International Journal of Computer Vision (2019) 127:719–742
https://doi.org/10.1007/s11263-019-01160-w

Detecting and Mitigating Adversarial Perturbations for Robust Face Recognition

Gaurav Goswami[1] · Akshay Agarwal[1] · Nalini Ratha[2] · Richa Singh[1] · Mayank Vatsa[1]

Received: 22 February 2018 / Accepted: 29 January 2019 / Published online: 22 March 2019

Abstract
Deep neural network (DNN) architecture based models have high expressive power and learning capacity. However, they are essentially a black box method since it is not easy to mathematically formulate the functions that are learned within its many layers of representation. Realizing this, many researchers have started to design methods to exploit the drawbacks of deep learning based algorithms questioning their robustness and exposing their singularities. In this paper, we attempt to unravel three aspects related to the robustness of DNNs for face recognition: (i) assessing the impact of deep architectures for face recognition in terms of vulnerabilities to attacks, (ii) detecting the singularities by characterizing abnormal filter response behavior in the hidden layers of deep networks; and (iii) making corrections to the processing pipeline to alleviate the problem. Our experimental evaluation using multiple open-source DNN-based face recognition networks, and three publicly available face databases demonstrates that the performance of deep learning based face recognition algorithms can suffer greatly in the presence of such distortions. We also evaluate the proposed approaches on four existing quasi-imperceptible distortions: DeepFool, Universal adversarial perturbations, l_2, and Elastic-Net (EAD). The proposed method is able to detect both types of attacks with very high accuracy by suitably designing a classifier using the response of the hidden layers in the network. Finally, we present effective countermeasures to mitigate the impact of adversarial attacks and improve the overall robustness of DNN-based face recognition.

Keywords Face recognition · Deep learning · Adversarial · Dropout · Adversarial learning · Attack detection · Attack mitigation

1 Introduction

With the convenience of obtaining large training data, availability of inexpensive computing power and memory, and utilization of cameras at multiple places, *deep learning* paradigm has seen significant proliferation in face recognition. Several algorithms such as DeepFace (Taigman et al. 2014), DeepID (Sun et al. 2015), FaceNet (Schroff et al. 2015), and Liu et al. (2015) are successful examples of application of deep learning to face recognition. These deep CNN based architectures with many hidden layers and millions of parameters can obtain very high accuracies when tested on databases such as the LFW database (Huang et al. 2007) and NIST's face recognition test (NIST face recognition vendor test ongoing 2018). While unprecedented improvements in the reported accuracy of machine learning algorithms continue, it is also known that they are susceptible to *adversaries* which can cause the classifier to yield incorrect results. Most of the time these adversaries are unintentional and are in the form of outliers. However, such attacks may also be intentionally executed by specifically targeting the *blind spots* of classifiers and have been explored in the literature in the context of many applications of machine learning such as malware detection (Laskov and Lippmann 2010).

✉ Mayank Vatsa
mayank@iiitd.ac.in

Gaurav Goswami
gauravgs@iiitd.ac.in

Akshay Agarwal
akshaya@iiitd.ac.in

Nalini Ratha
ratha@us.ibm.com

Richa Singh
rsingh@iiitd.ac.in

[1] IIIT-Delhi, New Delhi, India

[2] IBM, TJ Watson Research Center, Yorktown Heights, USA

Fig. 1 Illustrating how an image can be attacked with perceptible and quasi-imperceptible adversarial perturbations to create false accepts (match between different individuals) and false rejects (non-match between two images of the same individual). Such errors compromise the reliability of automated face recognition

Creating adversarial samples that can deceive/attack algorithms has become easy lately with the application of the same deep learning techniques. Recently, it has been shown that *fooling adversarial images* can be generated in such a manner where humans can correctly classify the images but deep learning algorithms misclassify them (Goodfellow et al. 2015; Nguyen et al. 2015). There are several algorithms to generate such images, for instance evolutionary algorithms (e.g. Genetic Algorithm) (Nguyen et al. 2015) or adversarial sample crafting using the fast gradient sign method (Goodfellow et al. 2015). Threat models by creating *perturbed eye-glasses* are also explored to fool face recognition algorithms (Sharif et al. 2016). Inspired by recent studies, it is our assertion that deep learning based face recognition algorithms are also susceptible to adversarial attacks and such attacks can be detrimental to recognition algorithms applied in real world applications. In other words, if a deep learning based recognition engine is being used, an attacker can use synthetic deception approaches to either deceive one's own identity (in law enforcement applications) or impersonate someone else's identity (in access control applications).

Even though adversarial attacks primarily pertain to deep network based algorithms, there do exist other forms of attacks against face recognition systems. Ratha et al. (2001) have identified multiple potential attack points for any biometric system; e.g. presenting false biometrics to the sensor level and injecting modified biometrics in between the acquisition and feature extraction levels. Spoofing or presentation attacks at the sensor level are similar to adversarial attacks where the goal is to make the face recognition system perform a misclassification of the input. While extensive research has been conducted on evaluating the vulnerabilities to spoofing attacks and associated countermeasures (Chingovska et al. 2016), *handling* adversarial attacks is relatively less explored in the literature.

The focus of this paper[1] is to demonstrate that the performance of deep learning based face recognition algorithms can be significantly affected due to adversarial attacks. As shown in Fig. 1, we also postulate that it is not required to attack the system with sophisticated learning based attacks; attacks such as adding random noise or horizontal and vertical black grid lines in the face image cause reduction in face verification accuracies. The first key step in taking countermeasures against such adversarial attacks is to be able to reliably determine which images contain such distortions. Once identified, the distorted images may be rejected for further processing or rectified using appropriate preprocessing techniques to prevent degradation in performance. Further, such proposed solutions should be able to operate well in a cross-attack (tested on attack types that are not included in the training data) and cross-database (trained on a different database than the ones used in testing) protocol to be applicable in a live environment where many new attacks and different images may be used with the network. In this paper, we propose a deep network based approach to perform both detection and mitigation procedures. The key contributions of this paper are:

- Design and evaluate image processing based adversarial attacks towards off-the-shelf deep learning based face recognition algorithms.
- Propose and evaluate a methodology for automatic detection of such attacks using the response from hidden layers of the DNN.
- Propose a novel technique of selective dropout in the DNN to mitigate the effect of these adversarial attacks.
- The proposed algorithms have been evaluated using cross-database protocols and have also been evaluated in cross-attack scenarios.

We believe that being able to not only automatically detect but also correct adversarial samples at runtime is a crucial ability for a deep network that is deployed for real world applications. With this research, we aim to present a new perspective on potential attacks as well as a different methodology to limit their performance impact beyond simply including adversarial samples in the training data.

[1] A shorter version of the manuscript was presented at AAAI2018.

2 Related Work

The existing literature on attacks against face recognition and associated defense strategies can be divided into four categories: face spoofing, and the generation, detection, and mitigation of adversarial sample based attacks. We briefly describe the existing work in each of these categories in the following subsections. Table 1 lists some recent adversarial example generation, detection, and mitigation algorithms. Recently, Akhtar and Mian (2018) have presented the survey of adversarial generation, detection, and mitigation algorithms.

2.1 Face Spoofing

Attacks on face recognition systems have been studied in the past focusing on presentation attacks on remote unsupervised face recognition. Among the first attacks on face biometrics that have come into focus are spoofing or presentation attacks. The presentation attack involves *presenting* a fake face to the biometric sensor using a printed photograph, worn mask, or even an image displayed on another electronic device. The presentation might not be just a static face image, rather it could be previously captured or otherwise obtained video of a face that can also be played back to the sensor using an electronic device. Chingovska et al. (2016) present a review of the vulnerabilities of a face based system in the presence of these attacks as well as how multispectral systems can mitigate some of the risk. However, Raghavendra et al. (2017) and Agarwal et al. (2017b) have prepared a database for multispectral spoofing and reported that even such systems are not immune to presentation attacks. Recent efforts in designing presentation attack detection methodologies include software level solutions such as color texture analysis based detection (Agarwal et al. 2016; Boulkenafet et al. 2016; Siddiqui et al. 2016) and hardware level solutions such as light polarization analysis using a novel hardware extension (Rudd et al. 2016). Biggio et al. (2017) have presented a method based on meta-level statistical analysis to assess the vulnerability of multi-biometric systems against presentation attacks. Patel et al. (2015) have proposed a detection methodology based on the moire pattern analysis for mobile phones. Smith et al. (2015) propose a reflection watermark challenge-response based detection methodology for consumer devices. Recently, Boulkenafet et al. (2017) have proposed a detection methodology using Fisher vector encoding and speeded-up robust features (SURF) (Bay et al. 2006) for spoofing attack detection with limited training data for a generalizable methodology that works well on unseen databases. For detecting silicone mask based face presentation attacks, Manjani et al. (2017) propose a dictionary learning based approach that shows state-of-the-art results on spoofing databases. Deep learning based approaches for face spoofing detection have also been proposed recently that utilize CNNs in conjunction with texture features and other types of deep networks (Akbulut et al. 2017; Gan et al. 2017; de Souza et al. 2017).

2.2 Adversarial Example Generation

With increasing usage of deep learning algorithms for complex and popular tasks such as object recognition and face recognition, researchers are also attempting to understand the limitations of deep learning algorithms. Szegedy et al. (2014) have investigated the properties of deep neural networks and concluded that the input–output mappings that are learned by them can be fairly discontinuous and can be exploited to create an adversarial perturbation. Goodfellow et al. (2015) have expanded on the research presented in Szegedy et al. (2014) and further investigated adversarial attacks on a deep network. They explain the existence of adversarial examples for a neural network based on the limited precision (0–255 in case of image pixels) of input data combined with the implications of a high-dimensional dot product. Sabour et al. (2016) generate adversarial samples by minimizing the distance between the internal feature representations of images belonging to different classes. Moosavi-Dezfooli et al. (2016) have presented a methodology to create adversarial examples called DeepFool that works by computing the minimal perturbation such that the distance between the correct decision hyperplane and a given data point is minimized, converging to 0. Papernot et al. (2017) have demonstrated a practical scenario for using an adversarial attack against a black-box DNN without any knowledge of the network's hyperparameters. Rozsa et al. (2016) discuss adversarial attacks on a deep CNN method that extracts soft biometric attributes from facial images (such as gender). They demonstrate that certain attributes are inherently more robust towards adversarial attacks than others. They also demonstrate that naturally adversarial samples exist which can be correctly classified by adding a perturbation in a kind of reverse adversarial attack. They construct an auxiliary substitute deep model by emulating the input–output mapping observed by the target DNN and then craft adversarial examples based on the auxiliary model. Moosavi-Dezfooli et al. (2017) have extended their DeepFool perturbations by aggregating the learned perturbations across an entire collection of images to determine a "universal" perturbation pattern that can be applied to any image to fool the targeted network. Carlini and Wagner (2017) have devised a set of attacks specifically targeted at rendering defensive distillation ineffective using l_p distance metric optimization to make them quasi-imperceptible. Rauber et al. (2017) have crafted blackbox attacks using domain-agnostic image transformations that can modify the texture of the image to attack deep networks. Rozsa et al. (2017a) have drafted a strategy to generate adversarial sam-

Table 1 Literature review of adversarial attack generation, detection, and mitigation algorithms

Adversary	Authors	Description
Generation	Szegedy et al. (2014)	L-BFGS: $L(x+\rho, l) + \lambda\|\|\rho\|\|^2\ s.t.\ x_i + \rho_i \in [b_{min}, b_{max}]$
	Goodfellow et al. (2015)	FGSM: $x_0 + \epsilon * (\nabla_x L(x_0, l_0)$
	Kurakin et al. (2016)	I-FGSM: $x_{k+1} = x_k + \epsilon * (\nabla_x L(x_0, l_0)$
	Papernot et al. (2016a)	Saliency Map: l_0 distance optmization
	Moosavi-Dezfooli et al. (2016)	DeepFool: $for\ each\ class, l \neq l_0, minimize\ d(l, l_0)$
	Rozsa et al. (2016)	Adversarial attacks on biometric attribute predicting deep CNNs
	Carlini and Wagner (2017)	C & W: l_p distance metric optimization
	Moosavi-Dezfooli et al. (2017)	Universal: Distribution based perturbation
	Rauber et al. (2017)	Blackbox: Uniform, Gaussian, Salt and Pepper, Gaussian Blur, Contrast
	Rozsa et al. (2017a)	LOTS: Layerwise Target-Origin Synthesis method to attack deep feature based systems
	Rozsa et al. (2016, 2017b)	Fast flipping attribute based on inverting classifier score
	Chhabra et al. (2018)	Facial attribute anonymization using adversarial noise
	Tramèr et al. (2018)	R+FGSM $x' + (\epsilon - \alpha) * sign(\nabla'_x J(x', y_{true})$
	Addad et al. (2018)	Clipping free Centered Initial Attack
	Alaifari et al. (2018)	Gradient descent based deformation
	Athalye and Sutskever (2018)	Expectation Over Transformation
Detection	Grosse et al. (2017)	Statistical test for attack and genuine data distribution
	Gong et al. (2017) and Metzen et al. (2017)	Neural network based classification
	Feinman et al. (2017)	Randomized network using Dropout at both training and testing
	Liang et al. (2017)	Quantization and smoothing based image processing
	Lu et al. (2017)	Quantize ReLU output for discrete code + RBF SVM
	Meng and Chen (2017)	Learned manifold based classification of adversarial and clean images
	Li and Li (2017)	Convolutional filter statistics with cascaded classifier
	Tramèr et al. (2018)	Ensemble training
	Akhtar et al. (2017)	Perturbation Rectifying Network
	Goswami et al. (2018)	Filter responses of CNN
	Agarwal et al. (2018)	Image Pixels + PCA + SVM
Mitigation	Miyato et al. (2017)	Virtual adversarial training
	Dziugaite et al. (2016)	JPEG compression based mitigation for FGSM attacks
	Das et al. (2017)	JPEG compression to reduce the effect of adversary
	Bhagoji et al. (2017)	Compressing the data using PCA before testing
	Luo et al. (2015)	Applying the network to different regions of the image
	Xie et al. (2017)	Random resizing and random padding of the input images
	Gu and Rigazio (2014)	Deep Contractive Networks with smoothness penalty
	Ross and Doshi-Velez (2018)	Gradient regularization based on relative change in output and input
	Papernot et al. (2016b)	Using class probability vectors from trained network to re-train the original model
	Nayebi and Ganguli (2017)	Using highly non-linear activation functions
	Cisse et al. (2017)	Layer-wise regularization by maintaining a small global Lipschitz constant
	Akhtar et al. (2017)	Add a pre-input perturbation rectification network to the target network
	Lee et al. (2017)	Generative adversarial network framework to perform adversarial training
	Ye et al. (2018)	Model compression using pruning + LOGITS Augmentation
	Ranjan et al. (2017)	Bounding the feature maps close to each other by power convolution
	Kurakin et al. (2016)	Naive adversarial training

Table 1 continued

Adversary	Authors	Description
	Rakin et al. (2018)	Quantization of activation function
	Prakash et al. (2018)	Pixel deflections + wavelet denoising
	Goswami et al. (2018)	Dropout of filter responses
	Tramèr et al. (2018)	R+FGSM adversarial training
	Guo et al. (2018)	Input transformations
	Xie et al. (2018)	Input randomization
	Song et al. (2018)	Purifies images using PixelCNN
	Samangouei et al. (2018)	Generative Adversarial Networks based defense

ples by targeting the perturbations such that the layer-wise features of the adversarial image closely resemble the features of a sample from a different class. They showcase that biometric systems using deep features along with some distance metric are more vulnerable to such attacks as compared to end-to-end networks that directly predict the output label. Athalye and Sutskever (2018) have presented the algorithm to generate the physical adversarial examples using Expectation Over Transformation (EOT).

