PROPERTIES OF LEARNING MULTIPLICATIVE UNIVERSAL ADVERSARIAL PERTURBATIONS IN IMAGE DATA

Alexandros Zamichos, Vasileios Mygdalis, Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Greece

ABSTRACT

Adversarial attacks in image classification are optimization problems that estimate the minimum perturbation required for a single input image, so the neural network misclassifies it. Universal adversarial perturbations are adversarial attacks that target a whole dataset, estimated by e.g., accumulating the perturbations for each image using standard adversarial attacks. This work treats the universal adversarial perturbation as a problem of transformation estimation. As such, we propose to learn an iterative transformation that maps "clean" images to a "perturbed" domain, by exploiting adversarial attacks. Our experiments show that the proposed formulation leads to easy generation of the adversarial perturbation, while it introduces less noise in the perturbed images, when compared to the state-of-the-art. Finally, this formulation allows us to explore additional properties, notably reversibility of the transformation and attainability of the transformation by using dataset samples.

***Index Terms*—** Multiplicative, Universal, Adversarial attack

1. INTRODUCTION

Adversarial attacks in deep neural network-based image classification involve estimating carefully crafted perturbations to an input image, in order to change the model output. These perturbations can be estimated using the additive noise paradigm [1], while typically remain almost imperceptible to the human eye. The perturbation is generated by exploiting gradient flow towards the input image, using standard neural network optimization functions. Since their original introduction, several works have been proposed in literature that construct adversarial attacks in different settings, depending on the adversary knowledge about the dataset (e.g., labels) and/or access to the neural network parameters (e.g., targeted, un-targeted, white-box, black-box). The reader is referred for more details in the review papers [2] [3] [4]. Another common grouping of adversarial attacks is based on their attack scope. That is, they can be distinguished in *image-specific* adversarial attacks, where methods compute a unique perturbation for each single input image, and in *universal adversarial attacks*, where the perturbation is universal i.e., is the same for any give image in a dataset.

Universal adversarial attacks typically employ image-specific adversarial attack constraints in order to cumulatively calculate a perturbation that generalizes for different (almost all) instances of the dataset. On one hand, the advantage of universal adversarial attacks is that the same calculated perturbation can be employed for attacking a classification system, thus decreasing the attack complexity, as only access to a single vector is required during inference. It has been shown that this perturbation is transferable, as it generalizes well to different classification systems [5]. This property is valuable in adversarial-based privacy protection systems [6], [7]. On the other hand, universal adversarial attacks produce more noisy images when compared to image-specific ones. Furthermore, due to the additive noise paradigm of the adversarial attack formulation, that they can be easily perceived by reverse engineering, thus, by employing a pair of input and perturbed images, the perturbation can be attained by a third party.

This work addresses the problem of universal adversarial attack generation in deep neural network classification as a transformation estimation one. We examine the simplest case where the transformation is linear. As such, we can unify two forms of introduced perturbation (multiplicative and additive) in the same optimization procedure. In the proposed problem formulation, existing universal adversarial attacks can be viewed as special cases where only the bias term of the transformation is estimated. It is shown that exploiting the multiplicative part of the transformation leads to adversarial attacks that are as effective as the additive ones, while introducing less perturbation to the final result. Moreover, our experimental results have shown that the proposed multiplicative transformation is reversible, thus can be applied as easy as additive noise in privacy protection settings. Finally, the multiplicative transformation is not easily attainable, by a single clean-adversarial image pair. The proposed method can be incorporated either as means for testing the robustness of machine learning models, or by being a part of privacy protection methods against automated classification systems.

2. ADDITIVE NOISE MODEL FOR ADVERSARIAL ATTACK GENERATION

Let $\mathbf{x} \in \mathbb{R}^D$ be a vectorized image sample of dimensions D (D is equal to the image's height $\times$ width) having a true label index y from a set $\mathcal{Y} = \{y|\ y \in \mathbb{N}, 1 \leq y \leq C\}$. A deep neural network classifier $f(\mathbf{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the model trainable parameters, has learned to classify images by training the operation $\mathcal{X} \mapsto \mathcal{Y}$ in the representative dataset $\mathcal{S} = \{\mathcal{X}, \mathcal{Y}\}, |\mathcal{S}| = N$, $\mathcal{X} = \{\mathbf{x}|\mathbf{x} \in \mathbb{R}^D\}$. The goal of adversarial attacks for the classification task can be represented as the problem of determining a perturbation vector $\mathbf{n} \in \mathbb{R}^D$ within a noise margin ϵ, so that to change the trained classifier decision for sample $\mathbf{x}$ i.e.:

