Advances in Reliability, Safety and Security, Part 4 – Kołowrocki, Dąbrowska (eds.)

Advances in Reliability, Safety and Security

ESREL 2024
Monograph Book Series

Improving Resilience And Robustness In Artificial Intelligence Systems Through Adversarial Training And Verification

Stefan Brunner, Monika Reif, Martin Rejzek

Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland

Abstract

This contribution presents a comprehensive review of the applicability of adversarial perturbations in the training and verification of neural networks. Adversarial perturbations, designed to deliberately manipulate inputs, have emerged as a powerful tool for improving the robustness and generalization of neural networks. This review systematically examines the utilization of adversarial perturbations in both the training and verification phases of neural network development. In the training phase, adversarial perturbations have been harnessed to enhance model resilience against adversarial attacks by augmenting the training dataset with perturbed examples. Various techniques, such as adversarial training and robust optimization, are explored for their effectiveness in fortifying neural networks against both traditional and advanced adversarial attacks. In the verification realm, adversarial perturbations offer a novel approach for assessing model reliability and safety. Adversarial examples generated during verification expose vulnerabilities and aid in uncovering potential shortcomings of neural network architectures. This review delves into the evaluation of various state-of-the-art adversarial perturbations for different model architectures and datasets and provides a comprehensive analysis of their applications in both training and verification of neural networks. By providing a thorough overview of their benefits, limitations, and evolving methodologies, this review not only contributes to a deeper understanding of the pivotal role adversarial perturbations play in enhancing the robustness and resilience of neural networks, but also provides a basis for selecting the appropriate perturbation for specific tasks such as training or verification.

Keywords: adversarial perturbations, neural network resilience, deep learning, machine learning.

1. Introduction

The phenomenon of adversarial perturbations (also referred to as adversarial noise) plays a critical role in Artificial Intelligence (AI), especially in safety-critical systems. Understanding and mitigating these perturbations is essential to developing AI systems that are robust, resilient and safe, particularly in environments where errors or failures can have severe consequences.

Adversarial perturbations are intentional modifications of input data designed to deceive Machine Learning (ML) models. Their significance is twofold. Firstly, they serve as an effective diagnostic tool for identifying vulnerabilities in AI systems. These perturbations expose potential avenues of exploitation, thereby highlighting system weaknesses (Carlini and Wagner, 2017a). Secondly, these perturbations play a critical role in adversarial training of AI systems. By incorporating them during the training phase, AI systems develop enhanced robustness, preparing them to more effectively face real-world adversarial scenarios (Bai et al., 2021). As such, they play a critical role in securing AI applications in scenarios where safety is paramount (Huang et al., 2020).

Adversarial perturbations are small yet deliberate changes to input data that serve to trigger misclassifications in ML models. Unlike environmental perturbations, this type of perturbation does not occur naturally in measurement devices but is maliciously introduced into ML models by an attacker. The small, intentional changes to the input data (adversarial noise) are derived by solving a non-convex optimization problem (trigger a misclassification while keeping the noise as small as possible). The idea of generating adversarial noise based on an optimization problem to trigger a misclassification in ML models was first introduced by Szegedy et al. (Szegedy et al., 2014). Depending on the formulation of the non-convex optimization problem and the strategy

for solving the problem, a wide range of different adversarial patterns can be generated. Kurakin et al. (Kurakin et al., 2018) described the potential security risk of adversarial patterns in the physical world, if an attacker feeds adversarial patterns into network components of high-risk applications, such as autonomous driving, to cause an accident.

Since this type of perturbation has introduced new risks to AI systems, the robustness properties against adversarial perturbations should be sufficiently proven for high-risk applications.

Zhao et al. (Zhao et al., 2022) conducted a systematic review based on 78 scientific papers related to adversarial perturbations for the adversarial training or verification. Thereby, they analyzed the perturbations in terms of computational complexity (including computation time) and accuracy of triggering a misclassification. Adversarial methods with a low computational complexity and a medium to high accuracy are suitable for the adversarial training while methods with a low to high computational complexity and a high accuracy should be applied for the verification of AI systems.

The research by Zhao et al. refered to the Fast Gradient Sign Method (FGSM) by Goodfellow et al. (Goodfellow et al., 2015) as applicable method for the adversarial training due to the low computational time and medium accuracy. In addition, the Projected Gradient Descent (PGD) by Madry et al. (Madry et al., 2018) and Basic Iterative Method (BIM) by Kurakin et al. (Kurakin et al., 2018) also have a low computational time and medium accuracy, but they both slightly outperform the (FGSM) in terms of accuracy. Therefore, these three white-box methods are suitable for the adversarial training. White-box methods for the verification of AI systems include the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) by Szegedy et al. (Szegedy et al., 2014) and the Jacobian-based Saliency Map Attack (JSMA) by Papernot et al. (Papernot et al., 2016), both of which have high computational complexity and high accuracy. Due to the high computational complexity, they are only suitable for verification unlike the Auto-PGD by Croce et al. (Croce and Hein, 2020), which has a low computational complexity while also achieving a high accuracy. The Wasserstein Attack (WA) by Wong et al. (Wong et al., 2019) has the same properties as the Auto-PGD. Wong et al. also showed that Wasserstein adversarials can improve the overall robustness of AI systems when used in the adversarial training.

The research of Zhao et al. only included one gray-box method (Carlini and Wagner Attack (C&W) by Carlini et al. (Carlini and Wagner, 2017b)), which is suitable for the verification of AI systems due to its high accuracy and its very high computational complexity. In addition, the Descision-based Boundary Attack by Brendel et al. (Brendel et al., 2018) can also be used for the verification, where the accuracy and computational complexity are lower than the C&W, although it is a black-box method. For the adversarial training, the Square Attack (SA) by Andriushchenko et al. (Andriushchenko et al., 2020) and the HotSkipJump Attack (HSJA) by Chen et al. (Chen et al., 2020) are referred due to the low computational complexity while performing a medium accuracy. A black-box method that is suitable for both (training and verification) is the Geometric Decision-based Attack (GeoDA) by Rahmati et al. (Rahmati et al., 2020) (medium computational complexity and high accuracy).

