ADVREVGAN: ON REVERSIBLE UNIVERSAL ADVERSARIAL ATTACKS FOR PRIVACY PROTECTION APPLICATIONS

Stefania Altini, Vasileios Mygdalis and Ioannis Pitas

Informatics Departments, Aristotle University of Thessaloniki, Thessaloniki, Greece

ABSTRACT

Different adversarial attack methods have been proposed in the literature, mainly focusing on attack efficiency and visual quality, e.g., similarity with the non-adversarial examples. These properties enable the use of adversarial attacks for privacy protection against automated classification systems, while maintaining utility for human users. In this paradigm, when privacy restrictions are lifted, access to the original data should be restored, for all stakeholders. This paper addresses exactly this problem. Existing adversarial attack methods cannot reconstruct the original data from the adversarial ones, leading to significant storage overhead for all privacy applications. To solve this issue, we propose AdvRevGAN, a novel Neural Network architecture that generates reversible adversarial examples. We evaluate our approach in classification problems, where we examine the case where adversarial attacks are constructed by a neural network, while the original images are reconstructed using the reverse transformation from the adversarial examples. We show that adversarial attacks using this approach maintain and even increase their efficiency, while the classification accuracy of the model in the reconstructed data can almost totally be restored.

***Index Terms*—** Adversarial Attacks, Reversible Adversarial Examples, Reversible GANs, AdvRevGan, Privacy protection

1. INTRODUCTION

Traditional research on image privacy protection often assumes human adversaries. In other words, privacy risks are usually quantified by how effectively the information contained in images can be picked up by human eyes and brains. As a result, “blurring”, “pixelation”, and “mosaic” are still the most widely used techniques to protect privacy in images, even while their effectiveness against automatic analysis tools is limited [1], [2]. On the other hand, the field of privacy protection against automatic analysis tools is gaining increased value in social media settings, where we assume that human users are not adversaries, while automatic image crawlers might want to collect images of a specific social media users. To this end, deidentification methods based on universal adversarial attacks have been proposed to disable automatic face detection/recognition [3], or adversarial attack methods that guarantee the principles of k-anonymity [4, 5], while introducing the minimum possible perturbation to the original images, maintaining the utility of the data for human viewers [6].

Nevertheless, an important privacy protection aspect is not only to maintain the utility of the deidentified data but to be able to completely restore the original data, upon request. To this end, the most straightforward approach is to maintain a local copy of the original data. However, such a solution severely increases the storage overhead; therefore, it would be a lot more useful if we only had a single function for calculating the privacy protection transformation. Universal adversarial perturbations could be used to this end [7, 8], however, the actual transformation to the images is merely additive noise, and most importantly, it is the same for any given input image. Thus, a third party with access to a single original and perturbed image pair can easily uncover the perturbation. Therefore, in privacy protection applications, it is essential that this transformation must be unique for a given input image. To this end, transformation-based adversarial attacks have been proposed [9], where the universal perturbation is based on a linear multiplicative transformation, thus it is indeed unique for each image. However, the parameters of the transformation matrix can still be approximated by using a sufficient number of original-adversarial image pairs.

In this work, we extend our previous work in transformation-based universal adversarial perturbations [9] to the nonlinear case. The role of the transformation function is assigned to a deep Generative Neural Network, which is composed of multiple nonlinear activation functions within its architecture. Therefore, the output perturbation is unique for each given input, whereas the parameters of the network cannot be attained by third parties. Specifically, we propose the Adversarial Reversible Generative Network (AdvRevGAN) architecture, which produces reversible adversarial examples that work in various input sizes. In contrast with a simple transformation, where input size directly affects the number of parameters (transformation matrix), AdvRevGAN is able to handle different input sizes and perform well without any change in the size of the model.

2. BACKGROUND AND RELATED WORK

2.1. Universal Adversarial Attacks

Universal Adversarial Attacks (UAP) calculate a perturbation that generalizes for different (almost all) instances of the dataset by employing image-specific adversarial attack constraints. The usage of the same calculated perturbations can decrease the attack complexity by accessing a single vector during inference. This perturbation has been an important contribution to different systems' generalizations [7]. On the other hand, universal adversarial attacks produce more noisy images when compared to image-specific ones [5]. According to to [7], the overall optimization function is formulated as follows:

$$\arg\min_{\mathbf{n}} \|\mathbf{n}\|_2 \text{ s.t. } f\left(\mathbf{x}_i + \mathbf{n}\right) \neq f\left(\mathbf{x}_i\right), \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is a dataset sample, $\mathbf{n}$ is the perturbation vector and $f(\cdot)$ is the classifier such that the target model misclassifies the adversarial sample $\mathbf{y}_i = \mathbf{x}_i + \mathbf{n}$.

