Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.026961
Article

Tech Science Press

An Optimised Defensive Technique to Recognize Adversarial Iris Images Using Curvelet Transform

K. Meenakshi[1,*] and G. Maragatham[2]

[1]School of Computing, Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, India
[2]School of Computing, Department of Computational Intelligence, SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, India
*Corresponding Author: K. Meenakshi. Email: meenaksk@srmist.edu.in

Received: 07 January 2022; Accepted: 24 February 2022

Abstract: Deep Learning is one of the most popular computer science techniques, with applications in natural language processing, image processing, pattern identification, and various other fields. Despite the success of these deep learning algorithms in multiple scenarios, such as spam detection, malware detection, object detection and tracking, face recognition, and automatic driving, these algorithms and their associated training data are rather vulnerable to numerous security threats. These threats ultimately result in significant performance degradation. Moreover, the supervised based learning models are affected by manipulated data known as adversarial examples, which are images with a particular level of noise that is invisible to humans. Adversarial inputs are introduced to purposefully confuse a neural network, restricting its use in sensitive application areas such as biometrics applications. In this paper, an optimized defending approach is proposed to recognize the adversarial iris examples efficiently. The Curvelet Transform Denoising method is used in this defense strategy, which examines every subband of the adversarial images and reproduces the image that has been changed by the attacker. The salient iris features are retrieved from the reconstructed iris image by using a pre-trained Convolutional Neural Network model (VGG 16) followed by Multiclass classification. The classification is performed by using Support Vector Machine (SVM) which uses Particle Swarm Optimization method (PSO-SVM). The proposed system is tested when classifying the adversarial iris images affected by various adversarial attacks such as FGSM, iGSM, and Deepfool methods. An experimental result on benchmark iris dataset, namely IITD, produces excellent outcomes with the highest accuracy of 95.8% on average.

Keywords: Adversarial attacks; biometrics; curvelet transform; CNN; particle swarm optimization; adversarial iris recognition

1 Introduction

Person in many machine learning tasks, Deep Neural Networks (DNN) have grown increasingly successful and popular. They've had great success in various applications like image recognition, text synthesis, speech recognition, etc. They are capable of recognizing objects with near-human accuracy in the image processing domain [1]. Despite the fact that Deep Neural Network models attain state-of-the-art performance in many situations, they have drawbacks that restrict their usage in critical applications. In adversarial situations, the vulnerability remains a problem: Without a doubt, it is relatively simple for an attacker to change a model's output by manipulating its input. The manipulated inputs are called adversarial examples that an attacker intentionally introduces to a DNN classification model, especially a deep learning based biometrics system. It is known as an adversarial attack, and there are two types (i) Black box attack (ii) White box attack [2].

Recently few research studies have been published to find countermeasures to protect deep neural networks from the threat of adversarial examples. The defending mechanism is categorized into three types. (A) Gradient Masking–Most adversarial attacks are performed by changing the classifier's gradient information, so this defending mechanism hides or masks the gradient values, which results in the attacking mechanism failing [3]. (B) Few studies [4] illustrate how to build a robust classifier that can accurately classify adversarial examples using robust optimization. (C) Adversary detection: Before introducing training data to deep learning models, the techniques try to determine whether it is normal or adversarial data. Thus, it can be viewed as a strategy of avoiding adversarial examples. DNN's robustness to adversarial examples is enhanced by these strategies [5]. Biometrics aim to determine individuals using physical traits such as fingerprints, iris, face, retina, and so on. Iris is a distinctive pattern that varies from person to person. Furthermore, although visible from the outside, the iris is a well-protected vital organ that remains remarkably unchanged throughout time. As a result, among the various biometric features, the most effective person identifying trait is iris images. Consequently, iris classification algorithms are used in many identification and security applications, and adversarial examples attempt to confuse these applications. In turn, this poses a serious threat to security systems [6]. It is an important research direction to develop a proper defensive technique to protect the iris recognition system. In this paper, the analysis of various adversarial attack mechanisms on iris images is studied, and further the efficient defensive mechanism is proposed to protect the iris recognition system.

This study proposes the following contributions:

a) Curvelet transform based Image denoising: Curvelet transform approach is used to reconstruct an adversarial iris example into a denoised image in an efficient way.
b) DNN based Feature Extractor: The extraction of required essential key features from the data determines the feature extractor's efficiency. The pre-trained CNN model termed as VGG16 is used to extract all major important features.
c) Classification: PSO-SVM is used for multiclass classification, which identifies the adversarial image efficiently with optimised SVM parameters.
d) The proposed framework is resistant to adversarial attacks, and its results are analysed with existing state of the art strategies.

The remainder of the paper is organized in the following manner. In the second section, related works are discussed. The techniques utilized to create adversarial examples are described in Section 3, and the adversarial detection methodology employed in our investigation is described in Section 4. Section 5 summarizes the results of our experimental effort, whereas Section 6 summarizes our findings and conclusion.

2 Related Works

2.1 Adversarial Attacks

Artificial neural networks based Deep Neural Networks (DNN) has been proven to be resistant to random noise [7], but they are more susceptible to adversarially constructed perturbations [8]. Several papers have tried to explain why DNNs are vulnerable to adversarial samples. Goodfellow et al. [9] proposed an effective basic hypothesis, claiming that low-probability adversarial points are widely dispersed in input data space. So, they claim, any point in image space is near a large number of adversarial points and may be easily manipulated to produce the desired model output. L-BFGS is the name of the adversarial attack method invented by the authors in [10]. It is a computationally expensive technique. To compensate for this limitation-the Fast Gradient Sign Method (FGSM) has been developed. The sign of the gradient of the classification loss with respect to the input sample is used to calculate the perturbation in FGSM attack. Papernot et al. have used a Jacobian matrix of class prediction with reference to the input pixels to lower the attack's computational cost. By estimating the saliency map of the input space, they were able to limit the amount of pixels that needed to be changed during the attack. By iteratively translating input data toward the closest decision boundary, Moosavi-Dezfooli et al. [11] identifies Lp minimum perturbations. The 'boundary tilting' perspective, proposed by Tanay et al., claims that adversarial samples might be located in places where the classification boundary is nearer to the manifolds of training samples [12].

