Deep Learning Adversarial Examples - A Survey

Nandini Krupa Krishnamurthy
Clemson University
nandink@clemson.edu

***Abstract*—Deep learning is the heart of machine learning. It is in usage everywhere from Artificial Intelligence models to Computer Vision. Deep Learning has witnessed some phenomenal success in solving problems that humans have not been able to. But as huge is its usage so are the threats to it. This survey paper discusses one of these - adversarial examples. These are very small perturbations in an image caused by adversaries that are invisible to human eye but can cause great damage to the success of a deep learning model function by fooling them. Adversarial examples is a major topic of interesting research. This paper starts by discussing some common terms related to adversarial examples in usage. Later on, it discusses briefly about some of the common methods of generating adversarial examples, methods to detect and defend against them and also some issues associated with the methods of dealing with adversarial examples.**

I. Introduction

Deep Learning has probably never been as important to mankind as it is now. Of course, it is an important part of Data Science, Machine Learning, Artificial Intelligence, and Technology, but more than that it is an important part of our everyday life. Going deeper into the meaning, let us see what Deep Learning is. Learning simply means a process, which is automatic, that helps in finding ways to better represent or better understand the data for analyzing. So if there is a way to represent the data from multiple angles, and come up with the best conclusion possible, then yes, the process has learned to deal with the data better. The process, in other words, has learned a way to come up with a better way of dealing with the data and has also learnt to draw optimum conclusions from it. So, that is what learning means. deep in deep learning refers to the ways of deriving an output by passing our input data through various layers suggesting some transformation on the input in each layer. The accuracy is expected to be higher because the output is not dependent on a single linear formula, but depth of transformations [1].

Deep Learning has brought about some radical changes in the Computer Vision and Speech Recognition Systems. In fact, CNN has become the go-to option for a wide variety of Computer Vision problems [2] [3]. The availability of large-scale datasets, such as ImageNet [4] along with added computational capabilities from GPUs, it is possible to obtained different visual patterns by training a hierarchical deep network. Some other areas of their massive uses include image classification, instance retrieval, object detection, visual concept discovery, semantic segmentation, boundary detection [5] and the list goes on.

However, though they are very successful, they have their own shortcomings. These are very sensitive to even minute perturbations in the input image [5]. These perturbations are often invisible to the human eye but their existence in the input data can guide a network to produce incorrect results. These perturbations are called adversarial examples. These tend to fall in such places on input data that are not much explored during training. They can cause a network to completely change the result on a prediction. Worse still, the model attacked reports high confidence in wrong prediction [6].

The four main goals of an adversarial are:

1) Confidence reduction - by the introduction of class ambiguity
2) Misclassification - change the output of an image to any other class than the actual target class
3) Targeted misclassification - modify inputs so that the output is forced to a target class
4) Source/target misclassification - force the output of a specific input to be classified into a specific target class

Therefore a study on adversarial importance becomes an area of prime concern because one, these are unrecognizable and two, they pose a threat to security. Thus this survey paper will focus more on better understanding adversarial examples along with some their detection techniques and defenses.

II. Main Techniques

A. Taxonomy of Adversarial Examples

Before moving on to understand some of the methods of generating adversarial examples, let us see the taxonomy of adversarial examples. This section mainly talks about the categories that these adversarial examples can be classified into threat model, perturbation and benchmark [7].

1) **Threat Model:** Threat model refers to the type of possible attacks based on some approach for example black-box attack/white-box attack. Attacks can only happen during training/deploying stage. The input data can only be hampered after the model is trained. Based on various parameters such as specific requirements, assumptions and scenarios adversarial examples are generated. The thread model can be broadly classified into four aspects - adversarial falsification, adversarys knowledge, adversarial specificity, and attack frequency.
   - Adversarial Falsification
     - False Positive attacks: These are used to create a negative sample which the model classifies as positive one i.e., a Type I error. For example in image classification, an adversarial example that

goes unrecognized by humans can be classified by a DNN model with high confidence level.
  - False Negative attacks: These are used to create a positive sample which the model classifies as negative one i.e., a Type II error. For example, in malware detection procedure a trained model cannot identify it.
- Adversarys Knowledge
  - White-box: This type of attack assumes that the adversary knows everything about the neural network model. In fact, many of the adversarial examples are generated by estimating model gradients.
  - Back-box: this type of attack assumes that the adversary has no knowledge about the existing neural network model. In this case, the adversary only knows the output.
- Adversarial Specificity
  - Targeted: This type of attack occurs mostly in multi-class classification problems. This is where an adversary misguides a network to a target class.
  - Non-targeted: This is just the opposite case of targeted attacks. Here the choice of a class is arbitrary but cannot be the actual class. For example, in object recognition, the neural networks always misclassify images.
- Attack Frequency
  - One-time attacks: these attacks take a single occurrence to optimize the adversarial examples.
  - Iterative attacks: These attacks take more than one time (iterative) to optimize the adversarial examples.

