



FACE AND IRIS RECOGNITION IN A VIDEO SEQUENCE USING DBPNN AND ADAPTIVE HAMMING DISTANCE

¹S. Revathy, ²Mr. L. Ramasethu

¹PG Scholar, Hindusthan College of Engineering and Technology, Coimbatore, India.

Email id: revathysreeni14@gmail.com

²Assistant Professor, Hindusthan College of Engineering and Technology, Coimbatore, India.

Abstract: - Dense feature extraction is becoming increasingly popular in face recognition. Face recognition is a vital component for authorization and security. In earlier days, CCA (Canonical Correlation Analysis) and SIFT (Scale Invariant Feature Transforms) was used for face recognition. Since multi scale extraction is not possible with these existing methods, a new approach to dense feature extraction is developed in this project. The proposed method combines dense feature extraction and decision based propagation neural network (DBPNN). Neural network algorithm is presented to recognize the face at different angle, and it is used for training and learning and leading to efficient and robust face recognition. Finally Iris matching is done by using Iterative randomized Hough transform for detecting the pupil region with number of iteration counts. Experimental results show that the proposed method is providing effective recognition rate with accuracy in comparing with existing methods.

Keywords: - face detection, Decision based neural network, Feature extraction, Classification

I. INTRODUCTION

The human face plays an important role in our social interaction for conveying people's identity. Face recognition is an ability to recognize people by their unique facial characteristics. The traditional authentication methods of person's identity include passwords, PINs, smart cards, plastic cards, token, keys and so forth. These could be hard to remember or retain, and passwords can be stolen or guessed, tokens and keys can be misplaced and forgotten. However an individual's biological traits cannot be misplaced, forgotten, stolen or forged.

Biometric-based technologies include identification based on physiological characteristics such as face, fingerprints, finger geometry, hand geometry, hand veins, palm, iris, retina, ear and voice and behavioural traits such as gait, signature and keystroke dynamics. Face recognition can be done passively without any explicit action or participation on the part of the user since face images can be acquired from a distance by a camera. Iris and retina identification require expensive equipment and are much too sensitive to any body motion. Voice recognition is susceptible to background noises in public places and auditory fluctuations on a phone line or tape recording. Signatures can be modified or forged. However, facial images can be easily obtained with a couple of inexpensive fixed cameras. Face recognition is totally non-intrusive and does not carry any such health risks.

II. RELATED WORKS

Indexing consists of storing SIFT keys and identifying matching keys from the new image. Lowe used a modification of the k-d tree algorithm in this method and is called the Best-bin first search method that can identify the nearest neighbors with high probability using only a limited amount of computation [12]. The BBF algorithm uses a modified search ordering for the k-d tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This search order requires the use of a heap-based priority queue for efficient determination of the search order. The best candidate match for each key point is found by identifying its nearest neighbour in the database of key points from training images.

When an image contains complex background, the SIFT descriptors tend to spread over the entire image rather than being concentrated in the object region. As a result, the actual object can be neglected in the matching process. Since the number of extracted SIFT descriptors is typically large, the computational cost to match extracted key-points is very high.

Adaptive matching algorithm is used to match the person according to the trained values for the time of testing the test image features are taken. The test image feature is compare with the trained image dataset if the test image feature and trained image feature is similar the person is matched person else the person is not matched. Thus the trained values from the database are used for authenticating the person.

When the probe image is given, the algorithm may compare the features of the input image with the trained database and determines the results as authenticated [18]. The algorithm may conclude its results only after all the features in the image are matched and hence this process is much superior to others. Different poses are not applicable in this method.

The template matching is simple technique for image processing topics like feature extraction, edge detection, object extraction. Template matching can be subdivided between two approaches: feature-based and template-based matching. The feature-based approach uses the features of the search and to recognize a human face, some special features need to be extracted. These special features include eyes, nose, mouth, and chin along with the shape of the face.

To locate these features, researchers have proposed a variety of methods based on symmetry of faces [1], facial geometry and luminance [2], and template matching [3]. Generally, the vision based face recognition can be explained as follows. Initially, the subject image is enhanced and segmented. Then the contour features of the face are extracted by contour extraction method, and compares with the extracted features of the database image [4]. If there is a match then the person in the subject image is recognized.

III PROPOSED METHOD

1. FACE RECOGNITION FROM VIDEO

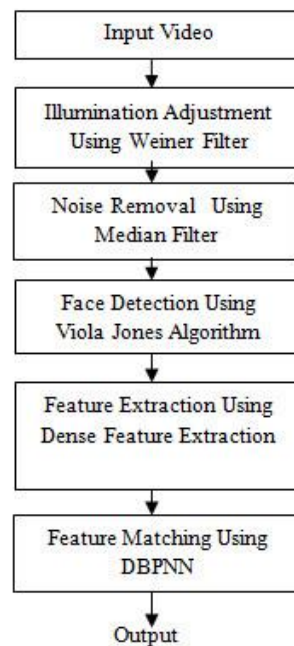


Figure 1. Block diagram for face recognition

1.1 ILLUMINATION ADJUSTMENT

An important goal of this adjustment is to render specific colours particularly neutral colours correctly; hence, the general method is sometimes called gray balance, neutral balance, or white balance.