2.2 Adversarial Defensive Techniques

Developing a framework that is resistive to manipulated data, has proven to be a difficult task. It is an on-going challenge. A DNN model's robustness can be improved through adversarial training or model distillation [13]. In order to train robust DL models, regularization strategies have been developed. Yan et al. [14] proposed that the classification objective include a perturbation-based regularizer that penalizes the standard of adversarial perturbations. Additionally, a binary classifier can be trained to determine if an input image data is an original or adversarial image. Yang et al. [15] developed a detection approach, which computes the threshold value of the feature attribution scores. Based on this threshold value, adversarial examples are detected by the trained model. The K-Nearest Neighbours (K-NN) algorithm is used to determine the score based on the intermediate training data set, which measures the confidence of a DNN classification model. This score can subsequently be used to clean out both adversarial examples and genuine mistakes [16]. Various Radial Basis Function kernel Support Vector Machine (RBF-SVM) is used on each deep layer to detect the adversarial example [17]. SafetyNet, [18] a detection framework which used SVM classifiers, was presented by Lu et al. It was using an RBF-SVM to identify regular images from adversarial samples by examining quantized codes derived from ReLUs' outcome. Input reconstruction algorithms have transformed the adversarial samples into original data, allowing them to be classified into their respective classes. By removing the adversarial perturbations, Meng et al. developed a denoising auto-encoder network that can convert adversarial samples into benign ones [19]. Network analysis techniques have evaluated whether an input image infringes a neural network's characteristics or not; based on that, the model determines the adversarial images efficiently [20]. Goswami et al. [21] have proposed a novel framework to detect the adversarial face image in order to protect the facial recognition system. Whereas in their approach, proactive defensive strategy is used to build a robust model. To our understanding, soleymani et al. is the most significant study on detecting adversarial iris image recognition, in which the researchers targeted iris recognition systems in a classification context and established three defensive approaches to determine adversarial image then reconstruct the original input. These defensive approaches rely on wavelet domain denoising of the input samples, which involves analysing every wavelet sub-band and eliminating the ones which are highly changed by the attacker [22,23]. The authors in [24,25] proposed a defensive technique that analyses the wavelet's denoised image and reconstructs it into the original image. Then a DL based classifier is used to determine the adversarial image with high accuracy.

3 Background and Study

3.1 Adversarial Attack

An adversary's common goal is to provide a sample that looks identical to a normal sample, but it should be incorrectly classified by the target model. Consider the input image, *a*, to generate an adversarial image. This can be done by adding minimal perturbation η to a which results in, $\hat{a} = a + \eta$. where â is an adversarial image. A wide range of adversarial approaches have been created in recent years to fool the classifier. Fig. 1 shows the illustration of an adversarial attack in iris classification. In this section the most popular and recent adversarial attacks are explained.

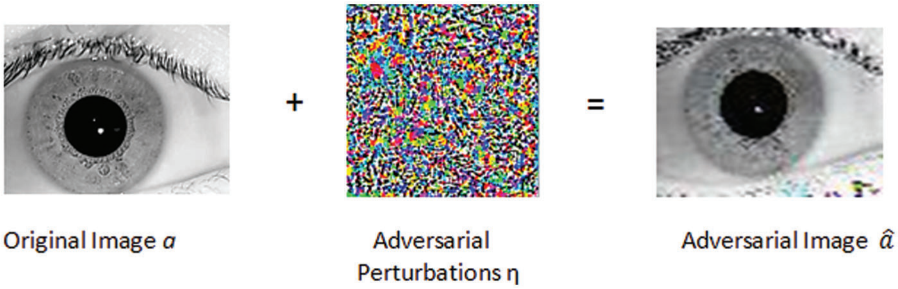

Figure 1: Generation of adversarial iris image

(a) Fast Gradient Sign Method (FGSM): Goodfellow et al. have introduced the Fast Gradient Sign Method and uses the derivative of the classifiers model's loss function based on the input feature vector to create adversarial perturbations. The strategy is to perturb each feature by magnitude □ in the direction of the gradient given a base input, where ε is a parameter which specifies scale of perturbation. The classification model I's loss is represented as:

$$\text{Model's Loss } = J(\varnothing, a, b) \tag{1}$$

where

$\varnothing$ - model parameters, *a* – input, *b* – label of *a*

Using Eq. (1), the adversarial sample generated by this FGSM method is

$$\hat{a} = a + \varepsilon\, sign(\nabla a\; J(\varnothing, a, b)) \tag{2}$$

The success rate of generating adversarial examples are mainly dependent on the perturbation rate (i.e.) ε.

(b) Iterative Gradient Sign Method: The iterative version of the FGSM is the IGSM. Instead of applying adversarial noise with one large perturbation size and clipping all the pixels after each iteration, this method applies FGSM multiple times with modest perturbation sizes to ensure that the results stay in the ε-neighbourhood of the input image *a*. In this method the l_2 normalization method is used, so in each iteration the l_2 version of IGSM moves in the direction of normalized gradient and effective adversarial examples are generated. On the ImageNet data set, it was proven that IGSM's attack was better than the FGSM method [5].

$$\hat{a}_0 = a, \ \ \hat{a}_{(n+1)} = \prod \mathrm{a}, \varepsilon \left(\hat{a}_n + \frac{\nabla a \, J(\emptyset, \hat{a}_n, b)}{||\nabla a \, J(\emptyset, \hat{a}_n, b)||} \right) \tag{3}$$

(c) Deepfool: Deepfool is a non-targeted attack approach for iteratively perturbing an image to build an adversarial example. It calculates the shortest distance between the original input and the Adversarial Attack decision boundary. It uses an iterative technique with a linear approximation to resolve nonlinearity in high dimensions. When the altered image changes the Deep Learning Model's classification, the process ends and the corresponding image is considered as an adversarial image. Tab. 1. summarizes various adversarial example generation techniques and datasets used in the attacks.

Table 1: Adversarial attack techniques

Attack Type	Attack Name	Dataset used
White box approach	Fast Gradient Sign Method (FGSM)	MNIST, CIFAR-10
	Iterative Gradient Sign Method	Imagenet
	Jacobian Saliency Map Attack (JSMA)	MNIST, CIFAR-10
	Deepfool (DF)	MNIST, CIFAR-10, Imagenet
Black Box approach	One Pixel Attack (OPA	CIFAR-10, Imagenet
	Natural GAN (NGAN)	MNIST, Textual Entailment
	Boundary Attack (BA)	MNIST, CIFAR-10, Imagenet
	Greedy Search Algorithm	Textual datasets Trec07p, Yelp,News

3.2 Curvelet Transform

The Curvelet Transform [26] was proposed by Donoho et al. as one type of multi-resolution image analysis tool. It is efficient in detecting singularities and therefore has been widely used in pattern recognition and image processing applications. The wavelet transform succeeds at depicting point discontinuities in the case of one and two-dimensional data (both signal and image), but this transform has difficulties with detecting curve singularities. The standard wavelet transform extracts the features of horizontal, vertical and diagonal orientation of the signal or image. However, images do not always have isotropic scaling, and so for an effective process-it requires the use of other multi-scale representation methods. The Curvelet Transform is appropriate because it was created specifically to depict things that have pattern smoothness. Curvelet Transforms are different from wavelets in that they use an asymmetrical scaling notion, whereas wavelets use an isotropic scaling approach [27].

