Are Image-Agnostic Universal Adversarial Perturbations for Face Recognition Difficult to Detect?

Akshay Agarwal, Richa Singh, Mayank Vatsa, and Nalini Ratha
IIIT-Delhi, India and IBM TJ Watson, USA
Email: {akshaya, rsingh, mayank}@iiitd.ac.in, and ratha@us.ibm.com

Abstract

High performance of deep neural network based systems have attracted many applications in object recognition and face recognition. However, researchers have also demonstrated them to be highly sensitive to adversarial perturbation and hence, tend to be unreliable and lack robustness. While most of the research on adversarial perturbation focuses on image specific attacks, recently, image-agnostic Universal perturbations are proposed which learn the adversarial pattern over training distribution and have broader impact on real-world security applications. Such adversarial attacks can have compounding effect on face recognition where these visually imperceptible attacks can cause mismatches. To defend against adversarial attacks, sophisticated detection approaches are prevalent but most of the existing approaches do not focus on image-agnostic attacks. In this paper, we present a simple but efficient approach based on pixel values and Principal Component Analysis as features coupled with a Support Vector Machine as the classifier, to detect image-agnostic universal perturbations. We also present evaluation metrics, namely adversarial perturbation class classification error rate, original class classification error rate, and average classification error rate, to estimate the performance of adversarial perturbation detection algorithms. The experimental results on multiple databases and different DNN architectures show that it is indeed not required to build complex detection algorithms; rather simpler approaches can yield higher detection rates and lower error rates for image agnostic adversarial perturbation.

1. Introduction

With the availability of large databases, computing resources, and newer optimization techniques, deep learning algorithms have seen huge success in several domains ranging from text processing to speech processing to visual understanding to complex task of autonomous driving. However, one of the major criticism of deep neural network (DNN) algorithms is lack of strong mathematical foundations. This limitation has motivated researchers to exploit the working of deep learning algorithms to *fool* the classifier for incorrect prediction. As shown in Figure 1, to fool a face recognition model (VGG-16 in this case), the input images are perturbed in such a way that the human can still predict the actual class but the network will classify it in the wrong class. This perturbation of the input image is popularly known as an adversarial perturbation.

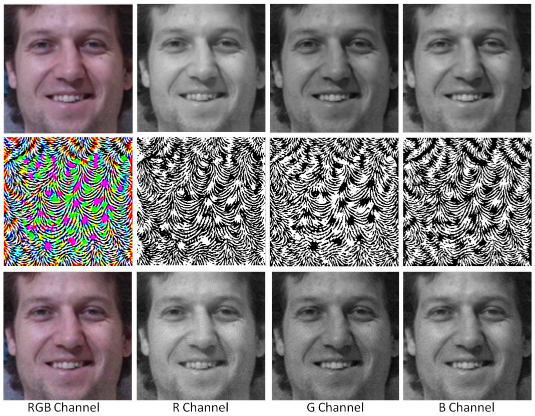

Figure 1: Visualization of original face image, perturbation vector, and perturbed image. First row is original face image, second row is the perturbation vector of VGG-16, and last row is the perturbed image (Better view at magnification factor 4).

Adversarial perturbation can be defined as the addition of a minimal vector r such that with addition of this vector into the input image x, i.e. $(x + r)$, the deep learning model s misclassifies the input. The impact of adversarial perturbation was first coined by Szegedy et al. [33]. It was shown that a minimal pixel change in the input image could lead to misclassification. While the primary objective of creating adversaries is misclassification, it is equally important that the changes are imperceptible and undetectable. Due to the wide spread applicability of deep learning algorithms, adversarial samples can cause severe damage in real world scenarios [19]. For instance, in "autonomous" driving, if the signboard is perturbed, adversarial samples can risk the car, people walking on the road, and other automobiles.