In the same vein, a variant of the UAP method, namely SGD-UAP was first introduced by [8]. According to [10], the creation of UAPs is based on using a variance of the Projected Gradient Descent (PDG) attack. In particular, it is proved that SGD can lead to better evasion rates and as a result, it was chosen over other methods [11]. Moreover, it has better convergence compared to UAP. In more detail, it optimizes the objective $\sum_i \mathcal{L}_f(\mathbf{x}_i + \mathbf{n})$ over batches rather than single inputs where $\mathcal{L}_f$ is the model's training loss and $\mathbf{x}_i$ can be batches of input images, and $\mathbf{n} \in \mathbb{R}^D$ are the set of the determined perturbations. The gradients updates towards $\mathbf{n}$ are calculated in batches in the direction of $-\sum_i \nabla\mathcal{L}_f(\mathbf{x}_i+\mathbf{n})$. It has been proven that the SGD-UAP method can create UAPs in a more effective way than the originally proposed method. In both cases, the derived perturbation vector is the same for any given input image.

2.2. Transformation-based Universal Adversarial Attacks

The adversarial attack optimization problem can also be viewed as a transformation estimation one, that is expressed as follows:

$$\begin{aligned} &\underset{|\boldsymbol{\phi}|}{min}\colon f\left(\boldsymbol{g}(\mathrm{x};\boldsymbol{\phi});\theta\right) \neq f(\mathrm{x};\theta), \\ &\text{s.t.: } \ \|\mathbf{x} - \boldsymbol{g}(\mathbf{x};\boldsymbol{\phi})\|_p < \epsilon, \quad p \in [1,\infty), \end{aligned} \tag{2}$$

where $\boldsymbol{g}(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an iterative transformation that maps the data samples of the clean domain $\mathcal{X}$ to an adversarial domain $\mathcal{Y}$, while $\boldsymbol{\phi}$ are the parameters of the transformation. Here, it should be noted that any type of function can be employed in order to solve the proposed optimization problem, i.e., $\boldsymbol{g}(\cdot)$ could represent any linear/non-linear transformation or even a whole neural network. This formulation allows more flexibility in the definition of additional optimization constraints. For instance, the constraint of reversibility, which is very useful in privacy protection settings, could be expressed as an additional optimization constraint, i.e., $\boldsymbol{g}^{-1}(\mathbf{y}) = \mathbf{x}$.

2.3. Multiplicative Universal Adversarial Attacks

The Multiplicative Universal Adversarial Transformation (MUAT) [9] is a method that exploits the Transformation-based universal adversarial attack definition. It examines the simplest case where $\mathbf{g}(\mathbf{x}) = \mathbf{T}\mathbf{x}$ is a linear transformation, where $\mathbf{T} \in \mathbb{R}^{d\times d}$ is a matrix. The original image can be obtained from the adversarial image, if matrix $\mathbf{T}$ is invertible. Specifically, this matrix stores the transformation parameters for clean sample perturbations. While in standard additive noise-based universal adversarial attacks, a simple subtraction using a single adversarial-clean image pair attains perturbations, in the case of multiplicative noise, the analogous is to reverse engineer the matrix $\mathbf{T}$ from the data, which cannot be obtained, using just a pair of clean-adversarial samples, since the rank of $\mathbf{T}$ is supposed to be larger than 1. The overall optimization function of the MUAT method is the following:

$$\min_{\mathbf{T}} \lambda\mathcal{L}_f(f(\mathbf{T}\mathbf{x};\theta),t) + 1 - s(\mathbf{x},\mathbf{T}\mathbf{x}), \tag{3}$$

where $\mathbf{T}$ is the learnable transformation matrix, $t \neq f(\mathbf{x};\theta)$ is a target class, $1 - s(\cdot,\cdot)$ is an additional constraint based on a similarity-based loss function according to the CW-SSIM metric [12] and λ is a hyper-parameter for controlling the significance of the adversarial attack term of the loss function.