The Adversarial Robustness Toolkit (ART) by Nicolae et al. (Nicolae et al., 2019) is a toolkit that provides a large set of implemented adversarial perturbations such as the PGD, the Auto-PGD and the FGSM.

However, the complexity and variety of potential adversarial attacks pose a significant challenge. Not all adversarial scenarios are equally relevant or effective for testing or improving the resilience of a safety-critical AI system. The selection and design of these attacks depends on the architecture of the AI system and the specific context in which it operates. This complexity is further highlighted by the variety of attack methodologies, ranging from gradient-based to score-based and decision-based attacks, each with its own specific approach and effectiveness in generating adversarial examples (Li et al., 2022). Thus, this paper evaluates various combinations of state-of-the-art model architectures, datasets, and adversarial attacks to support the decision process.

2. Theoretical foundations

Szegedy et al. (Szegedy et al., 2014) demonstrated the possibility of causing misclassification in neural networks for image classification by introducing a small, imperceptible perturbation to the input image. This perturbation, designed to maximize the prediction error of the network, is derived by identifying and modifying key features of the input data that significantly influence the model predictions, thereby inducing errors. Notably, these perturbations are not random, but exhibit systematic patterns that reveal underlying weaknesses in the model classification. Furthermore, it has been observed that a perturbation generated for a specific network can also induce misclassifications in different networks, suggesting a broader applicability. These perturbations, commonly referred to as adversarial perturbations, can be generated by various methods, including gradient-based optimization and evolutionary algorithms.

The process of identifying an adversarial perturbation is formalized in Equation (1). In this equation, the adversarial perturbation, represented by δ, is determined by maximizing the loss function $l(.)$. This maximization is subject to the condition that the perturbation triggers a misclassification, denoted by $f(x + \delta) \neq y$. Additionally, the model, typically a neural network, is denoted by $f(.)$ and serves as the basis for this optimization process.

$$\max_{||\delta<\varepsilon||} l(f(x + \delta), y) \text{ with } f(x + \delta) \neq y. \quad (1)$$

Adversarial perturbations can be categorized as either targeted or untargeted, as shown in Figure 1. Targeted perturbations are specifically designed to cause the model to classify an input as a particular incorrect label. In essence, the perturbation is designed to shift the input data point toward a particular mislabel, not just to cause any misclassification. For example, a targeted perturbation can be designed to cause a model to misclassify an image of a frog as a dog as visualized in Figure 1(a). Conversely, untargeted perturbations aim to induce any misclassification. An example of this is creating a perturbation that causes a frog to be misclassified as any category other than frog (Figure 1(b)).

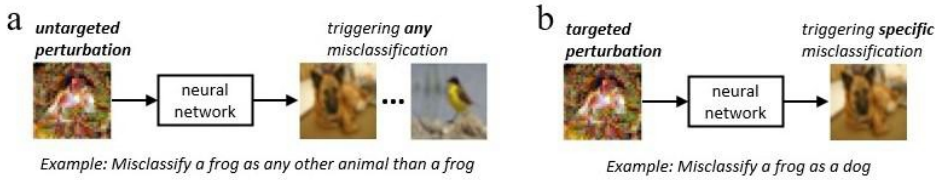

Fig. 1. Difference between (a) untargeted and (b) targeted adversarial perturbations where targeted ones try to trigger a specific misclassification and untargeted ones try to trigger any misclassification.

In addition, adversarial perturbations can be classified into black-box and white-box types, as shown in Figure 2. White-box adversarial perturbations, shown in Figure 2(a), require complete knowledge of the architecture and parameters of the neural network. This access to the network internals allows the use of the gradient information of the neural network $f(.)$ to compute the most effective perturbation, thereby efficiently solving the optimization problem outlined in equation (1). In contrast, black-box adversarial perturbations, shown in Figure 2(b), are based solely on knowledge of the network's input and output. Without access to the gradient information of the network, the optimization problem in equation (1) cannot be addressed directly, resulting in perturbations that may appear more random due to the lack of detailed network information.

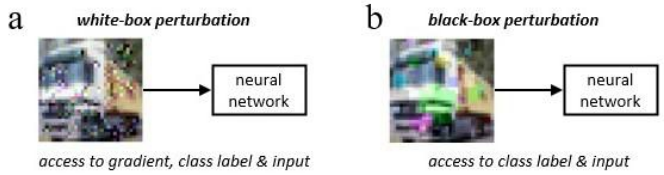

Fig. 2. Difference between (a) white-box and (b) black-box adversarial perturbations, where the white-box perturbation has also access to the input, predicted class label, and gradients of the prediction of the neural network while the black-box perturbation has only access to the input and predicted class label.

In conclusion, white-box adversarial perturbations tend to be more precise and effective as they benefit from comprehensive knowledge of the neural network architecture, which facilitates the generation of sophisticated perturbations. Although black-box attacks are somewhat constrained by limited information, they can still be quite effective in numerous scenarios. This distinction also results in different patterns of adversarial perturbations, as shown in Figure 2. Figure 2(a) shows a white-box perturbation pattern that uses the gradient information of a neural network, typically focusing on the most relevant features for the network's prediction. Meanwhile, the black-box perturbation in Figure 2(b) tends to exhibit a more randomized pattern due to the absence of gradient information.

3. Method

In this paper, we present a comprehensive evaluation of adversarial generation methods applied to three Deep Learning (DL) models on four benchmark datasets. The methodology, depicted in Figure 3, outlines the systematic approach employed to evaluate the state-of-the-art adversarial perturbations described in Table 2 across diverse datasets from different application domains (traffic sign classification, skin lesion detection, digit classification and benchmark classification) and three state-of-the-art DL models. For the model training process, we employ a transfer learning and fine-tuning strategy. Initialization involves leveraging pre-trained weights from ImageNet, with the subsequent exclusion of classification layers. The architecture is then extended by incorporating classification layers, a process outlined in Table 1. This table provides information on the transfer learning setup, fine-tuning setup, image format of the benchmark datasets, and classification accuracy based on the test sets.

