Wireless Personal Communications (2021) 118:3403–3433
https://doi.org/10.1007/s11277-021-08186-9

Performance Evaluation of Machine Learning Based Face Recognition Techniques

Sahil Sharma[1] · Vijay Kumar[2]

Accepted: 28 January 2021 / Published online: 16 February 2021

Abstract
The robustness of machine-learning model-based face recognition techniques to image processing attacks using the quantization of extracted features is presented. Recently developed face recognition techniques based on machine learning models have been outperformed over traditional face recognition techniques. An efficient face recognition technology should be able to resist various image processing attacks. This paper presents the simulation results by evaluating ten variants of machine-learning-based face recognition techniques on ten well-known image processing attacks. The quality of face recognition techniques has been assessed on recognition accuracy. The performance has been evaluated on two well-known face databases viz. Bosphorus and University of Milano Bicocca (UMB) face database. The experimental results reveal that the Subspace discriminant ensemble-based face recognition model has consistently performed in most image processing attacks. All image processing attacks have been visually verified and presented.

Keywords Enhancement attacks · Geometric attacks · Noise attacks · Classification · Quantization · HOG · Face recognition

1 Introduction

Face recognition is one of the widely researched topics in the field of computer vision for decades now. Currently, face recognition has reached mobile devices for unlocking of phones and surveillance purposes using drones [1]. Some common face recognition challenges are occlusion, make-up, illumination, image processing attacks etc. [2]. Face recognition has been studied under different attacks viz. stealth attacks [3], spoof attack [4], presentation attack [5], backdoor attacks [6].

✉ Vijay Kumar
vijaykumarchahar@gmail.com

Sahil Sharma
sahil301290@gmail.com

1 Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India

2 Computer Science and Engineering Department, National Institute of Technology, Hamirpur, India

This work is the extension of Sharma and Kumar [7]. The earlier work was done without the mathematical modelling and the pseudo-codes of the image processing attacks presented in this paper. In previous work, the focus was on literature already existing in addition to the empirical evaluation of the attacks. Zangeneh and Moradi [8] proposed a method to recognize the facial expressions using the differential geometric features. Geometric features are extracted by identifying the changes in the facial landmark values after the change in expression. Ahmad et al. [9] presented a pre-processing technique using independent component analysis to separate the single image's illumination and reflectance component for a face recognition system. Hsia et al. [10] proposed a backlight compensation technique to improve face recognition accuracy. The brightness and contrast of a face image favourably impact the quality of the face recognition system. Parubochyi and Shuwar [11] presented a self-quotient image method based on globally modified Gaussian filter kernel for light normalization. The most significant advantage of the self-quotient image technique is that it uses a single shot of an image. Sharma and Patterh [12] presented a review of feature extraction and recognition techniques for faces. The main methods that have been highlighted in this paper are Support Vector Machine based machine learning for face recognition, Latent Dirichlet Allocation and Discrete Cosine Transform feature engineering techniques. There are different researches in the field of face recognition that link with adversarial attacks [55, 57], make-up [56, 58], expression-based [60], age-based [63], handling bias [62], presentation attacks based [59], and based on explainable artificial intelligence [61].

Image processing attacks are classified into three broad classes, namely, image enhancement attacks, geometric image attacks, and image noise attacks [13]. Enhancement and noise attacks do not affect the number of pixels in an image but modify them. In geometric attacks, the number of pixels is involved. Machine learning plays a vital role when working in pattern recognition and image classification. After the features have been extracted from a face image, they are quantized using the rounding-up technique and then given as input to the machine learning algorithm for training purpose. Quantization is a signal processing technique that converts the given input into smaller sets most commonly by rounding up technique or modulus technique. Quantization can also be seen as a compression technique as the original features are being reduced. Four classes of machine learning viz. support vector machine, k-nearest neighbour, decision trees and discriminant analysis along with ensemble modelling have been explored for training and testing of image attacks invariant face recognition system [14–19].

There are many facial datasets available publicly. In the presented work, two datasets, namely Bosphorus face dataset [20], and University of Milano Bicocca (UMB) face dataset [21] have been used. We investigate the image processing attacking from a new perspective: how they affect the machine-learning-based face recognition techniques. In our knowledge, this is the first attempt to study the impact of different machine learning algorithms on the face recognition system under image processing attacks. The ten well-known image processing attacks are discussed with their time complexities. These are blurring, sharpening, median filtering, histogram equalization, resizing, rotation, cropping, Gaussian noise, Poisson noise and speckle noise attacks. They are evaluated in conjunction with ten machine-learning variant based face recognition techniques over two face databases. The rest of the paper is structured as follows: Sect. 2 presents the preliminary concepts of image processing and face recognition systems. Sect. 3 introduces the machine-learning models-based face recognition system. The experimental results and discussions are mentioned in Sect. 4. The visual verification of the system is shown in Sect. 5. The concluding remarks are drawn in Sect. 6.

2 Preliminaries

This section discusses the theory and mathematics related to the subject of image processing attacks and face recognition.

2.1 Basic Concepts of Face Recognition System

Training and testing are the two significant phases in face recognition. While training the face recognition system, a certain portion of the dataset is considered out of the full dataset. Face registration, pre-processing, feature extraction and machine learning are performed gradually till the classification model is trained for face recognition. Testing is done using the probe image by completing the registration, pre-processing, feature extraction and training generated model validation. The phases which are responsible for face recognition under different challenges can be seen in Fig. 1.

2.1.1 Dataset Collection

The face images can be collected with two methods, namely primary and secondary approach. Dataset is primary when the researcher collects data for novel use else; it is secondary [22]. The collected face images are correctly labelled for the right usage. Two dimensional (2D), two and a half dimensional (2.5D) or depth images and three dimensional (3D) [52–54] are the three type of face images that can form a dataset in single or multiple repositories.

2.1.2 Training Images

In the training phase, multiple images are read into the face recognition system being built. When training and testing phases are in the face recognition system's development phase, the training-testing ratio is set. When the best approach is found for creating the face recognition system, a full dataset is used to train the system. When a probe image comes for face

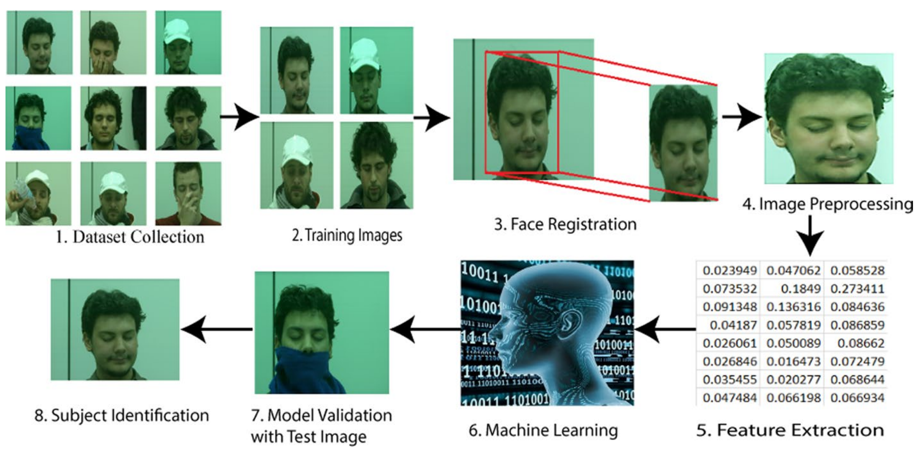

Fig. 1 Phases of face recognition [7]

identification or verification, it is processed and matched for correlation with the images trained in the system, returning the identified or verified person of interest.

2.1.3 Face Registration

After reading the dataset, the next task is to do the segmentation of face from the image. The main reason behind the face's segmentation is to focus on the pixels of face only and discard the rest of the image for better training purposes. This process of focusing on the face is known as face registration. It can be enhanced by using multiple techniques viz. iterative closest point (ICP) algorithm, spin images, simulated annealing and intrinsic coordinate system for the three-dimensional face registration process [23].

2.1.4 Image Pre-processing

This phase improves the quality of the image being processed. It can be either of the enhancement, geometric or noise attacks. Pre-processing an image is necessary for making the image ideal for feature extraction. This phase takes place in both cases of training and testing of the face recognition system.