$$\begin{aligned} &\min_{|\mathbf{n}|}: f(\mathbf{x} + \mathbf{n}; \boldsymbol{\theta}) \neq y, \\ &\text{s.t.:}\ \ \|\mathbf{n}\|_p < \epsilon, \quad p \in [1, \infty) \end{aligned} \tag{1}$$

where $||.||_p$ denotes the ℓ_p-norm and the adversarial sample $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$ must remain in the same domain as $\mathbf{x}$, i.e., be an image. This problem is NP-hard and it cannot be optimized in this setting by existing methods. To this end, equivalent alternative optimization problems have been proposed in the literature.

For instance, the so-called L-BFGS attack [1], is a white-box adversarial attack that assumes access to a continuous loss function denoted by L_f, e.g., the cross-entropy loss function, associated with the outputs of classifier model f to be deceived. By selecting a target label $\hat{y} \in \mathcal{Y}$ for the adversarial example $\tilde{\mathbf{x}}$, it employs the following optimization procedure in an iterative manner:

$$\min_{\mathbf{n}}: \quad \epsilon\|\mathbf{n}\|_2 + L_f(f(\tilde{\mathbf{x}}; \boldsymbol{\theta}), \hat{y}), \tag{2}$$

until the condition $f(\tilde{\mathbf{x}}; \boldsymbol{\theta}) = \hat{y}$ is satisfied, i.e., the label of the classifier changed successfully when the perturbation vector $\mathbf{n}$ is applied to the input image $\mathbf{x}$. Fast Gradient Sign (FGS) method [8] is a significantly faster alternative method that estimates $\boldsymbol{n}$ in a single optimization step along the direction of the gradient sign at each image pixel, at the expense of producing more noisy examples than L-BFGS. DeepFool [9] is an un-targeted adversarial attack method that produces adversarial examples containing even less noise than L-BFGS. It works by approximating the decision boundaries of deep neural networks with linear/affine classifiers. The perturbation is estimated by the orthogonal projection of the sample $\boldsymbol{x}$ to the closest decision boundary, in an iterative optimization process. Other influential works in this area include the Carlini-Wagner (C & W) attack [10], the Jacobian-based Saliency Map Attack [11], the one pixel attack [12]. A detailed list of image-specific adversarial attacks can be found in the review papers [2] [3] [4].

The universal adversarial perturbation (UAP) [5] is an adversarial attack with the additional constraint that equation (1) must be satisfied by all $\mathbf{x} \in \mathcal{X}$. The optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{|\mathbf{n}|}: f(\mathbf{x} + \mathbf{n}; \boldsymbol{\theta}) \neq y, \quad \forall \mathbf{x} \in \mathcal{X}, \\ &\text{s.t.:}\ \ \|\mathbf{n}\|_p < \epsilon, \quad p \in [1, \infty), \end{aligned} \tag{3}$$

where ϵ a parameter for controlling the magnitude of the perturbation. In practice, the perturbation is calculated by accumulating the outputs of DeepFool for all samples $\mathbf{x} \in \mathcal{X}$. As a stopping condition, the function $P(f(\mathbf{x}+\mathbf{n}; \boldsymbol{\theta}) \neq f(\mathbf{x}; \boldsymbol{\theta})) \leq 1 - \delta$ is introduced, where $P(\cdot)$ is a probability function and $0 < \delta < 1$ is a parameter that denotes the target fooling rate to be achieved ($\delta = 0$ denotes a fooling rate of 100%).

Another method that was proposed in [13] is the SGD-UAP method. In this work the authors achieved to create universal adversarial attacks using a variation of the Projected Gradient Descent (PDG) attack [14]. They used the Stochastic Gradient Descent (SGD) algorithm since it has been observed that is can lead to better evasion rates [15]. The SGD method optimizes the objective $\sum_i L_f(\mathbf{x_i} + \mathbf{n})$ over batches rather than single inputs where 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 $-\sum_i \nabla L_f(\mathbf{x_i} + \mathbf{n})$. More detailed description of other universal adversarial attacks can be found in the recent review papers [16].