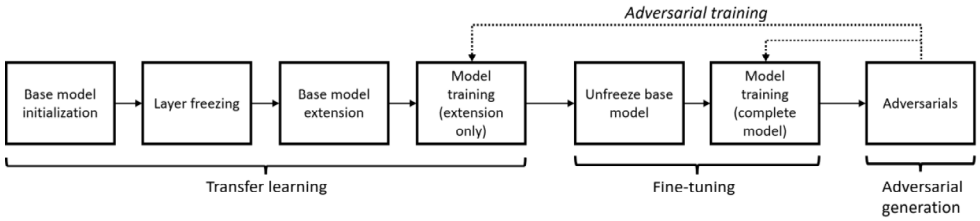

Fig. 3. Methodology for evaluating adversarial methods based on three state-of-the-art DL models across four application domains. A transfer learning and fine-tuning approach is utilized to train these three models. The misclassification rate and mean elapsed time per image is measured for each experimental setup and it is decided whether the respective adversarial method is suitable for adversarial training and or verification.

The trained models, thus configured, serve as the basis for generating adversarial examples. To gauge the efficacy of the various adversarial methods, we measure the mean elapsed time (MET) per image and the misclassification rate (MR). This evaluation is conducted on a subset of the correctly classified test set, comprising 250 randomly selected samples. Based on the MET and MR, the adversarial methods are evaluated for their suitability for adversarial training and/or adversarial verification. Moreover, the adversarial methods are initialized for untargeted attacks. For the selection of suitable methods for adversarial training, a low generation time and a medium to high misclassification rate are required, while for adversarial verification, primarily a high misclassification rate is required. Based on these evaluation results, a decision tree is proposed in order to support the selection of adversarials for training and verification of DL models in different application domains.

Models

In developing this contribution to the field of image classification, a careful selection of relevant models was made based on the authoritative rankings of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) leaderboard (Russakovsky et al., 2014). This leaderboard, a benchmark in the field of visual recognition, provides a robust and empirical basis for the selection of three relevant state-of-the-art models that are used in different applications (Papers with Code, 2023).

The Xception model is a convolutional neural network that extends the Inception architecture (Chollet, 2017). It is based on depth-separable convolutions consisting of two stages: a spatial depth convolution for each input channel, followed by a pointwise convolution that combines the outputs of the depth convolution. This design effectively separates the learning of spatial features and cross-channel correlations, and has 36 convolutional layers organized into 14 modules, with all but the first and last modules structured with linear residual connections. This architecture increases the feature extraction efficiency, making it highly effective in complex image recognition tasks.

ConvNeXt applies Transformer-inspired architectural principles to CNNs to optimize image classification performance (Liu et al., 2022). It differs from traditional ResNet architectures by incorporating Gaussian Error Linear Unit (GELU) activations and using larger kernel sizes. The model structure is characterized by empirically driven optimization of layer configurations and dimensionalities, providing improved scalability and adaptability in processing diverse image datasets. ConvNeXt's design represents a synthesis of Transformer and CNN features aimed at improving image classification accuracy and computational efficiency.

EfficientNetV2 (Tan and Le, 2021) is an iteration of the EfficientNet architecture (Tan and Le, 2019) that focuses on optimizing training efficiency and model scalability. It introduces a mixed-resolution training

method, called progressive learning, and advances compound scaling, which scales network width, depth, and resolution uniformly. This approach results in a model that balances the optimization of these dimensions, leading to faster training and high image classification accuracy while maintaining parameter efficiency.

Table 1. Experimental setup of the three DL models for the four datasets. A transfer learning approach is used, where for each model classification layers are built on top of the model. Moreover, all models are initialized with the ImageNet weights. For the MNIST dataset, three standard Convolutional Neural Networks (CNNs) are initialized instead of the three DL models since the DL models can only be applied to color images. Subsequently, the cells in the table for the MNIST dataset describe the layers of each CNN.

	EfficientNetV2 (small)	ConvNeXt (small)	Xception	Transfer learning	Fine-tuning	Test accuracy (EfficientNetV2; ConvNeXt; Xception)
MNIST image size (28, 28, 1)	3 Convolution layers (filter size of 16, 32, 64), 3 Maxpooling (poolsize 2), 3 MLP layers (unit size of 256, 128, and 10), ReLU activation, Batch normalization, softmax classification	3 Convolution layers (filter size of 8, 16, 32), 3 Maxpooling (poolsize 2), 3 MLP layers (unit size of 128, 64, and 10), ReLU activation, Batch normalization, softmax classification	3 Convolution layers (filter size of 32, 64, 128), 3 Maxpooling (poolsize 2), 3 MLP layers (unit size of 256, 128, and 10), ReLU activation, Batch normalization, softmax classification	20 epochs, batch size 256, Adam (learning rate = 1e-3), normal training (no transfer learning or fine-tuning)	-	0.99; 0.98; 0.99;
CIFAR10 image size (128, 128, 3)	3 MLP layers (unit size of 256, 128, and 10), ReLU activation, Batch normalization, softmax classification	3 MLP layers (unit size of 256, 128, and 10), ReLU activation, Batch normalization, softmax classification	3 MLP layers (unit size of 256, 128, and 10), ReLU activation, Batch normalization, softmax classification	10 epochs, batch size 64, Adam (learning rate = 1e-3)	3 epochs, batch size 32, Adam (learning rate = 1e-4)	0.95; 0.97; 0.94;
GTSRB image size (128, 128, 3)	3 MLP layers (unit size of 512, 256, and 43), ReLU activation, Batch normalization, softmax classification	3 MLP layers (unit size of 512, 256, and 43), ReLU activation, Batch normalization, softmax classification	3 MLP layers (unit size of 512, 256, and 43), ReLU activation, Batch normalization, softmax classification	10 epochs, batch size 32, Adam (learning rate = 1e-3)	3 epochs, batch size 32, Adam (learning rate = 1e-4)	0.97; 0.98; 0.95;
ISIC-2019 image size (128, 128, 3)	3 MLP layers (unit size of 256, 128, and 8), ReLU activation, Batch normalization, softmax classification	3 MLP layers (unit size of 256, 128, and 8), ReLU activation, Batch normalization, softmax classification	3 MLP layers (unit size of 256, 128, and 8), ReLU activation, Batch normalization, softmax classification	10 epochs, batch size 64, Adam (learning rate = 1e-3)	5 epochs, batch size 32, Adam (learning rate = 1e-4)	0.78; 0.81; 0.76;

Datasets

The MNIST (Modified National Institute of Standards and Technology) dataset (LeCun and Cortes, 2005) consists of 70,000 grayscale images, each of 28x28 pixels, typically divided into a training set of 60,000 images and a test set of 10,000 images. These images represent handwritten digits from 0 to 9, making it a 10-class classification problem. Each image is a single-channel (grayscale) array of 28x28 pixels, where the intensity of each pixel is represented as a value from 0 (black) to 255 (white).