2.4. Adversarial Examples with Generative Adversarial Networks

Generative Adversarial Networks (GANs) introduced by Goodfellow [13] for creating generative models P_G which model the data distribution P_{data} used in the training set.
More specifically, GANs consist of two DNNs that are trained simultaneously, a generator network $G : Z \rightarrow Y$ and a discriminator network $D : Y \rightarrow [0,1]$. The generator G is fed with random noise z generating instance y_{adv} from a probability distribution P_G. Then fake y_{adv} and real instance x are fed to discriminator D that tries to differentiate fake from real instances. From the classification procedure of y_{adv}, the discriminator produces a label that indicates whether y_{adv} belongs to the P_{data} (real input) or P_G distribution (adversarial input).

In a nutshell, generator G is trained in a way that maximizes the probability of discriminator D being deceived. In this case, GANs manage well enough to generate instances, almost identical to the original samples. In adversarial attack settings, GANs aim at misleading a pretrained classifier,

$f : Y \rightarrow C$ in a given dataset using a generator that transforms the input image. In particular, in the work of [14] the generator outputs the noise which is being added to the input image generating the adversarial attack in an efficient way. Training in a black box-attack context, losses are based on the input and output of the classifier without any knowledge of the inner function of the classifier. Thus the Loss function is defined as follows:

$$\mathcal{L}^{f}_{adv} = E_x \ell_f(G(\mathbf{x}), t) \tag{4}$$

$$\mathcal{L} = \mathcal{L}^{f}_{adv} + \alpha \mathcal{L}_{GAN} + \beta \mathcal{L}_{hinge}, \tag{5}$$

where α and β control the relative importance of each objective. Note that $\mathcal{L}_{GAN}$ here is used to encourage the perturbed data to appear similar to the original data x, while $\mathcal{L}^{f}_{adv}$ is leveraged to generate adversarial examples, optimizing for the high attack success rate.

3. PROPOSED FRAMEWORK

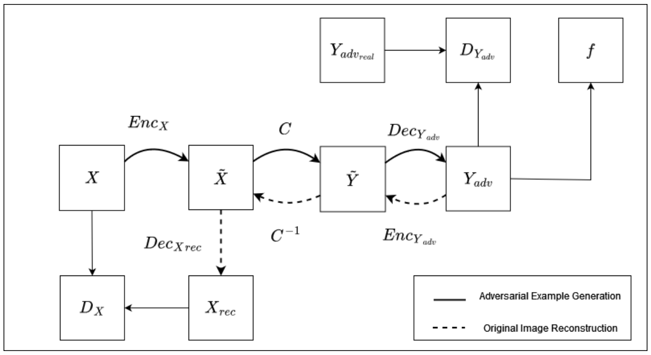

Fig. 1. Architecture of the proposed AdvRevGAN model.

3.1. Reversible Generative Adversarial Networks

Inspired by Image-to-image translation (I2I), our work considers the case where $\mathcal{X}$ is the clean image domain and $\mathcal{Y}$ is the adversarial image domain. The adversarial image domain can be obtained implicitly, by training a generator to produce adversarial examples, or explicitly, by using any adversarial attack. Then, our goal is to create an image-to-adversarial image translation model which is approximately invertible by design. The image-to-image translation aims at transferring images from a source to a target domain while retaining content representations [15] [16]. According to [17], the goal is to find the appropriate mapping between two given domains $\mathcal{X}$ and $\mathcal{Y}$, while minimizing the corresponding loss functions for unpaired training data. To this end, two mappings $G : X \rightarrow Y$ and $G^{-1} : Y \rightarrow X$ are learned, following the cycle-consistency.

In a similar fashion, we create a generator $G : \mathcal{X} \rightarrow \mathcal{Y}$ such that $G(\mathbf{x}_i) = \mathbf{y}_i^{adv}$ in order to generate adversarial examples such that $f(\mathbf{y}_i^{adv}) \neq f(\mathbf{x}_i)$ (untargeted attack). Also we design an "inverse" generator, $G^{-1} : \mathcal{Y} \rightarrow \mathcal{X}$. Then, G^{-1} is another architecture that produces $\mathbf{x}_i^{rec}$ as approximations of $\mathbf{x}_i$. Figure 1 depicts the architecture of our model.