2) **Perturbation:** Perturbations are the essence of adversarial examples. A small perturbation is necessary for an image to cause an adversarial attack. These are so created that they are invisible to human eye but can cause huge damage to the performance of a deep neural network. In this section we deal with the aspects of perturbation - perturbation scope and perturbation limitation.
   - Perturbation Scope
     - Individual: In this case, each input has some different kinds of perturbation generated.
     - Universal: This type of attack creates perturbations universally to all the clean input.
   - Perturbation limitation
     - Optimized perturbation: The goal of the perturbation is optimization. The main aim of these is to minimize perturbations so that humans would be incapable of recognizing them.
     - Constraint perturbation: Here, a perturbation is set as the constraint of the optimization problem.
3) **Benchmark:** Adversarial examples are generated based on the datasets and/or models. Since both of these are so diverse in nature, it becomes a difficult task to know whether the adversary was because of a dataset or the model. This section talks about some of the datasets and models.
   - Datasets: Though there are various datasets available online, MNIST, CIFAR-10, and ImageNet are the ones that are frequently used to study adversarial examples. Out of these MNIST and CIFAR-10 are small in size and therefore are easy to attack as well as defend. Therefore ImageNet is the best dataset to evaluate adversarial examples.
   - Victim Models: The common models that are attacked to generate adversarial examples are VGG, GooLeNet, LeNet, AlexNet, ResNet, and CaffeNet.

B. Generating an adversarial example

The generation of adversarial examples can be broadly classified into those that are used for classification such as fooling neural networks and the attacks beyond these as discussed in the upcoming sections.

*1) **Attacks for Classification**:* These attacks are devised to target deep neural network performing tasks of classification or recognition.

1) **L-BFGS Attack:** This was introduced by Szegedy et al.[8] as a method to generate adversarial examples on a neural network to solve a general targeted problem. This was a linear search and to find an appropriate value say c , as in the equation (1). This searches lines for values greater than 0, i.e., for c¿0.

$$\min_{x'} \quad c\|\eta\| + J_\theta(x', l') \qquad (1)$$
$$s.t. \qquad x'\epsilon[0,1]$$

2) **Fast Gradient Sign Method (FGSM):** This is an improvement on L-BFGS attack which was a linear search algorithm that was time consuming as well as expensive. This method, proposed by Goodfellow et al., involved one step gradient update in the direction of the gradient [9]. The equation (2) shows the perturbation where ϵ is its magnitude. If X is the resulted image, the $X = x + \eta$

$$\eta = \epsilon sign(\nabla_x J_\theta(x', l')) \qquad (2)$$

3) **Basic Iterative Method (BIM):** BIM is an approach wherein a finer optimization is applied for multiple iterations and every iteration has pixel values clipped so that there is no large change in each pixel. The equation (3) represents BIM.

$$Clip_{x,\xi}\{x'\} = min\{255, x+\xi, max\{0, x-\epsilon, x'\}\} \quad (3)$$

4) **Iterative Least-Likely Class Method (ILLC):** All the above-mentioned methods work well for those datasets where the number of classes is small and distinct. But these methods do not talk about those incorrect classes which the model can select. We can think of Iterative

Least-Likely Class Method as a way to devise more interesting errors. For this a target class exists which may be treated as the one which is the least likely to be chosen by the model of trained network on an Image say x.

$$y_{ll} = argmin_y\{p(y|x)\}$$

The procedure is explained by the equation (4).

$$x_{n+1} = Clip_{x,\epsilon}\{x_n - \epsilon sign(\nabla_x J(x_n, y_{LL}))\} \quad (4)$$

The main difference between fast methods and iterative methods is that fast methods are robust to photo transformation whereas iterative methods are not.