The CIFAR datasets, developed by the Canadian Institute for Advanced Research (Krizhevsky et al., 2009), consist of two distinct collections: CIFAR-10 and CIFAR-100. The CIFAR-10 dataset, utilized in this paper, contains 60,000 32x32 pixel color images divided into 10 classes, with each class (such as animals and vehicles) containing 6,000 images. This dataset is further divided into a training set of 50,000 images and a test set of 10,000 images. Each image in CIFAR-10 and is a three-channel (RGB) color composition, with the intensity of each pixel in each channel ranging from 0 (no color) to 255 (full intensity). This color depth, combined with the relatively low resolution of the images, presents a sophisticated challenge to image classification algorithms.

The ISIC-2019 dataset, curated by the International Skin Imaging Collaboration (Codella et al., 2019), contains over 25,331 high-resolution images, each of which varies in size but is typically larger than standard medical images to capture detailed skin lesion characteristics. These images represent a wide variety of skin lesions (8 different classes). Each image in the dataset is a full-color (RGB) photograph, where coloration is critical to accurately represent the different types of skin lesions. The dataset is split into 20,265 training and 5,066 test samples for training the base models and evaluating the adversarial methods.

The GTSRB (German Traffic Sign Recognition Benchmark) dataset (Stallkamp et al., 2011) is a specialized collection for traffic sign recognition, which is critical for the development of autonomous driving systems. It consists of more than 50,000 images (39,209 training samples and 10,791 test samples of varying size and high resolution. These images are categorized into 43 different classes, each representing a unique type of traffic sign, such as speed limits and prohibition signs. Each image in the GTSRB dataset is a full-color (RGB) photograph,

which is critical for accurately capturing the various colors and characteristics of traffic signs. In addition, the dataset provides metadata, including the physical dimensions and geographic locations of the traffic signs. As a pre-processing step, the images are brightened by 0.2 (pixel intensity from zero to one) and then clipped to the range from zero to one if the average pixel intensity of each image is less than 0.25.

Metrics

The MR and MET per image are used to measure the performance of the adversarial methods. Thereby, the MR is derived by subtracting the accuracy from one, where the accuracy measures the correct predicted classes of the DL models. The MET per image measures the adversarial generation time, which is an important factor for the application of the respective adversarial, since a low generation time is preferred for adversarial training. For a fair MET measurement, each adversarial method is executed on Tesla V100-PCIE-32GB (32GB GPU).

Adversarial Methods

The adversarial methods illustrated in Table 2 were selected based on the type of method (e.g., white-box), the applied norm, if suitable, and the applicability for adversarial training and/or adversarial verification, using the meta-study (analysis of 78 scientific publications related to adversarials) by Zhao et al. (Zhao et al., 2022) is utilized to assesses the applicability. In addition, we extend the existing research with our own research on adversarial methods.

Table 2. State-of-the-art adversarial methods used within the different experiments.

White-box methods	Gray-box methods	Black-box methods
Auto Projected Gradient Descent (Auto-PGD) – 1, 2, inf norm	Carlini and Wagner (C&W) – 2, inf norm	Decision-based/Boundary Attack (DBA)
Basic Iterative Method (BIM)		Geometric Decision-based Attack (GeoDA) – 1, 2, inf norm
DeepFool		HopSkipJump Attack (HSJA) – 2, inf norm
Elastic Net Attack (ENA)		Square Attack (SA) – 1, 2, inf norm
Fast Gradient Signed Method (FSGM) – 1, 2, inf norm		Zeroth Order Optimisation (ZOO)
Interative Frame Saliency (IFS) – 1, 2, inf norm		
NewtonFool		
Projected Gradient Descent (PGD) – 1, 2, inf norm		
Universal adversarial perturbation (UAP) – 1, 2, inf norm		
Virtual Adversarial Training (VAT)		
Wasserstein Attack (WA) – 1, 2, .., n p-distance		

4. Results

The MR and MET per image are listed in Table 3 for all initialized adversarial methods for all datasets, with the best performance of each method in terms of high MR and low MET per image highlighted in bold. PGD and Auto-PGD (inf norm) achieve a high MR while maintaining a low MET per adversarial generated across all domains for all initialized state-of-the-art models. In contrast to the inf norm, these methods applied with the first norm achieve a low MR in all setups. However, PGD and Auto-PGD initialized with the second norm achieve an acceptable MR combined with a very low MET per image for color images. White-box adversarial methods with the first or second norm or with a p-distance greater than or equal to one, only achieve a high MR for the WA, regardless of the p-distance applied, but the WA has a higher MET per image compared to the other methods, which is particularly evident for medium-sized images with the format (128, 128, 3), as in the CIFAR10, ISIC-2019 and GTSRB datasets. Other good performing white-box adversarial methods across all domains are BIM and DeepFool, which achieve MR close to one while keeping the MET per adversarial generated low. Adversarial methods such as IFS (inf norm) and UAP (inf norm) also have MRs close to one, but METs much higher than BIM and DeepFool. Furthermore, the UAP (inf norm) only generates acceptable adversarials based on color images, as evaluated in Table 3, since there is a strong drop in MR performance for grayscale images (MNIST).

Table 3. Evaluation of adversarial methods using three state-of-the-art DL models and four benchmark datasets. This table categorizes the results based on two performance metrics: MR and MET per image. Each entry details the results for the three DL models (EfficientNetV2, ConvNeXt, and Xception), separated by semicolons within the cells (e.g., MR EfficientNetV2; MR ConvNeXt; MR Xception). The * marks the attacks, where the 250 randomly selected samples (correctly classified) are reduced to 100 samples due to the high generation time of the respective attack.