Curvelet transformations can be implemented in two ways: the first uses Unequally Spaced Fast Fourier Transforms (USFFT), wrapping of specific Fourier samples are selected in the second implementation. The only difference between these two implementations is the geographic locational space used to transform curvelets at each level and direction with respect to curve, and they still generate a curvelet coefficients in a table format having the following fields: ŝ dimension level i.e. scale, direction, and a geographical location value. The wrapping curvelet transform is the quickest technique which is chosen for this proposed work. The steps to acquire the curvelet coefficients are depicted in Fig. 2.

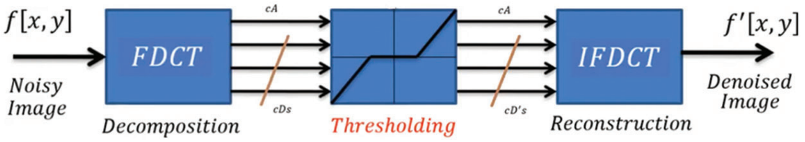

Figure 2: Curvelet transform denoising method

3.3 Deep Convolutional Neural Networks

Deep convolutional neural networks have made numerous ground-breaking advances in the field of image classification. Convolutional Neural Networks (CNN) follow the deep neural network concept, which extracts the high level features of the input data automatically whereas in classical Machine Learning (ML) the hand crafted features are supplied for processing the data [28]. The CNN has multiple functional blocks as layers with trainable parameters. Back propagation technique is used to develop an efficient model to perform complex tasks like image classification, object detection and tracking effectively [29]. The various layers of a CNN model are depicted in Fig. 3. Convolutional block is the initial layer, followed by a pooling layer for dimension reduction. The pooling function has various types like max pooling, average pooling etc. The dense layer is used to allow widespread learning among the neurons that have been activated by the previous levels. The dense layers introduce nonlinearity to the data, allowing it to generate complex mathematical models [30].

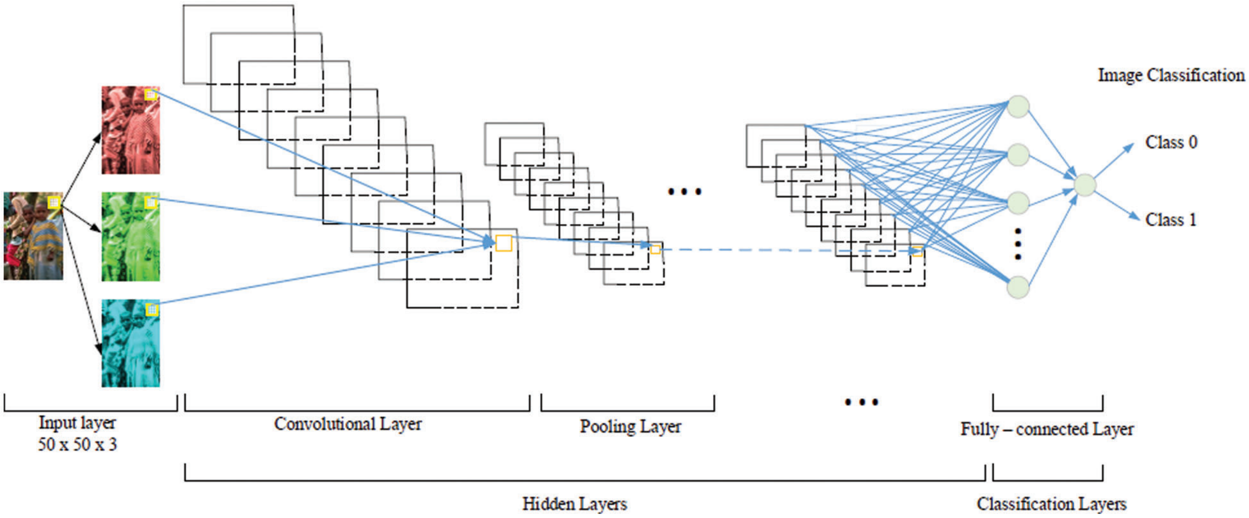

Figure 3: Architecture of CNN for binary classification of images

3.4 Particle Swarm Optimization-Support Vector Machine (PSO-SVM)

The PSO-SVM, a parallel evolutionary computation approach, was introduced to Kennedy et al. [31]. It's an optimised search algorithm based on the social behaviour of flocks of birds looking for food. Each particle, or population member in PSO flies about the multidimensional search space looking for better regions. The leader particle (the best solution) communicates to the other particles to reach the optimum solution.

In this work, the final classification is performed using SVM technique. To optimize the SVM parameters, the PSO method is applied. Fig. 4 visualizes the PSO algorithm to get the best parameters. The particles and the parameters in the PSO algorithm are initialized with random values. In each iteration the velocity and position of each particle is updated based on the best particle value. The equations below are used to update the values:

$$v_i^{t+1} = \omega * v_i^t + c_1 * r_1 * \left(P_{best} - X_i^t\right) + c_2 * r_2 * \left(G_{best} - X_i^t\right) \tag{4}$$

$$X_i^{t+1} = X_i^t + v_i^{t+1} \tag{5}$$

where

v_i^t – Velocity of particle i in t^{th} iteration, ω – Inertia weight, c_1,c_2, r_1, r_2 – PSO Parameters

P_{best}, G_{best} – individual best value, Group best value, X_i^t – Position of particle i in t^{th} iteration