5) **Jacobian-based Saliency Map Attack (JSMA):** This is caused by first calculating the Jacobian matrix for any input, say x. Here, F is the output from the softmax layer. It was observed that even a minute change in the input features can cause large differences to the output. In other words, small perturbations in the input can fool a network by causing a large change in the output. Equation (5) denotes this.

$$J_F(x) = \frac{\partial F(x)}{\partial x} = \left[\frac{\partial F_j(x)}{\partial x_i}\right]_{ixj} \quad (5)$$

6) **DeepFool:** DeepFool was devised to find the closest distance from the input to the decision boundary of the adversarial examples. Since high dimensions are usually non-linear, a method was devised to perform an iterative attack from a linear approximation. At each iteration a small perturbation is added to the image by a small vector. These perturbations are added to calculate the final perturbation. The minimum perturbation is calculated using

$$f(x_i) + \nabla f(x_i)^T \eta_i = 0 \quad (6)$$

The equation (6) above is for a single class. But this can also be extended to multiple classes classifiers by identifying the closest hyperplanes

7) **Compositional Pattern-Producing Network-encoded (CPPN EA Fool):** EA in the name refers to Evolutionary Algorithms. These adversarial examples are classified by DNNs with very high confidence. EAs are utilized to generate Adversarial Examples. This is an example of a False-Positive Attack.

8) **C&Ws Attack:** This type of adversarial examples can make them immune to most of the adversarial detecting defenses. Their objective function, g, is as follows:

$$min_\eta \|\eta\|_p + c.g(x+n) \quad (7)$$
$$s.t. \quad x + \eta \in [0,1]^n$$

Using this better optimization of distance and the penalty term can be obtained.

9) **Universal Perturbation:** This method follows the equation(8) which was formulated to find a universal perturbation vector. Using this method any image can be fooled with high probability

$$\|\eta\|_p \leq \epsilon$$
$$P\big(x' \neq f(x)\big) \geq 1 - \partial \quad (8)$$

par For each iteration, the DeepFool method is used so that a negligible perturbation is obtained for each input data. This iteration is continued until all samples have some perturbations in them.

10) **One-pixel Attack:** In this type of attack a single pixel is modified. The optimization problem is:

$$\min_{x'} \quad J(f(x'), l') \quad (9)$$
$$s.t. \quad \|\eta\| \leq \epsilon_0$$

*2) **Attacks beyond classification/recognition** :* These attacks happen beyond the task of classification or recognition [6].

1) **Attacks on autoencoders and generative models:** The experiments of Tabacof et al. [10] suggested that it is possible to mislead an autoencoder making it rebuild a completely different image. The adversarial attack, in this case, is creating images that are similar to the target image. Autoencoders are more robust to adversarial attacks when compared to classifier networks.
2) **Attacks on Recurrent Neural Networks:** There adversarial examples for the recurrent neural network were successfully created by Papernot et al. [11]. They suggested that those algorithms that were used to compute adversarial examples for feedforward networks can also be used to fool the recurrent neural networks.
3) **Attacks on Deep Reinforcement Learning:** Two different types of adversarial attacks on deep reinforcement learning models were found by Lin et al. strategically-times attack and enchanting attack [12]. In a strategically-times attack, the attack is carried out in small subsets of time such that the attack doesnt get detected. In enchanting attacks, the input is misguided to a wrong target by the use of two aspects a generative model and planning algorithm. These are integrated and generative model is used for prediction of the future states of the input and the planning algorithm is used to create actions that are used to misguide them.Huang et al. [15] showed that in deep reinforcement learning, FGSMs can be used to deteriorate performance of trained policies. In their experiments they made use of perturbations in the raw input of policies.
4) **Attacks on semantic segmentation and object detection:** Semantic segmentation and object detection are the two main aspects of computer vision. The experiments of Moosavi-Dezfooli et al. [17] and Metzen et al. [19] suggested that a DNN can be fooled to change the segmentation of the images by the usage of image-agnostic quasi-imperceptible perturbations. They were also able to show that it was indeed possible to calculate

noise vectors that can be used to remove a class from a segmented class while keeping most of the segmentations undisturbed.

5) **Attacks on Face Attributes:** Face attributes are one of the most famously emerging biometrics for security systems. They are treated as an important problem in computer vision.
Rozsa et al. [16] used a technique - Fast Flipping Attribute to demonstrate the robustness of DNNs against adversarial attacks significantly differs between facial attributes when a DNN is attacked. Adversarial examples have the ability to change the label of a target attribute to a correlated attribute. Mirjalili and Ross [18] were able to change the gender of a target by modifying the face image minutely. On similar grounds Shen et al. [20] proposed two methods that could be used to evaluate DNNs for face attractiveness i.e., those which can have high 'attractiveness scores' but low 'subjective scores'.

