IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 June 2023, accepted 5 September 2023, date of publication 26 September 2023, date of current version 2 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3319389

RESEARCH ARTICLE

Exploring Transferability on Adversarial Attacks

ENRIQUE ÁLVAREZ, RAFAEL ÁLVAREZ, AND MIGUEL CAZORLA, (Senior Member, IEEE)
Department of Computer Science and AI, University of Alicante, 03690 Alicante, Spain

Corresponding author: Enrique Álvarez (enrique.alvarez@ua.es)

This work was supported in part by MCIN/AEI/10.13039/501100011033 under Grant PID2022-138453OB-I00 and in part by ERDF A way of Making Europe.

ABSTRACT Despite of the progress that has been made in the field, the problem of adversarial attacks remains unresolved. The most up-to-date models are still vulnerable, and there is not a simple way to defend against these kinds of attacks; even transformers can be affected by this problem, although they have not been extensively studied yet. In this paper, we study transferability, which is a property of adversarial attacks in which images generated for one architecture can be transferred to another and still be effective. In real-world scenarios like self-driving cars, malware detection, and face recognition authentication systems, transferability can lead to security issues. In order to conduct a behavioral analysis, we select a diverse set of networks and measure how effectively the images produced by various attacks can be transferred among them. We generate adversarial samples for each network and then evaluate them with other networks to determine the corresponding transferability performance. We can observe that all networks are susceptible to transferability attacks, albeit in some cases at the expense of severely distorted images.

INDEX TERMS Adversarial attacks, convolutional neural networks, deep learning, GeoDA, HopSkipJump, SurFree, transferability.

I. INTRODUCTION

The evolution and expansion of artificial intelligence and machine learning (ML) are impressive, especially when considering new text-to-image models such as StableDiffusion [1] and Dall-E2 [2], natural language processing models that are currently used to create virtual shopping assistants and automate customer interaction, as well as applications in fraud detection, face recognition, agricultural process automation, and video games.

Despite the benefits of machine learning, adversarial attacks remain a relevant and unresolved issue. These are algorithms that use information from the target network to generate input images for the purpose of causing misclassifications. Since Szegedy et al. discovered this vulnerability in 2013 [3], a vast amount of research has been conducted on the subject. Defenses like image denoising, gradient masking, adversarial training, gradient regularization, or input reconstruction can help mitigate the attacks, but at the cost of accuracy or performance.

The variety of attacks is vast; there are white-box and black-box, physical and digital, targeted and non-targeted attacks, as detailed in Section III. In the white box setting, the adversary has complete or partial knowledge of the target model; however, black box attacks are closer to a real-world context because they do not require information about the target model and only require an output label. In addition, transferability allows the attacker to use the same outputs against other exposed models.

Adversarial attacks can create insecure situations, such as on self-driving cars, by altering the meaning of the traffic signals captured by sensors, or at gates controlled by facial recognition systems, where a malicious actor can use a printed adversarial exampleÂ to gain access. This study focuses on the transferability property of these attacks because it represents the primary success vector in a realistic setting. To generate adversarial images, an adversary can train a similar new model and launch an attack. The attack is then transferred from the adversarial model to the target

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Cheng.

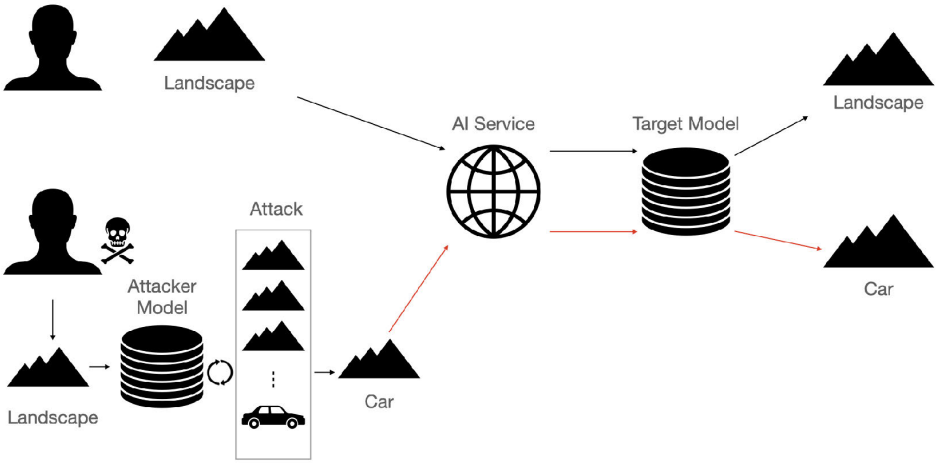

FIGURE 1. Bad actor performing an attack on an open AI service and using transferability.

model by sending the images to the target to achieve incorrect predictions, as shown in Fig. 1.