WIENER FILTER

The most important technique for removal of blur in images due to linear motion is the Wiener filter. The first process that was performed was creating a point spread function to add blur to a video. The blur was implemented by first creating a PSF filter in Mat Lab that would approximate linear motion blur. This PSF was then convolved with the original image to produce the blurred video. Convolution is a mathematical process by which a signal, in this case the image is acted on by a system the filter in order to find the resulting signal. The amount of blur added to the original video depends on two parameters of the PSF are the length of blur in pixels, and the angle of the blur. After a known amount of blur was introduced into the video, an attempt was made to restore the new blurred video to its original form. The Wiener filter is an inverse filter that employs a linear deconvolution method. Linear deconvolution means that the output is a linear combination of two common types of impulse noise are the salt-and-pepper noise and the random-valued noise. For images corrupted by salt-and-pepper noise respectively random-valued noise, the noisy pixels can take only the maximum and the minimum values respectively any random value in the dynamic range. There are many works on the restoration of images corrupted by impulse noise. See, for instance, the nonlinear digital filters.

1.2 NOISE REMOVAL USING MEDIAN FILTER

The median filter was once the most popular nonlinear filter for removing impulse noise, because of its good denoising power and computational efficiency. However, when the noise level is over 50%, some details and edges of the original image are smeared by the filter. Different remedies of the median filter have been proposed, like the adaptive median filter the multi-state median filter or the median filter based on homogeneity information. These so-called decision-based or switching filters first identify possible noisy pixels and then replace them by using the median filter or its variants, while leaving all other pixels unchanged. These filters are good at detecting noise even at a high noise level. Their main drawback is that the noisy pixels are replaced by some median value in their vicinity without taking into account local features such as the possible presence of edges. Hence details and edges are not recovered satisfactorily, especially when the noise level is high.

For images corrupted by Gaussian noise, least-squares methods based on edge-preserving regularization functional have been used successfully to preserve the edges and the details in the images. These methods fail in the presence of impulse noise because the noise is heavy tailed. Moreover the restoration will alter basically all pixels in the image, including those that are not corrupted by the impulse noise. Recently, non-smooth data-fidelity terms have been used along with edge preserving regularization to deal with impulse noise.

We propose a powerful two-stage scheme which combines the variation method proposed in with the adaptive median filter. More precisely, the noise candidates are first identified by the adaptive median filter, and then these noise candidates are selectively restored using an objective function with a data fidelity term and an edge-preserving regularization term. Since the edges are preserved for the noise candidates, and no changes are made to the other pixels, the performance of our combined approach is much better than that of either one of the methods. Salt and pepper noise with noise ratio as high as 90% can be cleaned quite efficiently.

1.3FACE DETECTION

Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Any facial feature changes in the database will invalidate the matching process.

VIOLA-JONES ALGORITHM

The Viola-Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time primarily by the problem of face detection. The human can do this easily, but a computer needs precise instructions and constraints. To make the task more manageable, Viola-Jones requires full view frontal upright faces. The algorithm has four stages, Haar Feature Selection, Creating an Integral Image, Adaboost Training, and Cascading Classifiers. The features of the detection framework involve the sums of image pixels within rectangular areas and it bears some resemblance to Haar basis functions, which have been used previously in the realm of image-based object detection. However, since the features used by Viola and Jones all rely on more than one rectangular area, they are generally more complex. The value of any given feature is always simply the sum of the pixels within clear rectangles subtracted from the sum of the pixels within shaded rectangles. The rectangular features of this sort are rather primitive when compared to alternatives such as steerable filters. Although they are sensitive to vertical and horizontal features, their feedback is considerably coarser.

The Haar features are the features which are used to match the similar properties of the human faces. Some of the similar properties related to human face are generally considered as, the eye region is darker than the upper-cheeks and the nose bridge region is brighter than the eyes. Composition of properties forming matchable facial features is, Location and size are eyes, mouth, and bridge of nose Value is oriented gradients of pixel intensities.

Rectangle features considered is given by the equation Value = Σ (pixels in black area) - Σ (pixels in white area). Three types: two-, three-, four-rectangles, Viola & Jones used two-rectangle features. For example: the difference in brightness between the white & black rectangles over a specific area where each feature is related to a special location in the sub-window

An image representation called the integral image evaluates rectangular features in constant time, which gives them a considerable speed advantage over more sophisticated alternative features. Because each feature's rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four-rectangle feature in ten.