III. Detection

Detection can be viewed as a process that alerts when an adversarial example is detected and is rejected from any further processing [6].

1) ***SafetyNet****:* SafetyNet was proposed by Lu et al. [26]. They said that dissimilar patterns are produced by adversarial examples and clean images on a network with ReLU activation is used. Based on this hypothesis they made Radial Basis Function SVM discrete codes calculated by final stage ReLUs in a network. When these codes are compared to that of the clean images in an SVM, perturbations can be detected.

2) ***Detector Subnetwork****:* Adding a subnetwork to the internal layers of a target network which is trained to do binary classification and training it for adversarial examples can help detect perturbations as proposed by Metzen et al. [27]. This method can be used to detect perturbations generated by FGSM, BIM, and DeepFool methods.

3) ***Exploiting Convolution filter Statistics****:* Usage of a cascaded classifier can be used to detect more than 85% of the adversarial examples by computing statistics of the convolutional filters in CNN based network as proposed by Li and Li [28].

4) ***Feature Squeezing****:* A technique to detect the perturbations in images by feature squeezing was proposed by Xu et al. [33]. Two models that could reduce the color of each pixel and also could carry out special smoothing on images were added to the classifier. The predictions between the original images and the squeezed images are compared. If the presence of large difference between these two is detected, then there is said to be existence of adversarial perturbations.

5) ***MagNet****:* The framework for MagNet was proposed by Meng and Chen [32]. This framework utilizes external detectors to make a classification between clean and perturbed images. The aim of the framework is to learn the manifold of clean images. During the testing phase, if an image is not near to the manifold is considered to be perturbed. The images that do not lie on the manifold but are close to it are reformed and these are sent to the classifier. The name of the framework, MagNet, goes with the process - attracting clean images onto the manifold and rejecting the ones that are perturbed.

IV. Defenses

In this section let us look at some of the important defenses that can be applied to a DNN and help it achieve the original goal of classifying adversarial examples correctly. For example, in classification classifying an adversarial image with a label to its actual class is one of the goals. Similar to detection these defenses can be broadly classified depending on three aspects [6]:

1) Modified input/training
2) Modified network
3) Network add-ons

A. Modified input/training

These make use of modified training during the learning phase or modified input during testing phase.

1) ***Brute-force adversarial training****:* There is a general notion that the robustness of a neural network improves with training. Therefore several contributions that introduce adversarial examples also introduce methods to defend against them. Though adversarial training improves robustness, it requires strong attacks and requires the network to be expressive. Since this adversarial training necessitates an increased data size, it is referred to a brute-force approach.

2) ***Data Compression as defense****:* Since most of the image datasets are made up of JPG files Dziugaite et al. [21] studied the effects of JPG compressions on perturbations by FGSM. Reports suggested that for FGSM perturbations, JPG compressions can actually increase the classification accuracy. But it was concluded that compression cannot be used as a complete defense. A limitation is this type is that larger compressions can decrease classification accuracy whereas smaller compressions do not sufficiently remove adversarial perturbations.

3) ***Foveation based defense****:* Foveation is a technique in which neural network is applied in different regions of the network. This method is still in its infant stage and its effectiveness is yet to be demonstrated. However, CNN based approaches are robust to scale and translation of objects in images. But this case doesnt apply to adversarial perturbations. So in essence foveation is a good option to defend again adversarial attacks

B. Modified Network

This kind of defense lies in that change of the network structure like modifying activation/loss function and/or adding/removing layers and/or adding subnetworks.

1) ***Deep Contractive Networks****:* Deep Contractive Networks introduced by Gu and Rigazio [24] showed that Denoising Auto Encoders helps reduce adversarial noise effectively, however, when they are stacked with the original network they increase the risk of adversarial perturbations. Following this observation, DCNs were trained with a smoothness penalty

and with this, a good amount of robustness was obtained against L-BGFS attacks.

*2) **Gradient Regularization/Masking**:* The adversarial robustness for gradient regularization was studied by Ross and Doshi-Velez [23]. They trained various DNN models while penalizing the degree of variation in the output with respect to change in the input. In other words, following this method, a small perturbation becomes inefficient to change the output majorly. This method if combined with brute-force can provide great robustness. A limitation of this is that each of the methods increases the complexity of the model by two times.

