 This face-bank is retained in the server.

The client module captures the image of the patient through a web camera and sends the patient’s face image to the server as a request in the form of a JSON file after encoding the image by using the pybase64 package of python. The server module receives the patient’s image as JSON file from the client using REST API. Then, the contrast enhancement is employed on the given image. To prove the need for contrast enhancement, we have considered two feature-based face detection methods: Viola-Jones, Local Binary Pattern (LBP) [42], and our deep learning framework for an experiment to verify the need for the contrast enhancement on the image to increase the detection rate of the face. The statistics of the experimental

results are listed in Table 1. The experimental results reveal that good performance is demonstrated by the deep face detection method with contrast enhancement.

After the contrast enhancement, patient recognition is done by using the procedure explained in section *Facial feature extraction and recognition*. The detected face and recognized face in the server as shown in Figure 5 & Figure 6.

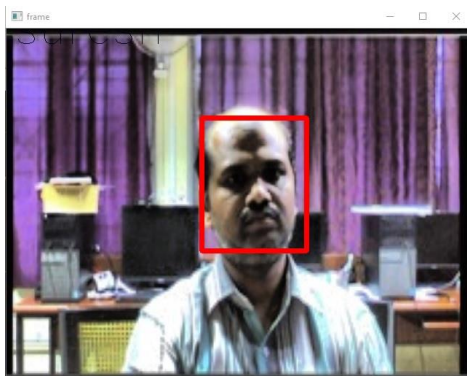
Then, the patient name is converted into a JSON description and send it to the client as an HTTP response. The client receives the response and retrieves the patient's name from the JSON description. It will be displayed on the graphical user interface as a front-end tool for the hospital staff as shown in Figure 7. Table 2 shows the recognition performance of the various methods. It is observed from Table 2 that the proposed deep learning framework achieves better performance compared to featured based methods.

**Table 1.** Experimental results for the need for contrast enhancement on the image

1297 images are considered for the experiment	Viola-Jones [41]		LBP [42]		Proposed	
	Contrast Enhancement		Contrast Enhancement		Contrast Enhancement	
	without	With	without	With	without	With
Detected Faces	1199	1087	1182	1203	898	1224
Multi-bounding boxes	90	187	12	24	-	-
Mis-detections	1	5	8	7	1	3
Not detected faces	7	12	94	62	398	70

**Table 2.** Patient recognition performance

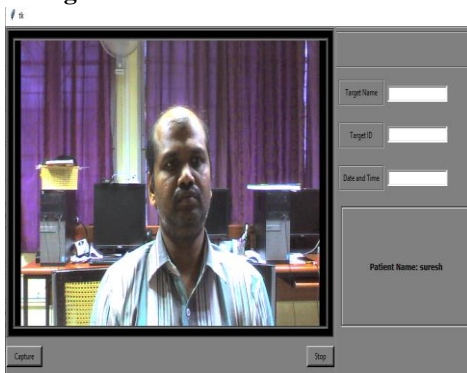
1297 images are considered for the experiment	Viola-Jones [41]		LBP [42]		Proposed	
	Contrast Enhancement		Contrast Enhancement		Contrast Enhancement	
	without	With	without	With	without	With
Faces Recognized	1199	1087	1182	1203	898	1224
Faces Not Recognized	98	210	115	94	399	73
Recognition %	92.4	83.9	91.1	92.7	69.2	94.3



**Figure 5.** Detected face in the Server



**Figure 6.** Sample of patient recognition in the server



**Figure 7.** Patient Name provided on front-end tool(GUI) at the client



**Figure 8.** Identifying closest faces when Unknown face recognized



One of our contributions is generating the 6 closest scores to the score of the test image for retrieving the closest matched 6 patients' names as shown in Figure 8. If the patient's face is not detected due to light illumination, occlusion, pose variations, that would be an advantage. In conclusion, the proposed framework can be used for patient/person identity which can be used to store and retrieve patients' medical records which in turn use to improve health care quality. The proposed framework can also be used in other applications.