2.1.5 Feature Extraction

There is a plethora of feature extraction techniques in the image processing literature viz. histogram of oriented gradients (HOG), speeded up robust features (SURF), local binary pattern (LBP) features, haar-like features, haralick features etc. [24–28]. All types of feature extraction techniques depend on the pixels of an image. Different distance metrics viz. Euclidean distance, city block distance, Minkowski distance, Mahalanobis distance etc. are available for the features to interact with each other during different machine learning classifications [29–31].

Feature extraction is done in both the training and the testing phases. Based on training images, features are extracted for the machine learning phase. Based on testing images, features are extracted for the model validation phase.

2.1.6 Machine Learning

During the face recognition system training, the machine learning phase is implemented after feature extraction of the image. This phase includes the crunching of features into the algorithms which uses different parameters for building the mathematical equations and correlations for the prediction of discrete class in case of classification or a real number in regression.

2.1.7 Model Validation

When the testing phase is under process, probe image is read, pre-processed and feature extracted for the prediction to be done by the machine learning trained model. The output of this phase gives the probable class of the person to which the photo belongs.

2.1.8 Subject Identification

The result of the model validation phase is compared to the expected output for the matter of subject identification. If the model validation phase output matches exactly the expected output, it is said to be true positive. If the model validation phase output does not match the expected output, it is true negative [32].

2.2 Image Processing Attacks

There are three classes of image processing attacks viz. enhancement attacks, geometric attacks and noise attacks.

2.2.1 Enhancement Attacks

Image enhancement attacks are the form of attacks that do not affect an image's size but modifies the existing pixels. There are four types of enhancement attacks chosen to be discussed viz. blurring, sharpening, median filtering and histogram equalization [13]. These can be seen in Fig. 2. The face used in Fig. 2 has been taken from the Bosphorus dataset [20].

Pseudo codes and time complexities of each enhancement

Blur Attack

Input: I[m, n] : Original Image, K[w, h] : Kernel

Output: B[m, n] : Blurred Image

$For\ \ \mathrm{i}=1\ \text{to}\ \mathrm{n}$

$\quad For\ \ j=1\ \text{to}\ m$

$\quad\quad Value = I[(i+1)*m+j+1]$

$\quad\quad For\ \ Ki=1\ \text{to}\ \mathrm{w}$

$\quad\quad\quad For\ \ Kj=1\ \text{to}\ \mathrm{h}$

$\quad\quad\quad\quad Value = Value + (I[(i+Ki)*m+j+Kj]*K[Ki*w+j]$

$\quad\quad\quad End\ \ for$

$\quad\quad End\ \ for$

$\quad\quad \mathrm{B[i,j]} = clamp(\mathrm{Value}, 0, 255)$

$\quad End\ \ for$

$End\ \ for$

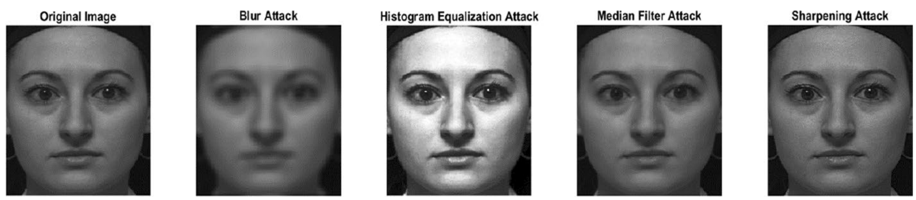

Fig. 2 Image enhancement attacks [20]

The time complexity of the blurring pseudo code is $O(m * n * w * h)$, where m is the width of the original image, n is the height of the original image, w is the width of the blurring kernel and h is the height of the blurring kernel.

Blurring is one of the image processing techniques in which the image's pixels are affected by the surrounding pixels [33]. This method is used for smoothing and edge detection. When blurring is increased, it drastically affects the recognition rate in case of face recognition.

Histogram Equalization Attack

Input: I[m, n] : Image, G : Maximum grey level

Output: HE[m, n] : Histogram equalization

```
Calculate histogram H for an image I
T[0] = H[0]
For i = 1 to G
    T[i] = T[i-1] + H[i]
End for
Calculate histogram equalization HE for image I
For y = 0 to m
    For x = 0 to n
        j = I[y, x]
        HE[y, x] = S[j]
    End for
End for
```

Time Complexity of the histogram equalization attack is $O(m * n)$, where m, n are the dimensions of the original image. Histogram equalization technique improves the overall quality of the image by increasing the intensity of all the pixels.

Median Filter Attack

Input: $\mathrm{I[m,n]}$: Original Image, $K\mathrm{[w,h]}$: Kernel

Output: $M\mathrm{[m,n]}$: Median Filtered Image

$Kr = (h-1)/2$

$For\ \ y = 1$ to n

$\quad Up = Max(y - Kr, 0)$

$\quad Down = Min(y + Kr, n-1)$

$\quad Array = a(w*h)$

$\quad For\ \ x = 1$ to m

$\quad\quad L = Max(x - Kr, 0)$

$\quad\quad R = Min(x + Kr, w-1)$

$\quad\quad Array = a[w*h]$

$\quad\quad count = 0$

$\quad\quad For\ \ \mathrm{i} = Up$ to $Down$

$\quad\quad\quad For\ \ j = L$ to R

$\quad\quad\quad\quad a[count] = I[i][j]$

$\quad\quad\quad\quad count = count + 1$

$\quad\quad\quad End\ \ for$

$\quad\quad End\ \ for$

$\quad\quad I[x, y] = Median[a]$

$\quad End\ \ for$

$End\ \ for$

The time complexity of the median filter attack is $O(m * n * w * h)$ where m, n are the dimensions of the original image and w, h are the dimensions of the kernel filter. Median filtering is an enhancement attack used for reducing the noise in an image. In this method, full image convolution is done for attenuating the noise signal.

Sharpening Attack

Input: I[m,n]: Original Image, K[w,h]: Sharpening Filter
Output: S[m,n]: Sharpened Image

$For\ i=1$ to m
    $For\ j=1$ to n
        $S[i][j]=I[i][j]$
    $End\ for$
$End\ for$
$For\ i=1$ to n
    $For\ j=1$ to m
        $Pixel=0$
        $For\ k=-h/2$ to $h/2$
            $For\ l=-w/2$ to $w/2$
                $Pixel=Pixel+K[k+1][l+1]*I[i+k][j+1]$
            $End\ for$
        $End\ for$
        $NewVal=(\text{int})(I[i][j]-Pixel)$
        $S[i][j]=clamp(NewVal,0,255)$
    $End\ for$
$End\ for$

The time complexity of the sharpening attack is $O(m*n*w*h)$ where m,n are the dimensions of the original image and w,h are the dimensions of the kernel filter. Addition of the original image and the signal proportional to high pass filtering version of the original image is known as sharpening. This is a technique of increasing the pixel intensities of an image for enhancing fine details and edges of the image [34].

2.2.2 Geometric Attack

Image geometric attacks can be defined as those attacks which affect the number of pixels in an image. Experimentation has been done on three geometric attacks: viz. resize, rotation and cropping [35]. These can be seen in Fig. 3.

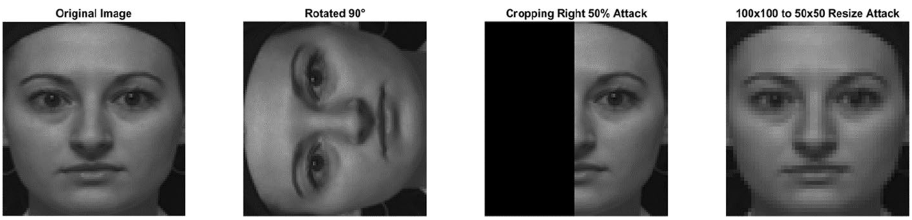

Fig. 3 Image Geometric Attacks [20]

Pseudo code and time complexities of each geometric attack are as follows:

Rotation Attack

```
Input: I[m, n]: Original Image, Ra: {90, 180, 270}
Output: R[m, n]: Rotated Image
For i = 1 to m
    For j = 1 to n
        R[i][j] = 0
    End for
End for
For i = 0 to m-1
    For j = 0 to n-1
        If Ra = 90
            R[j][n-1-i] = I[i][j]
        Else If Ra = 180
            R[m-1-j][n-1-i] = I[i][j]
        Else If Ra = 270
            R[m-1-j][i] = I[i][j]
        End if
    End for
End for
```

Time Complexity of the rotation attack is $O(m * n)$, where m, n are the dimensions of the original image. Rotation as an image processing attack is defined as a geometric transformation which deals with moving the whole image to given angle moving along the base in an anticlockwise or clockwise direction [36]. Image padding is applied to an image before being rotated.