3. TRANSFORMATION-BASED UNIVERSAL ADVERSARIAL ATTACKS

As stated in the introduction Section, the adversarial attack optimization problem can also be viewed as a transformation estimation one, that is expressed as follows:

$$\begin{aligned} &\min_{|\boldsymbol{\Phi}|}: f\left(\boldsymbol{g}(\mathbf{x}; \boldsymbol{\Phi}); \boldsymbol{\theta}\right) \neq y, \\ &\text{s.t.:}\ \ \|\mathbf{x} - \boldsymbol{g}(\mathbf{x}; \boldsymbol{\Phi})\|_p < \epsilon, \quad p \in [1, \infty) \end{aligned} \tag{4}$$

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 $\tilde{\mathcal{X}}$, 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 be 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., $\boldsymbol{g}^{-1}(\tilde{\mathbf{x}}) = \mathbf{x}$.

This work examines the simplest possible case, i.e., $\boldsymbol{g}(\cdot)$ denotes a linear transformation that perturbs clean samples from their domain to an adversarial one, such that they are

misclassified by the model f. This definition makes more sense in the universal adversarial attack optimization problem. The transformation parameters in this case include a matrix $\mathbf{T} \in \mathbb{R}^{D \times D}$ and a bias term $\mathbf{b} \in \mathbb{R}^{D}$. Therefore, adversarial samples can be represented as follows:

$$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x} + \mathbf{b}. \tag{5}$$

By using this definition, existing adversarial attack methods, including universal adversarial attacks, have only considered the special case where $\mathbf{T} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix and the bias term $\mathbf{b}$ is the analogous of the noise vector $\mathbf{n}$. Therefore, it could be argued that existing adversarial attacks have so far explored many different *additive* perturbations, using a wide range of optimization problems. Hereafter, we define the proposed method as a *multiplicative* perturbation generator.

3.1. Multiplicative Universal Adversarial Transformation (MUAT)

We examine the special case where $\mathbf{b} = \mathbf{0}$, where $\mathbf{0}$ is a vector of zeros. The proposed Multiplicative Universal Adversarial Transformation (MUAT) method, is a multiplicative noise generator formulated as follows:

$$\begin{aligned} &\min_{\|\mathbf{T}\|} f\left(\mathbf{T}\mathbf{x}; \boldsymbol{\theta}\right) \neq y, \\ &\text{s.t.:} \ \ \|\mathbf{x} - \mathbf{T}\mathbf{x}\|_p < \epsilon, \quad p \in [1, \infty), \\ &\qquad \mathbf{x} = \mathbf{T}^{-1}\tilde{\mathbf{x}}, \end{aligned} \tag{6}$$

where an additional constraint requiring that the matrix $\mathbf{T}$ is invertible is also imposed. In the standard additive noise-based universal adversarial attacks, the perturbation is attainable by a single adversarial-clean image pair, by a simple subtraction. However, in the multiplicative noise case, 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 proposed method can be optimized in the same manner as the UAP method, by using any standard adversarial attack (the L-BFGS attack was employed in all our experiments). As initialization values, we have employed $\mathbf{T} = \mathbf{I}$. In order to limit the amount of perturbation introduced in the optimization process, we also introduce a similarity-based loss function as in [17] $s(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, according to the CW-SSIM metric [18]. Thus, we introduce an additional constraint $1 - s(\mathbf{x}, \mathbf{T}\mathbf{x})$ to the proposed objective function to be minimized, and λ, a hyper-parameters for controlling the significance of each term of the loss function. Overall, the proposed optimization problem is the following:

$$\min_{\mathbf{T}} \quad \lambda L_f(f(\mathbf{T}\mathbf{x}; \boldsymbol{\theta}), \hat{y}) + 1 - s(\mathbf{x}, \mathbf{T}\mathbf{x}). \tag{7}$$

In our experiments, we refer to the above mentioned methodology as the MUAT method. Since the matrix $\mathbf{T}$ is defined along the image dimensionality, the computational complexity required for its derivation scales with the dimensionality of the employed images. A faster alternative can also be devised, by requiring that the matrix $\mathbf{T}$ is diagonal, thus limiting the number of learnable parameters from D^2 to D. This variant of the proposed method is refered as the MUAT(diag) method.