## 6. CONCLUSION

We have studied and investigated a deep learning framework to automate patient identity verification, which enables improvement in health care quality through storing and retrieving health records automatically for better medical treatment. The deep learning-based framework could contribute greatly to address the issues of identifying and verifying the patient in low light and shadow problems. This framework utilizes efficient and effective deep facial features to determine patient identity and verification with the help of a face-bank. Automatically retrieving the electronic medical health record are useful to doctors to make decisions for medical treatments. If the patient is unresponsive, the framework has the ability to identify the patient that quickly enables healthcare facilities by considering the medical history of the patient.

## ACKNOWLEDGMENT

I thank my institute for providing GPU resources.

## REFERENCES

- [1] Latham, T., Malomboza, O., Nyirenda, L., Ashford, P., Emmanuel, J., M'baya, B., Bates, I. (2012). Quality in practice: Implementation of hospital guidelines for patient identification in Malawi. *International Journal for Quality in Health Care*, pp. 626-633. <https://doi.org/10.1093/intqhc/mzs038>
- [2] Tase, T.H., Lourenção, D.C.A., Bianchini, S.M., Tronchin, D.M.R. (2013). Patient identification in healthcare organizations: An emerging debate. *Article in Portuguese*, 34(3): 196-200. <https://doi.org/10.1590/s1983-14472013000300025>
- [3] Ajami, S., Rajabzadeh, A. (2013). Radio Frequency Identification (RFID) technology and patient safety. *Journal of Research In Medical Sciences: The Official Journal of Isfahan University of Medical Sciences*, 18(9): 809-813.
- [4] Bolton-Maggs, P., Poles, D. (2017). Serious Hazards of Transfusion (SHOT) Steering Group. The 2015 Annual SHOT Report.
- [5] Ferguson, C., Hickman, L., Macbean, C., Jackson, D. (2019). The wicked problem of patient misidentification: How could the technological revolution help address patient safety? *Journal of Clinical Nursing*, 28(13-14): 2365-2368. <https://doi.org/10.1111/jocn.14848>
- [6] Thomas, P., Evans, C. (2004). An identity crisis? Aspects of patient misidentification. *Clinical Risk*, 10: 18-22. <https://doi.org/10.1258/135626204322756556>
- [7] Kanade, T. (1973). Picture processing by computer complex and recognition of human faces. Ph. D. Thesis, Kyoto University. [https://doi.org/10.1016/0146-664X\(73\)90002-6](https://doi.org/10.1016/0146-664X(73)90002-6)
- [8] Chen, D., Ren, S., Wei, Y., Cao, X., Sun, J. (2014). Joint cascade face detection and alignment. *European Conference on Computer Vision*, Springer, Cham, pp. 109-122. [https://doi.org/10.1007/978-3-319-10599-4\\_8](https://doi.org/10.1007/978-3-319-10599-4_8)
- [9] Leung, T.K., Burl, M.C., Perona, P. (1995). Finding faces in cluttered scenes using random labeled graph matching. In *Proceedings of Fifth IEEE International Conference Computer Vision*, pp. 637-644. <https://doi.org/10.1109/ICCV.1995.466878>
- [10] Yow, K.C., Cipolla, R. (1997). Feature-based human face detection. *Journal of Image and Vision Computing*, 15(9): 713-735. [https://doi.org/10.1016/S0262-8856\(97\)00003-6](https://doi.org/10.1016/S0262-8856(97)00003-6)
- [11] Dai, Y., Nakano, Y. (1996). Face-texture model based on SGLD and its application in face detection in a color scene. *Journal of Pattern Recognition*, 29(6): 1007-1017. [https://doi.org/10.1016/0031-3203\(95\)00139-5](https://doi.org/10.1016/0031-3203(95)00139-5)
- [12] Yang, J., Waibel, A. (1996). A real-time face tracker. In *Proceedings Third IEEE Workshop on Applications of Computer Vision*, WACV'96, Sarasota, FL, USA, pp. 142-147. <https://doi.org/10.1109/ACV.1996.572043>
- [13] Kjeldsen, R., Kender, J. (1996). Finding skin in color images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 312-317. <https://doi.org/10.1109/AFGR.1996.557283>
- [14] Turk, M., Pentland, A. (1998). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1): 71-86. <https://doi.org/10.1162/jocn.1991.3.1.71>
- [15] Sung, K.K., Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1): 39-51. <https://doi.org/10.1109/34.655648>
- [16] Rowley, H.A., Baluja, S., Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1): 23-38. <https://doi.org/10.1109/34.655647>
- [17] Osuna, E., Freund, R., Girosit, F. (1997). Training support vector machines: An application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 130-136. <https://doi.org/10.1109/CVPR.1997.609310>
- [18] Schneiderman, H., Kanade, T. (1998). Probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proceedings 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pp. 45-51. <https://doi.org/10.1109/CVPR.1998.698586>
- [19] Rajagopalan, A.N., Kumar, K.S., Karlekar, J., Manivasakan, R., Patil, M.M., Desai, U.B., Poonacha, P.G., Chaudhuri, S. (1998). Finding faces in photographs. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pp. 640-645. <https://doi.org/10.5555/938978.939148>
- [20] Yang, G., Huang, T.S. (1994). Human face detection in a complex background. *Pattern recognition*, 27(1): 53-63. [https://doi.org/10.1016/0031-3203\(94\)90017-5](https://doi.org/10.1016/0031-3203(94)90017-5)
- [21] Craw, I., Tock, D., Bennett, A. (1992). Finding face features. In *European Conference on Computer Vision*,