In particular, we study the transferability of non-targeted attacks in black-box and white-box settings by creating and evaluating adversarial images for five distinct attacks (two white-box and three black-box) and seven target networks in order to assess their transferability and gain a thorough understanding of the problem.

When reviewing the relevant literature, the first mention of transferability can be found in [3], where Szegedy et al. researched how adversarial samples could successfully transfer between models trained with the same dataset. Later, Goodfellow et al. [4] observed that the transferable images were highly aligned with the weights of the model, and that different models learned similar weights for similar tasks. However, they show [5] that this is not true in ImageNet-based models. It was demonstrated in [6] that different models trained on the same task share a fraction of subspaces, allowing transferability. Finally, Petrov et al. [7] study the transferability properties of three white-box attacks across two network families, concluding that similar architectures also have similarities in terms of adversarial attacks.

The rest of the paper is structured as follows. First, in Section II, we briefly explain the targeted architectures. Then an introduction to adversarial attacks and their types can be found in Section III. Next, in Section IV, we describe the study conducted in this research, discussing the obtained results in Section V. Finally, in Section VI we state some conclusions and future research directions.

II. TARGET ARCHITECTURES

Among the selected architectures for the study, there are six families:

- Very Deep Convolutional Networks (VGG). Simonyan and Zisserman described in [8] a new architecture with small convolutional filters that lets 16- or 19-layer convolutional neural networks be trained with good classification performance.
- Residual Networks (ResNet). In 2015, the ResNet framework was proposed in [9]. Residual networks are more complex than VGG models but easier to optimize and have greater depth. Typically, deep networks experience a degradation issue that impacts their precision. ResNets use a fundamental building block that consists of stacked layers that correspond to an underlying mapping (the identity mapping layer), where the identity layer has no parameters and is connected to the previous layer in order to sum its output and feed it to the subsequent layers.
- Inverted Residuals (MobileNetV2). This architecture [10] employs the same concept as [9] but changes the layer shortcut connections. They introduce the ''inverted residual with linear bottleneck'' module. This module expands the input to a higher dimension before applying a lightweight convolution filter. With these modules, the network requires fewer operations to produce results, making it suitable for use in constrained environments such as mobile phones.
- EfficientNet. This network was conceived on the premise that convolutional architectures can scale as more resources become available. Therefore, they focus on depth, width, and resolution in order to achieve superior classification performance compared to conventional networks. Tan and Le [11] proposed a new method for the scalability of dimensions (resolution, depth, and width) using an effective compound coefficient.

- Dense Convolutional Network (DenseNet). Following the concepts of inverted residuals and residual networks, DenseNet networks modify the connections between layers so that the output is close to the inputs of all subsequent layers. With this configuration, the network will have $L(L+1)/2$ connections. The authors note that this configuration reduces the problem of vanishing gradients. In addition, the architecture reduces the number of parameters and improves the propagation of features (see [12]).
- Inception. Szegedy et al. [13] introduced this architecture in 2014. The primary feature of this network architecture is the conservation of computer resources. The authors expanded the width and depth of the network while conserving resources. Internally, Inception is based on the Hebbian principle and multi-scale processing intuition.

For our research, we have chosen to evaluate at least one model of each of the aforementioned architectures: ResNet50, ResNet152V2, DenseNet201, EfficientB0, VGG16, VGG19, MobileNetV2, as well as GoogleNet. We used pre-trained models from Keras[1] and PyTorch[2] (with an average Top-1 accuracy of 75% for Keras).