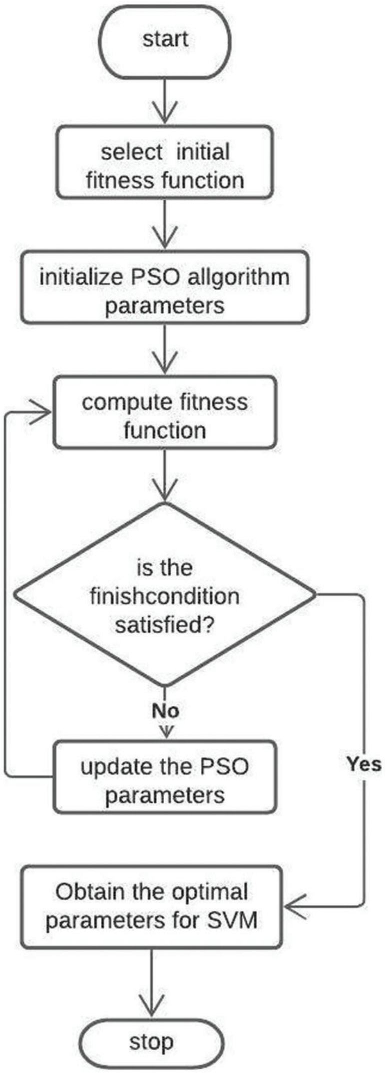
start
select initial fitness function
initialize PSO allgorithm parameters
compute fitness function
is the finishcondition satisfied?
No
update the PSO parameters
Yes
Obtain the optimal parameters for SVM
stop

Figure 4: SVM parameters optimization based on particle swarm optimization

4 Methodology

4.1 Dataset Description

The proposed work utilizes the two benchmark iris datasets which are publicly available. (I) IITD iris database (ii) CASIA-Iris-Interval. Tab. 2 summarizes the specifications of both datasets. In the proposed work, the adversarial attacks like FGSM, iGSM and Deepfool methods are performed on both the IITD and Casia datasets to generate adversarial iris images. These images are vulnerable to iris recognition systems which reduce the accuracy of the model. This proposed methodology is illustrated in Fig. 5. There are four steps. (i) Pre-processing step (ii) Retrieval of reconstructed image using curvelet transform (iii) Feature Extraction step (iv) Classification step. After the preprocessing step, the Curvelet Transform technique is applied to reconstruct the adversarial iris image where the adversary introduced the noise in benign iris images. The features are extracted from reconstructed images by using a Convolutional Neural Network. Finally PSO-SVM technique is used for classification of whether the image is an adversarial image or not. The output of the proposed work has been compared with different state of art models.

Table 2: Details of dataset

Dataset	Number of subjects	Number of images	Image size	Image format	Number of classes
IITD	224	1120	320x240	BMP	224
Casia Iris Interval	249	2639	320x280	JPEG	395

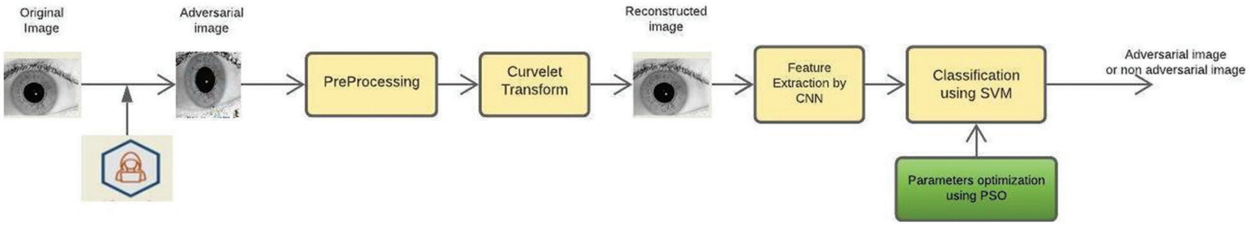

Figure 5: Over all flow of the proposed model

4.2 Preprocessing

Iris segmentation and normalization are carried out in this preprocessing step. The circular Hough transform is used to detect the boundaries of the iris and pupil in the iris segmentation procedure. In segmentation, the first step is to build an edge map by using Canny Edge detection technique. The Canny Edge detection consists of the following steps (a) Noise reduction by smoothing (b) Gradient calculation (c) Non maximum suppression (d) Double thresholding (e) Tracking the edges by Hysteresis. Fig. 6 shows the result of edge detection and segmentation of the adversarial iris image. The rubber sheet model is utilized in the iris normalization procedure. Each point in the localized iris image is represented as polar coordinate (i, θ) from Cartesian format (a,b), with the i value ranges from 0 to 1 and angle (θ) lies in 0 to 2Π. The mapping is performed by the following equations.