LEARNING ALGORITHM

The speed with which features may be evaluated does not adequately compensate for their number, however. For example, in a standard 24x24 pixel sub-window, there is a total of $M=162,336$ possible features and it would be prohibitively expensive to evaluate them all when testing an image. Thus, the object detection framework employs a variant of the learning algorithm Adaboost to both select the best features and to train classifiers that use them. This algorithm constructs a "strong" classifier as a linear combination of weighted simple "weak" classifiers.

$$h(x) = \sin\left(\sum_{j=1}^M \alpha_j h_j(x)\right) \quad (1)$$

Each weak classifier is a threshold function based on the feature f_j

$$h_j(x) = \begin{cases} -s_j & \text{if } f_j < \theta_j \\ s_j & \text{otherwise} \end{cases} \quad (2)$$

The threshold value θ_j and the polarity $s_j \in \pm 1$ are determined in the training, as well as the coefficients α_j .

Here a simplified version of the learning algorithm is reported as,

Input: Set of N positive and negative training images with their labels

(x^i, y^i) . If image i is a face $y^i = 1$, if not $y^i = -1$.

Initialization: assign a weight $w1^i = \frac{1}{N}$ to each image i .

For each feature F_j with $J = 1, \dots, M$

Renormalize the weights such that they sum to one.

Apply the feature to each image in the training set, and then find the optimal threshold and polarity θ_j, s_j that minimizes the weighted classification error. That is

$$\theta_j, s_j = \arg \min \sum_{i=1}^N w_j^i c_j^i \quad (3)$$

Where,

$$c_j^i = \begin{cases} 0 & \text{if } y^i = h_j(x^i, \theta_j, s_j) \\ 1 & \text{otherwise} \end{cases}$$

Assign a weight α_j to h_j that is inversely proportional to the error rate. In this way best classifiers are considered more.

The weights for the next iteration, i.e. w_{j+1}^i , are reduced for the images i that were correctly classified.

$$h(x) = \sin\left(\sum_{j=1}^M \alpha_j h_j(x)\right) \quad (4)$$

1.4 LOCAL FEATURE EXTRACTION

A feature is defined as an interesting part of an image, and features are used as a starting point for many computer vision algorithms. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is repeatability whether or not the same feature will be detected in two or more different images of the same scene.

DENSE FEATURE EXTRACTION

In computer vision and image processing the concept of dense feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain often in the form of isolated points, continuous curves or connected regions a feature is defined as an interesting part of an image and features are used as a starting point for many computer vision algorithms. Since features are used as the starting point and main primitives for subsequent algorithms, the overall algorithm will often only be as good as its feature detector. Consequently, the desirable property for a feature detector is repeatability whether or not the same feature will be detected in two or more different images of the same scene.

Scale-space theory is a framework for multi scale signal representation developed by the Computer vision, image processing and signal processing communities with complementary motivations from physics and biological vision. It is a formal theory for handling image structures at different scales, by representing an image as a one-parameter family of smoothed images, the scale-space representation, parameterized by the size of the smoothing kernel used for suppressing fine-scale structures.

The parameter t in this family is referred to as the scale parameter, with the interpretation that image structures of spatial size smaller than about \sqrt{t} have largely been smoothed away in the scale-space level at scale t .

1.5 DECISION BASED PROBABLISTIC NEURAL NETWORK

A Probabilistic Neural Network is defined as an implementation of statistical algorithm called Kernel discriminate analysis in which the operations are organized into multilayered feed forward network with four layers: input layer, pattern layer, summation layer and output layer. A PNN is predominantly a classifier since it can map any input pattern to a number of classifications. Among the main advantages that discriminate PNN is

Fast training process, an inherently parallel structure, guaranteed to converge to an optimal classifier as the size of the representative training set increases and training samples can be added or removed without extensive retraining. Accordingly, a PNN learns more quickly than many neural networks model and have had success on a variety of applications. Based on these facts and advantages, PNN can be viewed as a supervised neural network that is capable of using it in system classification and pattern recognition. The main objective of this paper is to describe the use of PNN in solving protein classification problem.

Pattern layer/unit: there is one pattern node for each training example. Each pattern node/unit forms a product of the input pattern vector x (for classification) with a weight vector W_i , $Z_i = x \cdot W_i$, and when perform a non linear operation on Z_i before outputting its activation level to the summation node/unit. Here instead of the sigmoid activation function of the nonlinear operation is used,

$$\exp[(Z_i - 1)/\sigma^2] \quad (5)$$

Both x and W_i are normalized to unit length, this is equivalent to using the

$$\exp[-(W_i - x)^T(W_i - x)/2\sigma^2]. \quad (6)$$

Summation layer/unit: each summation node/unit receives the outputs from pattern nodes associated with a given class. It simply sums the inputs from the pattern units that correspond to the category, from which the training pattern was selected,

$$\sum_i \exp\left[-\frac{(W_i - x)^T(W_i - x)}{2\sigma^2}\right]. \quad (7)$$

Output layer/unit: the output nodes/units are two-input neurons. These units produce binary outputs, related to two different categories $\Omega_r, \Omega_s, r \neq s, r, s = 1, 2, \dots, q$, by using the classification criterion,

$$\sum_i \exp\left[-\frac{(W_i - x)^T(W_i - x)}{2\sigma^2}\right] > \sum_j \exp\left[-\frac{(W_j - x)^T(W_j - x)}{2\sigma^2}\right] \quad (8)$$