2.3 Adversarial Example Detection

As new methods of creating adversarial examples have been proposed, research has also been conducted in utilizing adversarial examples for training more robust networks to counter adversarial attack as well as improve the overall quality of learned representations. Grosse et al. (2017) have proposed a method to statistically model the distribution of attacked images and genuine images, and then checking the fit of each image to classify it into either category. Meng and Chen (2017) have proposed a similar approach but with manifold learning instead for the clean and adversarial images. Feinman et al. (2017) have proposed using the uncertainty estimates of dropout networks as features to train separate binary classifiers for detecting attacks. Liang et al. (2017) have suggested using smoothing and quantization based image processing techniques to detect the perturbations added to images. Lu et al. (2017) have proposed a SafetyNet framework using the difference in the pattern of the output of ReLU activations as features to a RBF kernel SVM classifier to detect adversarial examples. Li and Li (2017) have proposed a similar algorithm using the convolutional filter statistics as features instead of ReLU activations and a cascaded classifier instead of the RBF kernel SVM. Xu et al. (2018) have proposed another detection methodology based on the difference in features extracted using a full resolution image with that of a lower fidelity version (obtained by reducing color bit depth or spatial smoothing). While this approach is simple and effective for high resolution images which contain a lot of detail, it may not be effective for low resolution cropped faces which are often used in face recognition scenarios. Recently, Agarwal et al. (2018) have shown high detection accuracy of image agnostic perturbation using image pixels and dimensionality reduction using PCA with SVM classifier.

2.4 Adversarial Example Mitigation

As the existence of adversarial examples has gained attention in the literature, researchers have also proposed a few techniques to handle adversarial attacks and mitigate their effect on the performance of a targeted deep network. Radford et al. (2015) have utilized adversarial pair learning to compute unsupervised representations using convolutional neural networks where the generator model produces images with the intent to try and fool the discriminator model. They demonstrate that both the models learn useful feature representations by using them for object and face recognition. This model of learning called Generative Adversarial Network (GAN) has since become quite popular. Recently, GANs have been used in domain adaptation (Bousmalis et al. 2016) and cross-domain image generation tasks using weight-sharing coupling (Liu and Tuzel 2016). GANs have now also been used as part of defenses against adversarial attacks (Lee et al. 2017; Samangouei et al. 2018). Song et al. (2018) have proposed PixelCNN based generative model to purify the adversarial examples. Papernot et al. (2016b) have proposed a defense mechanism towards adversarial attacks. The authors propose that distillation (Hinton et al. 2015) can be performed to create a network that is resilient towards adversarial attacks and utilize perturbations targeting sensitive gradients. They report favorable results using this methodology on the MNIST and CIFAR-10 databases, improving results against the crafted adversarial data. Although distillation seems to greatly improve results when the adversarial attacks are based on such perturbations, we focus on the impact of adversarial examples that employ a different approach and do not depend on network gradient information. Bhagoji et al. (2017) have

proposed that using PCA based dimensionality reduction can reduce the effect of adversarial examples on network performance. With a similar idea, Das et al. (2017) have proposed using JPEG compression to pre-process the image before applying the deep network. Xie et al. (2017) have proposed using randomly resizing and padding the input images before processing them which can reduce the effectiveness of adversarial attacks. Ross and Doshi-Velez (2018) have proposed modifying the loss function of the network such that small changes in the input causing large changes in the output is penalized to improve the stability of the predictions made by the network in the presence of adversarial examples that have been created with a constrained l_p norm. Nayebi and Ganguli (2017) have proposed using highly non-linear activation functions that are biologically inspired to reduce the linearity of the network and counter adversarial examples. Akhtar et al. (2017) have proposed adding a pre-input layer rectification network to the target network which is trained to reconstruct clean images from their adversarial counterparts so that the image can be cleaned before extracting features. Recently, Goel et al. (2018) have prepared the SmartBox toolbox containing several existing adversarial generation, detection, and mitigation algorithms.

3 Adversarial Attacks on Deep Learning Based Face Recognition

In this section, we discuss the adversarial distortions that are able to degrade the performance of deep face recognition algorithms. We use both imperceptible and perceptible perturbations. The perceptible perturbations are modeled on commonly observed face domain distortions. For example, an old passport might contain a laminated face image with a different type of distortion compared to someone growing a beard. Let I be the face image input to a deep learning based face recognition algorithm, $\mathcal{D}$, and l be the output class label (in case of identification, it is an identity label and for verification, it is *match* or *non-match*). Let $a(\cdot)$ be an adversarial attack operator which perturbs the input image I such that the network $\mathcal{D}$ yields an incorrect class label l'. In other words, $\mathcal{D}(I) = l$ and $\mathcal{D}(a(I)) = l'$ and $l \neq l'$. In this research, we also evaluate the robustness of deep learning based face recognition in the presence of image processing based distortions. Based on the information required in their design, these distortions can be considered at image-level or face-level. We propose two image-level distortions: (a) grid based occlusion, and (b) most significant bit based noise, three face-level distortions: (a) forehead and brow occlusion, (b) eye region occlusion, and (c) beard-like occlusion. Further, the imperceptible perturbations are based on state-of-the approaches including DeepFool (Moosavi-Dezfooli et al. 2016), Universal Adversarial Perturbations (Moosavi-Dezfooli et al. 2017), l_2 attack (Carlini and Wagner 2017), and EAD (Chen et al. 2018). We have also performed the adversarial detection and mitigation experiments on these adversarial perturbations.

3.1 Image-Level Distortions

Distortions that are not specific to faces and can be applied to an image of any object are categorized as image-level distortions. In this research, we have utilized two such distortions, grid based occlusion and most significant bit change based noise addition. Figure 2b and c present sample outputs of image-level distortions.

3.1.1 Grid Based Occlusion

For the grid based occlusion (termed as Grids) distortion, we stochastically select a number of points $P = \{p_1, p_2, \ldots, p_n\}$ along the upper ($y = 0$) and left ($x = 0$) boundaries of the image according to a parameter ρ_{grids}. The parameter ρ_{grids} determines the number of grids that are used to distort each image with higher values resulting in a denser grid, i.e., more grid lines. For each point $p_i = (x_i, y_i)$, we select a point on the opposite boundary of the image, $p'_i = (x'_i, y'_i)$, with the condition if $y_i = 0$, then $y'_i = H$ and if $x_i = 0$ then $x'_i = W$, where, $W \times H$ is the size of the input image. Once a set of pairs corresponding to points P and P' have been selected for the image, one pixel wide line segments are created to connect each pair, and each pixel lying on these lines is set to 0 grayscale value. In this paper, the parameter ρ_{grids} is set to 0.4 which results in a minimum of 4 and maximum of 10 grid lines (of 1 pixel thickness each) on each perturbed image.

3.1.2 Most Significant Bit Based Noise

For the most significant bit based noise (xMSB) distortion, we select three sets of pixels $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ from the image stochastically such that $|\mathcal{X}_i| = \phi_i \times W \times H$, where $W \times H$ is the size of the input image. The parameter ϕ_i denotes the fraction of pixels where the ith most significant bit is flipped. The higher the value of ϕ_i, the more pixels are distorted in the ith most significant bit. For each $\mathcal{P}_j \in X_i, \forall i \in [1, 3]$, we perform the following operation:

$$\mathcal{P}_{kj} = \mathcal{P}_{kj} \oplus 1 \tag{1}$$

where, $\mathcal{P}_{kj}$ denotes the kth most significant bit of the jth pixel in the set and $\oplus$ denotes the bitwise XOR operation. It is to be noted that the sets $\mathcal{X}_i$ are not mutually exclusive and may overlap. Therefore, the total number of pixels affected by the noise is at most $|\mathcal{X}_1 + \mathcal{X}_2 + \mathcal{X}_3|$ but may also be lower depending on the stochastic selection. In this research, results are reported with $\phi = [0.03, 0.05, 0.1]$.

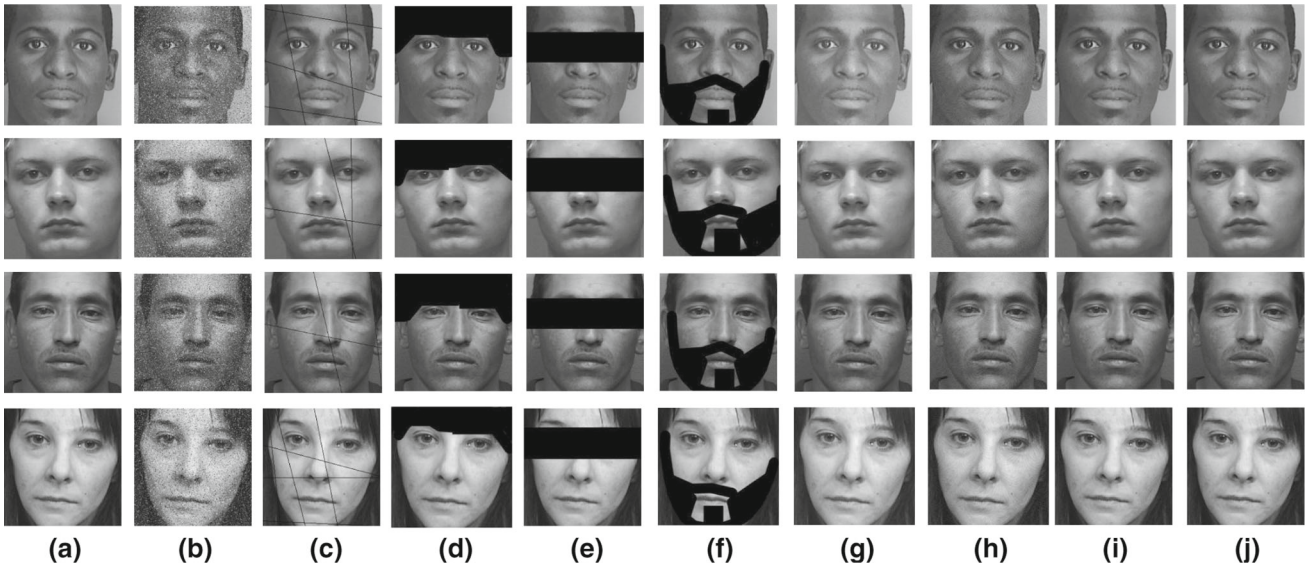


Fig. 2 Sample images representing the **b** grid based occlusion (Grids), **c** most significant bit based noise (xMSB), **d** forehead and brow occlusion (FHBO), **e** eye region occlusion (ERO), **f** beard-like occlusion (Beard), **g** DeepFool (Moosavi-Dezfooli et al. 2016), **h** Universal (Moosavi-Dezfooli et al. 2017), **i** L_2 (Carlini and Wagner 2017), and **j** Elastic-Net (EAD) (Chen et al. 2018) distortions when applied to the **a** original images

3.2 Face-Level Distortions

Face-level distortions specifically require face-specific information, e.g. location of facial landmarks. The three face-level region based occlusion distortions are applied after performing automatic face and facial landmark detection. In this research, we have utilized the open source DLIB library (King 2009) to obtain the facial landmarks. Once facial landmarks are identified, they are used along with their boundaries for masking. To occlude the eye region, a singular occlusion band is drawn on the face image as follows:

$$I\{x, y\} = 0, \forall x \in [0, W], y \in \left[y_e - \frac{d_{eye}}{\psi}, y_e + \frac{d_{eye}}{\psi}\right] \quad (2)$$

Here, $y_e = \left(\frac{y_{le}+y_{re}}{2}\right)$, and (x_{le}, y_{le}) and (x_{re}, y_{re}) are the locations of the left eye center and the right eye center, respectively. The inter-eye distance d_{eye} is calculated as: $x_{re} - x_{le}$ and ψ is a parameter that determines the width of the occlusion band. Similar to the eye region occlusion (ERO), the forehead and brow occlusion (FHBO) is created where facial landmarks on forehead and brow regions are used to create a mask. For the beard-like occlusion (Beard), outer facial landmarks along with nose and mouth coordinates are utilized to create the mask as combinations of individually occluded regions. Figure 2d–f illustrate the samples of face-level distortions.