Cropping Attack

Input: I[m][n]: Original Image, Cf[m][n]: Crop Filter
Output: CI[m][n]: Cropped Image

```
Rows = m
Columns = n
For i = 1 to m
    For j = 1 to n
        Cf[i][j] = 0
    End for
End for
If CropFilter = 25
    For i = 1 to m
        For j = 1 to n/4
            Cf[i][j] = 1
        End for
    End for
Else If CropFilter = 50
    For i = 1 to m
        For j = 1 to n/2
            Cf[i][j] = 1
        End for
    End for
Else If CropFilter = 75
    For i = 1 to m
        For j = 1 to (3*n)/4
            Cf[i][j] = 1
        End for
    End for
End If
For i = 1 to m
    For j = 1 to n
        CI[i][j] = I[i][j]*Cf[i][j]
    End for
End for
```

Time Complexity of the cropping attack is $O(m * n)$, where m, n are the dimensions of the original image. Cropping is a geometric attack similar to image segmentation. In cropping, image is partially filled with zeroes and the remaining part is left visible after the attack.

Resize Attack

Input: I[m,n]: Input Image, dsf: Downscaling factor
Output: S[a,b]: Scaled Image

```
a = m * dsf
b = n * dsf
batch = round(1 / dsf)
counta = countb = counts = count = sum = 0
For i = 1 to m
    For j = 1 to n
        count = count + 1
        sum = sum + I[i][j]
        If (count = batch)
            count = 0
            S[counta][countb] = round(sum / count)
            countb = countb + 1
            If (countb = b)
                countb = 0
            End if
            counts = counts + 1
            If (counts = a)
                counta = counta + 1
            End if
        End if
    End for
End for
```

Time Complexity of the resize attack in down-sampling is $O(m * n)$, where m, n are the dimensions of the original image. Resizing or scaling an image deals with up-sampling or down-sampling the number of pixels in an image [37]. Interpolation techniques are used in both the cases. When an image is up-scaled, the image quality decreases unless super resolution techniques are used. Face recognition accuracy drastically decreases when an image is up-scaled.

2.2.3 Noise Attacks

Image noise attacks are the attacks done directly on the pixels of an image. Generally, they are done based on density or the variance of their type. Direct changes are brought in an image by manipulating pixels. Image size is not affected by this attack. This work experimentation has been done using three types of noise attacks: gaussian noise attack, speckle noise attack, and poisson noise attack [36]. These can be seen in Fig. 4.

Pseudo code and time complexities of each noise attack are as follows:

Fig. 4 Image Noise Attacks [20]

Gaussian Attack

Input: I[m,n]: Original Image, Mean of Gaussian Attack: 0, Variance of Gaussian Attack: v

Output: G[m,n]: Gaussian White Noise Image

$N = m * n$

$Count = 0$

$For\ \ i = 1$ to m

  $For\ \ j = 1$ to n

    $If\ \ Count = (N / (v * 100)$

      $I[i][j] = 255$

      $Count = 0$

    $End\ \ if$

    $Count = Count + 1$

  $End\ \ for$

$End\ \ for$

Time Complexity of the gaussian noise attack is $O(m * n)$, where m, n are the dimensions of the original image. Gaussian noise attack is one of the most famous noise attack. In this attack, white pixels are added uniformly in the image. This method changes the original pixels throughout the image, making image corrupt.

Speckle Attack

Input: I[m,n]: Original Image, Mean of Speckle Attack: 0, Variance of Speckle Attack: v
Output: S[m,n]: Speckle Noise Image

$For\ \ i = 1$ to m
    $For\ \ j = 1$ to n
        $S[i][j] = 0$
    $End\ \ for$
$End\ \ for$
$For\ \ i = 1$ to m
    $For\ \ j = 1$ to n
        $S[i][j] = clamp(I[i][j] + (v * 100 * rand()), 0, 255)$
    $End\ \ for$
$End\ \ for$

Y = I + β*I; β = uniformly distributed random noise with values of mean and variance.

Time Complexity of the speckle noise attack is $O(m * n)$, where m, n are the dimensions of the original image. Speckle noise is multiplicative in nature. Noise and signal are statistically independent [38]. This noise has very prominent existence in ultrasound images. It deteriorates the edges and other fine details affecting the contrast of the image, which in return makes detection of lesions difficult [39].

Poisson Attack

Input: I[m,n]: Original Image
Output: P[m,n]: Poisson Noise Image

$For\ \ i = 1$ to m
    $For\ \ j = 1$ to n
        $\mathrm{P}[i][j] = 0$
    $End\ \ for$
$End\ \ for$
$For\ \ i = 1$ to m
    $For\ \ j = 1$ to n
        $\mathrm{P}[i][j] = I[i][j] * rand(1)$
    $End\ \ for$
$End\ \ for$

Time Complexity of the poisson noise attack is $O(m * n)$, where m, n are the dimensions of the original image.

Poisson noise is applied to an image in contrast to adding noise such as Gaussian. Poisson noise or Shot noise occurs when finite energy particles in electrical circuit generates measurable statistical fluctuations [40]. Poisson noise percentage is higher at darker pixels as compared to lighter pixels.

2.3 Need of Attacks Invariant Face Recognition System

Face recognition systems are prone to different forms of challenges including illumination, occlusion, make-up, age, enhancement, geometric and noise attacks. Three different forms of attacks have been considered viz. enhancement, geometric and noise attacks in the presented work.

3 Models used for Face Recognition

3.1 Motivation

The work presented in this paper makes use of the machine learning models with face recognition invariant of image processing attacks on such a wide scale. To the best of our knowledge, this work is being done the first time, including quantifying histogram of oriented gradients.

3.2 Mathematics of Models

Face recognition algorithms that have been used are Support Vector Machine, K-Nearest Neighbours, Discriminant and Bagged Tree Ensemble model. Table 1 presents the mathematics of machine learning models.

4 Experimentation and Result Discussions

This section presents the detail of experimentation of this research. Sub-sections have been made based on dataset detail, experimental setup and empirical evaluation.

4.1 Datasets Used

The Bosphorus face database [20] and the University of Milano Bicocca (UMB) face database [21] are two state-of-the-art face databases for all the experiments presented in this paper. Bosphorus database has a total of 4666 face images of 105 subjects. These images have good illumination and require less amount of pre-processing in the training and testing phases. UMBDB has a total of 1473 face images of 143 subjects clicked in multiple backgrounds and light illuminations.

4.2 Experimental Setup

The experimental platform has been developed on Dell Inspiron computer with Intel(R) Core(TM) i7-7500U CPU@2.70GHz and 16G RAM. The testing software is MALTAB

Table 1 Mathematics of Models

Model	Mathematics
Support Vector Machine (SVM)	Linear SVM: a) Decision boundary $h(\vec{u}) = w_1 u + w_2 v + b \geq 0$ Where (w, b) are Lagrangian parameters; u, v: 2D coordinates b) Class distinguishing margin $m = \frac{2}{\|\|\vec{w}\|\|}$, where $\|\|\vec{w}\|\| = \sqrt{\sum_i w_i^2}$ Where m: margin, w: width of margin. c) Solve for alphas, α $\sum_i \alpha_i v_i \vec{u}_i = \vec{w}$ Quadratic SVM: $K(\vec{u}, \vec{v}) = (\vec{u} \cdot \vec{v} + b)^2$ where K is Quadratic Kernel; u=(x, y). Cubic SVM: $K(\vec{u}, \vec{v}) = (\vec{u} \cdot \vec{v} + b)^3$, where K is Cubic Kernel; u=(x, y).Gaussian SVM: $K(\vec{u}, \vec{v}) = \exp\left(-\frac{\|\|\vec{u}-\vec{v}\|\|^2}{2\sigma^2}\right)$ Where K is Gaussian Kernel; Large σ^2 gets Gaussians flat and sharp otherwise
K-Nearest Neighbour (KNN)	a) Euclidean Distance Equation $d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + ... + (x_n - x'_n)^2}$ d is the distance of the point x and every other training observation. K points in close proximity to x is said to be set b) Cosine distance equation $\cos D = \frac{A \cdot B}{\|\|A\|\|_2 \|\|B\|\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$ Where A_i and B_i are the components of vectors A and B. c) Weighted distance equation $wD_{x,y} = \sqrt{\sum_{j=1}^{J} w_j (x_j - y_j)^2}$ Where $w_j = 1/s_j^2$ is inverse of j-th variance and s_j is sample standard deviation. Fine KNN: Number of Neighbours = 1. Interclass fine distinctions. Medium KNN: Number of Neighbours = 10. Interclass medium distinctions. Cosine KNN: Number of Neighbours =10. Interclass medium distinction with cosine distance metric. Weighted KNN: Number of Neighbours =10. Interclass medium distinction with weighted distance. Cubic KNN: Number of Neighbours = 10. Interclass medium distinction with cubic distance metric. $P(y = j \| X = x) = \frac{1}{K} \sum_{i \epsilon B} I(y^{(i)} = j)$ Calculating the conditional probability of points in set B corresponding to each class. Max probability class gets assigned to each test point