These units have only a single corresponding weight C , given by the loss parameters the prior probabilities and the number of training pattern in each category. Concretely, the corresponding weight is the ratio on a priori probabilities, divided by the ratio of samples and multiplied by the ratio of losses. This ratio can be determined only from the significance of the decision. There were developed non parametric techniques for estimating univariate probability density function from random samples. By using the multivariate Gaussian approximation, that is a sum of multivariate Gaussian distributions centered at each training sample, we obtain the following form of the pdf and i is the vector pattern number, m is the total number of training patterns, x_i is the i^{th} training pattern from category class, Ω_r, p is the input space dimension and σ is an adjustable smoothing parameter using the training procedure. The network is trained by setting the W_i weight vector in one of the pattern units equal to each x pattern in the training set and then connecting the pattern unit's output to the appropriate summation. In decision based PNN, if else condition is applied for the classification process where the classification can be done by the else condition format and the authentication is also based on the if else condition with more accuracy.

ALGORITHM FOR DBPNN:

owned by one of the training sample and the weight vector

initialize the objective function randomly generated contemporary parameter vector $X_i (i = 1, 2, \dots, N)$ and x_i

normalized samples training DBPNN, for videos, the objective function value for comparison with the objective function value of the current local optimal pests profile pests is, the objective function value is larger

each particle, its objective function value of the objective function value of the current global optimum best compare update best objective function value larger

According to the formula updates the values of X_i and x_i .

Termination of discrimination: If the number of iterations is the maximum number of times or the objective function reaches optimal if the loop is terminated, otherwise it returns Step3.

2. IRIS RECOGNITION FROM VIDEO

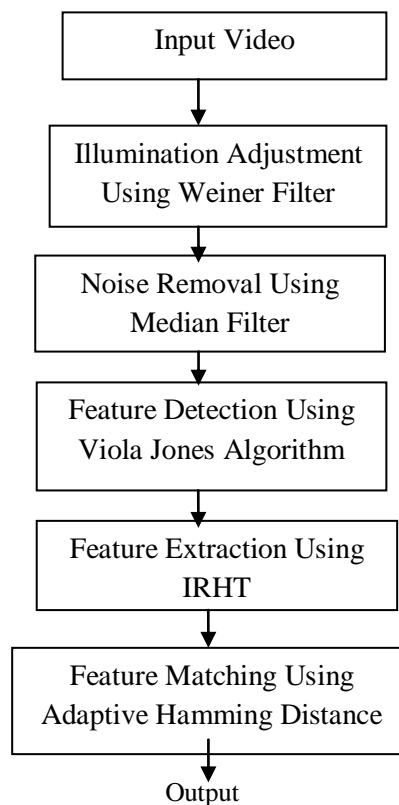


Figure 2. Block diagram for iris recognition

2.1 FEATURE EXTRACTION USING IRHT

An iterative randomized Hough transform (IRHT) is developed for detection of incomplete ellipses in images with strong noise. The IRHT iteratively applies the randomized Hough transform (RHT) to a region of interest in the image space. The region of interest is determined from the latest estimation of ellipse parameters. The IRHT zooms in on the target curve by iterative parameter adjustments and reciprocating use of the image and parameter spaces. During the iteration process, noise pixels are gradually excluded from the region of interest, and the estimation becomes progressively close to the target. The IRHT retains the advantages of RHT of high parameter resolution, computational simplicity and small storage while overcoming the noise susceptibility of RHT. Indivisible, multiple instances of ellipse can be sequentially detected. The IRHT was first tested for ellipse detection with synthesized images. It was then applied to fetal head detection in medical ultrasound images. The results demonstrate that the IRHT is a robust and efficient ellipse detection method for real-world applications.

ALGORITHM STEPS:

Iteration count

Mapping of edge points to the Hough space and storage in an accumulator.

Interpretation of the accumulator to yield lines of infinite length. The interpretation is done by thresholding and possibly other constraints.

Conversion of infinite lines to finite lines.

2.2 FEATURE MATCHING USING ADAPTIVE HAMMING DISTANCE

The weighted Hamming distance has been used for image retrieve, including Hamming distance weighting and the Anno Search. In each bit of the binary code is assigned with a bit-level weight, while the aim is to weight the overall Hamming distance of local features for image matching.

Only a single set of weights is used to measure either the importance of each bit in Hamming space, or to rescale the Hamming distance for better image matching. Jiang et al. propose a query-adaptive Hamming distance for image retrieve which assigns dynamic weights to hash bits, such that each bit is treated differently and dynamically. They harness a set of semantic concept classes that cover most semantic elements of image content. Then, different weights for each of the classes are learned with a supervised learning algorithm.