3.3 Learning Based Adversaries

Along with the proposed image-level and face-level distortions, we also analyze the effect of adversarial samples generated using four existing adversarial models: DeepFool (Moosavi-Dezfooli et al. 2016), Universal Adversarial Perturbations (Moosavi-Dezfooli et al. 2017), l_2 attack (Carlini and Wagner 2017), and EAD (Chen et al. 2018). DeepFool (Moosavi-Dezfooli et al. 2016) calculates a minimal norm adversarial perturbation for a given image in an iterative manner. It initializes with the original image that lies in the feature space in a region within the decision boundaries of the classifier for the correct class. In each subsequent iteration, the algorithm perturbs the current image by a small vector that is designed to shift the resulting image further towards the boundary. The perturbations added to the image in each iteration are accumulated to compute the final perturbation once the perturbed image changes its label according to the original decision boundaries of the network. The Universal adversarial perturbations (Moosavi-Dezfooli et al. 2017) are 'universal' in the sense that they are designed to be able to utilize *any* image to fool a network with a high probability. These perturbations are also visually imperceptible to a large extent. These are learned by using a set of clean images and iteratively shifting all of them towards the decision boundary while limiting the l_2 norm and l_∞ norm of the perturbation to a fraction of the respective norms of the original image. The universal perturbation is computed by gradually accumulating the perturbations for each image in the training data

while maintaining the constraint on the perturbation norm. The l_2 attack proposed by Carlini and Wagner (2017) operates with a similar formulation where they attempt to apply a box constraint to the adversarial image using the l_2 distance while ensuring maximum deviation from the correct class during prediction. However, they consider the integrality constraint function as well as use multiple gradient descent in the optimization routine. The EAD attack (Chen et al. 2018) follows the same philosophy as the l_2 attack but instead of focusing on the l_2-norm to apply the box constraint it instead utilizes the l_2 and l_1 metrics to perform an elastic-net regularization to optimize the adversarial generation routine. For these learning based attacks, we have followed the training process defined in the respective papers, along with default parameters including strength parameter. In our experiments, no knowledge of attacked databases is used in training the models i.e., distortions specific to a deep learning modal are computed on ImageNet database (Deng et al. 2009) and then applied for face images.

The inherent difference between these learning based adversaries and the proposed attacks is that the perturbation caused by the learning based adversaries is smaller (visually imperceptible) and therefore harder to detect. On the other hand, the proposed image processing operations based distortions are completely network-agnostic and instead rely on the domain knowledge by targeting face-specific features. By evaluating the proposed approaches on all the learning based quasi-imperceptible adversaries and the proposed perturbations, we are able to assess its performance in a variety of possible real world scenarios.

4 Impact of Adversarial Perturbations on Existing DNNs

In this section, we first provide a brief overview of the deep face recognition networks, databases, and respective experimental protocols that are used to conduct the face verification evaluations. We attempt to assess how the deep networks perform in the presence of different kinds of proposed distortions to emphasize the need for addressing such attacks.

4.1 Existing Networks and Systems

In this research, we utilize OpenFace (Amos et al. 2016), VGG-Face (Parkhi et al. 2015), LightCNN (Wu et al. 2018), and L-CSSE (Majumdar et al. 2017) networks to measure the performance of deep face recognition algorithms in the presence of the aforementioned distortions. The OpenFace library is an implementation of FaceNet (Schroff et al. 2015) and is openly available to all members of the research community for modification and experimental usage. The VGG face network is a deep convolutional neural network (CNN) with 11 convolutional blocks where each convolution layer is followed by non-linearities such as ReLU and max pooling. The network has been trained on a dataset of 2.6 million face images pertaining to 2622 subjects (Parkhi et al. 2015). LightCNN is another publicly available deep network architecture for face recognition that is a CNN with maxout activations in each convolutional layer and achieves good results with just five convolutional layers. LightCNN has been trained on a combined database with 99,891 individuals. L-CSSE is a supervised autoencoder formulation that utilizes a class sparsity based supervision penalty in the loss function to improve the classification capabilities of autoencoder based deep networks. These deep learning approaches are used to extract features and as described in the original papers, normalization and recommended matching measures are used. In order to assess the relative performance of deep face recognition with a non-deep learning based approach, we compare the performance of these deep learning based algorithms with a commercial-off-the-shelf (COTS) matcher. The details of the COTS matching algorithm are unavailable but it is known that it is not deep learning based. No fine-tuning is performed for any of these algorithms before evaluating their performance on the test databases.

4.2 Databases

We use three publicly available face databases, namely, the Point and Shoot Challenge (PaSC) database (Beveridge et al. 2013), the Multiple Encounters Dataset (MEDS) (Multiple encounters dataset (MEDS) 2011), and the Multiple Biometric Grand Challenge (MBGC) database (Phillips et al. 2009). The PaSC database (Beveridge et al. 2013) contains still-to-still and video-to-video matching protocols. We use the frontal subset of the still-to-still protocol which contains 4,688 images pertaining to 293 individuals which are divided into equally sized target and query sets. Each image in the target set is matched to each image in the query set and the resulting 2344×2344 score matrix is used to determine the verification performance.

The MEDS-II database (Multiple encounters dataset (MEDS) 2011) contains a total of 1,309 faces pertaining to 518 individuals. Similar to the case of PaSC, we utilize the metadata provided with the MEDS release 2 database to obtain a subset of 858 frontal face images from the database. Each of these images is matched to every other image and the resulting 858×858 score matrix is utilized to evaluate the verification performance.

The still portion of the MBGC database (Phillips et al. 2009) contains a total of 34,729 faces pertaining to 570 individuals. These images are split into 10,687 faces in the query set and 24,042 faces in the target set. There are two versions for the target and query sets, where one version has an inter-eye distance of 90 pixels and is compressed to a 8 KB JPEG

Table 2 Verification performance of existing face recognition algorithms in the presence of different proposed distortions on the MEDS and PaSC databases

Database	System	Original	Grids	xMSB	FHBO	ERO	Beard
MEDS	COTS	24.1	20.9	14.5	19.0	0.0	24.8
	OpenFace	66.7	49.5	43.8	47.9	16.4	48.2
	VGG-Face	60.1	50.3	45.0	25.7	10.9	47.7
	LightCNN	89.3	80.1	71.5	62.8	26.7	70.7
	L-CSSE	89.1	81.9	83.4	55.8	27.3	70.5
PaSC	COTS	40.3	24.3	19.1	13.0	0.0	6.2
	OpenFace	39.4	10.1	10.1	14.9	6.5	22.6
	VGG-Face	31.2	3.2	1.3	15.2	8.8	24.0
	LightCNN	60.1	24.6	29.5	31.9	24.4	38.1
	L-CSSE	61.2	43.1	36.9	29.4	39.1	39.8

All values indicate genuine accept rate (%) at 1% false accept rate

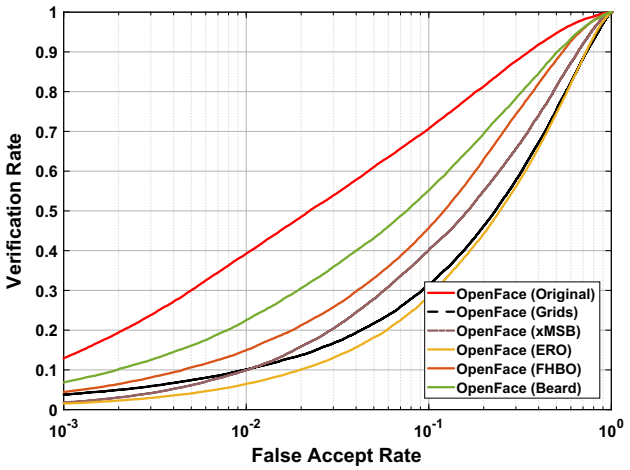
image, and the other has an inter-eye distance of 120 pixels and is compressed to a 20 KB JPEG image. We refer to the first set as MBGC (8 KB) or MBGC (8) and the other as MBGC (20 KB) or MBGC (20) while reporting the results. The 10687 $\times$ 24042 score matrix is used to determine the verification performance for both of these sets.

For evaluating performance under the effect of distortions, we randomly select 50% of the total images from each database and corrupt them with the proposed distortions separately. These distorted sets of images are utilized to compute the new score matrices for each case.

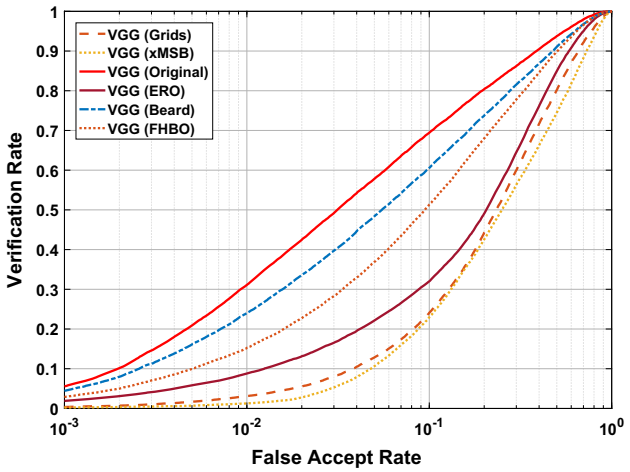
4.3 Results and Analysis

Effect of adversarial distortions on OpenFace, VGG-Face, LightCNN, L-CSSE, and COTS are summarized in Table 2. Figures 3 and 4 present the Receiver Operating Characteristics (ROC) curves on the PaSC and MEDS databases respectively with OpenFace, VGG-Face, and COTS. On the PaSC database, as shown in Fig. 3, while OpenFace and COTS perform comparably to each other at about 1% false accept rate (FAR), OpenFace performs better than the COTS algorithm at all further operating points when no distortions are present. However, we observe a sharp drop in OpenFace performance when any distortion is introduced in the data. For instance, with grids attack, at 1% FAR, the Genuine Accept Rate (GAR) drops from 39.4 to 10.1% which is a loss of 29.3% (OpenFace) and 31.2–3.2% which is a loss of 28.0% (VGG). On the other hand, the COTS performance only drops to 24.3% from 40.3% which is only about half the drop compared to what OpenFace and VGG experience. We notice a similar scenario in the presence of noise attack (xMSB) where OpenFace performance drops down to 10.1% which is a loss of 29.2% (29.9% in the case of VGG) as opposed to loss of 21.2% observed by COTS. In cases of LightCNN and L-CSSE, they both have shown higher performance with original images; however, as shown in Table 2, similar level of drops are observed. It is to be noted that for xMSB and grid attack, L-CSSE is able to achieve relatively better performance because L-CSSE is a supervised version of autoencoder which can handle *noise* better. We also observe that changing least significant bit (LSB) does not impact the performance of deep learning algorithms. In our experiments, we observe that single bit based perturbation has minimal impact and three most significant bit based perturbation yields the maximum impact. We observe similar results for the MBGC database with performance reducing substantially in the presence of adversarial attacks. Figure 5 shows the sample ROC of VGG based face recognition on the MBGC database. Overall, deep learning based algorithms experience higher performance drop as opposed to the non-deep learning based COTS. In the case of occlusions, however, deep learning based algorithms suffer less as compared to COTS. It is our assessment that the COTS algorithm fails to perform accurate recognition with the highly limited facial region available in the low-resolution PaSC images in the presence of occlusions.

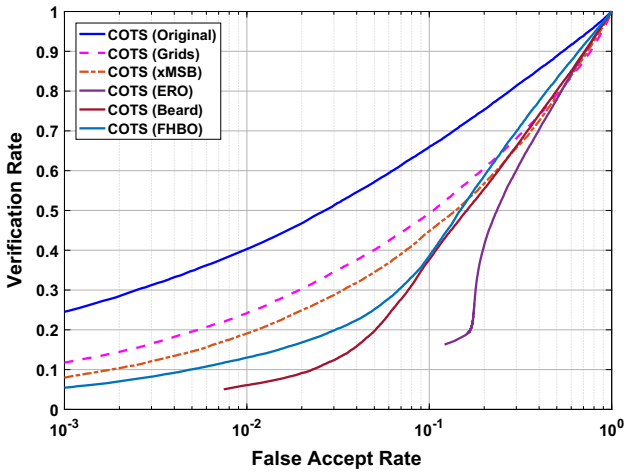
All deep learning based algorithms outperform the COTS matcher on the MEDS database with a GAR of 60–89% at 1% false accept rate (FAR) respectively as opposed to 24.1% by COTS. However, we observe that when the data is corrupted by the grids distortion, the performance of VGG and OpenFace drops by 9.83–50.28% and 17.1–49.5% respectively. In comparison, the performance of COTS drops to 21% which is only about a 3% drop. Similarly, we note that when the xMSB attack is applied, VGG and OpenFace performance drops to 45% and 43.8% showing a loss of 15% and 22.9% as opposed to 9.6% in the case of the COTS. In case of L-CSSE, noise level attacks have less impact compared to other deep learning models. As for the facial region occlusions, all the deep learning algorithms show similar trends when it comes to degradation in performance. VGG suffers a drop of 34.4% for FHBO and 12.4% for beard. OpenFace performance also degrades by 18.7% for FHBO and 18.5%