The forward mapping of generator G and the backward one of G^{-1} are broken down into three components. X is the original image domain, $Y_{adv_{real}}$ is the original adversarial image domain while Y_{adv} is the domain of adversarial generated images that are produced by G. We associate a feature space $\tilde{X}$ and $\tilde{Y}$ in higher dimensions for each domain respectively. Mappings between original and adversarial image space are individual and non-invertible. More specifically, for real image space X, we use an encoder $Enc_X : X \rightarrow \tilde{X}$ that extracts the image features of X, lifting the image into a higher dimensionality feature space and a decoder $Dec_{Xrec} : \tilde{X} \rightarrow X_{rec}$ that switch the image back to a lower image space in same dimensions as the initial. We follow the same procedure for generated adversarial image domain Y_{adv} using $Enc_{Y_{adv}} : Y_{adv} \rightarrow \tilde{Y}$ and $Dec_{Y adv} : \tilde{Y} \rightarrow Y_{adv}$.

Between feature spaces, we have an invertible core such that $C : \tilde{X} \rightarrow \tilde{Y}$ and $C^{-1} : \tilde{Y} \rightarrow \tilde{X}$. As a result, we demonstrate the full mappings that are:

$$G(X) = Dec_{Y adv} \circ C \circ Enc_X(X) \tag{6}$$

$$G^{-1}(Y_{adv}) = Dec_{X rec} \circ C^{-1} \circ Enc_{Y_{adv}}(Y_{adv}), \tag{7}$$

where $\circ$ denotes the composition of Enc_X, C, $Dec_{Y adv}$ for function G and $Enc_{Y_{adv}}$, C^{-1},$Dec_{X rec}$ for function G^{-1}. Also for each image space, X and Y_{adv} we use domain-specific discriminators D_X and $D_{Y adv}$ for training with the adversarial loss.

We first define loss for discriminator D_X to ensure that $\mathbf{x}_i$ and $\mathbf{x}_i^{rec}$ are close.

$$\mathcal{L}_{D_X}(G, G^{-1}, D_X) = mse(D(\mathbf{x}), p(\mathbf{x})), \tag{8}$$

where mse is the mean squared error and:

$$p(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \mathcal{X} \\ 1, & \mathbf{x} \notin \mathcal{X}. \end{cases}$$

Similarly we encourage the discriminator $D_{Y adv}$ that $\mathbf{y}^{adv}$ and $\mathbf{x}$ are indistinguishable with the following loss function:

$$\mathcal{L}_{D_{Y_{adv}}}(G^{-1}, G, D_{Y adv}) = mse(D(\mathbf{y}^{adv}), p(\mathbf{y}^{adv})), \tag{9}$$

where:

$$p(\mathbf{y}_{adv}) = \begin{cases} 0, & \mathbf{y}^{adv} \in \mathcal{Y} \\ 1, & \mathbf{y}^{adv} \notin \mathcal{Y}. \end{cases}$$

We define $\mathcal{L}1$ loss for the generator to ensure that $\mathbf{x}_i$, $\mathbf{x}_i^{rec}$ follow the same distribution. Additionally we introduce the $\mathcal{L}_{cycle}$ loss in order to measure the distance between $\mathbf{x}_i$ and $\mathbf{x}_i^{rec}$:

$$\mathcal{L}_{cycle}(G^{-1}, G, \mathbf{x}) = ||G^{-1}(G(\mathbf{x})) - \mathbf{x}||_1. \qquad (10)$$

Besides maintaining visual similarity, the generator network must also derive the actual adversarial examples. For these examples, we demand that they are misclassified by the classifier, formulating a loss function, that exploits some adversarial attack, e.g.,:

$$\mathcal{L}_{adv} = \mathcal{L}_f(f(G(\mathbf{x}), t), \qquad (11)$$

where $\mathcal{L}_f$ is a classification loss function, $f(\cdot)$ is a classifier and t is a target attack class index, that could be different from the original sample label. In fact, any adversarial attack can be employed. In our experiments, we have employed the C&W [18] loss function has been employed.