To compute the bit-level weights for a given query, a k nearest neighbour search is performed based on the original Hamming distance first, and then a linear combination of the weights of classes contained in the result list is used as the query-adaptive weights. In this section, we present the weighted Hamming distance ranking algorithm.

In most binary hashing algorithms, the distance between two points is simply measured by the Hamming distance between their binary codes. This distance metric is somewhat ambiguous, since for a K -bits binary code $H(p)$, there are Km different binary codes sharing the same distance m to $H(p)$. In most binary hashing algorithms, each hash bit takes the same weight and makes the same contribution to distance calculation. On the contrary, in our algorithm, we give different bits different weights.

With the bit-level weights, the returned binary codes can be ranked by the weighted Hamming distance at In this section, we present the weighted Hamming distance ranking algorithm. In most binary hashing algorithms, the distance between two points is simply measured by the Hamming distance between their binary codes. This distance metric is somewhat ambiguous, since for a K -bits binary code $H(p)$, Km there are K different binary codes sharing the same distance m to $H(p)$. In most binary hashing algorithms, each hash bit takes the same weight and makes the same contribution to distance calculation.

On the contrary, in our algorithm, we give different bits for different weights. With the bit-level weights, the returned binary codes can be ranked by the weighted Hamming distance at 1585 1587 a finer-grained binary code level rather than at the original integer Hamming distance level. The bit-level weight associated with hash bit K is denoted as ω_K . In the following, we will show that an effective bit-level weight is not only data dependent, but also query-dependent.

Note that, our algorithm is not to propose a new binary hashing method, but to give a ranking algorithm to improve the search accuracy of most existing binary hashing methods.

Some notations are given below to facilitate our discussion. Given a dataset $X = \{x(i)\}_{N, i=1 \in}$, the neighbor set of x is denoted as $N(x)$.

The paradigm of binary hashing is to first use a set of linear or non-linear hash functions $F = \{f_k: R^d \rightarrow R\}_{K, k=1}$ to map $x \in X$ to $F(x) \in R^K$, and then binarize,

$F(x) = (f_{1(x)} \dots f_{k(x)})^T$ By comparing each with $f_{k(x)}$ a threshold T_k to get a K-bit binary code $H(x) \in \{0,1\}^K$. Hence, the binary hash function is given by the, $h_{k(x)} = \text{sgn}(f_{k(x)} - T_k)$ we call the $f_{k(x)}$ unbinarized hash value. Each dimension of $H(x)$ is called a hash bit, and for a query q and its neighbour p , if the k -th bit of $H(q)$ and $H(p)$ is different, we call there is a bit-flipping on hash bit k . The weighted Hamming distance between two binary codes $h(1)$ and $h(2)$ is denoted as $D^w H(h(1), h(2))$.

IV. EXPERIMENTAL RESULTS

A. SIMULATION RESULTS OF FACE RECOGNITION

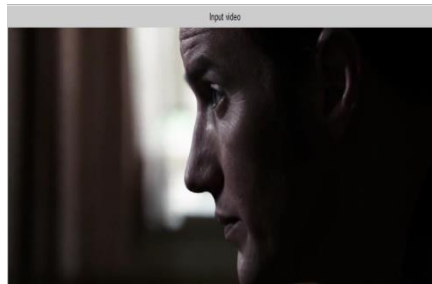


Figure 3. input video for face



Figure 4. Deblurred face

The above figure 3 shows the input video taken for face recognition from side pose and the figure 4. Shows the deblurred face and here the blur is implemented by first creating a PSF filter in Mat Lab that would approximate linear motion blur.

This PSF was then convolved with the original image to produce the blurred image the most important technique for removal of blur in images due to linear motion is the Wiener filter and this process is called the illumination adjustment.

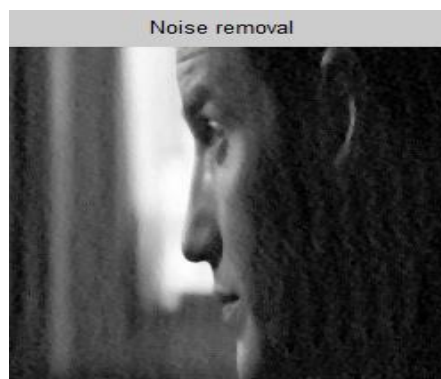


Figure 5. Removal of noise from face



Figure 6. Face detection

The figure 5 shows the removal of salt and pepper noise from face image using median filter and the figure 6. shows the image of face detection from video using viola-Jones algorithm.

The figure 7. shows extraction of local features from detected face using multi scale feature extraction method the features from chin, cheek and ear points are considered and in the figure. 8 the authentication of face image is done by using DBPNN method.