$$iris(a,b) \to iris(i,\theta) \qquad (6)$$

$$a(i,\theta) = (1-i)a_p(\theta) + ia_I(\theta) \tag{7}$$

$$b(i,\theta) = (1-i)b_p(\theta) + ib_I(\theta) \tag{8}$$

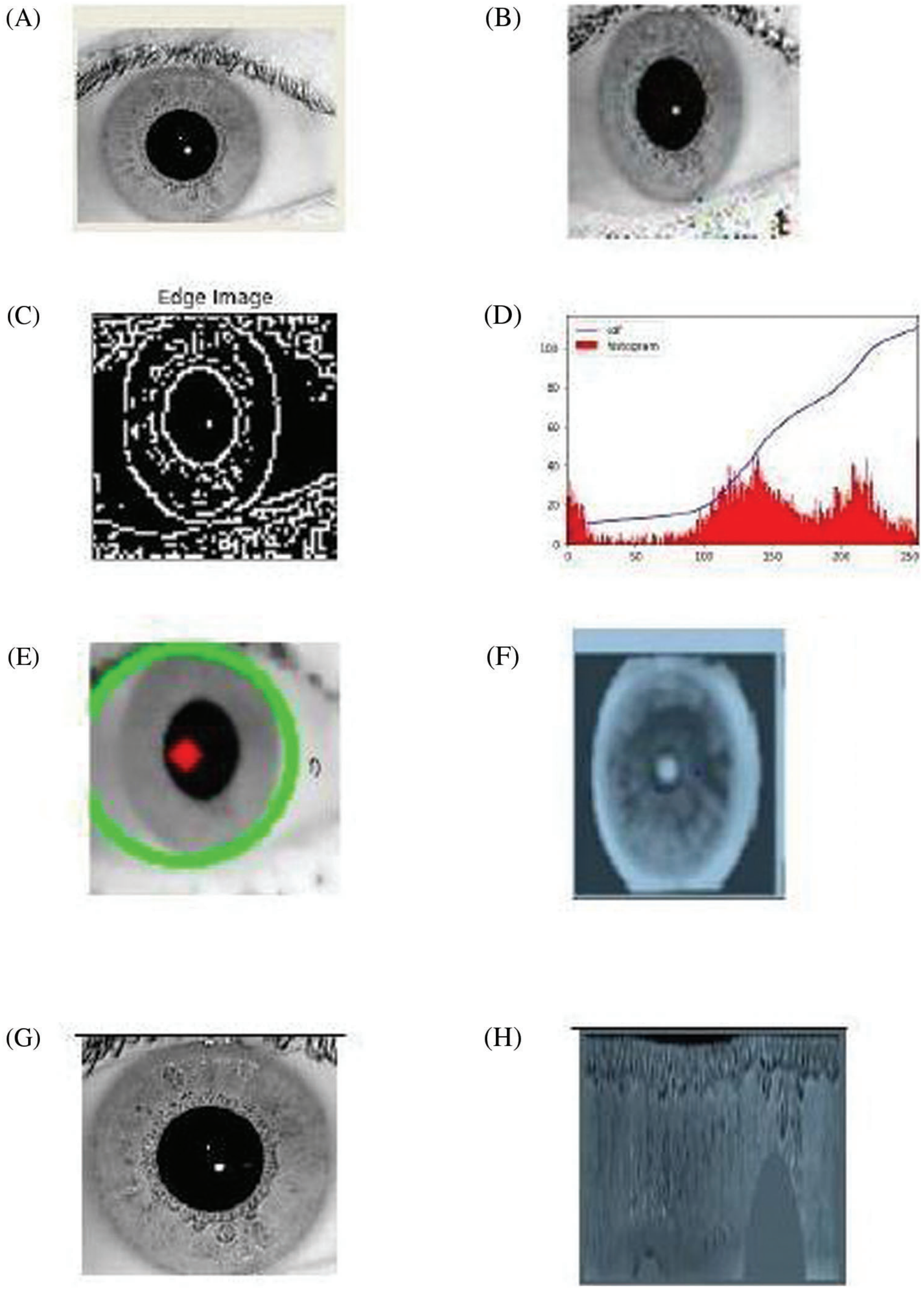

Figure 6: The result of preprocessing stage for (IITD Image 001-1.bmp). (A) Original image (B) FGSM image (C) Edge Detected image (D) Histogram output (E) Iris and pupil boundary detection (F) Histogram equalized image (G) Segmentation output (H) Image after Normalization

where

$iris(a, b)$ - cartesian format of iris region, $iris(i, \theta)$ - Polar format of the iris region.

$a_{p,}\ b_p$ - coordinates of pupil boundaries, $a_{I,}\ b_I$ - coordinates of iris boundaries, Θ - direction

4.3 Reconstruction of Adversarial Iris Image Using Curvelet Transform

The next step is, reconstruct the adversarial iris images by using a multi resolution analysis tool i.e. Curvelet Transform. After the pre-processing step, the Curvelet Transform via wrapping method is applied on the normalized images, in order to get the Curvelet coefficients. After applying the Curvelet Transform the image is analysed in terms of Approximation Curvelet coefficients and Detail curvelet coefficients. These coefficients are derived with different angles and scales. Consider the image size to be 256 × 256 and the angle value in the sub band set to be 16, then the number of curvelet coefficients is 1,84,985. If all coefficients are considered for further steps, then the computation load is too complex. So the IITD iris images are normalized from 320 × 240 to 100 × 100 with the above mentioned parameter values and it results in the coefficient size being 28,098. During the reconstruction step, the approximation coefficients are considered without any change, because they have deeper features of image and the detailed coefficients are chosen by using a soft thresholding technique. The soft thresholding technique produces more visually appealing images compared to hard thresholding. Mean Square Error, Signal to Noise Ratio (SNR) and Peak signal to noise ratio (PSNR) are the metrics used to evaluate the result of this reconstruction of images. It is observed that the curvelet transform reconstructs the original image from the adversarial attack efficiently. The reconstructed iris images are listed in Fig. 7.

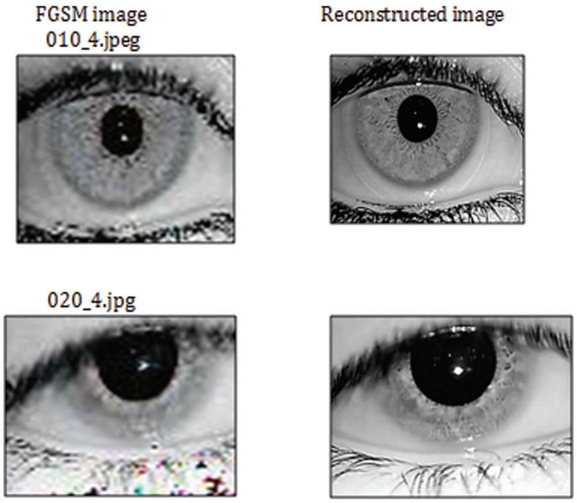

Figure 7: FGSM images (Left) corresponding reconstructed images (Right) - dataset IIID iris dataset

4.4 Feature Extraction

The feature extraction method is carried out with the help of transfer learning concept. VGG 16 is a pre-trained model which is based on Convolutional Neural Network and is used for extracting the features. The VGG-16 architecture is complex and the details of the architecture are shown in Tab. 3. This architecture has 16 layers in which 13 convolutional and 3 fully connected layers are utilized to extract the features. The convolutional layer performs 3 × 3 convolutions with stride size as 1 and same padding. The pooling layers in the architecture has 2 × 2 pooling layers with ŝ stride size of 2 and the kernel size is 3 × 3 for all the layers. The input image size is 224 × 224. The size of the feature map is halved after each pooling layer. The activation function used in these architectures is relu for all the layers except the final classification layer. Before the fully connected layers, the last feature map has 512 channels and is stretched into a vector with 25,088 (7 × 7 × 512) channels. Although the initial VGG-16 networks produced 1,000 classes, we only needed two classes, adversarial iris or not. So using VGG16, we have extracted features alone after that the classification is performed by the PSO SVM model.