	MNIST		CIFAR10		ISIC-2019		GTSRB	
Adversarial method	MR	MET [s]	MR	MET [s]	MR	MET [s]	MR	MET [s]
				White-box adversarials				
PGD (inf norm)	**0.98;0.97;0.98;**	**0.18;0.21;0.16;**	**1.0;0.99;1.0;**	**4.01;4.75;3.78;**	**0.99;1.0;1.0;**	**4.13;4.89;3.95;**	**0.98;0.96;0.99;**	**4.06;4.63;3.88;**
PGD (1 norm)	0.02;0.02;0.03;	0.18;0.22;0.16;	0.02;0.01;0.02;	3.89;4.21;3.69;	0.04;0.04;0.06;	4.06;4.33;3.81;	0.03;0.02;0.02;	3.97;4.38;3.86;
PGD (2 norm)	0.05;0.04;0.06;	0.18;0.21;0.17;	0.48;0.51;0.55;	3.92;4.37;3.54;	0.52;0.49;0.56;	4.11;4.74;4.08;	0.54;0.51;0.58;	4.02;4.69;3.91;
Auto-PGD (inf norm)	**1.00;0.99;1.00;**	**0.26;0.29;0.23;**	**1.0;0.98;0.99;**	**6.28;6.91;6.02;**	**1.0;0.99;1.0;**	**6.43;7.07;6.11;**	**0.99;1.0;0.97;**	**6.56;6.93;6.21;**
Auto-PGD (1 norm)	0.03;0.02;0.03;	0.25;0.28;0.23;	0.01;0.03;0.02;	6.07;6.54;5.87;	0.03;0.01;0.04;	6.28;6.79;6.03;	0.03;0.02;0.03;	6.51;6.89;6.14;
Auto-PGD (2 norm)	0.05;0.05;0.06;	0.25;0.28;0.22;	0.53;0.51;0.58;	6.14;6.31;5.96;	0.49;0.46;0.54;	6.23;6.37;5.81;	0.52;0.45;0.57;	6.26;6.58;6.01;
BIM	**0.96;0.95;0.97**	**0.37;0.42;0.35;**	**1.0;0.99;1.0;**	**3.98;4.21;3.67;**	**0.99;0.97;1.0;**	**4.04;4.38;3.82;**	**0.98;1.0;0.99;**	**4.09;4.47;3.90;**
DeepFool	**0.99;0.96;0.99;**	**0.14;0.19;0.10;**	**1.0;1.0;1.0;**	**3.57;3.76;3.21;**	**1.0;0.99;1.0;**	**3.60;3.89;3.34;**	**0.98;1.0;1.0;**	**3.64;3.82;3.42;**
ENA	0.63;0.67;0.66;	2.09;2.77;1.84;	0.42;0.45;0.51;	12.03;13.41;10.89;	0.39;0.37;0.46;	14.27;15.73;12.94;	0.45;0.41;0.58;	11.38;12.5;9.95;
FSGM (inf norm)	0.66;0.65;0.66;	0.01;0.01;0.01;	0.71;0.63;0.73;	0.40;0.49;0.38;	0.63;0.67;0.59;	0.37;0.51;0.33;	0.57;0.54;0.62;	0.42;0.46;0.40;
FSGM (1 norm)	0.02;0.02;0.04;	0.01;0.01;0.01;	0.01;0.03;0.02;	0.35;0.38;0.31;	0.02;0.01;0.02;	0.34;0.41;0.33;	0.02;0.01;0.04;	0.38;0.43;0.35;
FSGM (2 norm)	0.03;0.02;0.04;	0.01;0.01;0.01;	0.11;0.09;0.12;	0.30;0.34;0.28;	0.08;0.08;0.11;	0.32;0.37;0.31;	0.07;0.05;0.09;	0.29;0.32;0.27;
IFS (inf norm)	**0.98;0.99;0.99;**	**0.49;0.52;0.47;**	**1.0;0.97;0.99;**	27.83;29.50;24.12;	**0.98;0.94;0.96;**	31.29;34.69;29.43;	**1.0;1.0;0.97;**	29.71;31.62;25.48;
IFS (1 norm)	0.01;0.01;0.02;	0.57;0.64;0.51;	0.0;0.03;0.05;	58.04;61.46;53.45;	0.02;0.01;0.04;	52.19;54.33;49.71;	0.02;0.01;0.03;	55.86;57.42;52.97;
IFS (2 norm)	0.03;0.04;0.06;	0.57;0.62;0.49;	0.52;0.49;0.55;	58.01;62.58;55.91;	0.46;0.44;0.52;	53.52;56.74;50.02;	0.50;0.53;0.47;	57.29;60.51;54.14;
NewtonFool	0.10;0.12;0.09	0.31;0.35;0.28;	0.08;0.11;0.09;	12.51;13.99;11.78;	0.05;0.04;0.07;	14.82;18.52;13.96;	0.07;0.07;0.06;	13.37;14.47;11.83;
UAP (inf norm)*	0.23;0.25;0.31;	16.59;19.21;13.79;	**0.81;0.79;0.84;**	35.06;38.48;33.29;	**0.77;0.82;0.81;**	48.72;50.01;44.86;	**0.83;0.81;0.78;**	39.22;43.74;35.07;
UAP (1 norm)*	0.02;0.02;0.03;	18.45;21.92;15.71;	0.03;0.03;0.02;	90.83;102.18;86.90;	0.02;0.01;0.03;	101.83;127.65;94.38;	0.02;0.02;0.04	92.63;99.71;94.91;
UAP (2 norm)*	0.04;0.03;0.05;	12.33;15.46;10.05;	0.21;0.18;0.27;	85.57;91.33;81.96;	0.23;0.15;0.24;	89.69;97.92;85.31;	0.25;0.21;0.29;	78.23;83.49;73.35;
VAT	0.05;0.06;0.08;	1.46;1.87;1.23;	0.05;0.04;0.07;	11.73;12.29;10.58;	0.03;0.03;0.05;	12.07;13.16;10.94;	0.04;0.05;0.04;	10.81;11.64;10.08;
WA (1-distance)	**0.77;0.76;0.79;**	7.51;8.12;7.13;	0.65;0.68;0.68;	33.44;35.76;32.71;	0.56;0.52;0.61;	38.14;40.09;35.37;	0.62;0.54;0.57;	33.58;37.22;30.27;
WA (2-distance)	**0.80;0.82;0.84;**	6.39;6.88;6.08;	**0.71;0.73;0.75;**	39.89;41.05;37.63;	**0.76;0.79;0.81;**	35.13;37.21;32.79;	**0.82;0.80;0.73;**	32.47;34.01;30.53;
				Gray-box adversarials				
C&W (inf norm)*	0.71;0.69;0.73;	23.72;25.48;22.19;	**0.83;0.78;0.81;**	395.7;423.4;373.0;	**0.78;0.81;0.80;**	352.4;396.7;328.1;	**0.79;0.77;0.83;**	292.6;312.9;274.3;
C&W (2 norm)*	0.05;0.04;0.07;	5.81;6.35;5.30;	0.03;0.02;0.06;	124.5;139.8;110.2;	0.09;0.06;0.08;	110.2;124.9;101.1;	0.03;0.02;0.08;	98,4;108.7;90.5;
				Black-box adversarials				
DBA	**1.00;0.98;0.96;**	**2.37;2.89;2.02;**	**0.95;0.92;0.98;**	28.31;31.74;26.49;	**0.92;0.89;0.94;**	35.67;38.21;29.74;	**0.98;0.93;1.0;**	26.62;29.61;24.94;
GeoDA (inf norm)	**1.00;0.97;0.99;**	**0.95;1.27;0.88;**	**0.82;0.78;0.85;**	21.05;24.90;19.87;	**0.82;0.78;0.85;**	20.46;22.08;18.25;	**0.82;0.78;0.85;**	23.28;26.74;20.03;
GeoDA (1 norm)	**0.81;0.76;0.84;**	**0.81;0.92;0.76;**	**0.80;0.77;0.79;**	25.73;29.82;22.83;	**0.78;0.76;0.81;**	28.51;32.77;25.79;	**0.82;0.79;0.80;**	26.23;28.65;22.88;
GeoDA (2 norm)	**0.84;0.80;0.86;**	**0.83;0.97;0.79;**	**0.81;0.82;0.84;**	18.84;20.17;16.90;	**0.79;0.82;0.83;**	19.92;22.37;17.84;	**0.86;0.84;0.88;**	17.38;19.96;15.71;
HSJA (inf norm)*	**0.99;0.98;0.99;**	16.35;19.68;13.21;	**0.96;0.92;0.97;**	151.3;184.8;138.1;	**0.95;0.97;0.96;**	105.9;92.4;112.7;	**0.93;0.94;0.92;**	123.5;144.3;107.6;
HSJA (2 norm)*	**1.00;0.99;0.99;**	13.78;14.21;11.49;	**0.98;0.95;0.99;**	138.5;156.7;113.0;	**1.0;0.98;1.0;**	90.3;105.8;82.4;	**0.99;0.99;1.0;**	96.3;109.2;88.5;
SA (inf norm)	0.65;0.62;0.68;	0.08;0.10;0.07;	**0.92;0.85;0.94;**	**0.35;0.43;0.31;**	**0.90;0.88;0.91;**	**0.37;0.49;0.34;**	**0.96;0.92;0.98;**	**0.31;0.37;0.28;**
SA (1 norm)	0.02;0.02;0.03;	0.01;0.02;0.01;	0.03;0.03;0.04;	0.66;0.82;0.53;	0.01;0.03;0.02;	0.61;0.78;0.57;	0.02;0.01;0.02;	0.59;0.75;0.50;
SA (2 norm)	0.04;0.04;0.04;	0.11;0.17;0.09;	0.05;0.03;0.06;	2.95;3.38;2.49;	0.03;0.04;0.04;	2.78;3.51;2.65;	0.04;0.06;0.08;	3.07;3.52;2.74;
ZOO	0.02;0.03;0.02;	0.01;0.01;0.01;	0.01;0.03;0.01;	0.44;0.62;0.35;	0.02;0.01;0.04;	0.49;0.76;0.39;	0.03;0.02;0.04;	0.38;0.55;0.29;