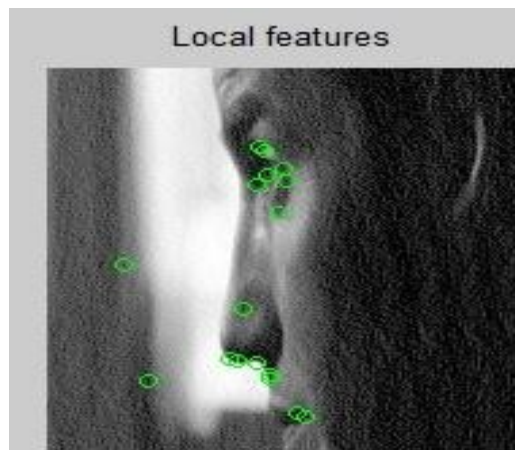


Figure 7. Extraction of local features

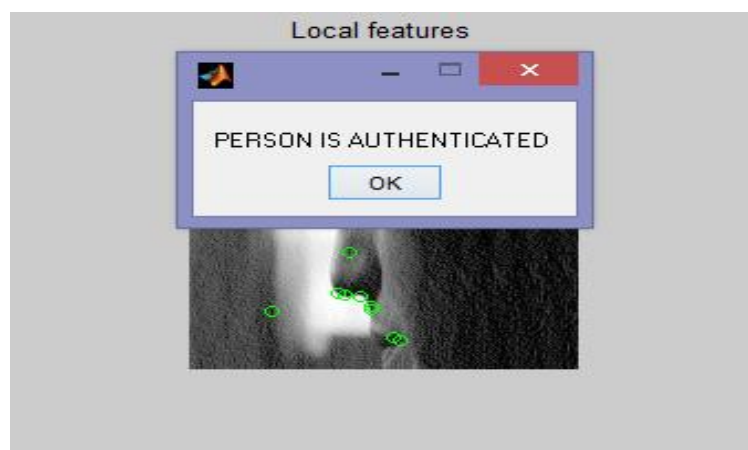


Figure 8. Face authentication

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

r1 =
    0.5886

Elapsed time is 0.253792 seconds.
Elapsed time is 0.202028 seconds.