(a) OpenFace

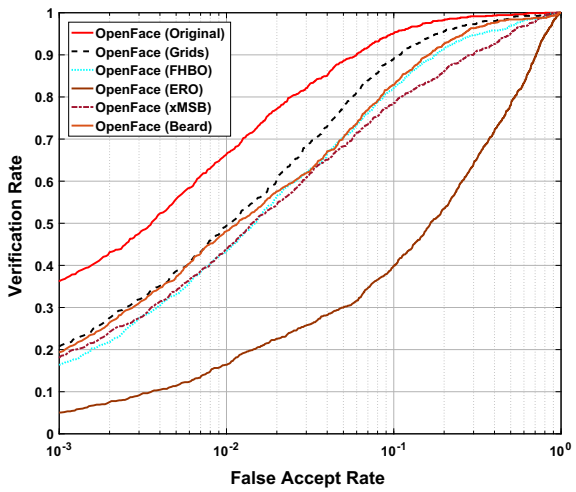


(b) VGG

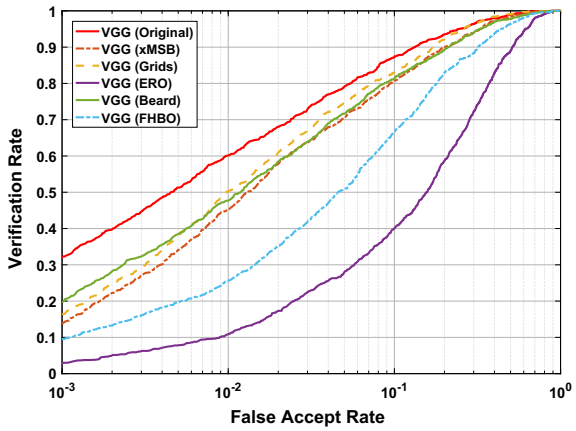


(c) COTS

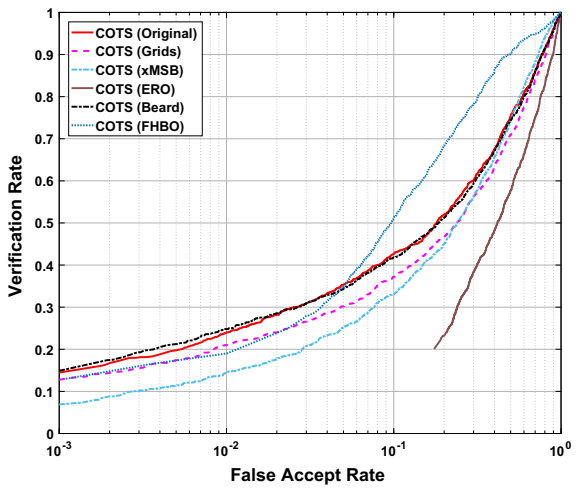
Fig. 3 Verification performance of OpenFace, VGG, and COTS under the effect of the adversarial distortions on the PaSC database



(a) OpenFace



(b) VGG



(c) COTS

Fig. 4 Verification performance of OpenFace, VGG, and COTS under the effect of the adversarial distortions on the MEDS database

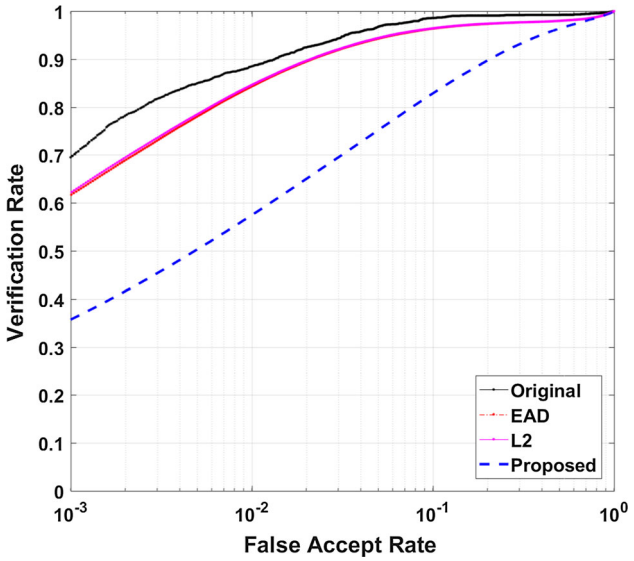


Fig. 5 Verification performance of VGG on the MBGC (20 KB) database under the effect of adversarial distortions

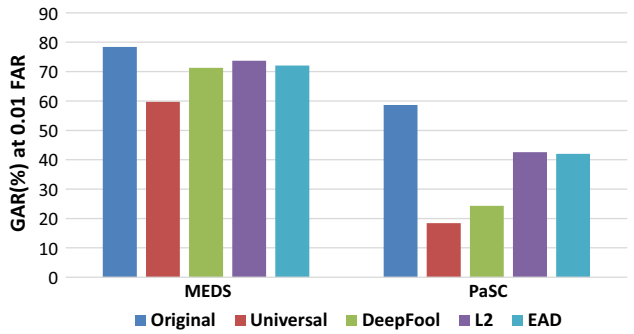


Fig. 6 Bar chart showing the effect of quasi-imperceptible adversarial perturbations on the MEDS and PaSC databases when the VGG face model is used

for beard. COTS performance drops by 5% for FHBO and notices an increase of 0.7% for the beard like occlusion. In the case of eye region occlusion, the COTS matcher suffers the most as in the case of the PaSC database, but high performance losses are also observed for both the deep learning algorithms: 50.3% for OpenFace and 49.2% for the VGG network. Similar trends are observed with Light-CNN and L-CSSE. Learning based distortions such as DeepFool and universal adversarial perturbations also have a similar effect on the performance of the VGG network as presented in Fig. 6. We notice that the performance drops significantly in the presence of distortions on the PaSC database but less so for the relatively higher quality MEDS database. This indicates that probably the effectiveness of such distortions depends on the resolution and inherent quality of the targeted images. In order to explore this further, we examine the effect of resolution where we progressively downscale the images from MEDS database by a scaling factor before applying the

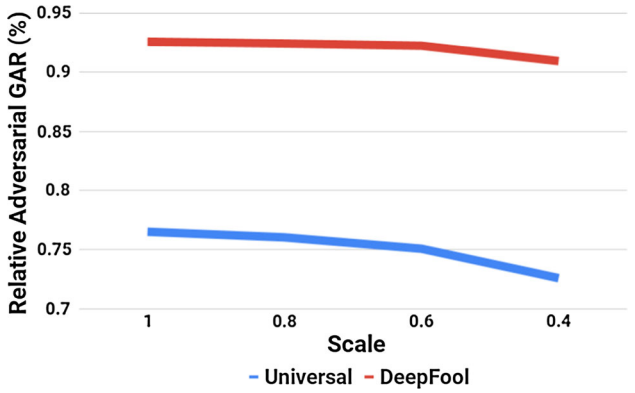


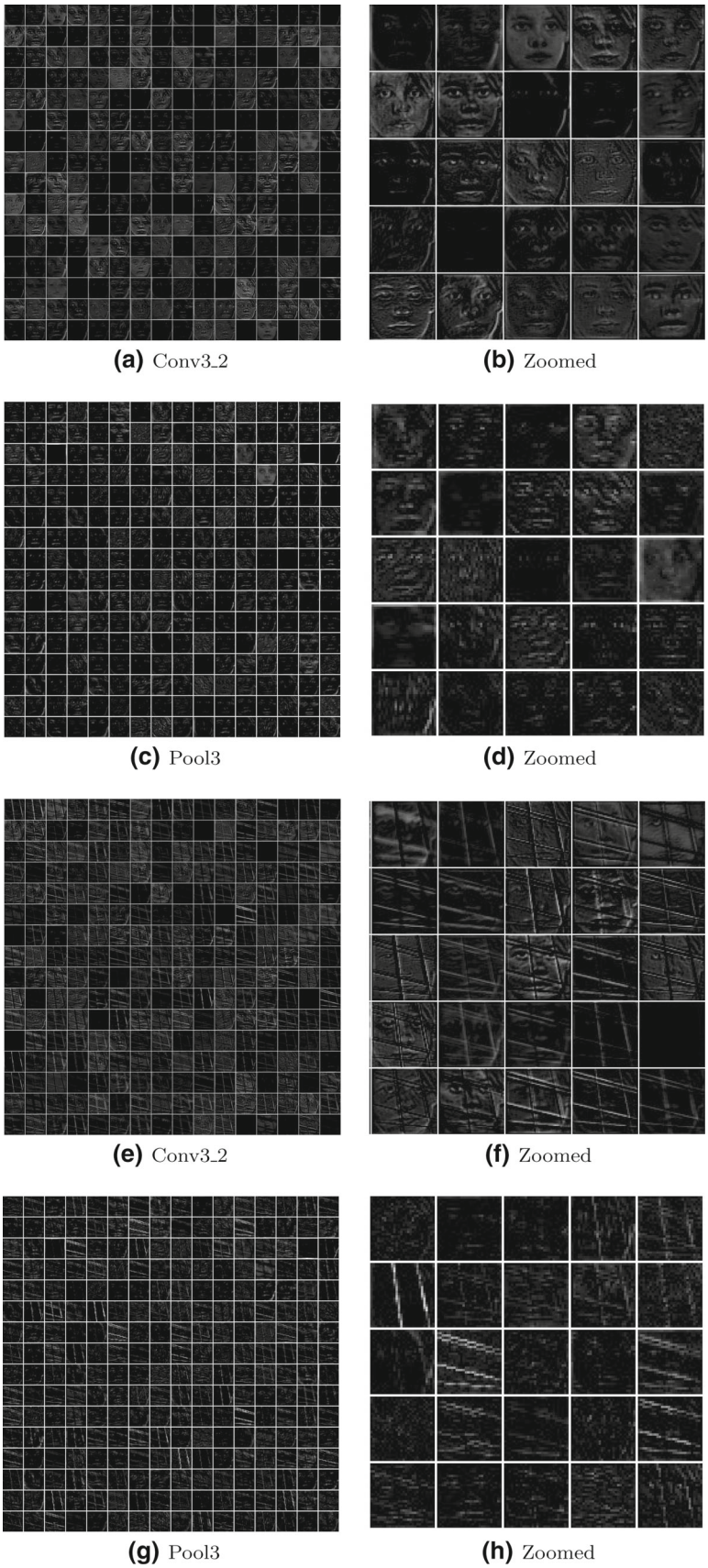
Fig. 7 Demonstrating the effect of image resolution on the impact of adversarial perturbations on the MEDS database when using the VGG face network. The relative adversarial GAR is reported at 0.01 FAR

adversarial perturbations. We compare the relative adversarial GAR at 1% FAR in each case where we define the relative adversarial GAR as: $\frac{GAR_{adv}}{GAR_{orig}}$. The results of this experiment are presented in Fig. 7. We observe that there is a consistent increase in the impact of adversarial distortions as the image resolution is reduced. Further, increasing the intensity of the perturbations by manipulating the parameter values may further deteriorate performance but the distortions will also become more visually noticeable.

5 Detection of Adversarial Attacks

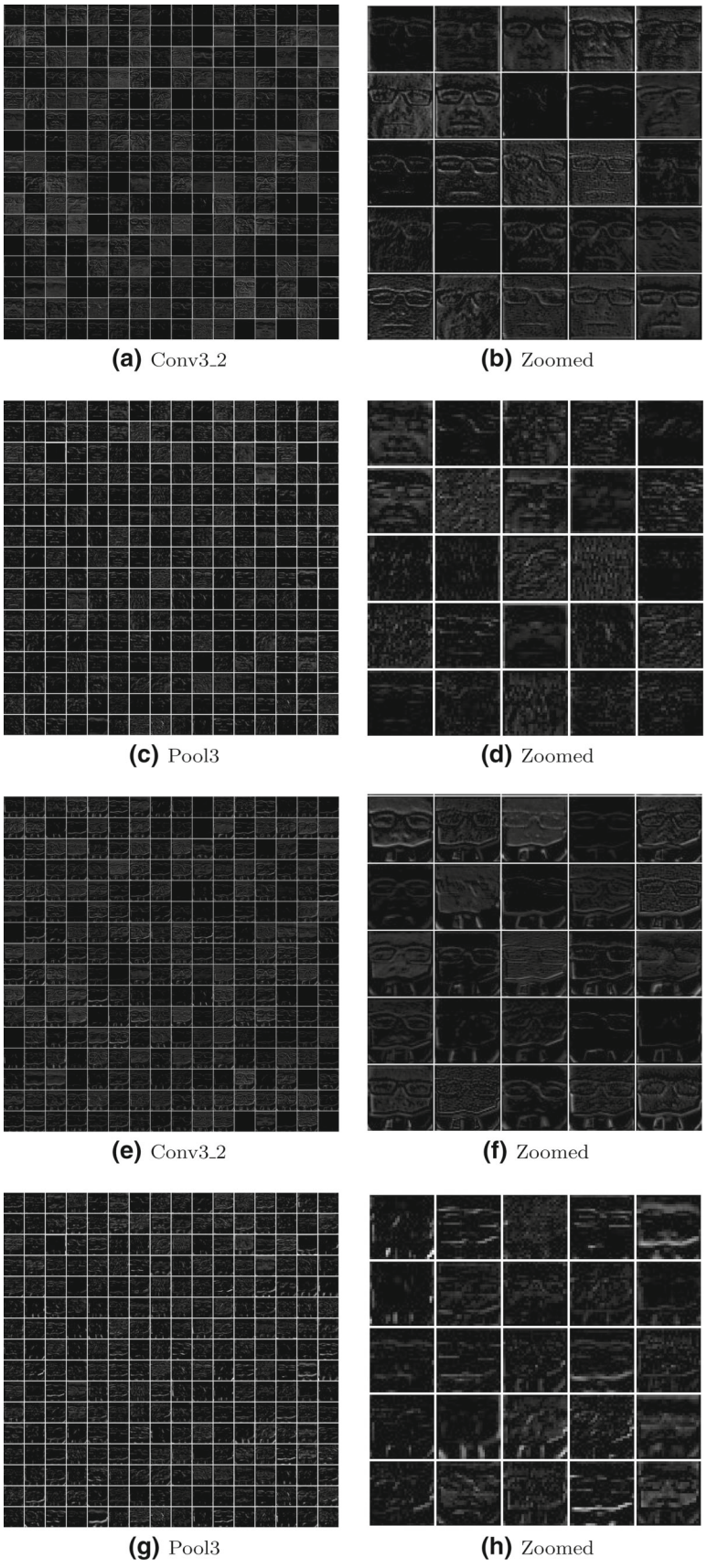
As observed in the previous section, adversarial attacks can substantially reduce the performance of usually accurate deep neural network based face recognition methods. Therefore, it is essential to address such singularities in order to make face recognition algorithms more robust and useful in real world applications. In this section, we propose novel methodologies for detecting and mitigating adversarial attacks. First, we provide a brief overview of a deep network followed by the proposed algorithms and their corresponding results.