- Springer, Berlin, Heidelberg, pp. 92-96. [https://doi.org/10.1007/3-540-55426-2\\_12](https://doi.org/10.1007/3-540-55426-2_12)
- [22] Lanitis, A., Taylor, C.J., Cootes, T.F. (1995). Automatic face identification system using flexible appearance models. *Journal of Image and Vision Computing*, 13(5): 393-401. [https://doi.org/10.1016/0262-8856\(95\)99726-H](https://doi.org/10.1016/0262-8856(95)99726-H)
- [23] Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4): 399-458. <https://doi.org/10.1145/954339.954342>
- [24] Singh, S., Prasad, S.V.A.V. (2018). Techniques and challenges of face recognition: A critical review. *Procedia Computer Science*, 143: 536-543. <https://doi.org/10.1016/j.procs.2018.10.427>
- [25] Nwosu, K.C., Igbonagwam, O.A. (2016). Facial recognition and mobile based system for patient identification/verification in medical emergencies for developing economies. *Journal for the Advancement of Developing Economies*, 5(1). <https://doi.org/10.13014/K2DF6PD6>
- [26] Shagholi, A., Charimi, M., Rakhshan, H. (2015). The effect of the distance from the webcam in heart rate estimation from face video images. In 2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA), pp. 1-6. <https://doi.org/10.1109/IPRIA.2015.7161622>
- [27] Pabiania, M.D., Santos, K.A.P., Villa-Real, M.M., Villareal, J.A.N. (2016). Face recognition system for electronic medical record to access out-patient information. *Jurnal Teknologi*, 78(6-3). <https://doi.org/10.11113/jt.v78.8935>
- [28] Naruniec, J., Wiczorek, M., Szlufik, S., Koziorowski, D., Tomaszewski, M., Kowalski, M., Przybyszewski, A. (2016). Webcam-based system for video-oculography. *IET Computer Vision*, 11(2): 173-180. <https://doi.org/10.1049/iet-cvi.2016.0226>
- [29] Mehra, S., Khatri, A., Tanwar, P., Khatri, V. (2018). Intelligent embedded security control system for maternity ward based on IoT and face recognition. In 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 49-53. <https://doi.org/10.1109/ICACCCN.2018.8748516>
- [30] Brunelli, R., Poggio, T. (1993). Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10): 1042-1052. <https://doi.org/10.1109/34.254061>
- [31] SatonkarSuhass, S., Kurhe Ajay, B., Prakash Khanale, B. (2012). Face recognition using principal component analysis and linear discriminant analysis on holistic approach in facial images database. *Int Organ Sci Res*, 2(12): 15-23. <https://doi.org/10.9790/3021-021241523>
- [32] Moghaddam, B., Pentland, A. (1997). Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7): 696-710. <https://doi.org/10.1109/34.598227>