4. EXPERIMENTS

This section describes the experiments conducted in order to evaluate the efficiency of the proposed method against the competition. The results of the proposed method were compared with the ones obtained by two recently proposed universal adversarial attack methods, namely the UAP [5] method and the SGD-UAP [13] method.

Three evaluation metrics were used, one for the evaluating how the accuracy of the classifiers is affected by the attack, and two metrics for evaluating the quality of the resulted images. Namely, the metrics that were used are a) the classification accuracy, b) the average Mean Square Error (MSE) and c) the average Structural Similarity Index Measure (SSIM) [18]. The reported MSE and SSIM values are within the scale of [0, 1], while the Accuracy values are scaled from [0, 1] to [0, 100], for readability purposes.

In order to be able to compare the attacks and due to the fact that our scope was both to evaluate the methods in means of accuracy and the quality of the perturbed images, we tried to reach a similar level of accuracy for the perturbed images on all methods and compare them based on the quality of the perturbed images.

Three publicly available datasets, MNIST [19], CIFAR-100 [20], and STL-10 [21], that are commonly used in the literature were employed to this end. Even if these datasets may be considered "easy" for the classification task (due to the high accuracy, usually, achieved by classification models), when it comes to the adversarial attacks, they are more challenging than the "difficult" datasets in which the classification models already fail. Also, it is more challenging to create adversarial perturbations that will lead to less noisy adversarial images for datasets that contain images of small dimensions than those of higher dimensions.

All methods were implemented in Python by using the PyTorch library. The conducted experiments in each respective dataset are detailed in Subsection 4.1. Finally, Subsection 4.2 describes the experiment conducted in order to evaluate how many clean-adversarial image pairs are necessary for determining the matrix $\mathbf{T}$ from the data.

4.1. Experimental results

In our first set of experiments, we employed The MNIST dataset [19], which contains 60,000 training samples and 10,000 test samples from 10 classes, depicting digits from 0

to 9. The size of the images is fixed on 28×28 pixel and the images are provided in gray scale. As baseline classification network, we trained a feed-Forward Neural Network [22], with the following architecture: one input layer (784 neurons), one hidden layer (500 neurons) with a ReLU [23] activation function and an output layer (10 neurons), that achieved an accuracy of 97.73%.

Table 1. Comparison results on MNIST dataset

	Accuracy (initial dataset)	Accuracy (attacked dataset)	MSE	SSIM
MUAT	97.73%	5.91%	0.0315	**0.8306**
MUAT (diag)	97.73%	9.12%	0.0997	0.806
UAP	97.73%	12.5%	0.1011	0.2499
SGD-UAP	97.73%	11.35%	**0.0206**	0.4030

Experimental results are presented in Table 1. As can be observed, the proposed MUAT method reduced the classifier's performance (Accuracy = 5.91%), while maintaining the quality of the perturbed images in a higher level (MSE=0.0315, SSIM=0.8306) when compared with the results of the UAP and SGD-UAP methods. For achieving the mentioned results, in all experiments, the Stochastic Gradient Descent (SGD) was utilized as optimizer. The hyper-parameters used for achieving the above results on MNIST were for MUAT: $\lambda = 0.005$, a learning rate (lr) of 0.0001, 1 epoch of training and 60.000 training samples, while for the MUAT(diag): $\lambda = 0.05$, $lr = 0.1$, training with 300 training samples, momentum$= 0.9$, and weight decay$= 0.0005$.

CIFAR-100 dataset includes 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. As baseline classifier, a ResNet-20 [1] was implemented whose accuracy on the test set of CIFAR-100 was 62.87%.

Table 2. Comparison results on CIFAR-100 dataset

	Accuracy (initial dataset)	Accuracy (attacked dataset)	MSE	SSIM
MUAT	62.87%	15.78%	0.0935	0.8129
MUAT (diag)	62.87%	14.98%	**0.0642**	**0.8584**
UAP	62.87%	17.58%	0.7958	0.3106
SGD-UAP	62.87%	14.85%	1.0340	0.2132

The results presented in Table 2 demonstrate that in this dataset the MUAT (diag) methods achieves the best results in MSE and SSIM metrics. Compared to the results achieved by the UAP and SGD-UAP methods, the MUAT and MUAT (diag) methods present better results in MSE and SSIM metrics. The hyper-parameters used, were, for MUAT: $\lambda = 0.05$, $lr = 0.01$, training with 100 training samples, momentum$=$ 0.9 , while for the MUAT(diag): $\lambda = 0.1$, $lr = 0.1$, training with 1050 training samples, momentum$= 0.9$.