Furthermore, we ensure that the perturbation on the image does not alternate entirely with the original image. For that reason, we define perturbation loss as follows

$$\mathcal{L}_{pert} = E_{\mathbf{x}}[||\mathbf{y}^{adv} - \mathbf{x}||_1]. \qquad (12)$$

Last, the two losses $\mathcal{L}_{adv}$ and $\mathcal{L}_{pert}$ constitute the loss functions for training G and G^{-1}.

4. EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed AdvRevGAN approach. As a baseline method for adversarial generation we have employed the SGD-UAP method, while for the reversible ones, MUAT is used. All methods have been implemented in Python using Pytorch. The training parameters used in AdvRevGAN and MUAT are the number of epochs, the training samples, and the learning step for the Adam optimizer [19]. For the SGD-UAP method, the parameters are more and consist of the number of epochs, the upper threshold for L_p norm of the attack, the pixel clamping value of the attack, the training samples and the learning step for SGD optimizer. As an evaluation dataset, we have employed MNIST [20], which is commonly used for evaluating adversarial attacks. Although it is a very easy dataset for classification, this is what makes it challenging for adversarial attacks, since adversarial attacks must generate more noise in order to fool the classifier. This way, generating the inverse image is also more difficult. Yet, due to its simplicity, the number of trainable parameters remains low for both the proposed and the competing methods, making the results easily reproducible. Adversarial attacks for both datasets were performed using the Carlini-Wagner L2 method [18].

Table 1. Comparison results on MNIST dataset

	Accuracy (initial dataset)	Accuracy (attacked dataset)	MSE(x, y^{adv})	SSIM(x, y^{adv})
AdvRevGAN	98.4%	0.09%	0.017	0.723
MUAT	98.4%	0.01%	0.056	0.384
SGD-UAP	98.4%	0.07%	0.106	0.300

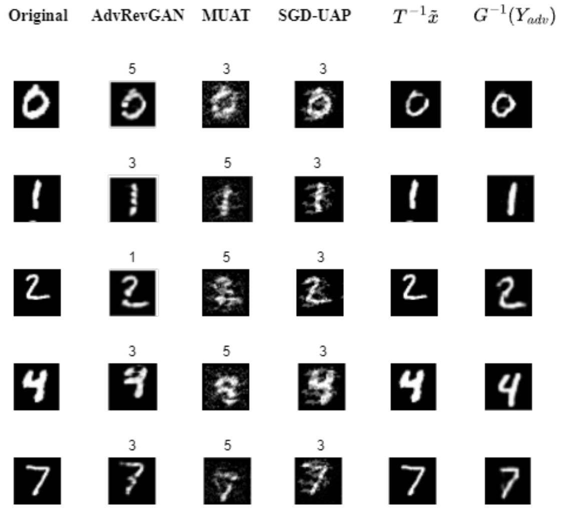

Fig. 2. Adversarial examples and reconstructed images on MNIST Dataset. The first column depicts original images $\mathbf{x}_i$, the next three columns are the corresponding adversarial examples $\mathbf{y}_i{}^{adv}$ generated by the proposed method, MUAT and UAP respectively while above them demonstrated the wrong class that predicted by the model. In the last two columns are demonstrated the reconstructed images $\mathbf{x}_i{}^{rec}$ derived by MUAT and our proposed method respectively.

Table 2. AdvRevGAN results on MNIST dataset

	Accuracy (initial attacked dataset)	Accuracy (classification x^{rec})	MSE(x, y)	SSIM(x, y)	MSE(x, x^{rec})	SSIM(x, x^{rec})
AdvRevGAN	98.4%	90.7%	0.019	0.723	0.010	0.949

The results obtained are analyzed in terms of the Mean Squared Error (MSE) and the Structural Similarity Index Measure (SSIM), which provide insights into the quality of the adversarial examples generated and the reconstructed images produced by the different methods. For implementing the experiments with the MNIST dataset, a LeNet-5 classifier was trained initially on the training set and evaluated on its test set. The accuracy of the classifier that was attacked in our experiments was 98.4%.