Several researchers have designed algorithms for adversarial image generation. Goodfellow et al. [12] proposed the generation of adversarial examples by adding the network gradient to the input image with the aim of *misclassification* and referred to it as "fast gradient sign method" (FGSM). The gradient method can be applied both using/not-using the sign of the gradient and iterative vs single time addition. Carlini and Wagner, and Chen et al. proposed algorithms based on the minimization of DNN loss function using L_1, L_2 norms [4], and Elastic-Net (L_1+L_2) [5]. Papernot et al. [29] have shown that in place of modifying every pixel of the input image, it is feasible to achieve adversarial effect by perturbing highly salient pixels that have high involvement in the classification task. Another adversarial approach, termed as DeepFool [27], computes the adversarial perturbation for each image. While most of the adversarial example generation algorithms are based on the minimization of particular optimization function over each image, recently Moosavi-Dezfooli et al. [26] proposed *image-agnostic* Universal Perturbation to fool deep networks. This image agnostic adversarial perturbation is named as "Universal" because it can successfully perturb any image. Experiments have shown the generalizability of the universal perturbation across three different DNN architectures, viz, VGG-16, GoogLeNet, and ResNet-152. Goel et al. [10] have developed a toolbox containing various algorithm corresponds to adversarial generation, detection, and mitigation.

It is intriguing to see the sensitivity of such accurate deep neural networks towards adversarial attacks. Therefore, the focus of this research is on addressing the challenge of adversaries by designing algorithm to detect adversarially perturbed samples. On analyzing this phenomenon in detail, Goodfellow et al. [12] observed that one of the reasons is the linearity of the hidden layers while Tanay and Griffin [34] highlighted that the sampled data points lie on a manifold. If the adversarial samples lie close to the submanifold, there is a high chance of misclassification. Since the adversary generation algorithms utilize the linearity of deep models, in this research we pose the question, *Is Adversarial Perturbation Challenging to Detect?* The contributions of this research are:

- The effectiveness of the solution is evaluated using two perturbation algorithms, Universal Perturbation [26] and Fast Feature Fool [28] (a variant of Universal Perturbation). The experiments are performed with four different DNN architectures, VGG-16 [31], GoogLeNet [32], ResNet-152 [16], and CaffeNet [18], and three different face databases namely MEDS [9], PaSC [2], and Multi-PIE [14].
- The results are reported using different combinations of the face databases for training and testing both in same-database and cross-database settings.
- The results are compared using two existing algorithms, Adaptive Noise Reduction [22] and Bayesian Uncertainty [8]. The detection accuracy is significantly better than these approaches with several orders of magnitude less in the computation requirement.
- Also, detailed analysis has been performed to showcase the performance of individual color channels in perturbation detection to select channels to reduce the computing requirement.

To the best of our knowledge, this is first reported work in this area with such high accuracy.

2. Related Work

Adversarial detection algorithms in literature can be classified into these four broad features: (i) measurement of distribution of original and adversarial class images, (ii) dimensionality reduction on the inner feature representations of Convolutional Neural Network (CNN), (iii) learning the classifier separately using the original and adversarial images or on the inner layer features of CNN, and (iv) image enhancement (enhancing the input image and providing the enhanced images to the classifier)

To detect the adversarial examples, Grosse et al. [15] introduced extra class label in the network and proposed to re-train the entire network. In place of retraining the entire network again, Gong et al. [11] learned a separate neural network using the examples from both the classes. Metzen et al. [25] applied the adversary detector at the intermediate layer outputs of ResNet model. Li and Li [21] presented two techniques: in the first approach, statistical features are calculated after applying the PCA on the feature maps of the CNN layers whereas, in the second approach, image enhancement is performed using mean-blur operations before passing the image to the network for classification. PCA is applied to each convolutional layers and cascade of linear Support Vector Machine (SVM) [6] classifiers are learned. At the test time, if each of the individual SVM classifier predicts the original class, then the input image is classified as original else adversarial. Hendrycks and Gimpel [17] applied PCA whitening to showcase that the variance of the principal components are much higher for adversarial images as compared to clean images. Lu et al. [23] quantized the output of ReLU layer to generate the binary code for the detection of adversarial examples. Goswami et al. [13] have presented adversarial detection and mitigation based on the response of the intermediate layers of deep CNN models.

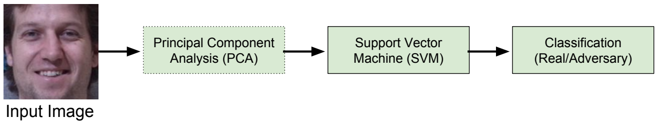
Input Image
Principal Component Analysis (PCA)
Support Vector Machine (SVM)
Classification (Real/Adversary)

Figure 2: Generic overview of the proposed universal adversarial perturbations detection algorithm.