The final dataset used for evaluating the performance of the proposed methods was the STL-10 [21]. The dataset is inspired by the CIFAR-10 dataset but it is modified so as the images to be of a higher resolution (96x96). There are samples of 10 different classes 500 training and 800 test images per class, meaning a dataset of 5,000 training images and 8,000 test images in total. The classifier used in the STL-10 experiments [2] achieved an accuracy of 77.58% on the test set.

Table 3. Comparison results on STL-10 dataset

	Accuracy (initial dataset)	Accuracy (attacked dataset)	MSE	SSIM
MUAT	77.58%	19.87%	0.0383	0.6219
MUAT (diag)	77.58%	21.17%	**0.0244**	**0.773**
UAP	77.58%	21.98%	0.2208	0.0934
SGD-UAP	77.58%	21.8%	0.2079	0.1956

Experimental resutls are drawn in Table 3. As can be observed, the MUAT methods compare favourably against the competition, both in terms of MSE and SSIM values, while providing decreased classification accuracy in the perturbed datasets as well. The hyper-parameters used, were, for MUAT: $\lambda = 0.05$, $lr = 0.01$, training with 1050 training samples, momentum$= 0.9$, while for the MUAT(diag): $\lambda = 0.1$, $lr = 0.1$, training with 5000 training samples, momentum$= 0.9$.

The obtained experimental results can be explained as follows. Improved SSIM metrics are mainly attributed to the application of the SSIM loss in the optimization procedure, thus guided the gradient in matrix $\mathbf{T}$ towards directions that maintain the structural similarity of the image. Moreover, the proposed MUAT method has more learnable parameters than the competition, which provides a significant advantage especially in STL-10 dataset, which contains larger images in size than MNIST and CIFAR-100 datasets. Finally, the MUAT(diag) method produced very good results in CIFAR-100 and STL-10 datasets, perhaps due to the fact that the diagonal matrix components are the most important for scaling the image visual features (pixel luminosities).

A qualitative example is shown in Figure 1. As can be observed, the images obtained by the proposed method appear less perturbed that the competition, especially in STL-10 dataset.

4.2. Recovering the transformation matrix T from the data

Finally, we conducted an additional experiment in order to determine the number of clean-adversarial image pairs requiring for obtaining a good approximation of the inverse transformation $\mathbf{T}^{-1}$. To this end, we have employed the the MNIST dataset. Different number of images (i.e., 1, 32, 64, 128, 256, 512, 784) were used in order to estimate the inverse transfor-

[1] https://github.com/chenyaofo/pytorch-cifar-models

[2] https://github.com/aaron-xichen/pytorch-playground

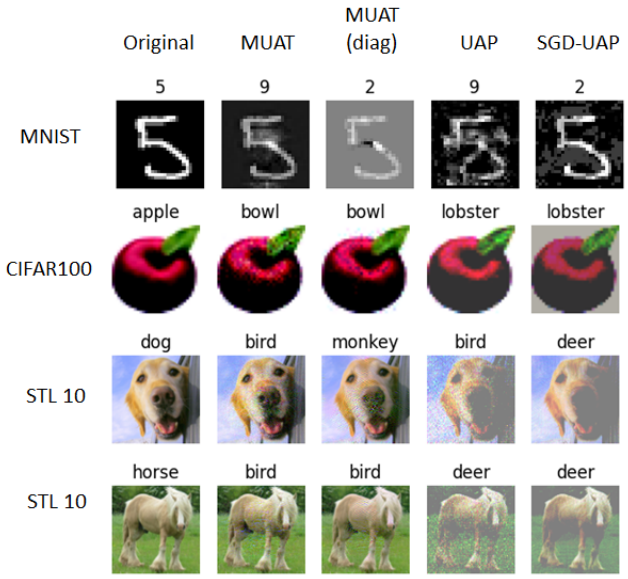

Fig. 1. Examples of perturbed images and their corresponding labels. On the first column, the dataset to which the sample belongs is mentioned, while on the first row it is depicted the method used for the generation of the perturbed images.