Table 3: Layer details of feature extraction model

Layer Name	Feature Map	Size	Stride
Input	1	224 × 224 × 3	–
2X Convolutions	64	224 × 224 × 64	1
Maxpooling	64	112 × 112 × 64	2
2X Convolutions	128	112 × 112 × 128	1
Maxpooling	128	56 × 56 × 128	2
2X Convolutions	256	56 × 56 × 256	1
Maxpooling	256	28 × 28 × 256	2
3X Convolutions	512	28 × 28 × 512	1
Maxpooling	512	14 × 14 × 512	2
3X Convolutions	512	14 × 14 × 512	1
Maxpooling	512	7 × 7 × 512	2
Fully connected 1	–	25088	–
Fully connected 2	–	4096	–
Fully connected 3	–	4096	–
Output	–	1000	–

4.5 Classification by PSO SVM

After feature extraction, the classifier is trained so that the trained model is used to identify the associated label for each input image. Support Vector Machine, Naïve Bayes, Neural Network and Softmax Regression are examples of classifiers that can be used for this purpose. In the proposed work SVM technique has been applied to classify the images as adversarial images or not. The kernel function is a key notion in SVM, as it allows operations to be carried out in the input space rather than the potentially high-dimensional feature space. There are four types of SVM kernel functions that are commonly used: 1. Linear Function 2. Polynomial function 3. Radial basis function 4. Sigmoid function. In our work we have used the RBF kernel SVM for classification. Initial values for the SVM parameters named Gamma and C (penalty) Parameters are defined in the prior step. Later Particle Swarm Optimization (PSO) method is used to find the best parameter values based on the fitness function. Root Mean Square Error (RMSE) is used as a fitness function in our methodology.

5 Experimental Results and Discussion

The proposed methodology has been experimented with two datasets named IITD iris and CASIA-Iris-Interval datasets. The details of the dataset explained in Section 4.1. The flow diagram of proposed work shown in Fig. 5. As in the workflow diagram, the first step is to construct the adversarial examples by using FGSM, iGSM and Deepfool techniques. These attacks can be implemented by using AdversariaLib which is a Python package for evaluating the security of machine learning (ML)-based classifiers in the face of adversarial attacks As discussed in the introduction section, the adversarial images affect the accuracy of the classification model. Tab. 4 shows the result of accuracy of the classifier Model which gets trained using original iris images and adversarial images. The Deep CNN model is applied on the IITD dataset for classification and the accuracy before FGSM attack is 98.01% whereas after the attack it reduces into 90.24%. The same has been represented for iGSM and Deepfool attacks in the Tab. 4. Fig. 8 visualizes the results.

Table 4: Accuracy of Deep CNN before and after the adversarial attack

Attack name accuracy (in %)	FGSM	iGSM	Deepfool
Before attack	98.01	97.6	98
After attack	90.24	86.7	93.83

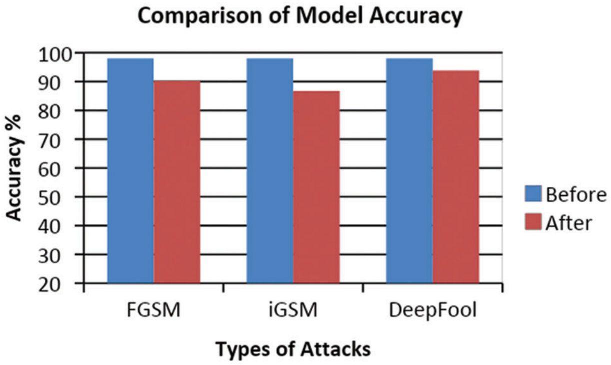

Figure 8: Deep CNN model accuracy before and after adversarial attack – IITD iris dataset

The next step in the proposed methodology is preprocessing of adversarial iris images. Segmentation and normalization are performed to enhance the iris image. Further the preprocessed image is reconstructed by using curvelet transform. Curvelet transforms are designed to handle curves using a small number of coefficients, hence this method handles curve discontinuities well. After performing curvelet transform, the resultant reconstructed images from the adversarial images are shown in Fig. 7. The effectiveness of the curvelet transform in terms of reconstruction are illustrated in the Tab. 5. It shows the metrics of assessing this reconstruction step which produces the promising result.

Table 5: Metrics for assessing the Curvelet Transform - after reconstruction of various attacked adversarial examples

Types of attack	Average SNR (Increase)	Average PSNR (Increase)	MSE (Decrease)
FGSM	10%	8%	83%
iGSM	12%	10%	85%
Deep fool	11%	12%	85%

The SNR value of original to attack image is computed. Similarly the SNR value of original to reconstructed image is computed. The increased value shows that the image is reconstructed properly. Again the decreased value of Mean Square Error (MSE) depict the efficient reconstruction of images. In Tab. 5, the first row shows average SNR value is 10% increased it means that the SNR value of original to attack image and original to reconstructed image is increased by 10% in average. Similarly the PSNR and MSE values are tabulated in the corresponding attacks. The preprocessing step and reconstruction step by using curvelet transform are implemented using MATLAB 2017 on a laptop with Core i7 CPU running at 2.8 GHz.

The pretrained CNN based model VGG16 is used to extract the features from the reconstructed image. The default input size for this model is 224 × 224. In this case we resized the reconstructed image into required size. The model gets trained with the Adam optimizer with the batch size of 32. The Adam optimizer was chosen because of its low weight updates, which can improve learning throughout the training phase. With a certain number of epochs, it eventually converges satisfactorily with the global minimum. For simpler weight updates, smaller learning rates are desirable. In our experiment we have initialized the learning rate is 0.0001. The rectified linear unit (Relu) is a function that is used to activate neurons in each layer and induce non- linearity into the data. Relu activation function is used throughout all the layers except the final classification layer. We have used a Sparse categorical entropy loss function instead of the multinomial cross entropy loss function that was used for multilabel classification. For each image, a multilabel is assigned, and the output of the classification loss with the lowest classification loss is chosen, for better performance. The features are obtained from the last deeper dense layer, with the Relu activation function. The output of this layer is 4096 feature maps and they are arranged in column vector (CSV) format to speed up the process of PSO SVM. Figs. 9a and 9b shows the learned features on the first and last layer of the pretrained convolutional neural network model VGG 16. The first layer output largely consists of edges and colors, indicating that the first layer acts as edge detectors and color filters. The last layer learns the high level combinations of features of the image.