Table 1 (continued)

Model	Mathematics
Linear Discriminant	Two classes having mean μ_1 and μ_2, covariance $\sum_0$ and $\sum_1$. $\vec{w} \cdot \vec{x}$ be linear combination of features, mean = $\vec{w} \cdot \vec{\mu}_i$ and variance = $\vec{w}^T \sum_i \vec{w}$ for i = 0, 1s $S = \frac{(\vec{w} \cdot (\vec{\mu}_1 - \vec{\mu}_0))^2}{\vec{w}^T (\sum_0 + \sum_1) \vec{w}}$ Where S is the separation in two classes.

Table 2 Algorithms and their Parameters Initialization

Algorithm	Parameter	Values/Type
Fine K-Nearest Neighbour [41]	K (Neighbours)	1
	Distance Metric	City Block
	Distance Weight	Equal
Medium K-Nearest Neighbour [42]	K (Neighbours)	10
	Distance Metric	City Block
	Distance Weight	Equal
Cosine K-Nearest Neighbour [43]	K (Neighbours)	10
	Distance Metric	Cosine
	Distance Weight	Equal
Weighted K-Nearest Neighbour [44]	K (Neighbours)	10
	Distance Metric	City Block
	Distance Weight	Squared Inverse
Cubic K-Nearest Neighbour [45]	K (Neighbours) Distance	10
	Distance Metric	Minkowski
	Distance Weight	Equal
Linear Support Vector Machine [46]	Kernel	Linear
	C	1
Quadratic Support Vector Machine [47]	Kernel	Quadratic
	C	1
	γ	0.05
Cubic Support Vector Machine [48]	Kernel	Cubic
	C	1
	γ	0.05
Gaussian Support Vector Machine [49]	Kernel	RBF
	C	1
	γ	0.015
Linear Discriminant [50]	Covariance Structure	Diagonal
Subspace Discriminant [51]	Number of Learners	25 500
	Subspace Dimension	
All Models	Train-Test Partition	0.7-0.3

Table 3 Image processing attack class, name and variation handled in current research

Attack class	Attack name	Variation
Enhancement	Blurring	Kernel Size: [5 5], [9 9]
	Sharpening	Image Size: [50 50], [100 100]
	Median Filtering	Image Size: [50 50], [100 100]
	Histogram Equalization	Image Size: [50 50], [100 100]
Geometric	Resize	Image Size: [50 50], [100 100]
	Rotation	Anticlockwise: 90°, 180°, 270°
	Cropping	Right-Side: 25%, 50%, 75%
Noise	Gaussian	Mean: 0, Variance: 0.05, 0.15, 0.25, 0.35, 0.45
	Poisson	Image Size: [50 50], [100 100]
	Speckle	Mean: 0, Variance: 0.01, 0.04, 0.10, 0.20, 0.40

Table 4 Effect of blurring on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset			UMBDB dataset		
	Filter 5×5	Filter 9×9	Rank of model	Filter 5×5	Filter 9×9	Rank of model
Subspace Discriminant	80.1	78.1	1	77.2	76.5	1
Subspace KNN	76.0	69.2	2	72.9	69.3	2
Fine KNN	75.0	67.0	3	70.7	67.4	3
Weighted KNN	73.8	69.6	4	68.3	65.7	7
Quadratic SVM	73.3	69.0	5	69.5	63.3	5
Cubic SVM	72.9	68.4	6	68.6	61.6	6
Medium Gaussian SVM	71.9	68.1	7	70.0	66.2	4
Medium KNN	71.6	67.0	8	61.4	61.6	8
Cubic KNN	70.3	66.1	9	61.6	56.1	9
Cosine KNN	69.8	61.7	10	60.9	59.5	10

2017a licensed under Thapar Institute of Engineering and Technology and running on Windows 10.

Table 2 presents the machine learning model's initialisation table parameters, representing the basic parameters with their initial values when models were trained.

4.3 Empirical Evaluation

This sub-section presents an extensive analysis of image processing attacks by comparing ten variants of machine learning models. All attacks have been implemented after the quantization of HOG features. Variations of attacks presented in this research can be seen in Table 3.

4.3.1 Experimentation 1: Effect of Enhancement Attacks on Machine Learning based FR Systems

Multiple machine learning models have been tested for accuracy by varying the parameters of enhancement attacks. Effect of four different enhancement attacks have been shown as follows:

4.3.2 Effect of Blurring on Models

Table 4 shows the blurring effect on the face recognition accuracy on both datasets namely Bosphorus and UMBDB with model ranking. The variants of three classification models, namely support vector machine, k-nearest neighbor, and discriminant analysis, have been used to train and test blurring attacks in the face recognition system.

Subspace discriminant ensemble model achieves the best accuracy of 80.1% and 78.1% for 5x5 and 9×9 blurring filters respectively on Bosphorus dataset. Even for the UMBDB dataset, subspace discriminant ensemble outperforms other models with 77.2% and 76.5% accuracy for 5×5 and 9×9 blurring filters.

4.3.3 Effect of Sharpening on Models

Table 5 shows sharpening attack on face images of Bosphorus as well as UMBDB dataset in two parts. Comparing model accuracy between ten variants of SVM, KNN and discriminant analysis have been represented for both datasets.

Subspace discriminant ensemble model outperforms others with 85.5% and 84.8% accuracy with 50×50 and 100×100 size image for Bosphorus dataset face recognition. Similarly, for the UMBDB dataset, again subspace discriminant ensemble model outperforms other models with 86.7% accuracy for 50×50 image size and 86.3% accuracy for 100×100 image size.

Table 5 Effect of sharpening on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset		UMBDB dataset			
	50×50	100×100	Rank of model	50×50	100×100	Rank of model
Subspace Discriminant	85.5	84.8	1	86.7	86.3	1
Linear Discriminant	84.8	81.6	2	71.6	84.9	2
Quadratic SVM	81.2	80.8	3	79.1	79.4	4
Cubic SVM	80.9	79.3	4	78.1	77	5
Subspace KNN	79.7	79.3	5	76.6	78.7	7
Fine KNN	79.4	80.1	6	76.6	78.4	6
Linear SVM	78.9	79.3	7	75.5	76.7	8
Weighted KNN	78.3	79.5	8	79.5	78.9	3
Medium Gaussian SVM	77.5	74.6	9	69.8	70.7	10
Medium KNN	75.7	76.6	10	73.4	70.5	9

Table 6 Effect of median filtering on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset		UMBDB dataset			
	50×50	100×100	Rank of model	50×50	100×100	Rank of model
Subspace Discriminant	85.7	84.9	1	90.3	83.6	1
Linear Discriminant	85.3	84.5	2	76.3	66.5	8
Quadratic SVM	83.5	82.8	3	79.1	76.1	5
Cubic SVM	83.5	83.3	4	78.1	76.1	7
Fine KNN	83.4	81.1	5	82.4	77.5	3
Subspace KNN	83.1	81.1	6	82.7	77.9	2
Weighted KNN	82.1	80.7	7	82.4	78.1	4
Linear SVM	81	81.0	8	78.1	71.0	6
Medium Gaussian SVM	79.8	78.5	9	75.9	60.6	9
Medium KNN	79.2	78.7	10	75.5	64.4	10

4.3.4 Effect of Median Filtering on Models

Table 6 presents the median filtering enhancement attacks for Bosphorus and UMBDB datasets. Ten model variants of SVM, KNN and discriminant analysis have been used for accuracy comparisons and ranking of models for both datasets.

Subspace discriminant ensemble model performs best with 85.7% accuracy for 50x50 image size and 84.9% accuracy for 100×100 image size for Bosphorus dataset. For UMBDB dataset, subspace discriminant is best performing with 90.3% accuracy for 50×50 size image and 83.6% accuracy for 100×100 size image.