*3) **Defensive distillation**:* Distillation is a procedure in which the knowledge associated with a more complex network is transferred to a smaller network. Papernot et al. [22] used this idea to obtain class probability vectors of the training data and fed it back to the training model. By using this method they were able to prove that it is possible to improve the immunity of the network to small perturbations in images.

*4) **DeepCloak**:* Gao et al. [25] proposed to add a layer which is trained by forward-passing clean and adversarial pair of images and with an ability to encodes the differences between the output features of the previous layers for those image pairs before the layer handling classification. This acts as a masking layer. The principal weights of the network relate to the sensitive features in the network. During classification, these sensitive features are masked by the dominant weights.

C. Network add-ons

This is a method in which the external models are added to networks when classifying an unseen example.

*1) **Defense Against Universal Perturbations**:* The contributions of Akhtar et al. [29] suggested that by adding pre-input layers to a target image and training them to rectify a perturbed image, it is possible to develop a defense against adversarial examples. This method rectifies the perturbed image so that it becomes the same as the original clean image and gets predicted into its class properly. These pre-inputs were named as Perturbation Rectifying Network (PRN). The training is done without updating parameters in the network.

*2) **GAN-based defense**:* Lee et al. [30] proposed a method to defend against adversarial perturbations in Generative Adversarial Networks and trained a network against attacks similar to FGSM. Their idea was to train a target network along with the generator network that supposes to create adversarial images. During this process, the classifier would be continuously trying to classify clean and perturbed images.GAN based network can also be used to rectify an image as shown by Shen et al. [31].

V. Issues and Problems

This section deals with a few of the problems of defenses and the working of neural networks associated with adversarial examples.

1) Even though most of the works concentrate on adversarial examples in different types of deep neural networks, deep learning approaches are affected by adversarial examples in general.
2) Adversarial attacks are general in nature i.e., they can be transferred from one network to another. This property, widely known as transferability is very common, especially in cases where the networks have a similar structure. This is more than often seen in black-box attacks.
3) Deep Neural Networks are designed to behave linearly in high dimensional spaces. This makes them susceptible to adversarial attacks.
4) Though it is true that adversarial attacks can be defended using countermeasures, it is not very difficult to have a counter-counter measure to cause an attack on the defended network again.
5) Though it is always thought that adversarial examples appear in the large regions of the pixel space, it is also true that they can appear in the weaker places too.
6) One of the major issues may be that of data incompletion which roots to the existence of adversarial examples.
7) Though there are a variety of discoveries on adversarial examples and new inventions on the defenses against them, these codes are not publicly available [7]. This makes it difficult for researchers who want to replicate an issue

VI. Future Trends

Though there is a great deal of contribution in the field of deep learning adversarial examples, there are still a few topics of interest for which solutions can be expected in the near future. They are:

1) There is no universally accepted method of detecting/defending an adversarial attack that can be common to all networks [7]. Robustness is a very important factor for this task but since different types of deep neural networks have a different requirement, coming up with a common solution to this problem is still a future work to look out for.
2) Though there are a large number of researches going on in this field to counter adversarial examples, security in the deep neural networks is still a topic of immense interest for research and improvements because more is the growth of DNNs, higher is the concern for security.
3) One of the basic concerns is the property of the Transferability of adversarial examples. Some of the questions that still need more research are why do adversarial examples transfer? Is there a method to stop transferability?
4) Robustness is the most important requirement for defense. But there is not a real accepted benchmark for robustness [7]. Some of the topics to ponder over are How to we evaluate robustness? Do we have a universal methodology to conduct a robustness test? How to we actually compare adversarial example generating methods under different threat models?

5) Some of the areas of interest in the future include adoption open analytics methods, usage, and adoption of Artificial Intelligent for various tasks, greater availability of faster codes and adoption of simplified frameworks.

VII. Conclusion

This survey paper talks about some of the important methods of generating adversarial examples. As we saw, adversarial examples are minute perturbations, invisible to human eye, but cause great damage to the output of deep learning models by fooling networks, producing incorrect results with great confidence. This paper also discusses about why the concept of adversarial examples is important to the deep learning field and why has it become an important area of study. We also saw the general terms in usage related to this. It also deals with some of the interesting detection and defense mechanisms against adversarial examples. In future, there is a lot of scope to dig deeper into this field there is security always remains as a matter of concern to technology.