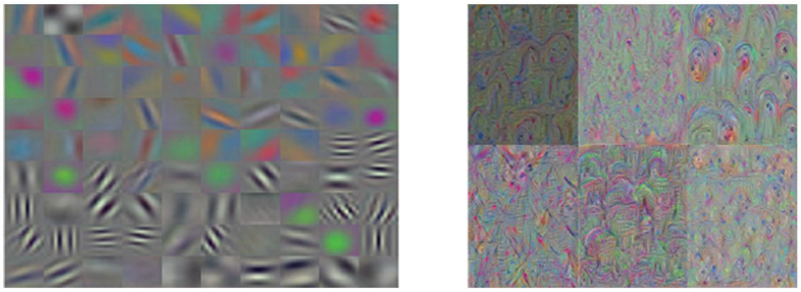

Figure 9: (a) Features Learned from first layer of VGG-16 model. (b) Features Learned from last layer of VGG-16 model

After extracting the features from the pre-trained model, the PSO SVM Model is used to recognise the adversarial iris images. For a high SVM recognition rate, a suitable kernel function with optimal parameters is essential. Therefore our proposed system incorporates the evolutionary based optimization method (PSO) with Support Vector Machine Algorithm. In this SVM recognition experiment, we have considered both the cases (i) the fixed parameters of SVM and (ii) optimized parameters of SVM using our proposed method. For the first case the values of parameters are fixed and they are C(Penalty) = 0.7 and gamma = 1. In the second case PSO method is used to find the best value for the parameters for each set of adversarial attack images. Clearly, the PSO-SVM can automatically determine the best parameters C and gamma for SVM, and the accuracy is clearly superior to Regular SVM in almost all types of attacks. Tab. 6 lists the optimal value of the parameters of each attack.

Table 6: Optimal SVM parameter values obtained from PSO-SVM of different attack images set

S. No.	Attack	C-value	gamma Value
1	FGSM	0.25	9.7
2	iGSM	0.27	9.2
3	Deepfool	0.25	9.7

The classification accuracy of the proposed model with other state of art models are tabulated in Tab. 7. The accuracy of the proposed model is 96.3% in the case of FGSM attack iris images. Similarly the accuracy of iGSM and Deepfool attack images are 95% and 96.3% respectively. Fig. 10 illustrates the accuracy of the proposed method with other state of art classification models with respect to FGSM adversarial attack. Fig. 11 shows the accuracy of the proposed model with other research methods related to iGSM attack. In the same way Fig. 12 represents the suggested model performance with respect to Deepfool attack. It is clear that our suggested technique is quite well suited to recognize the adversarial images with other traditional classification methods based on the IITD database. In comparison to other design trails, we discovered that the proposed model design performs better.

Table 7: Comparison of proposed model with other classification model in terms of accuracy

Classification Method	Accuracy (in %) Adversarial attacks		
	FGSM	iGSM	Deepfool
Model1-Curvelet+SVM	88	85	86.2
Model 2-Curvelet+PCA+SVM	92.4	89.7	90
Model 3-Curvelet+CNN+SVM	94.7	93.5	94
Model 4-Curvelet+CNN	95.6	94.3	95
Model 5-Curvelet+CNN+ PSO SVM (Proposed Method)	96.3	95	96.3

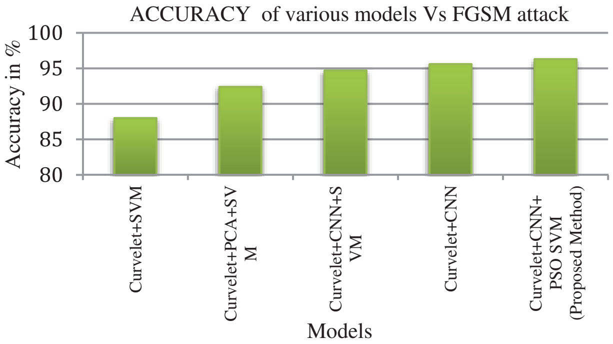

Figure 10: Comparison of classification methods with respect to FGSM attack

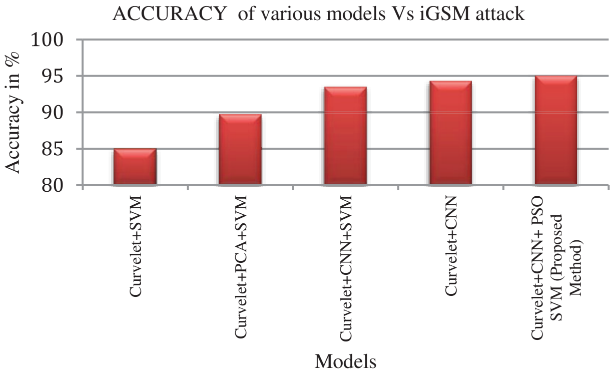

Figure 11: Comparison of classification methods respect with respect to iGSM attack

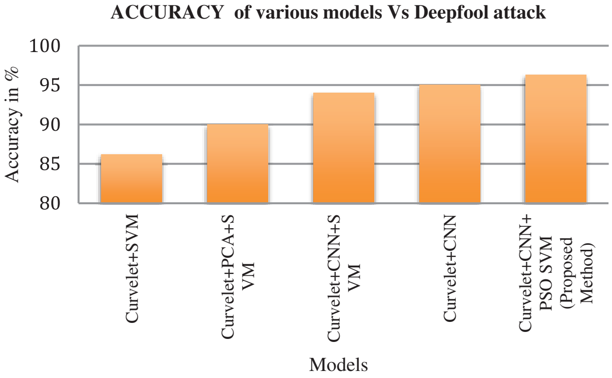

Figure 12: Comparison of classification methods with respect to Deepfool attack

6 Conclusion

In this research, we look at how to defend against adversarial attacks on Deep Convolutional Neural Networks, which are used in biometric systems as Iris classifiers. A novel defending method which addresses the adversarial iris images is proposed in this research. This defending technique uses the Curvelet Transform based denoising method which analyses each sub band of input image and reconstructs the image those are affected by the adversary. The salient iris features are extracted from the denoised iris image, by using a pre-trained convolutional Neural Network (VGG 16 model) followed by Multi class classification is performed by using PSO Support Vector Machine (PSO SVM). The presented method is evaluated using publicly available datasets (IITD iris databases and CASIA-Iris-Interval), and it achieves a high accuracy rate. Experiments show that our suggested method can generate useful and realistic iris features to recognise the adversarial attacks effectively with high accuracy and robustness. When compared to other existing defending models, this model has produced significant results with 96.5 percent accuracy in the test data. A number of novel ideas are included in this work. In the existing defending mechanism, wavelet transform is used for feature extraction and it is good at describing the point singularities but it struggles to detect curve singularities. Feature learning methods and Transfer learning are receiving a lot of attention these days. The reconstructed image is sent directly into the pre-trained CNN model which extracts the best aspects of the image. In terms of adversarial image detection, for a high SVM recognition rate, a suitable kernel function with optimal parameters is essential. Therefore our proposed system incorporates the evolutionary based optimization method (PSO) with Support Vector Machine Algorithm which produces the promising result compared with other conventional classification methods. In the future, we will test the performance of the proposed technique in other iris datasets and with other biometric recognition challenges using alternative pre trained CNN models. The proposed system may be tested with other types of adversarial attacks to build a more generalized defending framework.