Table 7 Effect of histogram equalization on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset		UMBDB dataset			
	50×50	100×100	Rank of model	50×50	100×100	Rank of model
Subspace Discriminant	85.3	84.3	1	87.1	82.3	1
Quadratic SVM	82.1	81.4	2	74.6	75.5	7
Cubic SVM	81.9	81.2	3	75.1	74.1	6
Fine KNN	80.9	81.1	4	78.2	76.3	3
Subspace KNN	80.1	80.6	5	79.4	78.2	2
Weighted KNN	79.2	80.7	6	78.2	77.2	4
Linear SVM	79.1	79.5	7	73.9	72.2	8
Medium Gaussian SVM	77.5	76.8	8	76.7	77.7	5
Medium KNN	76.1	77.9	9	70.5	68.8	9
Cubic KNN	75.1	77.0	10	68.6	60.9	10

4.3.5 Effect of Histogram Equalization on Models

Table 7 shows histogram equalization image enhancement attack results for Bosphorus as well as UMBDB face dataset. There are ten variants of machine learning models from class of SVM, KNN and discriminant analysis.

In the case of the Bosphorus face dataset, the subspace discriminant ensemble model is outperforming other models with 85.3% and 84.3% face recognition accuracy for 50×50 and 100×100 image size. In the UMBDB face dataset, the subspace discriminant ensemble model is outperforming other model variants with 87.1% accuracy for 50×50 image size and 82.3% accuracy for 100×100 image size.

4.3.6 Experimentation 2: Effect of Geometric Attacks on Machine Learning Based FR Systems

Rotation, cropping and resizing attacks have been performed under this section. Results are as follows:

4.3.7 Effect of Rotation on Models

Table 8 presents the rotation attacks on Bosphorus and UMBDB face datasets. In the case of Bosphorus face dataset, subspace discriminant ensemble model is holding rank 1 with 85.6% accuracy for 90° rotations, 84.8% accuracy for 180° rotations and 84.2% accuracy for 270° rotations. In case of UMBDB face dataset, subspace discriminant ensemble model is holding rank 1 with 83.5% accuracy for 90 ° rotations, 83.0% accuracy for 180° rotations and 83.9% accuracy for 270° rotations.

In this paper, only 90° variants have been studied for image rotation purposes. Accuracies of machine learning models are not varying much with the variation of the angle of rotation. It is believed that if the angle of rotation is acute, the accuracy of rotated faces will drop compared to 90° variations. Acute angle image rotation based face recognition would be included in future work.

Table 8 Effect of rotation on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset			UMBDB dataset				
	90°	180°	270°	Rank of model	90°	180°	270°	Rank of model
Subspace Discriminant	85.6	84.8	84.2	1	83.5	83.0	83.9	1
Cubic SVM	82.3	81.6	81.6	2	74.6	74.1	74.8	5
Quadratic SVM	81.9	81.3	81.3	3	74.6	73.4	75.5	4
Subspace KNN	81.0	80.7	80.3	4	74.8	73.9	74.1	3
Fine KNN	80.6	80.5	80.3	5	74.1	72.9	73.1	7
Weighted KNN	79.8	79.7	79.3	6	74.8	75.3	75.1	2
Linear SVM	78.9	79.0	79.0	7	73.9	72.4	73.9	8
Medium Gaussian SVM	77.7	77.1	76.8	8	74.1	72.9	72.9	6
Medium KNN	76.1	75.9	75.7	9	64.3	63.5	63.3	10
Cosine KNN	73.9	74.6	74.1	10	66.9	65.5	65.7	9

Table 9 Effect of cropping on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset				UMBDB dataset			
	Right 25%	Right 50%	Right 75%	Rank of model	Right 25%	Right 50%	Right 75%	Rank of model
Subspace Discriminant	78.3	82.5	88.1	1	70.1	84.2	83.1	1
Subspace KNN	77.8	81	84	2	66.5	74.1	71.6	5
Quadratic SVM	76.4	82.6	83.8	3	62.6	71.6	72.7	3
Cubic SVM	76.1	82.5	83.7	4	63.7	71.9	72.3	4
Fine KNN	76.4	79.8	83.2	5	65.5	72.7	71.2	6
Linear SVM	71	80.8	82.4	6	59	68.7	70.1	7
Weighted KNN	75.1	78.6	81.8	7	62.9	72.7	75.5	2
Medium Gaussian SVM	72.3	77.2	80.5	8	60.1	65.1	64.7	10
Medium KNN	73.1	75.3	78.9	9	52.5	64.7	67.3	9
Cubic KNN	69.2	73.5	75.9	10	51.8	65.5	68	8

4.3.8 Effect of Cropping on Models

Table 9 shows the cropping attack on Bosphorus dataset faces as well as UMBDB dataset faces. Three variants of cropping have been tested with variants of machine learning models. In case of Bosphorus faces, recognition accuracy is 78.3% for right 25% of the image cropped, 82.5% for right 50% of the image cropped, and 88.1% for right 75% of the image cropped respectively by using subspace discriminant ensemble model.

In the UMBDB face dataset, the best accuracy has been achieved by subspace discriminant ensemble model with 70.1% recognition accuracy for right 25% cropped image, 84.2% accuracy for right 50% cropped image and 83.1% accuracy for right 75% cropped image.

It is noteworthy from Table 9, the accuracy of face recognition is increasing when the cropped image is covering more percentage of the face.

4.3.9 Effect of Resize on Models

Table 10 presents the image resize attack on Bosphorus face dataset as well as UMBDB face dataset. In the Bosphorus dataset, the best performing model is a subspace discriminant ensemble model with 85.5% accuracy for 50×50 image size and 85.3% accuracy for 100×100 image size. In the case of UMBDB dataset, subspace discriminant ensemble model has outperformed all other models by achieving 88% accuracy for 50×50 size face images and 87.8% accuracy for 100×100 size face images.

Generally, face recognition accuracy drops when an image is resized from a smaller size to bigger due to interpolation. In Table 10, the accuracy of 50×50 and 100×100 size image are both at par rather than expected difference in them. The reason behind less accuracy-difference is that both times, the image's resizing was done from a larger original image, rather than resizing 50×50 image to a size of 100×100.

Table 10 Effect of resizing on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset			UMBDB dataset		
	50×50	100×100	Rank of model	50×50	100×100	Rank of model
Subspace Discriminant	85.5	85.3	1	88.0	87.8	1
Quadratic SVM	82.1	81.8	2	76.5	79.6	6
Cubic SVM	81.9	81.8	3	76.7	78.4	5
Subspace KNN	81.0	80.4	4	79.4	80.3	3
Fine KNN	80.9	80.6	5	78.2	80.3	4
Linear SVM	79.1	79.8	6	73.9	76.7	8
Weighted KNN	79.0	80.3	7	79.9	83.0	2
Medium Gaussian SVM	77.8	77.6	8	74.8	80.3	7
Medium KNN	75.7	77.8	9	71.9	76.0	9
Cosine KNN	74.3	76.7	10	70.0	73.4	10