Akhtar et al. [1] have proposed the Perturbation Rectifier Network (PRN) to detect the image-agnostic adversarial examples. SVM classifier is learned for the detection on the Discrete Cosine Transformation of the difference image of PRN and input image.

In a recent research, Carlini and Wagner [3] reviewed and listed the limitation of ten existing adversarial detection algorithms. Further, in literature, most of the defense techniques for the adversary have focused on adversary generated from individual images and to the best of our knowledge, there is a lack of detection algorithms for image-agnostic adversary, i.e. samples generated based on the distribution of the image dataset.

Another approach to address the adversarial attacks is defense against the attacks. Lu et al. [24], Papernot et al. [30], and Kurakin et al. [20] have presented different defense techniques against an adversary. The method proposed by Kurakin et al. [20] is effective for single step based adversary but fail for the black-box based adversary. Papernot et al. [30] proposed an improvement in the classification layer with the introduction of one extra parameter. The proposed adversarial defensive model produces both the actual label and soft label for the image in question. This defensive model is successful in reducing the success rate of adversarial images generated using simple algorithms form 96% to 0.5%. However, it does not work well for the adversarial examples generated from C&W's (L_2) approach [4].

3. Perturbation Detection Algorithm

To understand how easy or difficult it is to detect adversarial perturbation in the images, our hypothesis is that a linear classifier applied on either the pixel values or the projections obtained from principal component analysis (PCA) [35], can efficiently differentiate between perturbed and non-perturbed samples. Using this hypothesis, in this section, we describe two simple image-agnostic perturbation detection algorithms. The overview of the proposed algorithm is illustrated in Figure 2.

Pixel + SVM Classification: In the first approach, we apply a two-class SVM classifier on raw pixel values. The steps involved in detecting universal perturbations using raw pixel values and SVM classifier are:

1. Images in the training database are flattened to form a row vector and combined to make one large training matrix of dimension $k \times n$, where k is the total number of training images in both the classes and n is the image vector dimension.
2. Linear SVM classifier is trained for two-class classification on the training matrix using label $+1$ for original images and -1 for adversarial images.
3. Similarly, the test image is first converted into a row vector and then fed into the learned classification model for classification. The learned SVM model provides the classification score for each test image.

PCA+SVM Classification: Extending the first approach, we utilize PCA based dimensionality reduction followed by SVM classification. The steps involved in the second approach are as follows:

1. From the training database, images belonging to both original and adversarial classes are flattened to form a row vector and combined to make one large training matrix of dimension $k \times n$, where k is the total number of images of both the classes and n is the dimension of the image vector.
2. Linear projection vector on the training matrix is computed using Principal Component Analysis (PCA). PCA reduces the dimension of training matrix from $k \times n$ to $k \times p$ while preserving 99% Eigen energy.
3. Linear Support Vector Machine Classifier is trained for two-class classification on the reduced dimensional training matrix using label $+1$ for original images and -1 for adversarial images.
4. Similarly, the test image is first converted into a row vector and then dimensionality is reduced using PCA.
5. The reduced dimensional test vector is then fed into the SVM classifier to compute the classification score.

4. DNN Architectures, Database, and Evaluation Protocol

The experiments are performed on three databases using two attack algorithms and three deep neural network models. The details are discussed below.

DNN architectures: Three different DNN architectures, viz., VGG-16 [31], GoogLeNet [32], and ResNet-152 [16] are used to generate universal adversarial images. These three DNN models are 16, 22, and 152 layers deep architecture, respectively and yield state-of-the-art accuracies in face and object recognition challenges.

Attacks: The aim of universal perturbation [26] is to generate the single adversarial vector which can misclassify *any*

Table 1: Characteristics of the databases used for detecting universal adversarial perturbations.

Type	Database	Clean	Adversarial
Face	MEDS	836	2,508
	Multi-PIE	1,680	5,040
	PaSC	7,443	22,329
Total	—	**9,959**	**29,877**

image. The aim of the perturbation vector can be defined mathematically as following: $\hat{k}(x+v) \neq \hat{k}(x)$, for most $x \sim \mu$. Where μ denotes the data distribution, v denotes the perturbation vector, and $\hat{k}$ is the classification function. The perturbation is bound by the following conditions: (i) $\|v\|_p < \xi$, and (ii) P($\hat{k}(x+v) \neq \hat{k}(x)) \geq 1 - \delta$. The misclassification ratio of the classifier is controlled by δ and χ controls the magnitude of v.