mation, from the following formula:

$$\mathbf{T} = (\mathbf{X}^{\dagger}\tilde{\mathbf{X}})^{T}, \tag{8}$$

where $\mathbf{X}^{\dagger} = (\mathbf{X}^{T}\mathbf{X})^{-1}\mathbf{X}^{T}$ is the left pseudoinverse of matrix $\mathbf{X}$, which is a concatenation of a set of original images and $\tilde{\mathbf{X}}$ is the corresponding set of the perturbed images, occurred after applying the MUAT method on the original images.

The results of the above experiment are presented in Figure 2. As could be expected, the more image pairs used increased the approximation quality of the matrix $\mathbf{T}$. From the above experiment it is concluded that in order to achieve an accuracy similar to that before the attack, we should reconstruct $\mathbf{T}$ with almost the same number of image pairs of the rank of $\mathbf{T}$ (which is 784) in this case.

5. CONCLUSIONS

This work presented the problem of generating multiplicative universal adversarial attacks that are reversible but not easily attainable. Many works have been proposed during the past years in literature regarding both image-specific and universal adversarial attacks. In this work we proposed the Multiplicative Universal Adversarial Transformation (MUAT) in two variations, namely the MUAT method and the MUAT diag method. In the first variation, a matrix $\mathbf{T} \in \mathbb{R}^{D\times D}$ is learned during the training phase. In the second variation only the diagonal element of the matrix $\mathbf{T}$ are learnable, targeting thus to decrease the complexity of the attack. Both methods were

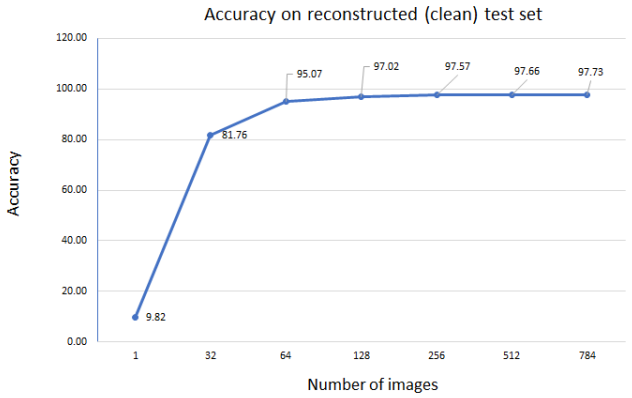

Fig. 2. Experimentas in MNIST dataset in order to determine the number of images needed for the estimation of ($\mathbf{T}$) and consequently of its inverse transformation ($\mathbf{T}^{-1}$) for recovering the original test set images.

evaluated and compared in three datasets with two existing state-of-the-art methods of the literature. The results demonstrate that the proposed attack can lead to a significant decrease of the accuracy of the machine learning models while producing images of better quality than the competition in the terms of MSE and SSIM metrics. Moreover, the proposed attack can be reverted, while due to its multiplicative nature can not be easily obtained by third parties. The proposed method can find application in privacy protection against automated recognition systems (e.g., face recognition in social media). After estimating an appropriate perturbation that generalizes well, it can be used to preserve human privacy on real-time recognition systems, without the need of calculating new perturbations for each image as in image-specific attacks. Moreover, those that have access to the attack's reverse transformation, can use it in order to recover the original images.

Future work could be focused towards expanding the proposed method in many directions. At first, instead of learning $\boldsymbol{g}(\cdot)$ as a linear transformation, non-linear transformations can be considered to be used, such as neural networks. Second, this method could be expanded to classification/regression problems, e.g., object/face detection problems and pixel-level image segmentation.

Acknowledgment

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] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[2] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[3] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[4] Naveed Akhtar and Ajmal Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14410–14430, 2018.

[5] 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.

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

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

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[10] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[11] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[12] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

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

[14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[15] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein, "Universal adversarial training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 5636–5643.

[16] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon, "A survey on universal adversarial attack," *arXiv preprint arXiv:2103.01498*, 2021.

[17] 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.

[18] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," 2009.

[21] Adam Coates, Andrew Ng, and Honglak Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[22] Jürgen Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[23] Abien Fred Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.