The evaluated gray-box adversarials show that C&W (inf norm) is able to achieve a high MR (around 0.8) in all domains, independent of the state-of-the-art DL models used. However, for medium-sized color images, the MET per image increases immensely compared to the small-sized grayscale images (MNIST), as shown in Table 3. C&W (second norm) also has a high MET, but also achieves MTs close to zero.

Compared to the gray-box methods, the black-box methods tend to give slightly better MTs and lower METs per image. DBA and GeoDA, regardless of the norm used, achieve an MT close to one, while the MET per image remains around 20 and 25 seconds, respectively, for medium-size color images with the format (128, 128, 3). The HSJA outperforms these two methods in terms of MT. However, the MET per image is also more than 100 seconds for most setups, except for small grayscale images. On the other hand, the SA (inf norm) has a massively shorter generation time compared to the HSJA (less than 0.5 seconds), while achieving almost the same MT for medium-sized color images. Nonetheless, for small grayscale images, the SA results in a MT close to 0.6.

The different adversarial patterns generated are illustrated in Figure 4, which visualizes the diversity of the pattern depending on the adversarial method used, including the applied norm or *p*-distance. The PGD and Auto-PGD (inf norm) generate adversarial patterns across the entire image in all domains, while the WA (2-distance) generates a small subset of pixels as shown in the MNIST and ISIC-2019 samples. In addition, WA is also able

to generate patterns that blur or darken the original image, as illustrated in the MNIST, CIFAR10 and GTSRB samples. A special case of the applied inf norm is the IFS, where only a small adversarial bar is generated, rather than an adversarial pattern across the image as in methods that employ the inf norm, such as PGD and Auto-PGD. While these methods utilize a norm or p-distance, the BIM does not. As a result, the adversarials generated by the BIM method have a different structure, as depicted in Figure 4 (ISIC-2019, GTSRB), where an adversarial blur effect is generated.