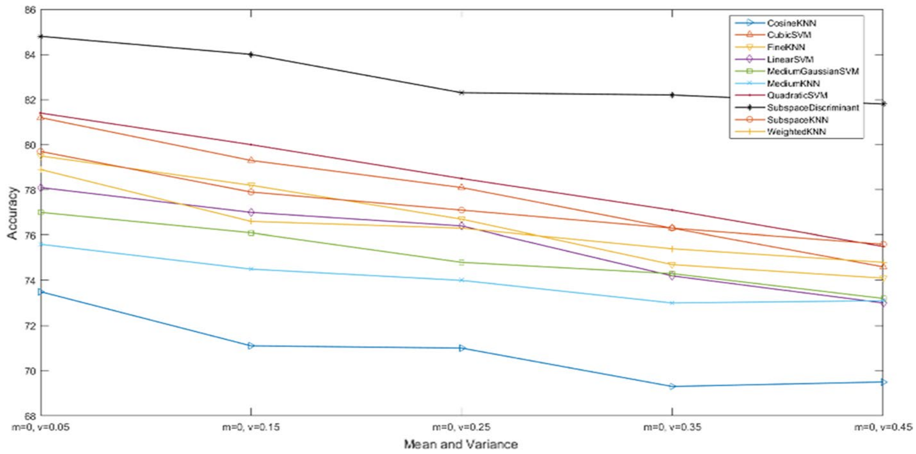

Fig. 5 Effect of Gaussian noise on models with Bosphorus dataset

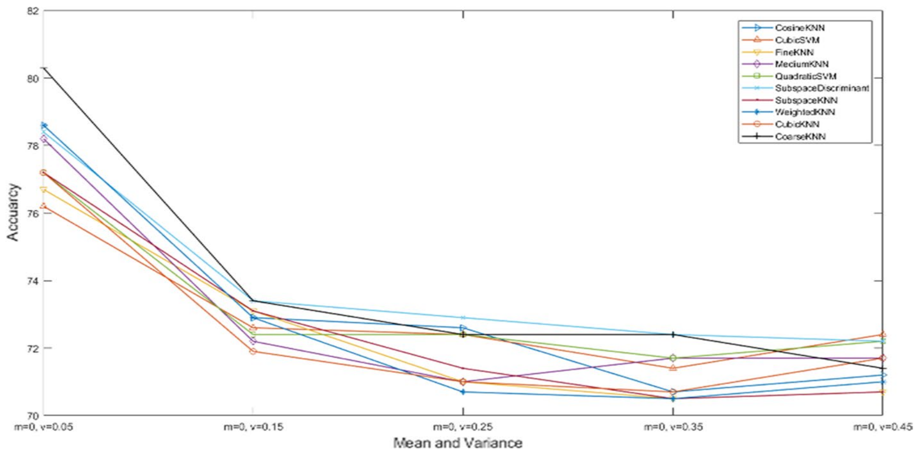

Fig. 6 Effect of Gaussian noise on models with UMBDB dataset

4.3.10 Experimentation 3: Effect of Noise attacks on Machine Learning Based FR Systems

Gaussian, speckle and poisson noise attacks have been implemented on the models in this sub-section. Results are as follows

4.3.11 Effect of Gaussian Attack on Models

Figures 5 and 6 show the graphical representations of the accuracy performances of ten different KNN, SVM, and discriminant analysis variations. Both figures show five Gaussian

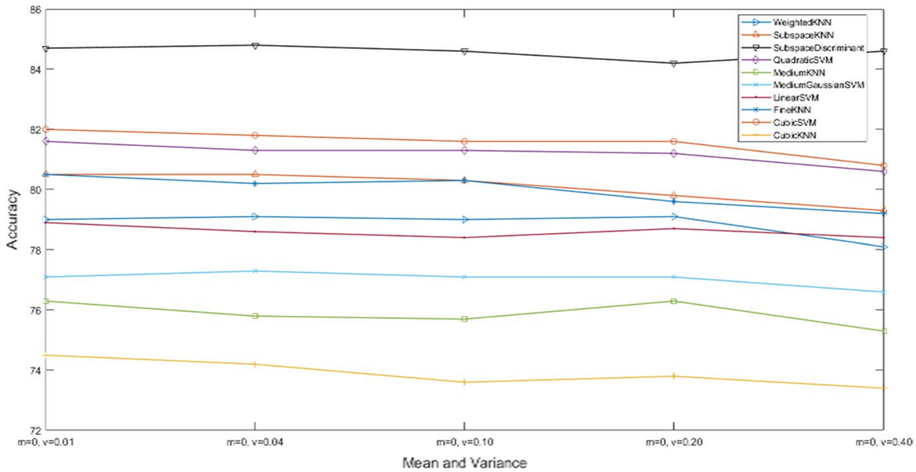

Fig. 7 Effect of speckle noise on models with Bosphorus dataset

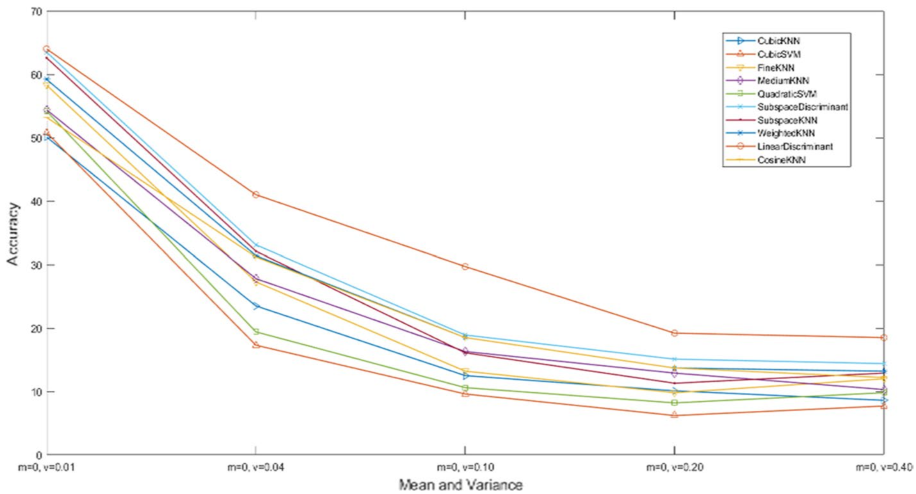

Fig. 8 Effect of speckle noise on models with UMBDB dataset

noise variations with mean 0 for each and variance as 0.05, 0.15, 0.25, 0.35 and 0.45 respectively.

In Fig. 5, Bosphorus face dataset under Gaussian noise attack, subspace discriminant ensemble model outperformed other models with the highest accuracy of 84.8% for v=0.05. In Fig. 6, UMBDB face dataset, coarse KNN model outperforms other models with an accuracy of 80.4% for v=0.05 in face recognition accuracies. Accuracy decreases gradually as the variance of Gaussian noise is increased.

Table 11 Effect of Poisson noise on models with Bosphorus and UMBDB datasets

Machine learning model	Bosphorus dataset			UMBDB dataset		
	50×50	100×100	Rank of model	50×50	100×100	Rank of model
Subspace Discriminant	78.6	73.8	1	71.9	63.3	1
Linear Discriminant	77.9	60.1	2	65.5	64	5
Medium Gaussian SVM	67.5	62.1	3	62.2	55.9	8
Quadratic SVM	66.8	68.8	4	64.7	52.5	6
Cubic SVM	66.5	66.9	5	62.6	35.3	7
Weighted KNN	65.2	71.4	6	69.4	60.7	2
Linear SVM	65	66.5	7	52.4	23.7	10
Medium KNN	63.2	69.5	8	65.8	54.7	4
Subspace KNN	63.1	73.8	9	68.7	65.5	3
Cubic KNN	60.7	67.2	10	62.2	52.3	9

4.3.12 Effect of Speckle Attack on Models

Figures 7 and 8 show the graphical representations of the accuracy performances of ten different KNN, SVM, and discriminant analysis variations. Both figures show five variations of Speckle noise with mean 0 for each and variance as 0.01, 0.04, 0.10, 0.20 and 0.40 respectively.

Figure 7, Bosphorus face dataset under Speckle noise attack, subspace discriminant ensemble model outperforms other models with the highest accuracy of 84.8% for v=0.04. In Fig. 8, for the UMBDB face dataset, linear discriminant model outperforms other models with accuracy of 64% for v=0.01 in face recognition. Accuracy decreases gradually as the variance of Gaussian noise is increased.

4.3.13 Effect of Poisson Attack on Models

Table 11 presents the Poisson noise attack on face database of Bosphorus and UMBDB. Poisson noise attack has been performed with two different sizes of the images.

In the case of Bosphorus, the best performing model is subspace discriminant ensemble model with 78.6% accuracy for 50×50 image size and 73.8% accuracy for 100×100 image size. In the case of UMBDB, subspace discriminant ensemble model has outperformed other models by achieving 71.9% accuracy for 50×50 size face images and 63.3% accuracy for 100×100 size face images.

It can be concluded that subspace discriminant ensemble model best handled 95% cases of image processing attacks trained and tested for face recognition system accurately.

Fig. 9 Visual Verification of Enhancement Attacks on Face Recognition System

5 Visual Verification of Image Attacks Invariant Face Recognition System

This section shows the input and output of all the image processing attacks on face recognition system visually. Three sub-sections have been made to show different image attacks belonging to enhancement, geometric and noise attacks, respectively.