Mopuri et al. [28] presented the data independent approach to generate the universal adversarial perturbation to fool the classifier on any image. The perturbation is generated for general object detection CNN models: VGG-16, GoogLeNet, and CaffeNet [18]. The data independent approach namely *Fast Feature Fool (F3)* is defined as following: $f(x+\delta) \neq f(x)$, such that $\|\delta\|_\infty < \xi$. Here, $x \in \chi$, χ defines the distribution of the images and f defines the CNN function of classification.

Databases: As shown in Table 1, in this research, we have used three face databases to showcase the adversarial detection performance. The face databases used in this research are: Point and Shoot Challenge (PaSC) database [2], a subset of CMU Multi-PIE database [14] (frontal only view), and the Multiple Encounters Dataset (MEDS-II) [9]. Table 1 shows the statistics of each of the databases used in this research.

Protocol: We have performed two different kinds of experiments for the generalizability of the detection algorithms. The first experiment represents the intra-database scenario where the training and testing sets belongs to the same database. In the second experiment, referred as inter-database (or cross-database) where the training and testing sets belongs to two different databases. For the intra-database experiments, 50% images of each class are used for training the detector and remaining 50% images are used for evaluating the classifier. For example, for the experiments on MEDS database, 418 images are used in training and remaining 418 images are used for testing (418 images are original and equal number of perturbed samples are generated). For inter-database experiments, all the images belonging to one database (e.g. entire MEDS) are used for training and all the images of other databases (e.g. Multi-PIE) are used for testing. We have also reported the detection performance of individual color channels with the motivation that: *is there any particular channel which is more critical in adversarial detection?*

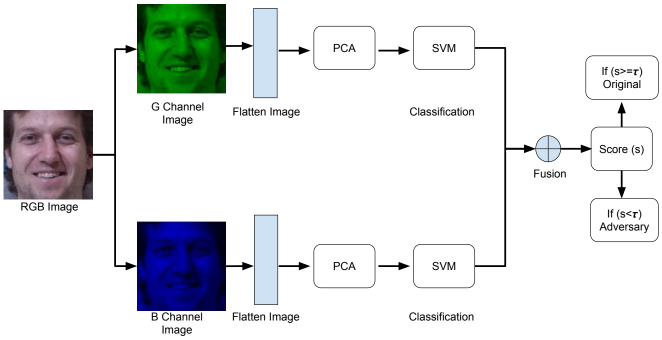

Figure 3: Proposed classification pipeline of detecting universal adversarial perturbations.

Evaluation Metric: In order to report the experimental results, we first define APCCER (Adversarial Perturbation Class - Classification Error Rate) and OCCER (Original Class - Classification Error Rate). APCCER is defined as the fraction of adversarial images that are wrongly classified to original class and OCCER is defined as the fraction of original images that are misclassified to the adversarial classes. Mathematically, $APCCER = score < \tau$ where score corresponds to original class samples, and $OCCER = score \geq \tau$ where score corresponds to adversarial class images.

To measure the above-mentioned error rates, the threshold (τ) is selected at Equal Error Rate (EER) where False Accept Rate (FAR) is equal to False Reject Rate (FRR). Using these two, we finally define ACER (Average Classification Error Rate) which is the average of OCCER and APCCER, use it for reporting the results. For a meaningful perturbation detection algorithm, ACER should be as low as possible.

5. Results and Analysis

In this section, we present the results of adversarial perturbation detection on face databases. The results are reported using the protocol defined in section 4 in terms of ACER (described in section 4). For demonstrating that linear projection using PCA can easily detect the **universal** adversarial perturbation and fast feature fool, extensive experiments are conducted on face databases using intra and inter (or cross) database protocols. The inter database protocol is necessary for the real world scenario where it is possible that detector is trained on one kind of images while at the time of testing, an input image might be captured in completely different environment. The steps involved in the proposed PCA +SVM followed color channel fusion algorithm are illustrated in Figure 3.

Table 2: Individual color channel detection performance of the proposed algorithm for Universal perturbation attack. The results are reported in terms of ACER (%) for both intra-database and inter-database detection experiments.