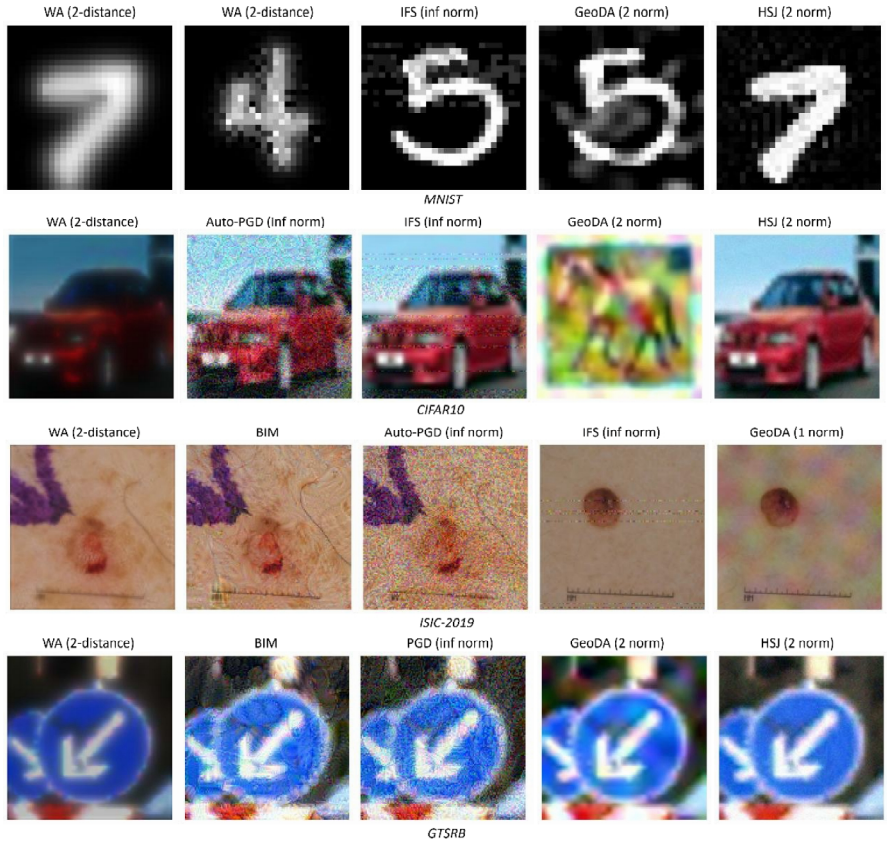

Fig. 4. Generated adversarials with different patterns depending on the attack and the utilized norm or p-distance used. Adversarials generated with the first or second norm or p-distance tend to create patterns with a small subset of adversarial pixels, while adversarials generated with the inf norm tend to create a broad adversarial pattern across the whole image.

In contrast to the white-box attack, the black-box attacks generate different patterns, as shown in the GeoDA (first and second norm) samples for all domains. A pattern is created across the image as in Auto-PGD and PGD. However, instead of creating a noisy pattern, it creates a kind of adversarial mask with different colors and corresponding smooth transitions between the colors, except for grayscale images where the transitions are between different grayscales. The HSJA (second norm) generates a small subset of adversarial pixels, as shown in the MNIST, CIFAR10 and GTSRB examples, which has similarities to the WA (2-distance), while using only the black-box access.

While the DeepFool method achieves a high MR and a low MER per image as shown in Table 3, it can also lead to strong overfitting, as depicted in Figure 5, where the generated adversarial pattern completely obscures the original image in all data domains. Another type of overfitting is illustrated in the generated SA (inf norm) adversarials, which tend to create a single colored area over the entire image, or it generates a pattern that inverts colors, smoothes the image and also creates the typical stripes of the attack. Another type of overfitting can be seen in the pattern generated by the UAP (inf norm) shown in Figure 5, where a pattern is created across the image that degrades the original image to the point of unrecognizability. Subsequently, these methods visualized

in the figure have a high MR due to the effect of overfitting, while the methods visualized in Figure 4 achieve a good performance with no or less overfitted adversarials generated.

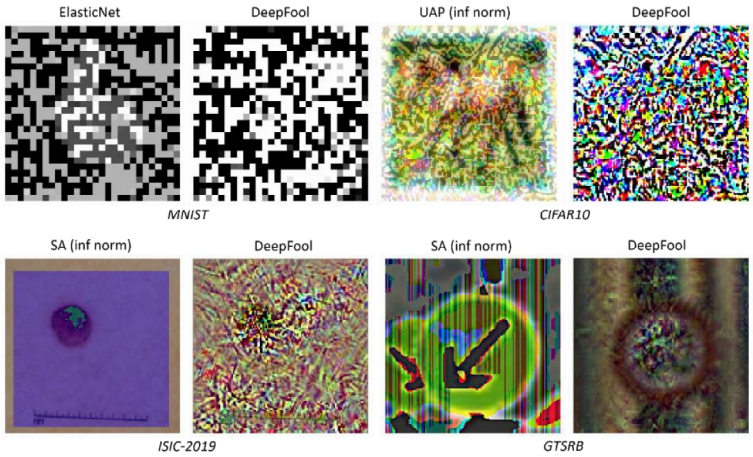

Fig. 5. Overfitted adversarial patterns across the four domains. A typical visual effect of an overfitted adversarial is the degradation of the original image to the point of unrecognizability.

Figure 6 illustrates our proposed decision tree for selecting suitable methods for adversarial training and/or adversarial verification. The decision tree is based on the evaluated adversarial methods across multiple data domains for three state-of-the-art DL models and the standard CNN. In addition, the decision tree considers the type of attack, such as white-box or black-box attack, since depending on the AI system, access to the internals (white-box) is not always granted. To improve the diversity of adversarials, suitable methods for respective norms or p-distances are also included in the decision tree, since different adversarial patterns are generated depending on the applied norm or distance, as shown in Figure 4. Methods that tend to strongly overfit are excluded from the decision tree, despite their high MT, because they degrade images to the point of unrecognizability.