5.1 Visual Verification of Enhancement Attacks on Face Recognition System

Figure 9 shows the visual input and output for different enhancement attacks viz. blurring, histogram equalization, median filter and sharpening. Blurring has been shown with 5×5 and 9×9 blur filter as attack in input. Histogram equalization, median filter and sharpening

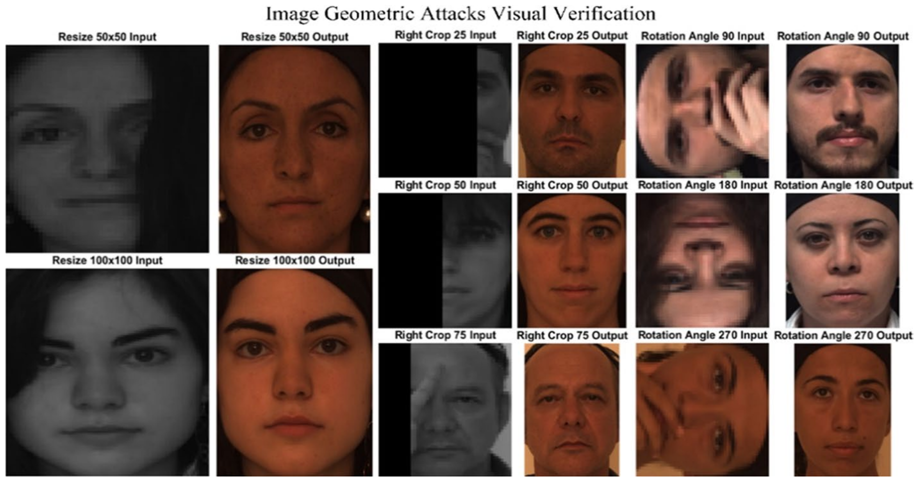

Fig. 10 Visual Verification of Geometric Attacks on Face Recognition System

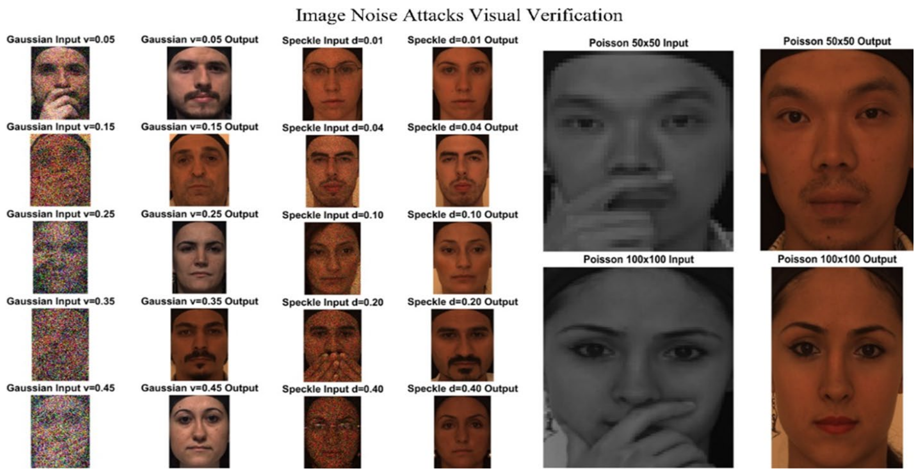

Fig. 11 Visual Verification of Noise Attacks on Face Recognition System

attack have been visually verified with inputs of 50×50 and 100×100 image sizes. All the inputs have been selected randomly out of occluded faces.

5.2 Visual Verification of Geometric Attacks on Face Recognition System

Figure 10 shows the visual input and output for different geometric attacks viz. resize, cropping and rotation. Resize attack has been demonstrated with 50x50 and 100x100 attacks. Cropping is shown with right 25%, right 50% and right 75% area cropped in input. Rotation is demonstrated with 90°, 180° and 270° anticlockwise rotations. All the inputs have been taken out of occluded faces randomly.

5.3 Visual Verification of Noise Attacks on Face Recognition System

Figure 11 shows the visual input and output for different noise attacks viz. Gaussian, Speckle and Poisson for visual verification.

Gaussian noise attack has been shown with five variations of mean and variance viz. (0,0.05), (0,0.15), (0,0.25), (0,0.35) and (0,0.45). Speckle noise attack has been shown with five variations of density viz. d = 0.01, d = 0.04, d = 0.10, d = 0.20, and d = 0.40. Poisson noise attack has been shown with image sizes 50x50 and 100x100. All the inputs with occlusion have been chosen randomly.

It can be cross-validated from Figs. 9, 10, and 11 that the face recognition system is invariant of image processing attacks built by training of various machine learning models. It can also be verified that all the test cases in visual verification have an occlusion in the image.

6 Conclusion

This paper presents the face recognition under different image processing attacks in great detail. Pseudo codes of all attacks have been given along with the time complexities of each attack. The mathematical of the machine learning algorithms, experimental setup with parameters initialization, and experimental results in extensive empirical form has been provided. Visual verification of image attacks is an attempt to demonstrate attacks invariant face recognition system. Ten image processing attacks viz. blurring, histogram equalization, sharpening, median filtering, resize, cropping, rotation, Gaussian noise, Speckle noise and Poisson noise have been discussed in this paper. All the attacks implemented done have used quantized-HOG features, hence compressing the original features.

This research is limited to two-dimensional face recognition systems. Work can be extended for three-dimensional face recognition. How image processing attacks work on voxel information and meshes would be an interesting research to work up on. An effort was made to extend this work on depth images or 2.5D images of the face but results were bad and were not included into this research. This work has an application in captcha-based recognition where these attacks are commonly used for objects identification.

In the last section, visual verification has been presented showcasing the robustness of the image processing attacks invariant face recognition system. In future, we intend to extend the current work to expression and occlusion identification, invariant of image processing attacks using deep learning techniques.

References

1. Hsu, H. J., & Chen, K. T. (2015) Face recognition on drones: Issues and limitations. In Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use; 39-44.
2. S. Gupta, M. K. Markey and A. C. Bovik. (2007) Advancing the state of the art in 3D human face recognition. SPIE Newsroom.
3. Sharif, M., Bhagavatula, S., Bauer, L., et al. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *SIGSAC Conference on Computer and Communications Security, 10*(1145/2976749), 2978392.

4. Akhtar, Z., & Foresti, G. L. (2016). Face spoof attack recognition using discriminative image patches. *Journal of Electrical and Computer Engineering*. https://doi.org/10.1155/2016/4721849.
5. Raghavendra, R., Raja, K. B., & Busch, C. (2015). Presentation attack detection for face recognition using light field camera. *IEEE Transactions on Image Processing., 24*(3), 1060–1075.
6. Chen, X., Liu, C., Li, B., Lu, K., & Song, D. (2017) Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526
7. Sharma, S., & Kumar, V. (2018). Performance evaluation of 2D face recognition techniques under image processing attacks. *Modern Physics Letters B*. https://doi.org/10.1142/S0217984918502123.
8. Zangeneh, Erfan, & Moradi, Aref. (2018). Facial expression recognition by using differential geometric features. *The Imaging Science Journal*. https://doi.org/10.1080/13682199.2018.1509176.
9. Ahmad, Fawad, Khan, Asif, Islam, Ihtesham Ul, et al. (2017). Illumination normalization using independent component analysis and filtering. *The Imaging Science Journal., 65*(5), 308–313. https://doi.org/10.1080/13682199.2017.1338815.
10. Hsia, S.-C., Chen, C.-J., & Yang, W.-C. (2016). Improvement of face recognition using light compensation technique on real-time imaging. *The Imaging Science Journal., 64*(6), 334–340. https://doi.org/10.1080/13682199.2016.1219117.
11. Parubochyi, Vitalius, & Shuwar, Roman. (2018). Fast self-quotient image method for lighting normalization based on modified Gaussian filter kernel. *The Imaging Science Journal*. https://doi.org/10.1080/13682199.2018.1517857.
12. Sharma, R., & Patterh, M. S. (2015). A broad review about face recognition – feature extraction and recognition techniques. *The Imaging Science Journal., 63*(7), 361–377. https://doi.org/10.1179/1743131X14Y.0000000071.
13. Kutter, M., & Petitcolas, F. A. (1999). Fair benchmark for image watermarking systems in security and watermarking of multimedia contents. *International Society for Optics and Photonics, 3657,* 226–240.
14. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning., 20*(3), 273–297.
15. Ruiz, E. V. (1986). An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters, 4*(3), 145–157.
16. Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies, 27*(3), 221–234.
17. Mika, S., Ratsch, G., Weston, J., et al. (1999) Fisher discriminant analysis with kernels. In Neural networks for signal processing IX. Proceedings of the 1999 IEEE signal processing society workshop. 41-48. IEEE.
18. Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning, 40*(2), 139–157.
19. Dietterich, T. G. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems* (pp. 1–15). Berlin, Heidelberg: Springer.
20. Savran, A., Alyüz, N., Dibeklioğlu, H., et al. (2008). Bosphorus database for 3D face analysis. *European Workshop on Biometrics and Identity Management* (pp. 47–56). Berlin, Heidelberg: Springer.
21. Colombo, A., Cusano, C., & Schettini, R. (2011) UMB-DB: A database of partially occluded 3D faces. IEEE International Conference onComputer Vision Workshops (ICCV Workshops). 2113-2119. IEEE.
22. Nicholson, S. W., & Bennett, T. B. (2009). Transparent practices: primary and secondary data in business ethics dissertations. *Journal of business ethics, 84*(3), 417–425.
23. Patil, H., Kothari, A., & Bhurchandi, K. (2015). 3-D face recognition: features, databases, algorithms and challenges. *Artificial Intelligence Review, 44*(3), 393–441.
24. Dalal, N., & Triggs, B. (2005) Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR. 1: 886-893. IEEE.
25. Bay, H., Tuytelaars, T., & Gool, L. (2006). Surf: Speeded up robust features. *European conference on computer vision* (pp. 404–417). Berlin, Heidelberg: Springer.
26. Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12,* 2037–2041.
27. Mita, T., Kaneko, T., & Hori, O. (2005) Joint haar-like features for face detection. Tenth IEEE International Conference on Computer Vision ICCV. 2: 1619-1626. IEEE.
28. Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics, 3*(6), 610–621.
29. Danielsson, P. E. (1980). Euclidean distance mapping. *Computer Graphics and image processing, 14*(3), 227–248.
30. Kruskal, J. B. (1964). Nonmetric multidimensional scaling: a numerical method. *Psychometrika, 29*(2), 115–129.