Each layer in a deep neural network essentially learns a function or representation of the input data. The final feature computed by a deep network is derived from all of the intermediate representations in the hidden layers. In an ideal scenario, the internal representation at any given layer for an input image should not change drastically with minor changes to the input image. However, that is not the case in practice as proven by the existence of adversarial examples. The final features obtained for a distorted and undistorted image are measurably different from one another since these features map to different classes. Therefore, it is implied that the intermediate representations also vary for such cases. It is our assertion that the internal representations computed at



(a) Conv3_2 **(b)** Zoomed **(c)** Pool3 **(d)** Zoomed **(e)** Conv3_2 **(f)** Zoomed **(g)** Pool3 **(h)** Zoomed

Fig. 8 Visualizing filter responses for selected layers from the VGG network when the input image is unaltered and affected by the grids distortion. The first two rows present visualizations for conv3_2 and pool3 layers for the original input images respectively. The next two rows present visualizations for the same layers when the input images are distorted using adversarial perturbations



(a) Conv3_2 **(b)** Zoomed

(c) Pool3 **(d)** Zoomed

(e) Conv3_2 **(f)** Zoomed

(g) Pool3 **(h)** Zoomed

Fig. 9 Visualizing filter responses for selected layers from the VGG network when the input image is unaltered and affected by the beard distortion. The first two rows present visualizations for conv3_2 and pool3 layers for the original input images respectively. The next two rows present visualizations for the same layers when the input images are distorted using adversarial perturbations

each layer are different for distorted images as compared to undistorted images. Therefore, in order to detect whether an incoming image is perturbed in an adversarial manner, we decide that it is distorted if its layer-wise internal representations deviate substantially from the corresponding mean representations.

5.1 Network Analysis and Detection

In order to develop adversarial attack detection mechanism, we first analyze the filter responses in CNN architecture. Visualizations in Figs. 8 and 9 showcase the filter responses for a distorted image at selected intermediate layers and demonstrate the sensitivity towards noisy data. The propagation of the adversarial signal into the intermediate layer representations is the inspiration for our proposed detection and mitigation methodologies. We can see that many of the filter outputs primarily encode the noise instead of the input signal. We observe that the deep network based representation is more sensitive to the input and while that sensitivity results in a more expressive representation that offers higher performance in case of undistorted data, it also compromises

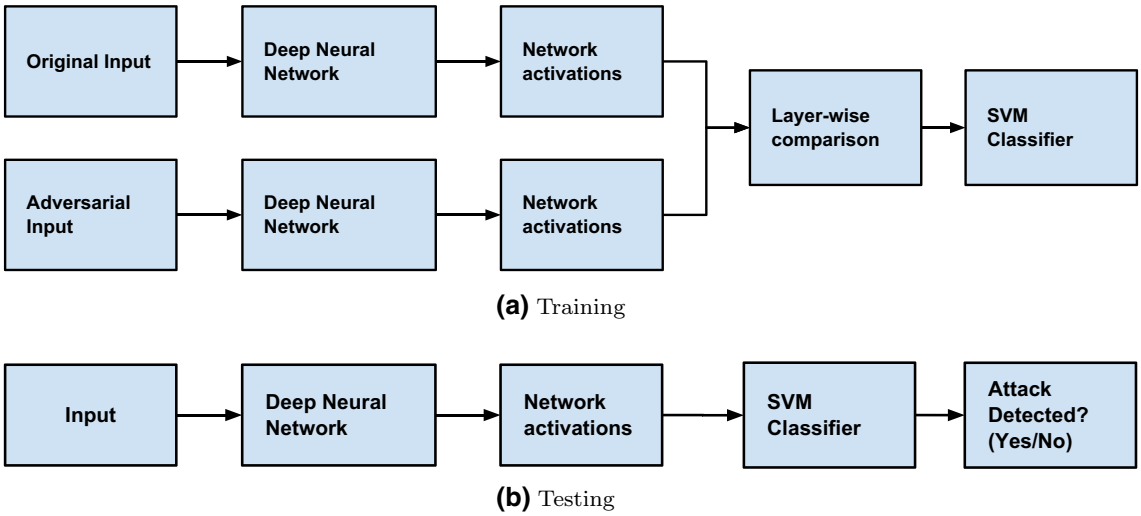


Fig. 10 **a** Training and **b** Testing view of the proposed detection framework. During training, the original input refers to the mean of the input data

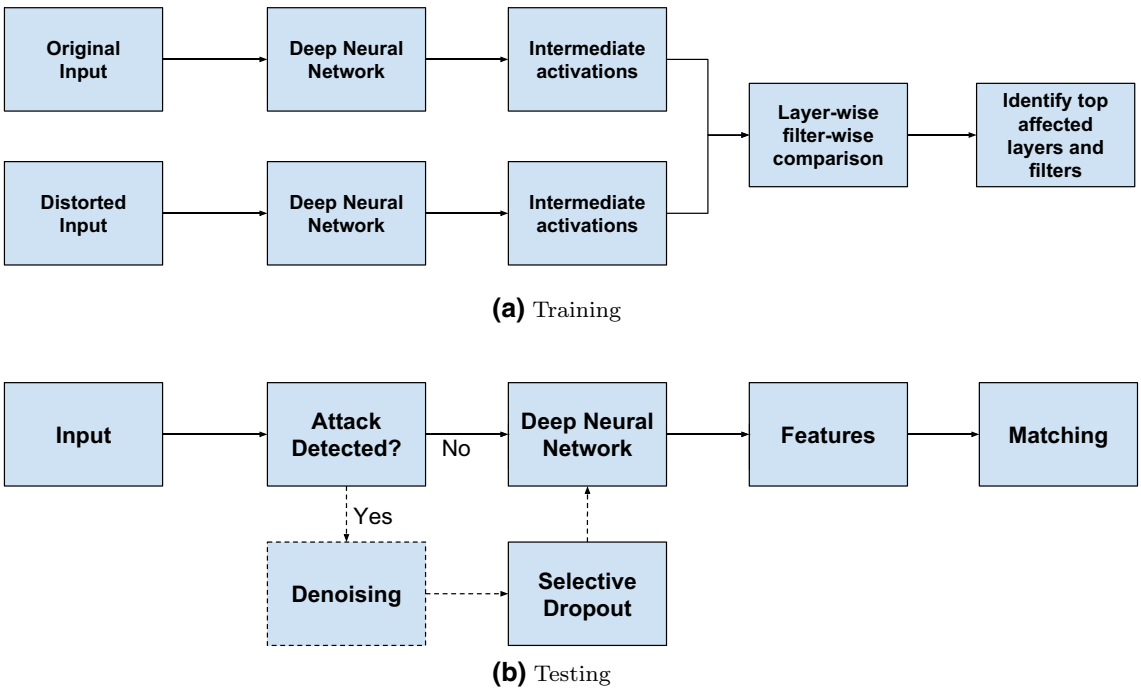


Fig. 11 **a** Training and **b** Testing view of the proposed mitigation framework

the robustness towards noise such as the proposed distortions. Since each layer in a deep network learns increasingly more complicated functions of the input data based on the functions learned by the previous layer, any noise in the input data is also encoded in the features thus leading to a higher reduction in the discriminative capacity of the final learned representation. Similar conclusions can also be drawn from the results of other existing adversarial attacks on deep networks, where the addition of a noise pattern leads to spurious classification (Goodfellow et al. 2015).

To counteract the impact of such attacks and ensure practical applicability of deep face recognition, the networks must either be made more robust towards noise at a layer level during training or it must be ensured that any input is preprocessed to filter out any such distortion prior to computing its deep representation for recognition.

In order to detect distortions we compare the pattern of the intermediate representations for undistorted images with distorted images at each layer. The differences in these patterns are used to train a classifier that can categorize an unseen input as an undistorted/distorted image. The overall flow of the detection[2] and mitigation algorithms is summarized in Figs. 10 and 11, respectively. In this research, we use the VGG (Parkhi et al. 2015) and LightCNN (Wu et al. 2018) networks to devise and evaluate our detection methodology. From the 50,248 frontal face images in the CMU Multi-PIE database (Gross et al. 2010), 40,000 are randomly selected and used to compute a set of layer-wise mean representations, μ, as follows:

$$\mu_i = \frac{1}{N_{train}} \Sigma_{j=1}^{N_{train}} \phi_i \left(I_j\right) \tag{3}$$

where, I_j is the jth image in the training set, N_{train} is the total number of training images, μ_i is the mean representation for the ith layer of the network, and $\phi_i(I_j)$ denotes the representation obtained at the ith layer of the network when I_j is the input. Once μ is computed, the intermediate representations computed for an arbitrary image I can be compared with the layer-wise means as follows:

$$\Psi_i(I, \mu) = \Sigma_z^{\lambda_i} \frac{|\phi_i(I)_z - \mu_{iz}|}{|\phi_i(I)_z| + |\mu_{iz}|} \tag{4}$$

[2] The algorithms proposed by Metzen et al. (2017) and Lu et al. (2017) have also used network responses for detecting adversarial attacks. As mentioned in Sect. 2, for real and adversarial examples, SafetyNet (Lu et al. 2017) hypothesize that the ReLU activation at the final stage of CNN follows different distributions. Based on this assumption they have discretized the ReLU maps and append an RBF SVM in the target model for adversarial examples detection. On the other hand, Metzen et al. (2017) have trained the neural network on the features of internal layers of CNN.

where, $\Psi_i(I, \mu)$ denotes the Canberra distance between $\phi_i(I)$ and μ_i, λ_i denotes the length of the feature representation computed at the ith layer of the network, and μ_{iz} denotes the zth element of μ_i. If the number of intermediate layers in the network is N_{layers}, we obtain N_{layers} distances for each image I. These distances are used as features to train a Support Vector Machine (SVM) Suykens and Vandewalle (1999) for two-class classification.

6 Mitigation of Adversarial Attacks

It is essential to take a corrective action after an adversarial attack is detected on the system. The simplest action can be to "reject" the input without any further processing and thus preventing a bad decision. In this section, we describe our mitigation approach. An ideal automated solution should not only automatically detect but also mitigate the effect of an adversarial attack so as to maintain as high performance as possible. Therefore, the next step in defending against adversarial attack is mitigation. Often a simple technique can be discarding or preprocessing (e.g. denoising) the affected regions. Our motivation comes from the same thought that there must be some excitations in the intermediate layers with highly anomalous behavior causing the final output to go out of control. If we can detect those rogue filters and layers and suppress them, we may succeed in mitigating the attack.

6.1 Mitigation: Selective Dropout

In order to accomplish these objectives, we again utilize the characteristics of the output produced in the intermediate layers of the network. We select 10,000 images from the Multi-PIE database that are partitioned into 5 mutually exclusive and exhaustive subsets of 2000 images each. Each subset is processed using a different distortion. The set of 10,000 distorted images thus obtained contains 2000 images pertaining to each of the five proposed distortions. Using this data, we compute a filter-wise score per layer that estimates the particular filter's sensitivity towards distortion as follows:

$$\epsilon_{ij} = \Sigma_{k=1}^{N_{dis}} \|\phi_{ij}(I_k) - \phi_{ij}(I_k^{'})\| \tag{5}$$

where, N_{dis} is the number of distorted images in the training set, ϵ_{ij} denotes the score and $\phi_{ij}(\cdot)$ denotes the response of the jth filter in the ith layer, I_k is the kth distorted image in the dataset, and $I_k^{'}$ is the undistorted version of I_k. Once these values are computed, the top η layers are selected based on the aggregated ϵ values for each layer. These are the layers identified to contain the most filters that are adversely affected by the distortions in data. For each of the selected η layers, the top κ fraction of affected filters are disabled by modifying the weights pertaining to 0 before computing

the features. We also apply a median filter of size 5×5 for denoising the image before extracting the features. We term this approach as *selective dropout*. It is aimed at increasing the network's robustness towards noisy data by removing the most problematic filters from the pipeline. We determine the values of parameters η and κ via grid search optimization on the training data with verification performance as the criterion.

6.2 Results of Adversarial Detection and Mitigation Algorithms

This section presents the results of the proposed detection and mitigation algorithms along with comparison with existing algorithms. For training the detection model, we use the 10,000 frontal face images from the CMU Multi-PIE database as undistorted samples. We generate 10,000 distorted samples using all five proposed distortions (discussed in Sect. 3.1 and 3.2) with 2000 images per distortion that are also randomly selected from the CMU Multi-PIE database. Each distortion based subset comprises of a 50% split of distorted and undistorted faces. These are the same sets that have been used for evaluating the performance of three face recognition algorithms.

As discussed previously, the proposed detection algorithm uses VGG and LightCNN. Since the VGG network has 20 intermediate layers, we obtain a feature vector of size 20 distances for each image. We perform a grid search based parameter optimization using the $20,000 \times 20$ training matrix to optimize and learn the VGG SVM model. Since LightCNN has fewer intermediate layers, we obtain a feature vector of size 13. Therefore, for the LightCNN SVM model, the training matrix is of size $20,000 \times 13$ and grid search based approach is used to train the SVM. Once the model is learned, any given test image is characterized by the distance vector and processed by the SVM. The score given by the model for the image to belong to the distorted class is used as the distance metric. We observe that the metric thus obtained is able to classify distorted images on unseen databases.

The mitigation algorithm is evaluated with both LightCNN and VGG networks on the PaSC, MEDS, and MBGC databases with the same experimental protocol as used in obtaining the verification results in Sect. 4. It should be noted that all of the experiments presented in the subsequent subsections are performed according to a cross-database protocol, i.e., training is performed only using the Multi-PIE database (original and distorted images) and testing is performed on the MEDS, PaSC, and MBGC databases.

6.3 Results and Analysis of Perturbation Detection

First, we present the results of the proposed algorithm in detecting whether an image contains adversarial distortions or not using the VGG and LightCNN networks. Figure 12 and Table 3 present the results of adversarial attack detection. In all the related tables and figures, the detection performance is reported in the form of detection accuracy which is the combined accuracy of correctly classifying both unperturbed and perturbed images. We choose these as the model definition and weights are publicly available. We also compare the performance of the proposed algorithm with three existing quality measures: Blind Image Quality Index (BIQI) (Moorthy and Bovik 2010), Spatial-Spectral Entropy-based Quality (SSEQ) (Liu et al. 2014), and a face-specific quality measure (Chen et al. 2015). The performance is also compared against two existing adversarial example detection algorithms: (i) Adaptive noise reduction (Liang et al. 2017), and (ii) Bayesian uncertainty (Feinman et al. 2017).