- [33] Ahmed, N., Natarajan, T., Rao, K.R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, 100(1): 90-93. <https://doi.org/10.1109/T-C.1974.223784>
- [34] Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 7: 711-720. <https://doi.org/10.1109/34.598228>
- [35] Yu, W.W., Teng, X.L., Liu, C.Q. (2006). Face recognition using discriminant locality preserving projections. *ELSEVIER Image and Vision Computing*, pp. 239-248. <https://doi.org/10.1016/j.imavis.2005.11.006>
- [36] Daugman, J.G. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Journal of Vision Research*, 20(10): 847-856. [https://doi.org/10.1016/0042-6989\(80\)90065-6](https://doi.org/10.1016/0042-6989(80)90065-6)
- [37] Bartlett, M.S., Movellan, J.R., Sejnowski, T.J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6): 1450-1464. <https://doi.org/10.1109/TNN.2002.804287>
- [38] Timotius, I.K., Setyawan, I., Febrianto, A.A. (2010). Face recognition between two person using kernel principal component analysis and support vector machines. *International Journal on Electrical Engineering and Informatics*, 2(1): 55-63. <https://doi.org/10.15676/ijeei.2010.2.1.5>
- [39] Jamil, N., Lqbal, S., Iqbal, N. (2001). Face recognition using neural networks. *International Multi Topic Conference (IEEE INMIC 2001)*, 3(3): 277-281. <https://doi.org/10.1109/INMIC.2001.995351>
- [40] Kshirsagar, V.P., Baviskar, M.R., Gaikwad, M.E. (2011). Face recognition using Eigenfaces. In 2011 3rd International Conference on Computer Research and Development, 2: 302-306. <https://doi.org/10.1109/ICCRD.2011.5764137>
- [41] Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA*, pp. I-1. <https://doi.org/10.1109/CVPR.2001.990517>
- [42] Tang, H.L., Yin, B.C., Sun, Y.F., Hu, Y.L. (2013). 3D face recognition using local binary patterns. *Journal of Signal Processing*, 93(8): 2190-2198. <https://doi.org/10.1016/j.sigpro.2012.04.002>
- [43] Hu, J., Lu, J., Tan, Y.P. (2014). Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1875-1882. <https://doi.org/10.1109/CVPR.2014.242>
- [44] Taigman, Y., Yang, M., Ranzato, M., Wolf, L. (2014). Closing the gap to human-level performance in face verification. *deepface*. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1701-1708. <https://doi.org/10.1109/CVPR.2014.220>
- [45] Sun, Y., Wang, X.G., Tang, X.O. (2015). Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2892-2900. <https://doi.org/10.1109/CVPR.2015.7298907>
- [46] Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815-823. <https://doi.org/10.1109/CVPR.2015.7298682>
- [47] Parkhi, O.M., Vedaldi, A., Zisserman, A. (2015). Deep face recognition. *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 41.1-41.12. <https://doi.org/10.5244/C.29.41>