accuracy4_proposed =
    96.6667

recograte_proposed =
    33.3333

accuracy4_exsiting =
    93.3333

recograte_existing =
    30
```

Figure 9. Command window from mat lab

The above figure shows the output command window obtained by running the program coding and these values are used for evaluating the performance of the proposed methods applied.

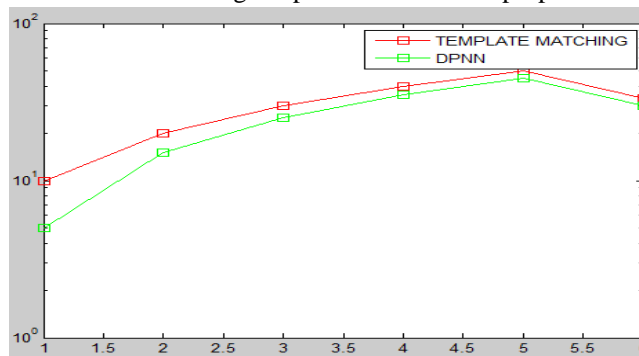


Figure.10 performance graph of template matching versus DBPNN

Figure.10 gives the performance evaluation of template matching with DBPNN where the graph is plotted between accuracy and recognition rate.

B.SIMULATION RESULTS OF IRIS RECOGNITION.

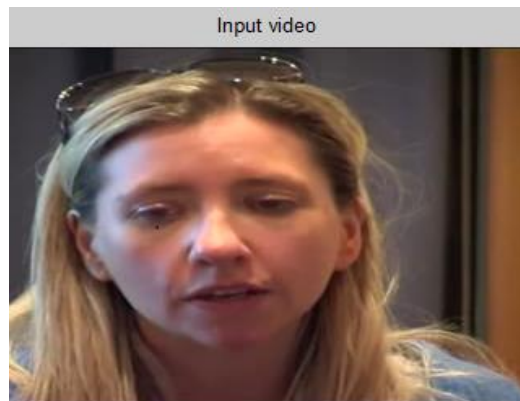


Figure 11. Input video for iris

The figure.11 is the input video considered for iris recognition process and the figure.12 is the deblurred image from the liner motion blurred image using wiener filter.

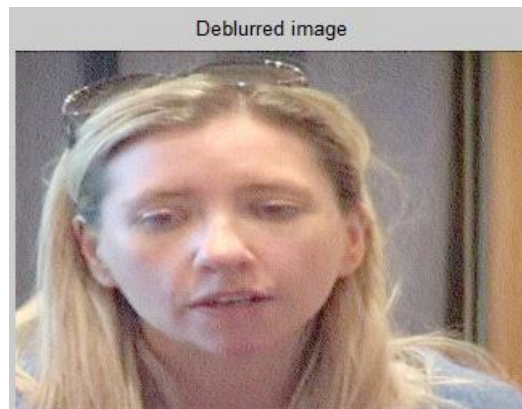


Figure 12 .deblurred images



Figure 13. Noise removals from video

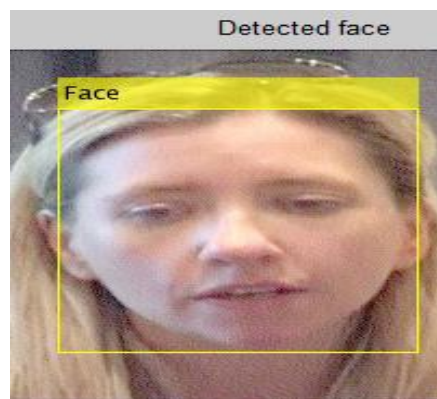


Figure 14. Face detection

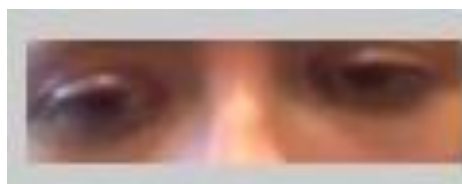


Figure 15.detected eyes

The figure 13 is the removal of salt and pepper noise from the input video using median filter and the figure 14, 15 shows the face and eyes detected from the input video using viola Jones algorithm.

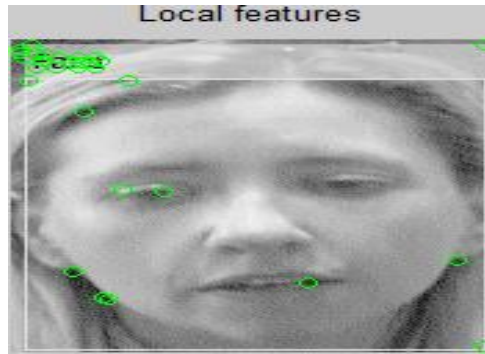


Figure 16. Feature extraction

The figure 16. Shows the feature extraction process where the features of the eyes are extracted using Iterative randomized hough transform.

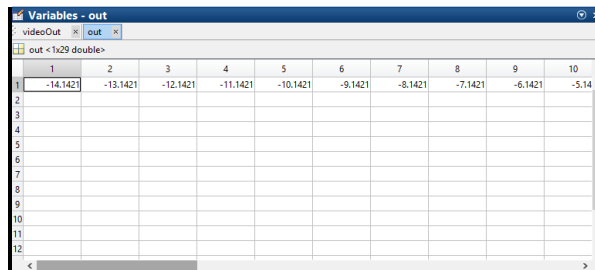
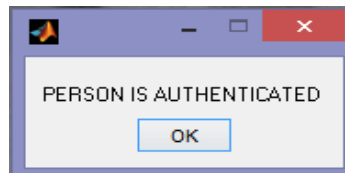


Figure 17 IRHT command window



Figures 18.authentication

Figure 17. Shows the output obtained from the features of the eyes using IRHT method and in the figure 18. Authentication of iris features can be obtained from IRHT.

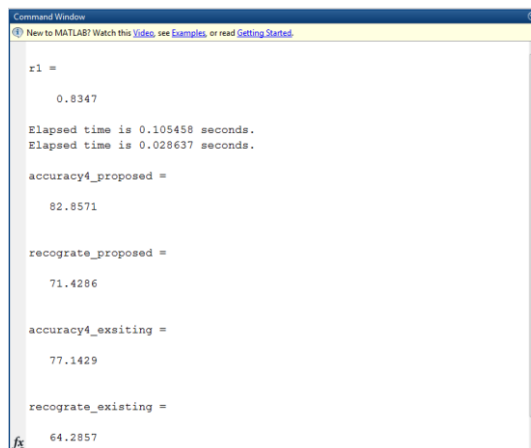


Figure 19. Command window of mat lab by executing the program coding.

The figure19. Gives the values needed for performance evaluation between adaptive matching and adaptive hamming distance.

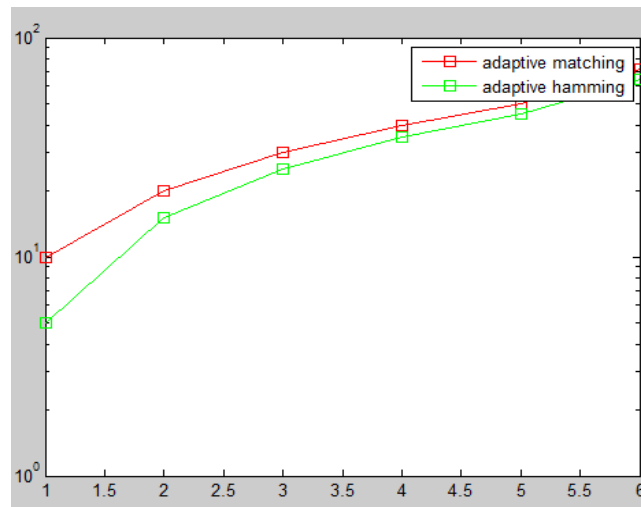


Figure 20. Performance graph of adaptive matching versus adaptive hamming

The figure 20. Shows the comparison between accuracy and recognition rate of adaptive matching and adaptive hamming distance method.

C. Comparison table for face recognition method.

ALGORITHM	ACCURACY	RECOGNITION RATE
TEMPLATE MATCHING (EXISTING)	93.33	30
DBPNN (PROPOSED)	96.66	33.33

Table 1. Comparison chart of output values of face recognition

D. Comparison table for iris recognition method

ALGORITHM	ACCURACY	RECOGNITION RATE
ADAPTIVE MATCHING (EXISTING)	77.14	64.20
ADAPTIVE HAMMING DISTANCE (PROPOSED)	82.55	71.42

Table Comparison chart of output values of Iris recognition Rate is calculated by,

$$\text{Recognition Rate} = \frac{\text{Total no. of. images}}{\text{correctly accepted images}}$$

Accuracy is calculated by,

$$\text{Accuracy} = 100 \times \text{True Positive rates}$$

Angle of the face is calculated by,

$$\text{Angle} = \text{obtained value} \times 180$$

V. CONCLUSION