Model	Algorithm	Channel	Training Database	Testing Database		
				MEDS	Multi-PIE	PaSC
VGG-16	Pixel + SVM	Red	MEDS	17.58	19.55	8.41
			Multi-PIE	23.98	18.39	14.69
			PaSC	11.18	13.07	6.63
		Green	MEDS	17.82	15.48	6.66
			Multi-PIE	13.10	18.45	7.80
			PaSC	10.94	9.35	3.36
		Blue	MEDS	19.14	13.04	8.60
			Multi-PIE	20.81	18.87	13.99
			PaSC	11.36	9.64	5.92
	PCA + SVM	Red	MEDS	12.08	15.27	5.15
			Multi-PIE	16.87	**9.35**	9.14
			PaSC	6.28	11.58	3.42
		Green	MEDS	**10.41**	10.30	**3.15**
			Multi-PIE	**10.47**	10.24	**4.27**
			PaSC	**6.04**	**4.43**	**1.48**
		Blue	MEDS	14.95	11.19	5.48
			Multi-PIE	13.04	10.12	5.90
			PaSC	8.13	6.90	3.04
GoogLeNet	Pixel + SVM	Red	MEDS	19.38	18.30	17.01
			Multi-PIE	16.62	21.37	8.70
			PaSC	17.04	20.12	9.41
		Green	MEDS	19.01	19.40	13.15
			Multi-PIE	19.32	18.57	12.29
			PaSC	15.19	16.66	5.39
		Blue	MEDS	17.82	11.07	6.19
			Multi-PIE	13.10	21.31	4.69
			PaSC	12.50	11.81	3.01
	PCA + SVM	Red	MEDS	11.54	11.28	7.56
			Multi-PIE	8.85	10.95	4.03
			PaSC	7.95	7.29	3.45
		Green	MEDS	**10.05**	9.94	5.77
			Multi-PIE	13.22	7.20	4.91
			PaSC	**7.48**	8.13	**1.83**
		Blue	MEDS	10.29	**7.02**	**2.78**
			Multi-PIE	**4.78**	**7.08**	**0.85**
			PaSC	8.43	**6.01**	**1.83**
ResNet-152	Pixel + SVM	Red	MEDS	22.85	18.39	10.79
			Multi-PIE	22.43	19.94	24.45
			PaSC	14.05	11.99	6.73
		Green	MEDS	22.01	13.24	9.55
			Multi-PIE	20.57	18.99	15.40
			PaSC	10.35	7.05	6.68
		Blue	MEDS	18.78	12.62	8.93
			Multi-PIE	9.15	20.00	8.52
			PaSC	9.92	19.11	6.30
	PCA + SVM	Red	MEDS	14.71	11.82	4.98
			Multi-PIE	12.92	**6.49**	10.83
			PaSC	10.23	7.41	2.50
		Green	MEDS	13.28	**5.00**	**4.26**
			Multi-PIE	14.71	7.62	8.06
			PaSC	**6.34**	**3.57**	**2.10**
		Blue	MEDS	**11.12**	8.54	5.22
			Multi-PIE	5**.56**	9.11	**3.97**
			PaSC	6.46	14.11	2.50

5.1. Analysis and Evaluation Studies

As mentioned in Section 4 we have used three different face databases and three different DNN architectures to generate the universal perturbed images. Results for both intra and inter database experiments are reported in Table 3 for VGG-16, GoogLeNet, and ResNet-152 architecture. For VGG-16 architecture when training is performed using MEDS database and testing is performed using Multi-PIE and PaSC databases, detection algorithm yields 4.79% and 1.79% ACER, respectively. When the universal perturbed images are generated using GoogLeNet architecture and perturbation detection training is performed using PaSC, the ACER is 4.01% and 4.29% on MEDS and Multi-PIE databases, respectively. The detection performance of individual color channels are reported in Table 2. It is evident that dimensionality reduction using PCA improves the error rates by atleast 5%, 7%, and 8% for VGG-16, GoogLeNet, and ResNet-152 model respectively. Similarly, as expected the cross-database error rates are higher than intra-database experiments for all the DNN models. Analyzing the score distributions show that "there is minimal overlap between the scores, which helps in efficiently detecting whether the image is perturbed or original".

Detection performance of the proposed color channel fusion algorithm on Fast Feature Fool adversarial algorithm are reported in Table 4. Similar to Universal adversarial detection performance, the proposed algorithm shows high detection performance and lower error rates across all databases and DNN models. The detection error rate in intra-database experiments on all three DNN models lies in the range of 3.51%–11.48%.