31. De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems, 50*(1), 1–18.
32. Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science, 240*(4857), 1285–1293.
33. Patidar, P., Gupta, M., Srivastava, S., & Nagawat, A. K. (2010). Image de-noising by various filters for different noise. *International journal of computer applications., 9*(4), 45–50.
34. http://www.nptel.ac.in/courses/117104069/chapter_8/8_32.html
35. Lin, C. Y., Wu, M., Bloom, J. A., et al. (2001). Rotation, scale, and translation resilient watermarking for images. *IEEE Transactions on image processing., 10*(5), 767–782.
36. Gershon, R. (1985). Aspects of perception and computation in color vision. *Computer vision, graphics, and image processing., 32*(2), 244–277.
37. Martucci, S. A. (1995). Image resizing in the discrete cosine transform domain. Proceedings on International Conference on Image Processing. 2: 244-247. IEEE.
38. Mather, P., & Tso, B. (2016) Classification methods for remotely sensed data. CRC press.
39. Benzarti, F., & Amiri, H. (2013) Speckle noise reduction in medical ultrasound images. arXiv preprint arXiv:1712.05526arXiv:1712.05526
40. Chandra, E., & Kanagalakshmi, K. (2011) Cancelable biometric template generation and protection schemes: A review. 3rd International Conference on Electronics Computer Technology (ICECT). 5: 15-20. IEEE.
41. Xu, Y., Zhu, Q., Fan, Z., et al. (2013). Coarse to fine K nearest neighbor classifier. *Pattern Recognition Letters, 34*(9), 980–986.
42. Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., et al. (2014) Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. arXiv preprint arXiv:1409.0919
43. Nguyen, H. V., & Cosine Bai, L. (2010). similarity metric learning for face verification. *Asian conference on computer vision* (pp. 709–720). Berlin, Heidelberg: Springer.
44. Goldberger, J., Hinton, G. E., Roweis, S. T., et al. (2005). Neighbourhood components analysis. *Advances in neural information processing systems, 17,* 513–520.
45. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM computing surveys (CSUR), 31*(3), 264–323.
46. Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery, 2*(2), 121–167.
47. Schittkowski, K. (2005). Optimal parameter selection in support vector machines. *Journal of Industrial and Management Optimization, 1*(4), 465.
48. Souza, C. R. (2010) Kernel functions for machine learning applications. Creative Commons Attribution-Noncommercial-Share Alike.3: 29.
49. Friedrichs, F., & Igel, C. (2005). Evolutionary tuning of multiple SVM parameters. *Neurocomputing, 64,* 107–117.
50. Huang, J., Yuen, P. C., Chen, W. S., et al. (2007). Choosing parameters of kernel subspace LDA for recognition of face images under pose and illumination variations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B Cybernetics, 37*(4), 847–862.
51. Binol, H., Cukur, H., & Bal, A. (2016). A supervised discriminant subspaces-based ensemble learning for binary classification. *International Journal of Advanced Computer Research, 6*(27), 209.
52. Sharma, S., & Kumar, V. (2020). Voxel-based 3D occlusion-invariant face recognition using game theory and simulated annealing. *Multimedia Tools and Applications, 79*(35), 26517–26547.
53. Sharma, S., & Kumar, V. (2020). Voxel-based 3D face reconstruction and its application to face recognition using sequential deep learning. *Multimedia Tools and Applications, 79,* 25–26.
54. Sharma, S., & Kumar, V. (2021). 3D landmark-based face restoration for recognition using variational autoencoder and triplet loss. *IET Biometrics, 10*(1), 87–98.
55. Goswami, G., Ratha, N., Agarwal, A., Singh, R. and Vatsa, M., (2018). Unravelling robustness of deep learning based face recognition against adversarial attacks. arXiv preprint arXiv:1803.00401
56. Zhu, Z.A., Lu, Y.Z. and Chiang, C.K., (2019), September. Generating adversarial examples by makeup attacks on face recognition. In 2019 IEEE International Conference on Image Processing (ICIP) (pp. 2516-2520). IEEE.
57. Goswami, G., Agarwal, A., Ratha, N., Singh, R., & Vatsa, M. (2019). Detecting and mitigating adversarial perturbations for robust face recognition. *International Journal of Computer Vision, 127*(6–7), 719–742.
58. Kotwal, K., Mostaani, Z., & Marcel, S. (2019). Detection of age-induced makeup attacks on face recognition systems using multi-layer deep features. *IEEE Transactions on Biometrics, Behavior, and Identity Science, 2*(1), 15–25.

59. Nguyen, D. T., Pham, T. D., Baek, N. R., & Park, K. R. (2018). Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors. *Sensors, 18*(3), 699.
60. Weitz, K., Hassan, T., Schmid, U. and Garbas, J.U., (2019) Deep-learned faces of pain and emotions Elucidating the differences of facial expressions with the help of explainable AI methods. tm-Technisches Messen, 86(7-8): 404-412.
61. Williford, J.R., May, B.B. and Byrne, J., (2020) August. Explainable Face Recognition. In European Conference on Computer Vision (pp. 248-263). Springer, Cham.
62. Wang, M. and Deng, W., 2020. Mitigating Bias in Face Recognition Using Skewness-Aware Reinforcement Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9322-9331).
63. Duong, C.N., Luu, K., Quach, K.G., Nguyen, N., Patterson, E., Bui, T.D. and Le, N., (2019). Automatic face aging in videos via deep reinforcement learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 10013-10022).

Sahil Sharma received his BTech in Computer Science and Engineering from Punjab Technical University in 2012, ME in Information Security from Thapar University in 2015. He has worked on machine learning project as Junior Research Fellow in Indian Institute of Technology, Mandi during the period of March – May 2015. Since January 2016, he is currently pursuing his PhD in the field of computer vision from Thapar Institute of Engineering and Technology, India. He has worked as a Lecturer at Computer Science and Engineering Department of Thapar Institute of Engineering and Technology during January 2018 – January 2020. Since January 2020, he is an assistant professor in the same institute. His main research interests include computer vision, image processing, and machine learning.

Vijay Kumar received his BTech in Information Technology from Kurukshetra University, Kurukshetra in 2005. He received his MTech in Computer Science and Engineering from Guru Jambeshwer University of Science and Technology, in 2008. He received his PhD in Computer Engineering from National Institute of Technology, in 2015. He has worked as an Assistant Professor at JCDM College of Engineering during 2008–2014, at Manipal University during 2014–2015, at Thapar University during 2015–2019, and since July 2019 at Computer Science and Engineering Department, National Institute of Technology, Hamirpur. His main research interests include data clustering, metaheuristic techniques, and pattern recognition.