To perform detection using a quality measure, we utilize the same training data and SVM classification protocol but replacing the features with the quality score of each image. Table 3 summarizes the detection accuracies[3] of our proposed solution for each of the different types of data distortions on both the MEDS and the PaSC databases. Results on the MBGC database are presented in both seen and unseen attack protocols in Figs. 13 and 15, respectively. It is evident that the proposed algorithm outperforms both the quality based approaches with both the deep networks. Figure 12 presents the detection ROCs for the proposed algorithm. These ROCs showcase the trade-off between the false accept rate (unperturbed image detected as adversarial) and the GAR (adversarial image correctly classified as adversarial) as the threshold of detection varies. The LightCNN network based detection, i.e., when the LightCNN network is the target for the detection algorithm, performs much better for the MEDS database with the sole exception of the grids distortion. The performance on the PaSC database is high for both networks but performance at lower false accept rates is poorer for the occlusion based distortions in the case of the LightCNN network. Quality based methods are unable to perform well as distortion detectors. This is especially true for the PaSC database which contains lower quality images that are misclassified by the quality based models as distorted, thereby increasing false rejects. BIQI is an algorithm that performs quality measurement in the wavelet domain and SSEQ utilizes the Discrete Cosine Transform (DCT) coefficients for determining the quality of an image. Therefore, we assess that methods based on detecting noisy patterns in transform domains such as wavelet and DCT are not trivial solutions to perform detection of images distorted using the proposed methodology. We have also conducted experiments using LBP and DSIFT as feature descriptors and SVM as the classifier. Using the same training data and experimental protocol, we observe that the texture approaches are at least 25% less

[3] Detection accuracies are reported at equal error rate (EER).

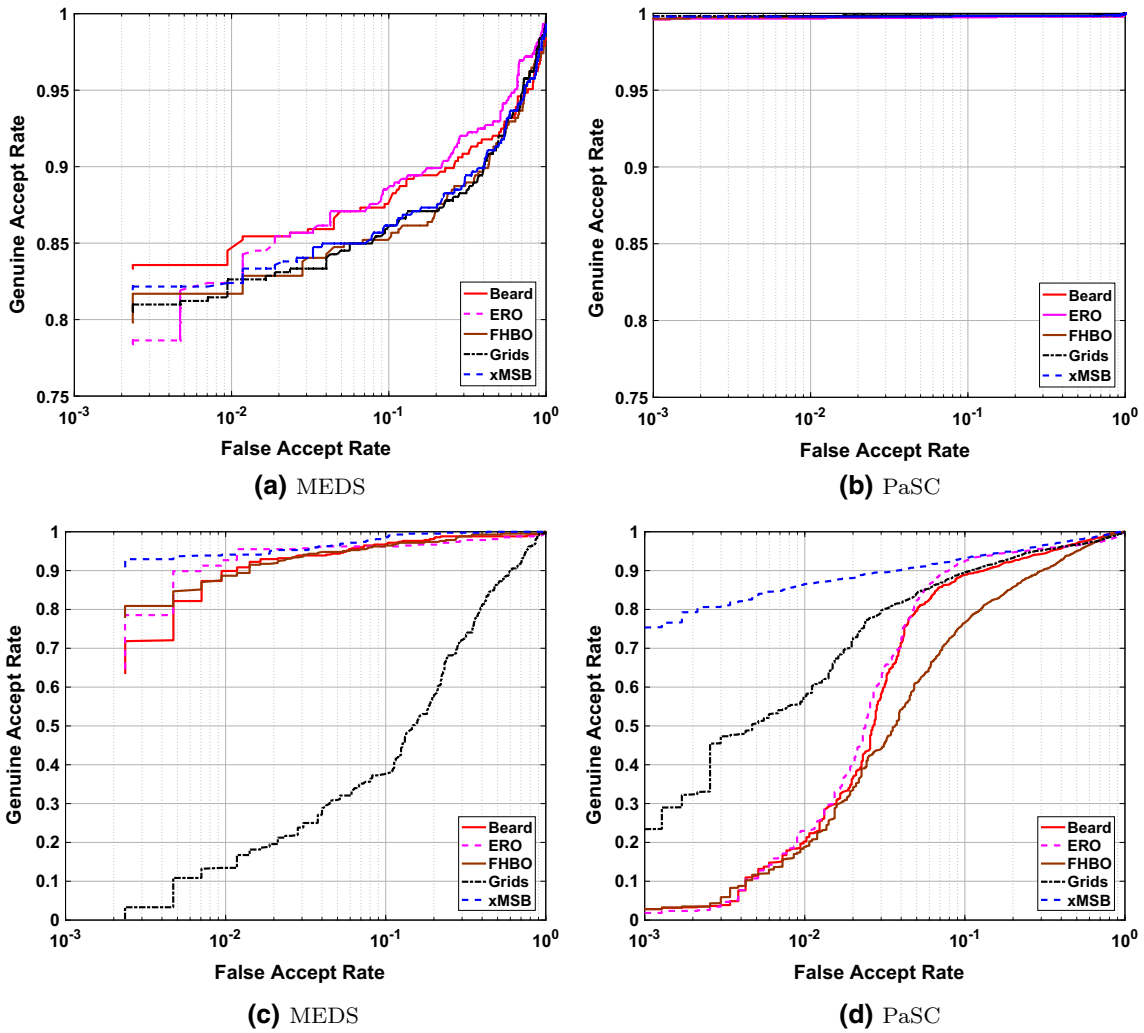


Fig. 12 ROCs for the proposed detection algorithm on the MEDS and PaSC databases with VGG (**a**, **b** upper row) and LightCNN (**c**, **d**) lower row)

Table 3 Performance of the proposed detection methodology (using LightCNN and VGG as the target networks) on MEDS and PaSC database

Database	Distortion	Face quality	BIQI	SSEQ	Adaptive noise (Liang et al. 2017)	Bayesian uncertainty (Feinman et al. 2017)	LightCNN	VGG
MEDS	Beard	60.0	64.0	43.2	81.2	80.9	**92.2**	86.8
	ERO	61.8	64.3	38.1	80.4	80.0	**91.9**	86.0
	FHBO	56.7	63.2	43.9	79.8	79.6	**92.9**	84.4
	Grids	60.7	63.7	44.4	62.1	62.4	68.4	**84.4**
	xMSB	54.3	66.6	40.9	80.2	80.9	**92.9**	85.4
PaSC	Beard	56.2	47.4	49.9	83.4	85.1	89.5	**99.8**
	ERO	56.2	48.7	51.2	84.9	84.6	90.6	**99.7**
	FHBO	53.5	52.5	51.4	78.3	77.8	81.7	**99.8**
	Grids	55.8	51.1	39.0	85.1	85.7	89.7	**99.9**
	xMSB	55.0	61.0	16.1	88.2	87.9	93.2	**99.8**

Bold values indicate the best performance value in each criterion
Grids grid based occlusion, *xMSB* most significant bit based noise, *FHBO* forehead and brow occlusion, *ERO* eye region occlusion, and *Beard* beard like occlusion

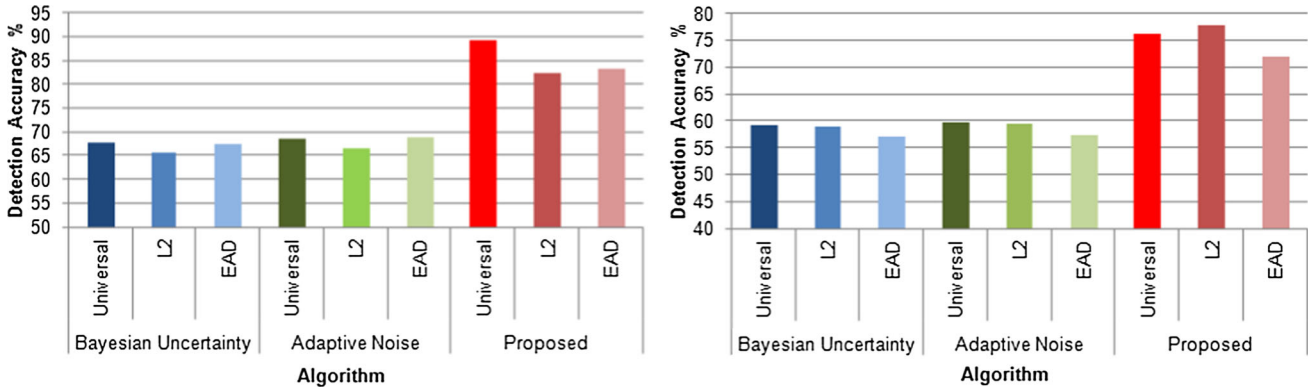


Fig. 13 Performance of the proposed detection methodology (using VGG as the target network) on MBGC 8 (Left) and MBGC 20 (Right) databases in 'intra' attack setting

accurate than the proposed algorithm. Furthermore, we have performed the comparative experiments with neural network classifier (in place of SVM) and observe that, across different attacks, SVM outperforms neural network classifier by 20–30%.

The proposed detection algorithm performs almost perfectly for the PaSC database with the VGG network and maintains accuracies of 81.7–93.2% with the LightCNN network. The lowest performance is observed on the MEDS database (classification accuracy of 68.4% with the LightCNN network). The lower accuracies with the LightCNN can be attributed to the smaller network depth which results in smaller size features to be utilized by the detection algorithm. It is to be noted that the proposed algorithm maintains high true positive rates even at very low false positive rates across all distortions on the three databases which is desirable when the cost of accepting a distorted image is much higher than a false reject for the system. We also observe that the quality based algorithms struggle with high resolution distorted images and low resolution undistorted images, classifying them as undistorted and distorted respectively. Besides exceptionally poor quality images that are naturally quite distorted, we observe that high or low illumination results in false rejects by the algorithm, i.e., falsely detected as distorted. This shows the scope of further improvement and refinement in the detection methodology. This is also another reason for lower performance with the MEDS database which has more extreme illumination cases as compared to PaSC. We observe both general no-reference image quality measures and face-specific quality measures to also be insufficient for attack detection. The Bayesian uncertainty and adaptive noise reduction algorithms do perform better than the quality-based metrics, but are outperformed by the proposed algorithm. We also test using the Viola Jones face detector (Viola and Jones 2004) and find that, on average, approximately 60% of the distorted faces pass face detection. Therefore, the distorted face images cannot be differentiated from undistorted faces on the basis of failing face detection.

We attempt to reduce the feature dimensionality to deduce the most important features using sequential feature selection based on classification loss by a SVM model learned on a given subset of features. For the VGG based model, using just the top 6 features for detection, we obtain an average accuracy of 81.7% on MEDS and 96.9% on PaSC database across all distortions. If we use only one most discriminative feature to perform detection, we obtain 79.3% accuracy on MEDS and 95.8% on PaSC on average across all distortions. This signifies that comparing the representations computed by the network in its intermediate layers indeed produces a good indicator of the existence of distortions in a given image. Finally, in Eq. 4, in place of Canberra distance, we experimented with other metrics such as l_1, l_2, and Cosine. For adversarial perturbation detection, Canberra distance shows the best performance over other measures. For example, on the MEDS database, it yields at least 4.6% better detection accuracy compared to l_1, l_2, and Cosine measures.

6.4 Performance on Quasi-imperceptible Attacks

In addition to the proposed adversarial attacks, we have also evaluated the efficacy of the proposed detection methodology on four existing attacks that utilize network architecture information for adversarial perturbation generation, i.e., DeepFool (Moosavi-Dezfooli et al. 2016), Universal adversarial perturbations (Moosavi-Dezfooli et al. 2017), l_2 (Carlini and Wagner 2017), and EAD (Chen et al. 2018). We have also compared the performance of the proposed detection algorithm with two recent adversarial detection techniques based on adaptive noise reduction (Liang et al. 2017) and Bayesian uncertainty (Feinman et al. 2017). The same training data and protocol was used to train and test all three detection approaches as specified in Sect. 4. The results of detection are presented in Figs. 13 and 14. We observe that the proposed methodology is at least 11% better at detecting DNN architecture based adversarial attacks

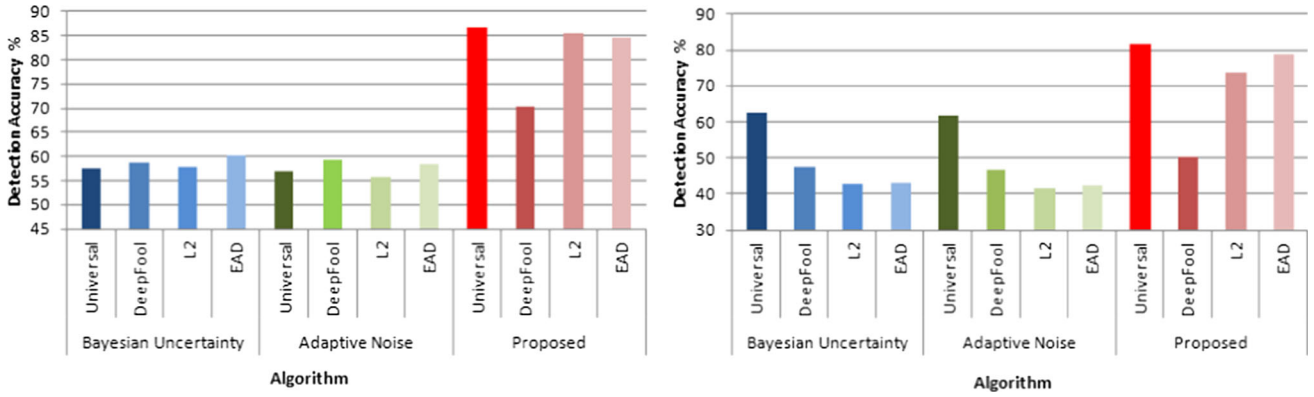


Fig. 14 Summarizing the results of the proposed and existing detection algorithms on the PaSC (Left) and MEDS (Right) databases

as compared to the existing algorithms for all cases except for detecting DeepFool perturbed images from the MEDS database where it still outperforms the other approaches by more than 3%. We believe that this is due to the fact that MEDS has overall higher image quality as compared to PaSC and even the impact of these near imperceptible perturbations on verification performance is minimal for the database. Therefore, it is harder to distinguish original images from perturbed images for these distortions for all the tested detection algorithms.