Face recognition is a complicated task that requires efficient handling of complex variations in facial appearance caused by a range of factors such as, illumination variation, expressions, and aging effects. In the existing method, a highly compact and discriminative feature descriptor and adaptive matching framework is used for enhanced face recognition. But the method is not suitable for the face in side pose or different angle. In order to overcome this, a new methodology is proposed which performs face recognition under the combined effects of deblurring and denoising. In this method, Viola-Jones algorithm is initially used for detecting the face and dense feature extraction is used for feature extraction. DBPNN is used to detect the face from side pose and finally Iris detection is done by using Iterative randomized Hough transform where the pupil of the eyes is detected with number of iteration. The proposed method significantly reduces the dimension of feature representation and improved computational efficiency without sacrificing recognition performance. The future work is to recognize the face in side pose even from occluded face.

REFERENCES

- [1] Ahonen.T, Hadid.A, and Pietikäinen.M, "Face recognition with local binary patterns," in Proc. ECCV, 2004, pp. 469–481.
- [2] Belhumeur .P.N, Hespanha.J.P, and Kriegman.D.J, "Eigenface vs. Fisher faces: Recognition using class specific linear projection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 711–720, Jul. 1997.
- [3] Bao.B.K, Liu.G, Hong.R, Yan.S, and Xu.C, "General subspace learning with corrupted training data via graph embedding," IEEE Trans. Image Process., vol. 22, no. 11, pp. 4380–4393, Nov. 2013.
- [4] Cao.X, Yin.Q, Tang.X, and Sun.J, "Face recognition with learning based descriptor," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 2707–2714.
- [5] Chen.D, Cao.X, Wen.F, and Sun.J, "Blessing of dimensionality: High dimensional Feature and its efficient compression for face verification," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 3025–3032.
- [6] Dalal.N and Triggs.B, "Histograms of oriented gradients for human detection," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2005, pp. 886–893.
- [7] Huang.G.B, Ramesh.M, Berg.T, and Learned-Miller.E, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.
- [8] Klare .B and Jain A.K, "Heterogeneous face recognition: Matching NIR to visible light images," in Proc. 20th Int. Conf. Pattern Recognit. (ICPR), Aug. 2010, pp. 1513–1516.
- [9] Klare.B.F, Li.Z, and Jain.A.K, "Matching forensic sketches to mug shot photos," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 3, pp. 639–646, Mar. 2011.
- [10] Lei.Z, Pietikainen.M, and Li.S.Z, "Learning discriminant face descriptor," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 2, pp. 289–302, Feb. 2014.
- [11] Lu.J, Plataniotis.K.P, and Venetsanopoulos A.N, "Face recognition using kernel direct discriminant analysis algorithms," IEEE Trans. Neural Netw., vol. 14, no. 1, pp. 117–126, Jan. 2003.

- [12] Lowe.D.G, "Object recognition from local scale-invariant features," in Proc. IEEE Int. Conf. Comput. Vis., Sep. 1999, pp. 1150–1157.
- [13] Liu.C, "Gabor-based kernel PCA with fractional power polynomial models for face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 5, pp. 572–581, May 2004.
- [14] Mäenpää.T and Pietikäinen.M, "Multi-scale binary patterns for texture analysis," in Proc. SCIA, 2003, pp. 885–892.
- [15] Simonyan.K, ParkhiO.M, Vedaldi.A, and Zisserman.A, "Fisher vector faces in the wild," in Proc. BMVC, 2013, pp. 8.1–8.12.
- [16] Turk and.M.A, Pentland.A.P, "Face recognition using eigenface," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 1991, pp. 586–591.
- [17] Wang.W and Tang.X, "A unified framework for subspace face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 9, pp. 1222–1228, Sep. 2004.
- [18] Zhifeng Li, Senior Member, IEEE, Dihong Gong, Xuelong Li, Fellow, IEEE, and Dacheng Tao, Fellow, IEEE "Learning Compact Feature Descriptor and Adaptive Matching Framework for Face Recognition", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 24, NO. 9, SEPTEMBER 2015