Through extensive experiments, we have observed that the performance of *Red* channel is comparatively weak in comparison to *Green* and *Blue* channels on face databases except intra database experiment on Multi-PIE database. In our experiments, we observe APCCER value of 0% over all the intra and inter face database experiments, which is highly desired in the "zero-intruders" passing security system. We have also observed that when PaSC database is used for training, we have achieved lowest ACER value. The prime reason for this is the number of training images in PaSC is larger than MEDS and Multi-PIE.

5.2. Effect on Color Channels

To understand the effect of perturbation and detection on color channels, a channel wise analysis is performed for both the detection algorithms. For both pixel+SVM and PCA+SVM algorithms, linear SVM classifier is trained for each color channel separately. Steps are repeated for each color channel independently, and the final classification is obtained by fusing the scores of green and blue channels. The results pertaining to individual color channels are reported in Table 2. As shown in Figure 1, we have observed that blue channel is receptive for adversarial detection and when combined with green channel, yields the lowest error rate. If we view adding perturbations as image watermark-

Table 3: Universal adversarial detection performance of intra and inter face database experiments in terms of ACER% for VGG-16, GoogLeNet, and ResNet-152

Algorithm	DNN Model	Training Database	Testing Database		
			MEDS	Multi-PIE	PaSC
Bayesian Uncertainty [8]	VGG-16	MEDS	19.7	20.7	21.5
		Multi-PIE	29.9	25.3	27.4
		PaSC	36.1	34.7	28.8
	GoogLeNet	MEDS	20.1	26.9	25.9
		Multi-PIE	29.1	30.2	21.0
		PaSC	37.2	34.7	26.4
	ResNet-152	MEDS	21.6	27.1	25.0
		Multi-PIE	27.0	29.7	29.8
		PaSC	34.8	33.1	24.2
Adaptive Noise [22]	VGG-16	MEDS	19.8	22.6	20.9
		Multi-PIE	29.1	24.5	28.3
		PaSC	34.7	32.2	28.0
	GoogLeNet	MEDS	20.8	29.5	28.2
		Multi-PIE	29.7	30.6	30.9
		PaSC	36.8	37.0	25.8
	ResNet-152	MEDS	21.1	28.7	27.1
		Multi-PIE	29.5	28.9	31.0
		PaSC	33.2	31.3	23.7
Proposed	VGG-16	MEDS	**6.46**	**4.79**	**1.52**
		Multi-PIE	**6.52**	**5.42**	**2.37**
		PaSC	**3.41**	**1.82**	**0.71**
	GoogLeNet	MEDS	**6.82**	**4.58**	**2.28**
		Multi-PIE	**4.37**	**3.15**	**0.93**
		PaSC	**4.01**	**4.29**	**0.84**
	ResNet-152	MEDS	**6.82**	**2.83**	**1.85**
		Multi-PIE	**5.44**	**3.99**	**2.71**
		PaSC	**3.83**	**3.66**	**0.98**

ing process, then this is a known information that detecting watermark in blue and green channels is easier [7].

5.3. Comparison with Existing Algorithms

The proposed color channel fusion algorithm is compared with three recently proposed adversarial detection algorithms: Intermediate CNN Filter Response [13], Adaptive Noise Reduction [22] and Bayesian Uncertainty [8]. The comparative results are reported in Table 3. When the adversarial detector is trained on MEDS database and tested on MEDS database the Adaptive Noise Reduction [22] and Bayesian Uncertainty [8] algorithms yields 19.7% and 19.8% detection error where adversarial examples are generated using VGG-16 model. Similarly, when GoogLeNet and ResNet-152 model is used to create the adversarial images, in the intra-database experiments of Multi-PIE and PaSC, the error rate of the existing algorithms are in the range 24.2– 30.6%. The algorithm proposed by Goswami et al. [13] yields 18.4% ACER on MEDS when VGG-16 adversary is used for perturbation. In the inter-database tests, the error rate of the existing algorithms lies in the range of 22.6%–34.7%. The error rate of the existing algorithms is at-least three times higher than the proposed algorithm. On F3 the error rates of the existing algorithms are in the range of 30.1%–20.1%. It is interesting to note that the highest error of the proposed algorithm on any of the experiments is 6.82%, whereas the minimum error of the existing algorithms is 19.7%.