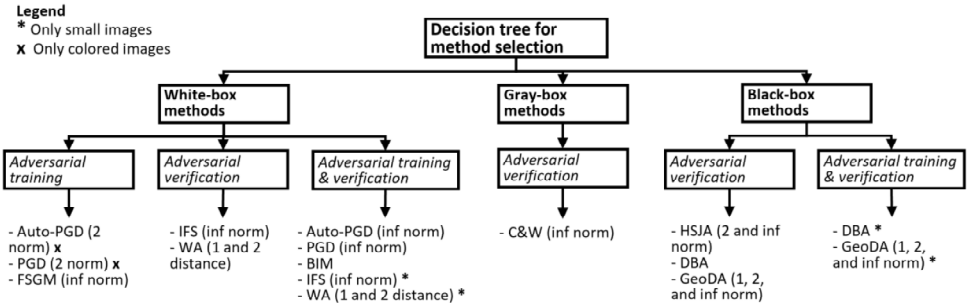

Fig. 6. Proposed decision tree based on the evaluated adversarial methods, providing a set of suitable methods depending on the type of attack (e.g., white-box) and the application of the respective method, such as adversarial training, verification, or both. In addition, the methods cover a wide range of different adversarial patterns to increase the resilience of AI systems.

5. Discussion and conclusions

This contribution highlights the multiple roles of adversarial perturbations in neural network training and verification. In training, these perturbations enhance resilience, but their effectiveness varies with network architecture and dataset type. In verification, they critically assess the robustness of the network and reveal vulnerabilities. Adversarial perturbations are not universally effective in all scenarios. Their suitability depends

on specific task requirements, network architecture, system openness, and dataset characteristics. For specialized applications, a tailored approach is required. This review provides a guide for selecting appropriate methods, but the dynamic nature of neural networks and adversarial strategies requires continuous research and adaptation.

References

Andriushchenko, M., Croce, F., Flammarion, N., Hein, M. 2020. Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search. In A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, 12368, 484–501. Springer International Publishing. https://doi.org/10.1007/978-3-030-58592-1_29

Bai, T., Luo, J., Zhao, J., Wen, B., Wang, Q. 2021. Recent Advances in Adversarial Training for Adversarial Robustness. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 4312–4321. https://doi.org/10.24963/ijcai.2021/591

Biggio, B., Fumera, G., Roli, F. 2014. Security Evaluation of Pattern Classifiers under Attack. IEEE Transactions on Knowledge and Data Engineering 26(4), 984–996. https://doi.org/10.1109/TKDE.2013.57

Brendel, W., Rauber, J., Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. Sixth International Conference on Learning Representations (ICLR 2018). http://arxiv.org/abs/1712.04248

Carlini, N., Wagner, D. 2017a. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 3–14. https://doi.org/10.1145/3128572.3140444

Carlini, N., Wagner, D. 2017b. Towards Evaluating the Robustness of Neural Networks. 2017 IEEE Symposium on Security and Privacy (SP), 39–57. https://doi.org/10.1109/SP.2017.49

Chen, J., Jordan, M. I., Wainwright, M. J. 2020. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. 2020 IEEE Symposium on Security and Privacy (SP), 1277–1294. https://doi.org/10.1109/SP40000.2020.00045

Chollet, F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1800–1807. https://doi.org/10.1109/CVPR.2017.195

Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., Kittler, H., Halpern, A. 2019. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). https://doi.org/10.48550/ARXIV.1902.03368

Croce, F., Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning 119, 2206–2216. PMLR.

Goodfellow, I. J., Shlens, J., Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. International Conference on Learning Representations (2015). http://arxiv.org/abs/1412.6572

Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., Wu, M., Yi, X. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review 37. https://doi.org/10.1016/j.cosrev.2020.100270

Krizhevsky, A., Nair, V., Hinton, G. 2009. CIFAR-10 and CIFAR-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html

Kurakin, A., Goodfellow, I. J., Bengio, S. 2018. Adversarial examples in the physical world. In Artificial intelligence safety and security. Chapman and Hall/CRC, 99–112.

LeCun, Y., Cortes, C. 2005. The mnist database of handwritten digits. https://api.semanticscholar.org/CorpusID:60282629

Li, Y., Cheng, M., Hsieh, C.-J., Lee, T. C. M. 2022. A Review of Adversarial Attack and Defense for Classification Methods. The American Statistician 76(4), 329–345. https://doi.org/10.1080/00031305.2021.2006781

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S. 2022. A ConvNet for the 2020s. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11966–11976. https://doi.org/10.1109/CVPR52688.2022.01167

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. 6th International Conference on Learning Representations, ICLR 2018. http://arxiv.org/abs/1706.06083

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. M., Edwards, B. 2019. Adversarial Robustness Toolbox v1.0.0 (arXiv:1807.01069). arXiv. http://arxiv.org/abs/1807.01069

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., Swami, A. 2016. The Limitations of Deep Learning in Adversarial Settings. 2016 IEEE European Symposium on Security and Privacy (EuroS&P), 372–387. https://doi.org/10.1109/EuroSP.2016.36

Papers with Code. 2023. State of the Art on Image Classification on ImageNet. https://paperswithcode.com/sota/image-classification-on-imagenet

Rahmati, A., Moosavi-Dezfooli, S.-M., Frossard, P., Dai, H. 2020. Geoda: A geometric framework for black-box adversarial attacks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8446–8455.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L. 2014. ImageNet Large Scale Visual Recognition Challenge. https://doi.org/10.48550/ARXIV.1409.0575

Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C. 2011. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. The 2011 International Joint Conference on Neural Networks, 1453–1460. https://doi.org/10.1109/IJCNN.2011.6033395

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. 2014. Intriguing properties of neural networks. 2nd International Conference on Learning Representations, ICLR 2014. http://arxiv.org/abs/1312.6199

Tan, M., Le, Q. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In K. Chaudhuri, R., Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning 97, 6105–6114. PMLR.

Tan, M., Le, Q. 2021. EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning 139, 10096–10106.

Wong, E., Schmidt, F., Kolter, Z. 2019. Wasserstein Adversarial Examples via Projected Sinkhorn Iterations. In K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning 97, 6808–6817. PMLR.

Zhao, W., Alwidian, S., Mahmoud, Q. H. 2022. Adversarial Training Methods for Deep Learning: A Systematic Review. Algorithms 15(8), 283. https://doi.org/10.3390/a15080283