We have also performed the experiments with a distortion-invariant protocol and compared the performance with two existing algorithms as well. The results of distortion-invariant protocol are given in Table 4. In these experiments, the training is done on all perturbations except for one and testing is done on the unseen perturbation not used in training. The cross-attack experiment is performed using the MPIE database for training and the MEDS, PaSC, and MBGC databases for testing so the experiment is also cross-database. Following this protocol, we observe that the proposed detection algorithm is still able to achieve 63.2% accuracy on the PaSC database (Table 4) when tested on the universal perturbation and trained on the other distortions. In comparison, the existing approaches [Adaptive Noise Reduction, Liang et al. (2017) and Bayesian Uncertainty, Feinman et al. (2017)] are only able to achieve a maximum of 41.3% accuracy on the MEDS and 47.1% accuracy on the PaSC database. The proposed algorithm outperforms these existing approaches for the other cases as well by a margin of at least 12% on the MEDS database and 16% on the PaSC database. As shown in Fig. 15, we observe similar results on the MBGC database on both the 8 KB and 20 KB sets. Thus we conclude that the proposed algorithm is able to better generalize its detection performance even in the case of attacks that it has never seen during training.

Table 4 Adversarial perturbation detection accuracy of the proposed detection methodology (using VGG as the target network) where all but one distortions are used for training and the remaining unseen distortion is used for testing

Distortion	Algorithm	Database	
		MEDS	PaSC
DeepFool	Proposed	56.1	50.6
	Bayesian (Feinman et al. 2017)	38.2	34.4
	Adaptive (Liang et al. 2017)	38.9	34.1
Universal	Proposed	53.4	63.2
	Bayesian (Feinman et al. 2017)	40.8	46.7
	Adaptive (Liang et al. 2017)	41.3	47.1
l_2	Proposed	55.5	63.6
	Bayesian (Feinman et al. 2017)	38.6	39.2
	Adaptive (Liang et al. 2017)	39.2	40.1
EAD	Proposed	59.2	62.7
	Bayesian (Feinman et al. 2017)	40.6	42.1
	Adaptive (Liang et al. 2017)	41.5	42.2
Proposed distortions	Proposed	58.1	53.9
	Bayesian (Feinman et al. 2017)	37.6	32.2
	Adaptive (Liang et al. 2017)	38.9	32.8

The proposed entry in the distortion column refer to the results on the proposed image-level and face-level distortions as detailed in Sects. 3.1 and 3.2

6.5 Results and Analysis of Mitigation Algorithm

The proposed technique of selective dropout shows interesting performance. Figure 16 and Table 5 present the results for the mitigation algorithm. Mitigation is a two-step process to enable better performance and computational efficiency. First, the detection algorithm is used to detect the perturbed/adversarial images. Secondly, the proposed mitigation algorithm is applied to only those images that are predicted as adversarial by the detection algorithm. Face verification

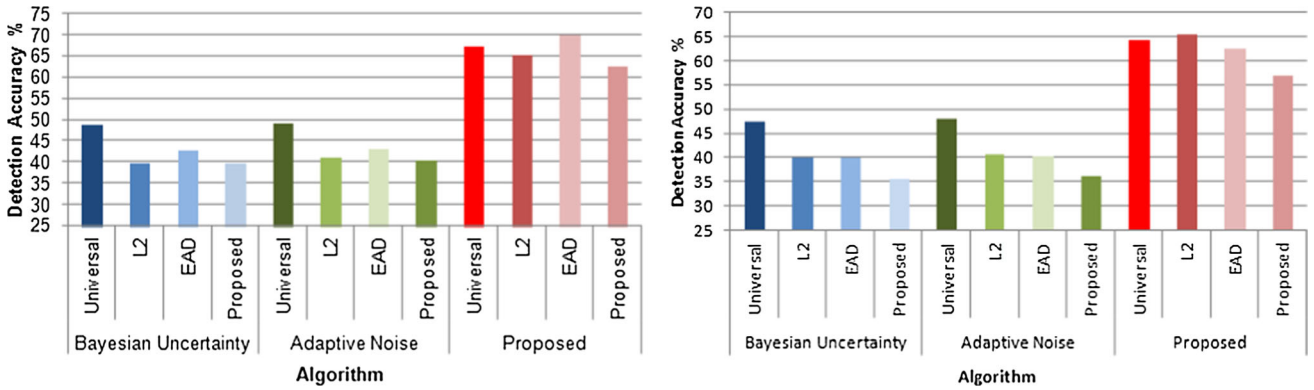


Fig. 15 Performance of the proposed detection methodology (using VGG as the target network) on MBGC 8 (Left) and MBGC 20 (Right) databases where all but one distortions are used for training and the remaining unseen distortion is used for testing

results after applying the proposed mitigation algorithm on the MEDS and PaSC databases are presented in Fig. 16. We can observe that the mitigation model is able to improve the verification performance with both the networks and bring it closer to the original curve. For instance, as shown in Table 5, in the case of the MBGC database (20 KB), the performance drops from 88.5 to 75.9%, which is almost a 13% decrease. The proposed mitigation algorithm is able to boost this performance back to 86.4% which is only a 2.1% drop in performance compared to the original. Thus, we see that even discarding a certain fraction of the intermediate network output that is most affected by adversarial distortions, results in better recognition than incorporating them into the obtained feature vector. We have conducted one more study, where we have used normalized inner product for mitigation in place of l_2-norm. The results of this study are presented in Table 6. We have observed that using normalized inner product on the larger and more challenging PaSC database in the mitigation algorithm reduces the mitigated verification performance at equal error rate (EER) by 1.5%.

To further analyze the contributions of the two different stages of the mitigation algorithm, we assess the mitigation performance when only one of them is applied in isolation. The results for this experiment are summarized in Table 7. We observe that selective dropout is comparatively more effective than just applying the median filter, but the combined result is much better than either of the stages on their own. We also evaluate how the two hyperparameters, η and κ, impact the performance of the proposed algorithm. These results are presented in Table 8. We observe that for the higher quality MEDS database, increasing the overall number of filters dropped per layer results in improved performance as long as η is not increased to 5. However, for the PaSC database, increasing the number of filters dropped per layer to 0.1 (or 10%) results in a substantial drop in performance, even lower than what median filter alone can accomplish in the case of $\eta = 3$ and $\eta = 5$. We assess that higher quality faces provide better scope for dropping more layers and filters per layer to improve the mitigation performance whereas the values of the parameters must be set carefully for lower quality faces. Finally, in a cross attack mitigation setting, we observe that the proposed mitigation algorithm can transfer to sim-

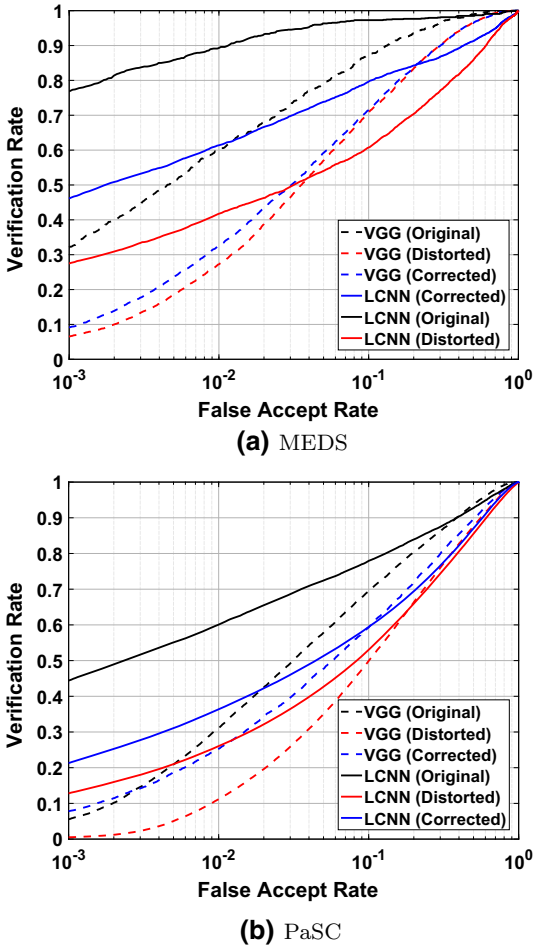


Fig. 16 ROCs for the proposed mitigation algorithm on the **a** MEDS and **b** PaSC databases

Table 5 Mitigation results on the MEDS, PaSC, and MBGC databases

Database	LightCNN			VGG		
	Original	Distorted	Corrected	Original	Distorted	Corrected
PaSC	60.5	25.9	**36.2**	54.3	14.6	**24.8**
MEDS	89.3	41.6	**61.3**	78.4	30.5	**40.6**
MBGC (8 KB)	86.9	75.4	**86.2**	51.8	44.1	**49.5**
MBGC (20 KB)	88.5	75.9	**86.4**	52.7	44.3	**50.3**

We report GAR (%) at 1% FAR
Bold values indicate the best performance value in each criterion

Table 6 Mitigation results for DeepFool perturbation on the MEDS and PaSC databases using l_2-norm and inner-product

Database	GAR (%) at EER	GAR (%) at 1% FAR
MEDS (Original)	93.3	78.4
MEDS (Perturbed)	93.2	78.8
MEDS (Corrected, l_2-norm)	93.4	78.7
MEDS (Corrected, Inner Product)	**93.8**	**79.8**
PaSC (Original)	84.8	54.3
PaSC (Perturbed)	79.8	28.6
PaSC (Corrected, l_2-norm)	**79.5**	28.8
PaSC (Corrected, Inner Product)	78.0	**29.1**

EER refers to Equal Error Rate
Bold values indicate the best performance value in each criterion

Table 7 Mitigation Results on the MEDS, PaSC, and MBGC databases when the median filter (denoted as Median) and selective dropout (denoted as Dropout) stages of the proposed mitigation algorithm are applied in isolation on the distorted data

Database	LightCNN			VGG		
	Median	Selective dropout	Combined	Median	Selective droput	Combined
PaSC	28.6	31.1	36.2	19.4	21.0	24.8
MEDS	52.5	57.4	61.3	33.9	36.7	40.6
MBGC (8 KB)	77.6	81.7	86.2	46.6	48.2	49.5
MBGC (20 KB)	78.4	82.1	86.4	45.7	47.6	50.3

We report GAR (%) at 1% FAR

Table 8 Evaluating the effect of the hyperparameters on the performance of the proposed mitigation algorithm. We report the GAR (%) at 0.01 FAR when the VGG network is used for the MEDS and PaSC databases as the values of η and κ are varied

	MEDS			PaSC		
	$\kappa = 0.03$	$\kappa = 0.05$	$\kappa = 0.1$	$\kappa = 0.03$	$\kappa = 0.05$	$\kappa = 0.1$
$\eta = 1$	34.1	35.7	36.9	19.7	20.4	20.8
$\eta = 3$	38.6	40.6	**41.2**	22.7	**24.8**	19.3
$\eta = 5$	40.1	39.4	37.5	20.3	19.1	18.7

Bold values indicate the best performance value in each criterion

ilar unseen image processing operations (e.g. grid based to xMSB) but requires further research in significantly dissimilar attacks.

7 Conclusion and Future Research Directions

To summarize, our work has four main contributions: (i) a framework to evaluate robustness of deep learning based face recognition engines, (ii) a scheme to detect adversarial attacks on the system, (iii) methods to mitigate adversarial attacks when detected, and (iv) perform the detection and mitigation in a cross-database and cross-attack scenario which closely resembles a real-world scenario. Playing the role of an expert level adversary, we propose five classes of image distortions in the evaluation experiment. Using an open source implementation of FaceNet, i.e., OpenFace, and the VGG-Face, LightCNN, and L-CSSE networks, we conduct a series

of experiments on the publicly available PaSC, MEDS, and MBGC databases. We observe a substantial loss in the performance of the deep learning based systems when compared with a non-deep learning based COTS matcher for the same evaluation data. In order to detect the attacks, we propose a network activation analysis based method in the hidden layers of the network. When an attack is reported by this stage, we invoke mitigation methods described in the paper to show that we can recover from the attacks in many situations. In the future, we will build more efficient mitigation frameworks to restore to normal level of performance. Further, there is a requirement to make the proposed defense (both detection and mitigation) robust to unseen attacks, both physical [for example, disguise Singh et al. (2019) and spoofing Ramachandra and Busch (2017)] and digital [for example, adversarial, morphing Agarwal et al. (2017a), and retouching Bharati et al. (2016)]. It is our assertion that with these findings, future research can be aimed at correcting such adversarial samples and incorporating various other kinds of countermeasures in deep neural networks to further increase their robustness.

Acknowledgements G. Goswami was partly supported through IBM PhD Fellowship, A. Agarwal is partly supported by Visvesvaraya PhD Fellowship, and M. Vatsa and R. Singh are partly supported through CAI@IIIT-Delhi. M. Vatsa is also partially supported through Department of Science and Technology, Government of India through Swarnajayanti Fellowship.

References

Addad, B., Kodjabashian, J.,& Meyer, C. (2018). *Clipping free attacks against artificial neural networks*. arXiv preprint arXiv:1803.09468.

Agarwal, A., Singh, R., & Vatsa, M. (2016). Face anti-spoofing using haralick features. In *2016 IEEE 8th international conference on biometrics theory, applications and systems* (pp. 1–6).

Agarwal, A., Singh, R., Vatsa, M., & Noore, A. (2017a). SWAPPED! Digital face presentation attack detection via weighted local magnitude pattern. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (pp. 659–665). https://doi.org/10.1109/BTAS.2017.8272754.

Agarwal, A., Singh, R., Vatsa, M., & Ratha, N. (2018). Are image-agnostic universal adversarial perturbations for face recognition difficult to detect? In *IEEE international conference on biometrics: Theory, applications, and systems*.