Table 4: Fast Feature Fool adversarial detection performance of intra and inter face database experiments in terms of ACER% for VGG-16, GoogLeNet, and ResNet-152

Attack	DNN Model	Training Database	Testing Database		
			MEDS	Multi-PIE	PaSC
Fast Feature Fool	VGG-16	MEDS	10.05	**9.23**	4.48
		Multi-PIE	10.41	10.12	5.82
		PaSC	**7.00**	11.19	**3.51**
	GoogLeNet	MEDS	11.48	7.75	8.04
		Multi-PIE	10.77	**7.44**	3.56
		PaSC	**7.42**	7.83	**3.54**
	CaffeNet	MEDS	9.69	8.15	**1.46**
		Multi-PIE	10.89	8.81	5.14
		PaSC	**5.26**	**5.95**	6.87

5.4. Computation Complexity

Computationally, the proposed approach (PCA+SVM) is efficient compared to existing approaches. On a desktop with 3.4 GHz Intel $i7$ processor and 16GB RAM, PCA+SVM classification requires less than 0.01 seconds to detect adversary, whereas existing algorithms [8], [22] are computationally expensive and take up to 10 seconds for the same task.

6. Conclusion

In this research, we focus on how to detect image-agnostic adversarial perturbation. We present a simple approach of PCA+SVM for detecting universal perturbations. Experiments are performed using various face databases such as MEDS, Multi-PIE, and PaSC. *Universal* and Fast Feature Fool adversarial images are generated using VGG-16, GoogLeNet, ResNet-152, and CaffeNet DNN architectures. Experiments using both intra and inter (cross) database settings show that the linear projection features using PCA are sufficient to detect image-agnostic adversarial perturbations. In addition, we observe that only two channels are enough to detect the perturbation in the image. We compare our detection rate and computational efficiency with the results from two recent research papers to show the superior performance.

7. Acknowledgement

Agarwal is partly supported by Visvesvaraya PhD Fellowship, and Vatsa and Singh are partly supported through CAI@IIIT-Delhi.

References

[1] N. Akhtar, J. Liu, and A. Mian. Defense against universal adversarial perturbations. *IEEE CVPR*, 2018.

[2] J. Beveridge, P. Phillips, D. Bolme, B. Draper, G. Given, Y. M. Lui, M. Teli, H. Zhang, W. Scruggs, K. Bowyer, P. Flynn, and S. Cheng. The challenge of face recognition from digital point-and-shoot cameras. In *IEEE BTAS*, pages 1–8, 2013.

[3] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *ACM WAIC*, 2017.

[4] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, pages 39–57, 2017.

[5] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*, 2017.

[6] C. Cortes and V. Vapnik. Support vector machine. *Machine Learning*, 20(3):273–297, 1995.

[7] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008.

[8] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[9] A. P. Founds, N. Orlans, W. Genevieve, and C. I. Watson. Nist special databse 32-multiple encounter dataset ii (meds-ii). *NIST Interagency/Internal Report (NISTIR)-7807*, 2011.

[10] A. Goel, A. Singh, A. Agarwal, M. Vatsa, and R. Singh. Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition. *IEEE BTAS*, 2018.

[11] Z. Gong, W. Wang, and W.-S. Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017.

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

[13] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa. Unravelling robustness of deep learning based face recognition against adversarial attacks. *AAAI*, 2018.

[14] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[15] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE CVPR*, pages 770–778, 2016.

[17] D. Hendrycks and K. Gimpel. Early methods for detecting adversarial images. *arXiv preprint arXiv:1608.00530*, 2016.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.

[19] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[20] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[21] X. Li and F. Li. Adversarial examples detection in deep networks with convolutional filter statistics. *ICCV*, 2017.

[22] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang. Detecting adversarial examples in deep networks with adaptive noise reduction. *arXiv preprint arXiv:1705.08378*, 2017.

[23] J. Lu, T. Issaranon, and D. Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *IEEE CVPR*, pages 446–454, 2017.

[24] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*, 2015.

[25] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[26] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *IEEE CVPR*, 2017.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE CVPR*, pages 2574–2582, 2016.

[28] K. R. Mopuri, U. Garg, and R. V. Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *BMVC*, 2017.

[29] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *IEEE ESSP*, pages 372–387, 2016.

[30] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE S&P*, pages 582–597. IEEE, 2016.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, pages 1–9, 2015.

[33] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[34] T. Tanay and L. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[35] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.