- [48] Parchami, M., Bashbaghi, S., Granger, E. (2017). Video-based face recognition using ensemble of haar-like deep convolutional neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 4625-4632. <https://doi.org/10.1109/IJCNN.2017.7966443>
- [49] Hu, Y.B., Wu, X., He, R., (2017). Attention-set based metric learning for video face recognition. In 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), pp. 97-102. <https://doi.org/10.1109/ACPR.2017.43>
- [50] Rao, Y.M., Lu, J.W., Zhou, J. (2017). Attention-aware deep reinforcement learning for video face recognition. In Proceedings of the IEEE International Conference on Computer Vision, pp. 3951-3960. <https://doi.org/10.1109/ICCV.2017.424>
- [51] Goswami, G., Vatsa, M., Singh, R. (2017). Face verification via learned representation on feature-rich video frames. *IEEE Transactions on Information Forensics and Security*, 12(7): 1686-1698. <https://doi.org/10.1109/TIFS.2017.2668221>
- [52] Yeung, H.W.F., Li, J., Chung, Y.Y. (2017). Improved performance of face recognition using CNN with constrained triplet loss layer. In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1948-1955. <https://doi.org/10.1109/IJCNN.2017.7966089>
- [53] Aseem, S., Ratrou, B.M., Litin, S.C., Ganesh, R., Croghan, I.T., Salerno, M.S., Majka, A.J., Chutka, D.S., Hurt, R.T., Lebdeh, H.S.A., Vincent, A. (2020). A process of acceptance of patient photographs in electronic medical records to confirm patient identification. *Mayo Clinic Proceedings: Innovations, Quality & Outcomes*, 4(1): 99-104. <https://doi.org/10.1016/j.mayocpiqo.2019.10.002>
- [54] Johnson Jr, J.E., Blanes, R., Sheng, D., Narayanan, A. (2019). Face Recognition for Fast Information Retrieval and Record Lookup, Defensive publication series.
- [55] Jeon, B., Jeong, B., Jee, S., Huang, Y., Kim, Y., Park, G.H., Kim, J., Wufuer, M., Jin, X., Kim, S.W., Choi, T.H. (2019). A facial recognition mobile App for patient safety and biometric identification: Design, development, and validation. *JMIR Mhealth and Uhealth*, 7(4): 11472. <https://doi.org/10.2196/11472>
- [56] <https://www.oezsoft.com/oezfr> - Ji Seung-hoon CEO of OezSoft company, Software tool for face recognition.
- [57] Sharma, S., Shanmugasundaram, K., Ramasamy, S.K. (2016). FAREC-CNN based efficient face recognition technique using Dlib. In 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 192-195. <https://doi.org/10.1109/ICACCCT.2016.7831628>
- [58] Boyko, N., Basystiuk, O., Shakhovska, N. (2018). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), pp. 478-482. <https://doi.org/10.1109/DSMP.2018.8478556>
- [59] Chaitanya, K., Karani, N., Baumgartner, C.F., Becker, A., Donati, O., Konukoglu, E. (2019). Semi-supervised and task-driven data augmentation. In International Conference on Information Processing in Medical Imaging, Cham, pp. 29-41. [https://doi.org/10.1007/978-3-030-20351-1\\_3](https://doi.org/10.1007/978-3-030-20351-1_3)
- [60] Zhang, K., Zhang, Z., Li, Z., Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10): 1499-1503. <https://doi.org/10.1109/LSP.2016.2603342>
- [61] Farfade, S.S., Saberian, M.J., Li, L.J. (2015). Multi-view face detection using deep convolutional neural networks. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 643-650. <https://doi.org/10.1145/2671188.2749408>
- [62] Chen, S., Liu, Y., Gao, X., Han, Z. (2018). MobileFaceNets: Efficient cnns for accurate real-time face verification on mobile devices. *Chinese Conference on Biometric Recognition*, Cham, pp. 428-438. [https://doi.org/10.1007/978-3-319-97909-0\\_46](https://doi.org/10.1007/978-3-319-97909-0_46)
- [63] Sasi, N.M., Jayasree, V.K. (2013). Contrast limited adaptive histogram equalization for qualitative enhancement of myocardial perfusion images. *Engineering*, 5(10): 326-331. <https://doi.org/10.4236/eng.2013.510B066>