Table 1 shows a comparison of the proposed method, SGD-UAP, and MUAT in terms of adversarial attack generation. As can be observed, the proposed method produces less noisy perturbations when compared to other methods, while it remains effective in reducing classification accuracy, as well. Table 2 presents the results of our proposed method, in terms of the reconstruction quality. As can be observed, the classification accuracy in the reconstructed data is restored to 90.7%, while the structural similarity of the reconstructed samples with the original ones is very high, while the MSE of the reconstructed data when compared to the original data is very low. Finally, Figure 2 shows a qualitative evaluation of the competing methods. As can be observed, the proposed method produces adversarial examples that look very similar to the original data while it is able to reconstruct the original data sufficiently well.

5. CONCLUSIONS AND FUTURE WORK

A reversible adversarial attack method has been described, that produces a reversible mapping function that uniquely maps given input images into an adversarial domain, where its inverse can almost reconstruct the original input. The proposed method allows the generation of untargeted adversarial examples that are also reversible for different dataset complexities using generative adversarial networks (GANs). The proposed AdvRevGAN generates adversarial attacks with less noise when compared to legacy adversarial attack methods. Last but not least, the transformation cannot be obtained by third parties, since it is non-linear, and requires access to the neural network architecture and parameters.

According to recent research [21], diffusion models are suggested as a promising alternative to GANs for generating diverse and realistic samples as they use a diffusion process to iteratively transform a noise vector into a sample that matches the data distribution, and they have shown to be more stable and easier to train than GANs. Their ability to capture complex multi-modal distributions makes them a viable alternative for generating synthetic data in scenarios where labeled data is limited or costly to obtain. Future work will consider extending the proposed architecture to also accommodate differential privacy constraints in the adversarial attack optimization problem using more complex datasets and include the diffusion models in our experiments.

Acknowledgement

This work has received funding from the European Union's European Union Horizon 2020 research and innovation programme under grant agreement 951911 (AI4Media). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

6. REFERENCES

[1] Panteleimon Chriskos, Rosen Zhelev, Vasileios Mygdalis, and Ioannis Pitas, "Quality preserving face de-identification against deep cnns," in *Proceedings of the 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6.

[2] Panteleimon Chriskos, Jonathan Munro, Vasileios Mygdalis, and Ioannis Pitas, "Face detection hindering," in *Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 403–407.

[3] Yujia Liu, Weiming Zhang, and Nenghai Yu, "Protecting privacy in shared photos via adversarial examples based stealth," *Security and Communication Networks*, vol. 2017, 2017.

[4] Vasileios Mygdalis, Anastasios Tefas, and Ioannis Pitas, "K-anonymity inspired adversarial attack and multiple one-class classification defense," *Neural Networks*, vol. 124, pp. 296–307, 2020.

[5] Vasileios Mygdalis, Anastasios Tefas, and Ioannis Pitas, "Introducing K-anonymity principles to adversarial attacks for privacy protection in image classification problems," in *Proceedings of the 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021, pp. 1–6.

[6] Seong Joon Oh, Mario Fritz, and Bernt Schiele, "Adversarial image perturbation for privacy protection–a game theory perspective," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1482–1491.

[7] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1765–1773.

[8] Kenneth T Co, Luis Muñoz-González, Leslie Kanthan, Ben Glocker, and Emil C Lupu, "Universal adversarial robustness of texture and shape-biased models," in *Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 799–803.

[9] Alexandros Zamichos, Vasileios Mygdalis, and Ioannis Pitas, "Properties of learning multiplicative universal adversarial perturbations in image data," in *Proceedings of the 2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2022, pp. 1–6.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proceedings of the International Conference on Learning Representations*, 2018.

[11] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S. Davis, and Tom Goldstein, "Universal adversarial training," *Association for the Advancement of Artificial Intelligence*, vol. 34, no. 04, pp. 5636–5643, Apr. 2020.

[12] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[14] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song, “Generating Adversarial Examples with Adversarial Networks,” *OpenReview*, Feb. 2018.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[16] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen, “Image-to-image translation: Methods and applications,” *IEEE Transactions on Multimedia*, vol. 24, pp. 3859–3881, 2022.

[17] Tycho FA van der Ouderaa and Daniel E Worrall, “Reversible gans for memory-efficient image-to-image translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4720–4728.

[18] Nicholas Carlini and David Wagner, “Towards evaluating the robustness of neural networks,” in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*. IEEE, 2017, pp. 39–57.

[19] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[20] Li Deng, “The MNIST database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[21] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah, “Diffusion models in vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2023.