Funding Statement: The authors received no specific funding for this research.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770–778, 2016.

[2] K. Meenakshi and G. Maragatham, "A review on security attacks and protective strategies of machine learning," in *Int. Conf. on Emerging Current Trends in Computing and Expert Technology*, Cham, Springer, pp. 1076–1087, 2019.

[3] A. Athalye, N. Carlini and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Int. Conf. on Machine Learning, PMLR*, Stockholm, Sweden, pp. 274–283, 2018.

[4] W. Xu, D. Evans and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," arXiv preprint arXiv: 1704.01155, 2017.

[5] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. of the 10th ACM Workshop on Artificial Intelligence and Security*, Texas, Dallas, USA, pp. 3–14, 2017.

[6] B. B. Bhaganagare and A. D. Harale, "Iris as biometrics for security system," in *Second Int. Conf. on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, IEEE, pp. 1–7, 2017.

[7] A. Fawzi, S. M. Moosavi-Dezfooli and P. Frossard, "Robustness of classifiers: From adversarial to random noise," arXiv preprint arXiv: 1608.08967, 2016.

[8] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik *et al.,* "The limitations of deep learning in adversarial settings," in *IEEE European Symp. on Security and Privacy*, Saarbruecken, Germany, IEEE, pp. 372–387, 2016.

[9] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv: 1412.6572, 2014.

[10] A. Rozsa, E. M. Rudd and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, USA, pp. 25–32, 2016.

[11] S. M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2574–2582, 2016.

[12] T. Tanay and L. Griffin, "A boundary tilting perspective on the phenomenon of adversarial examples," arXiv preprint arXiv: 1608.07690, 2016.

[13] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *IEEE Symp. on Security and Privacy*, San Jose, CA, USA, pp. 582–597, 2016.

[14] Z. Yan, Y. Guo and C. Zhang, "Deep defense: Training DNNs with improved adversarial robustness," arXiv preprint arXiv: 1803.00404, 2018.

[15] P. Yang, J. Chen, C. J. Hsieh, J. L. Wang and M. Jordan, "Ml-loo: Detecting adversarial examples with feature attribution," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 6639–6647, 2020.

[16] N. Papernot and P. McDaniel, "Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning," arXiv preprint arXiv: 1803.04765, 2018.

[17] A. Sotgiu, A. Demontis, M. Melis, B. Biggio, G. Fumera *et al.,* "Deep neural rejection against adversarial examples," *EURASIP Journal on Information Security*, vol. 1, pp. 1–10, 2020.

[18] J. Lu, T. Issaranon and D. Forsyth, “Safetynet: Detecting and rejecting adversarial examples robustly,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 446–454, 2017.

[19] D. Meng and H. Chen, “Magnet: A two-pronged defense against adversarial examples,” in *Proc. of ACM SIGSAC Conf. on Computer and Communications Security*, Texas, Dallas, USA, pp. 135–147, 2017.

[20] G. Katz, C. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer, “Reluplex: An efficient SMT solver for verifying deep neural networks,” in *Int. Conf. on Computer Aided Verification*, in *Proc.: Lecture Notes in Computer Science, Cham, Springer, pp. 97–117, 2017.*

[21] G. Goswami, A. Agarwal, N. Ratha, R. Singh and M. Vatsa, “Detecting and mitigating adversarial perturbations for robust face recognition,” *International Journal of Computer Vision*, vol. 127, no. 6, pp. 719–742, 2019.

[22] S. Soleymani, A. Dabouei, J. Dawson and N. M. Nasrabadi, “Adversarial examples to fool iris recognition systems,” in *2019 Int. Conf. on Biometrics (ICB)*, Crete, Greece, IEEE, pp. 1–8, 2019.

[23] S. Soleymani, A. Dabouei, J. Dawson and N. M. Nasrabadi, “Defending against adversarial iris examples using wavelet decomposition,” in *IEEE 10th Int. Conf. on Biometrics Theory, Applications and Systems*, Tampa, FL, USA, pp. 1–9, 2019.

[24] S. R. Tamizhiniyan, A. Ojha, K. Meenakshi and G. Maragatham, “DeepIris: An ensemble approach to defending iris recognition classifiers against adversarial attacks,” in *Int. Conf. on Computer Communication and Informatics*, Coimbatore, India, IEEE, pp. 1–8, 2021.

[25] K. Meenakshi and G. Maragatham, “A self supervised defending mechanism against adversarial iris attacks based on wavelet transform,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, pp. 564–569, 2021.

[26] D. L. Donoho and M. R. Duncan, “Digital curvelet transform: Strategy, implementation, and experiments,” *Wavelet applications VII, International Society for Optics and Photonics*, vol. 4056, pp. 12–30, 2000.

[27] T. Mandal, Q. M. J. Wu and Y. Yuan, “Curvelet based face recognition via dimension reduction,” *Signal Processing*, vol. 89, no. 12, pp. 2345–2353, 2009.

[28] S. Dargan, M. Kumar, M. R. Ayyagari and G. Kumar, “A survey of deep learning and its applications: A new paradigm to machine learning,” *Archives of Computational Methods in Engineering*, vol. 27, no. 4, pp. 1071–1092, 2020.

[29] Z. Zhou, Z. Cao and Y. Pi, “Dynamic gesture recognition with a terahertz radar based on range profile sequences and Doppler signatures,” *Sensors*, vol. 18, no. 1, pp. 10–15, 2018.

[30] E. Mehdi Cherrat, R. Alaoui and H. Bouzahir, “Convolutional neural networks approach for multimodal biometric identification system using the fusion of fingerprint, finger-vein and face images,” *PeerJ Computer Science*, vol. 6, no. 2, pp. e248, 2020.

[31] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. of ICNN'95-Int. Conf. on Neural Networks*, Perth, WA, Australia, vol. 4, pp. 1942–1948, 1995.