Agarwal, A., Yadav, D., Kohli, N., Singh, R., Vatsa, M., & Noore, A. (2017b). Face presentation attack with latex masks in multispectral videos. In *IEEE conference on computer vision and pattern recognition workshops* (pp. 275–283).

Akbulut, Y., Şengür, A., Budak, Ü., & Ekici, S. (2017). Deep learning based face liveness detection in videos. In *2017 international artificial intelligence and data processing symposium (IDAP)* (pp. 1–4). Malatya. https://doi.org/10.1109/IDAP.2017.8090202.

Akhtar, N., Liu, J., & Mian, A. (2017). *Defense against universal adversarial perturbations*. arXiv preprint arXiv:1711.05929.

Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, *6*, 14410–14430.

Alaifari, R., Alberti, G. S., & Gauksson, T. (2018). *Adef: An iterative algorithm to construct adversarial deformations*. arXiv preprint arXiv:1804.07729.

Amos, B., Ludwiczuk, B., Harkes, J., Pillai, P., Elgazzar, K., & Satyanarayanan, M. (2016). *OpenFace: Face recognition with deep neural networks*. http://github.com/cmusatyalab/openface. Accessed 10 Apr 2016.

Athalye, A., & Sutskever, I. (2018). Synthesizing robust adversarial examples. In *International conference on machine learning*.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404–417).

Beveridge, J., Phillips, P., Bolme, D., Draper, B., Given, G., Lui, Y. M., Teli, M., Zhang, H., Scruggs, W., Bowyer, K., Flynn, P., & Cheng, S. (2013). The challenge of face recognition from digital point-and-shoot cameras. In *IEEE conference on biometrics: Theory, applications and systems*

Bhagoji, A. N., Cullina, D., & Mittal, P. (2017). *Dimensionality reduction as a defense against evasion attacks on machine learning classifiers*. arXiv preprint arXiv:1704.02654.

Bharati, A., Singh, R., Vatsa, M., & Bowyer, K. W. (2016). Detecting facial retouching using supervised deep learning. *IEEE Transactions on Information Forensics and Security*, *11*(9), 1903–1913.

Biggio, B., Fumera, G., Marcialis, G. L., & Roli, F. (2017). Statistical meta-analysis of presentation attacks for secure multibiometric systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(3), 561–575.

Boulkenafet, Z., Komulainen, J., & Hadid, A. (2016). Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security*, *11*(8), 1818–1830.

Boulkenafet, Z., Komulainen, J., & Hadid, A. (2017). Face antispoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters*, *24*(2), 141–145.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain separation networks. *Advances in Neural Information Processing Systems*, *29*, 343–351.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *IEEE symposium on security and privacy* (pp. 39–57).

Chen, J., Deng, Y., Bai, G., & Su, G. (2015). Face image quality assessment based on learning to rank. *IEEE Signal Processing Letters*, *22*(1), 90–94.

Chen, P. Y., Sharma, Y., Zhang, H., Yi, J., & Hsieh, C. J. (2018). EAD: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*.

Chhabra, S., Singh, R., Vatsa, M., & Gupta, G. (2018). Anonymizing k-facial attributes via adversarial perturbations. In *International joint conferences on artificial intelligence* (pp. 656–662).

Chingovska, I., Erdogmus, N., Anjos, A., & Marcel, S. (2016). Face recognition systems under spoofing attacks. In T. Bourlai (Ed.), *Face recognition across the imaging spectrum*. Cham: Springer. https://doi.org/10.1007/978-3-319-28501-6_8.

Cisse, M. M., Adi, Y., Neverova, N., & Keshet, J. (2017). Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Advances in neural information processing systems* (pp. 6977–6987).

Das, N., Shanbhogue, M., Chen, S. T., Hohman, F., Chen, L., Kounavis, M. E., & Chau, D. H. (2017). *Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression*. arXiv preprint arXiv:1705.02900.

de Souza, G. B., da Silva Santos, D. F., Pires, R. G., Marana, A. N., & Papa, J. P. (2017). Deep texture features for robust face spoofing

detection. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *64*(12), 1397–1401.
Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F.-F. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition* (pp. 248–255).
Dziugaite, G. K., Ghahramani, Z., & Roy, D. M. (2016). *A study of the effect of jpg compression on adversarial images*. arXiv preprint arXiv:1608.00853.
Feinman, R., Curtin, R. R., Shintre, S., & Gardner, A. B. (2017). *Detecting adversarial samples from artifacts*. arXiv preprint arXiv:1703.00410.
Gan, J., Li, S., Zhai, Y., & Liu, C. (2017). 3d convolutional neural network based on face anti-spoofing. In *2017 2nd international conference on multimedia and image processing (ICMIP)* (pp. 1–5).
Goel, A., Singh, A., Agarwal, A., Vatsa, M., & Singh, R. (2018). Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition. In *IEEE International conference on biometrics: Theory, applications, and systems*
Gong, Z., Wang, W., & Ku, W. S. (2017). *Adversarial and clean data are not twins*. arXiv preprint arXiv:1704.04960.
Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International conference on learning representations*.
Goswami, G., Ratha, N., Agarwal, A., Singh, R., & Vatsa, M. (2018). Unravelling robustness of deep learning based face recognition against adversarial attacks. In *Association for the advancement of artificial intelligence*.
Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-PIE. *Image and Vision Computing*, *28*(5), 807–813.
Grosse, K., Manoharan, P., Papernot, N., Backes, M., & McDaniel, P. (2017). *On the (statistical) detection of adversarial examples*. arXiv preprint arXiv:1702.06280.
Gu, S., & Rigazio, L. (2014). *Towards deep neural network architectures robust to adversarial examples*. arXiv preprint arXiv:1412.5068.
Guo, C., Rana, M., Cissé, M., & van der Maaten, L. (2018). Countering adversarial images using input transformations. In *International conference on learning representations*.
Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *Stat*, *1050*, 9.
Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07–49, University of Massachusetts, Amherst.
King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758.
Kurakin, A., Goodfellow, I., & Bengio, S. (2016). *Adversarial examples in the physical world*. arXiv preprint arXiv:1607.02533.
Laskov, P., & Lippmann, R. (2010). Machine learning in adversarial environments. *Machine Learning*, *81*(2), 115–119.
Lee, H., Han, S., & Lee, J. (2017). *Generative adversarial trainer: Defense to adversarial perturbations with gan*. arXiv preprint arXiv:1705.03387.
Li, X., & Li, F. (2017). Adversarial examples detection in deep networks with convolutional filter statistics. In *International conference on computer vision*.
Liang, B., Li, H., Su, M., Li, X., Shi, W., & Wang, X. (2017). *Detecting adversarial examples in deep networks with adaptive noise reduction*. URL arXiv:1705.08378
Liu, J., Deng, Y., Bai, T., & Huang, C. (2015). *Targeting ultimate accuracy: Face recognition via deep embedding*. URL arXiv:1506.07310.
Liu, L., Liu, B., Huang, H., & Bovik, A. C. (2014). No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image Communication*, *29*(8), 856–863.
Liu, M. Y., & Tuzel, O. (2016). Coupled generative adversarial networks. *Advances in Neural Information Processing Systems*, *29*, 469–477.
Lu, J., Issaranon, T., & Forsyth, D. (2017). Safetynet: Detecting and rejecting adversarial examples robustly. In *IEEE international conference on computer vision* (pp. 446–454).
Luo, Y., Boix, X., Roig, G., Poggio, T., & Zhao, Q. (2015). *Foveation-based mechanisms alleviate adversarial examples*. arXiv preprint arXiv:1511.06292.
Majumdar, A., Singh, R., & Vatsa, M. (2017). Face verification via class sparsity based supervised encoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1273–1280.
Manjani, I., Tariyal, S., Vatsa, M., Singh, R., & Majumdar, A. (2017). Detecting silicone mask-based presentation attack via deep dictionary learning. *IEEE Transactions on Information Forensics and Security*, *12*(7), 1713–1723.
Meng, D., & Chen, H. (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 135–147).
Metzen, J. H., Genewein, T., Fischer, V., & Bischoff, B. (2017). On detecting adversarial perturbations. In *International conference on learning representations*.
Miyato, T., Dai, A. M., & Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. In *International conference on learning representations*.
Moorthy, A. K., & Bovik, A. C. (2010). A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, *17*(5), 513–516.
Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1765–1773).
Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2574–2582).
Multiple encounters dataset (MEDS). (2011). Retrieved October 6, 2017 from http://www.nist.gov/itl/iad/ig/sd32.cfm.
Nayebi, A., & Ganguli, S. (2017). *Biologically inspired protection of deep networks from adversarial attacks*. arXiv preprint arXiv:1703.09202.
Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE conference on computer vision and pattern recognition* (pp. 427–436).
NIST face recognition vendor test ongoing. (2018). Retrieved December 10, 2017 from https://www.nist.gov/programs-projects/face-recognition-vendor-test-frvt-ongoing.
Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia conference on computer and communications security* (pp. 506–519). ACM.
Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016a). The limitations of deep learning in adversarial settings. In *IEEE European symposium on security and privacy* (pp. 372–387).
Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016b). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy* (pp. 582–597).
Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *British machine vision conference* (vol. 1, p. 6).
Patel, K., Han, H., Jain, A. K., & Ott, G. (2015). Live face video vs. spoof face video: Use of moire patterns to detect replay video attacks. In *2015 international conference on biometrics* (pp. 98–105).

Phillips, P. J., Flynn, P. J., Beveridge, J. R., Scruggs, W., O'Toole, A. J., Bolme, D., Bowyer, K. W., Draper, B. A., Givens G. H., Lui, Y. M., Sahibzada, H., Scallan, J. A., & Weimer, S. (2009). Overview of the multiple biometrics grand challenge. In *Advances in biometrics*, (pp. 705–714).

Prakash, A., Moran, N., Garber, S., DiLillo, A., & Storer, J. (2018). Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8571–8580).

Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv preprint arXiv:1511.06434.

Raghavendra, R., Venkatesh, S., Raja, K., Cheikh, F., & Busch, C. (2017). On the vulnerability of extended multispectral face recognition systems towards presentation attacks. In *IEEE international conference on identity, security and behavior analysis*.

Rakin, A. S., Yi, J., Gong, B., & Fan, D. (2018). *Defend deep neural networks against adversarial examples via fixed anddynamic quantized activation functions*. arXiv preprint arXiv:1807.06714.

Ramachandra, R., & Busch, C. (2017). Presentation attack detection methods for face recognition systems: A comprehensive survey. *ACM Computing Survey, 50*(1), 8:1–8:37.

Ranjan, R., Sankaranarayanan, S., Castillo, C. D., & Chellappa, R. (2017). *Improving network robustness against adversarial attacks with compact convolution*. arXiv preprint arXiv:1712.00699.

Ratha, N. K., Connell, J. H., & Bolle, R. M. (2001). An analysis of minutiae matching strength. In *Audio- and video-based biometric person authentication: Third international conference, proceedings* (pp. 223–228).

Rauber, J., Brendel, W., & Bethge, M. (2017). *Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models*. URL arXiv:1707.04131.

Ross, A. S., & Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*.

Rozsa, A., Günther, M., & Boult, T. E. (2017a). LOTS about attacking deep features. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (pp. 168–176). Denver, CO. https://doi.org/10.1109/BTAS.2017.8272695.

Rozsa, A., Günther, M., Rudd, E. M., & Boult, T. E. (2016). Are facial attributes adversarially robust? In *International conference on pattern recognition* (pp. 3121–3127).

Rozsa, A., Günther, M., Rudd, E. M., & Boult, T. E. (2017b). Facial attributes: Accuracy and adversarial robustness. *Pattern Recognition Letters*. https://doi.org/10.1016/j.patrec.2017.10.024.

Rudd, E. M., Gunther, M., & Boult, T. E. (2016). Paraph: Presentation attack rejection by analyzing polarization hypotheses. In *The IEEE conference on computer vision and pattern recognition workshops*.

Sabour, S., Cao, Y., Faghri, F., & Fleet, D. J. (2016). Adversarial manipulation of deep representations. In *International conference on learning representations*.

Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International conference on learning representations*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *IEEE conference on computer vision and pattern recognition* (pp. 815–823).

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC conference on computer and communications security* (pp. 1528–1540).

Siddiqui, T. A., Bharadwaj, S., Dhamecha, T. I., Agarwal, A., Vatsa, M., Singh, R., & Ratha, N. (2016). Face anti-spoofing with multifeature videolet aggregation. In *IEEE international conference on pattern recognition* (pp. 1035–1040).

Singh, M., Singh, R., Vatsa, M., Ratha, N., & Chellappa, R. (2019). Recognizing disguised faces in the wild. *IEEE Transactions on Biometrics, Behavior, and Identity Science*. https://doi.org/10.1109/TBIOM.2019.2903860.

Smith, D. F., Wiliem, A., & Lovell, B. C. (2015). Face recognition on consumer devices: Reflections on replay attacks. *IEEE Transactions on Information Forensics and Security*, *10*(4), 736–745.

Song, Y., Kim, T., Nowozin, S., Ermon, S., & Kushman, N. (2018). Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International conference on learning representations*.

Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In *The IEEE conference on computer vision and pattern recognition*.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *International conference on learning representations*. URL arXiv:1312.6199.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *IEEE conference on computer vision and pattern recognition* (pp. 1701 – 1708).

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2018). Ensemble adversarial training: Attacks and defenses. In *International conference on learning representations*.

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137–154.

Wu, X., He, R., Sun, Z., & Tan, T. (2018). A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2884–2896.

Xie, C., Wang, J., Zhang, Z., Ren, Z., & Yuille, A. (2018). Mitigating adversarial effects through randomization. In *International conference on learning representations*.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., & Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In *IEEE international conference on computer vision*.

Xu, W., Evans, D., & Qi, Y. (2018). Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and distributed system security symposium*.

Ye, S., Wang, S., Wang, X., Yuan, B., Wen, W., & Lin, X. (2018). Defending DNN adversarial attacks with pruning and logits augmentation. In *International conference on learning representations workshop*.