References

[1] See: https://towardsdatascience.com/a-weird-introduction-to-deep-learning-7828803693b0

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, 2012.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 2015.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition. IEEE, 2009.

[5] Xie, Cihang, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie and Alan L. Yuille. Adversarial Examples for Semantic Segmentation and Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV) (2017): 1378-1387.

[6] Akhtar, Naveed and Ajmal S. Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. IEEE Access 6 (2018): 14410-14430.

[7] Yuan, Xiaoyong, Pan He, Qile Zhu and Xiaolin Li. Adversarial Examples: Attacks and Defenses for Deep Learning. IEEE Transactions on Neural Networks and Learning Systems 30 (2017): 2805-2824.

[8] 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.

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

[10] P. Tabacof, J. Tavares, and E. Valle. (2016). "Adversarial images for variational autoencoders." [Online]. Available: ttps://arxiv.org/abs/1612.00155

[11] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in Proc. IEEE Military Commun. Conf., 2016, pp. 49_54.

[12] 12. Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun.(2017). "Tactics of adversarial attack on deep reinforcement learning agents." [Online]. Available: https://arxiv.org/abs/1703.06748

[13] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 86_94.

[14] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Oct. 2017, pp. 2755_2764.

[15] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel. (2017)."Adversarial attacks on neural network policies." [Online]. Available:https://arxiv.org/abs/1702.02284

[16] A. Rozsa, M. Gnther, E. M. Rudd, and T. E. Boult. (2018). "Facial attributes: Accuracy and adversarial robustness." [Online]. Available: https://arxiv.org/abs/1801.02480

[17] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 86_94.

[18] V. Mirjalili and A. Ross, "Soft biometric privacy: Retaining biometric utility of face images while perturbing gender," in Proc. Int. Joint Conf. Biometrics, 2017, pp. 1_10.

[19] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Oct. 2017, pp. 2755_2764.

[20] 20. S. Shen, R. Furuta, T. Yamasaki, and K. Aizawa, "Fooling neural networks in face attractiveness evaluation: Adversarial examples with high attractiveness score but lowsubjective score," in Proc. IEEE 3rd Int. Conf. Multimedia Big Data, Apr. 2017, pp. 66_69.

[21] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. (2016). "A study of the effect of JPG compression on adversarial images." [Online]. Available: https://arxiv.org/abs/1608.00853

[22] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in Proc. IEEE Symp. Secur. Privacy (SP), May 2016, pp. 582597

[23] A. S. Ross and F. Doshi-Velez. (2017). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. [Online]. Available: https://arxiv.org/abs/1711.09404

[24] S. Gu and L. Rigazio. (2015). Towards deep neural network architectures robust to adversarial examples. [Online]. Available: https://arxiv.org/abs/1412.5068

[25] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi. (2017). DeepCloak: Masking deep neural network models for robustness against adversarial samples. [Online]. Available: https://arxiv.org/abs/1702.06763

[26] J. Lu, T. Issaranon, and D. Forsyth. (2017). SafetyNet: Detecting and rejecting adversarial examples robustly. [Online]. Available: https://arxiv.org/abs/1704.00103

[27] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. (2017). On detecting adversarial perturbations. [Online]. Available: https://arxiv.org/abs/1702.04267

[28] X. Li and F. Li, Adversarial examples detection in deep networks with convolutional filter statistics, in Proc. Int. Conf. Comput. Vis., 2017, pp. 19.

[29] N. Akhtar, J. Liu, and A. Mian. (2017). Defense against universal adversarial perturbations. [Online]. Available: https://arxiv.org/abs/ 1711.05929

[30] H. Lee, S. Han, and J. Lee. (2017). Generative adversarial trainer: Defense to adversarial perturbations with GAN. [Online]. Available: https://arxiv.org/abs/1705.03387

[31] S. Shen, G. Jin, K. Gao, and Y. Zhang. (2017). APE-GAN: Adversarial perturbation elimination with GAN. [Online]. Available: https://arxiv.org/abs/1707.05474

[32] D. Meng and H. Chen, MagNet: A two-pronged defense against adversarial examples, in Proc. ACM Conf. Comput. Commun. Secur. (CCS), 2017, pp. 135147.

[33] W. Xu, D. Evans, and Y. Qi. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. [Online]. Available: https://arxiv.org/abs/1704.01155