III. ADVERSARIAL ATTACKS

Since the discovery of adversarial attacks, an important number of algorithms and techniques have been proposed. Given their diversity and volume, it is necessary to categorize them. A common classification found in the literature could be targeted or non-targeted, white-box or black-box, and digital or physical:

- *White-box*. This category of attacks typically utilizes total or partial knowledge of the target network. It is assumed that the attacker knows the architecture, weights, activation functions, and hyperparameters utilized in the training process. It is common practice to employ model gradients in attacks (see [4], [14], [15], [16], [17], [18]).
- *Black-box*. In contrast to white-box attacks, the attacker has no internal knowledge of the target and can only access the network's output. In this scenario, the exposed network is typically protected by a service that restricts the number of queries, making an attack more difficult. One way to evaluate the effectiveness of this technique is to consider the number of successful queries required (see [19], [20], [21], [22]). In addition, black-box attacks can be classified as transfer, score-based, or decision-based. In the transfer category, the attack uses the output labels of the target network to label a new dataset, which is then used to create a new synthetic model [23]. The attacker then applies a white-box attack to this new model in order to generate adversarial images. Score-based attacks are focused on probability vectors or softmax logits, querying the targeted model in order to generate adversarial images (see [21], [24]). Decision-based attacks are likely the most plausible and challenging scenario, in which the attacker has only the output label of the targeted network to create new images (see [19], [22]).
- *Non-targeted*. Depending on the context, the adversary only needs to achieve a misclassification on the target network, regardless of the label result, as long as it does not match the original classification associated with the input image. These attacks can be classified as non-targeted. The benefit of this configuration is that the attack is typically simpler.
- *Targeted*. The goal of this attack is to find an adversarial image in which the target network outputs a desired label instead of the original, as opposed to merely causing a generic misclassification (i.e., the output is an image of a cat with a truck label). Almost every attack described in this study is capable of producing both targeted and non-targeted results.
- *Digital*. The results of an attack are typically digital files, such as images, executables, audio files, etc. They continue to exist in the digital realm and can serve directly as inputs for the targeted networks.
- *Physical*. Typically, physical attacks are generated in the digital domain using white-box or black-box techniques, but they are implemented in the physical world as stickers, clothing, or eyeglasses; e.g., Sharif et al. [25] impersonated a legitimate person while remaining undetected by a facial recognition network by putting printed frames into glasses. This attack is potentially quite dangerous because it can be used in the real world to fool facial recognition-based access control systems or, even worse, to trick autonomous vehicles into misinterpreting traffic signals, endangering human lives [26].

All of the attacks described in this research are implemented in the Adversarial Robustness Toolbox (ART,[3] see [27]) and Foolbox[4] libraries. Both are written in Python and support numerous attack and defense types. Next, we will briefly describe the nature of these attacks.

A. FAST GRADIENT SIGN METHOD

Probably the simplest attack described in the literature, it was developed by Goodfellow et al. [4]. It is a one-step attack that generates a vector with the same length as the original input and whose elements are derived from the cost function of the targeted network.

As shown below, FGSM calculates the input gradients, multiplies the result by a small multiplier, ϵ, and then adds the vector to the original image:

$$\tilde{x} = x + \epsilon \cdot sign(\nabla_x J(\theta, x, y))$$

[1] https://keras.io/api/applications/
[2] https://pytorch.org/vision/stable/index.html
[3] https://github.com/Trusted-AI/adversarial-robustness-toolbox
[4] https://github.com/bethgelab/foolbox

B. PROJECTED GRADIENT DESCENT

Projected Gradient Descent (PGD) [17] was developed in 2017. It is based on the Basic Iterative Method (BIM) [28] and incorporates a novel approach in which the algorithm begins at a random point within the ϵ norm ball [29]. BIM is an iterative variant of the FGSM attack, where a small step size is done in each iteration and the intermediate pixels are clipped, keeping them in the ϵ-neighborhood ball of the original input:

$$\tilde{X}_0 = X$$
$$\tilde{X}_{N+1} = Clip_{X,\epsilon}\{\tilde{X}_N + \alpha sign(\nabla_X J(\tilde{X}_N, y))\}$$

C. HOP SKIP JUMP ATTACK

Hop Skip Jump Attack (HSJA) [19] is a black box and decision-based attack that employs gradient direction estimation to perform adversarial attacks and achieves high success rates with a small number of queries in both targeted and non-targeted contexts. It is based on the Boundary Attack (BA) [30], has no hyperparameters, and controls boundary deviation errors. The attack is composed of three primary components: estimation of the gradient direction, step-size search via geometric progression, and boundary search. The attack attempts to address the following optimization issue:

$$\min_{x'} d(x', x^*)$$
$$s.t.\ \phi_{x^*}(x') = 1$$

Here, d is the distance function that specifies the distance between the adversarial and the original samples. This is represented in Fig 2.

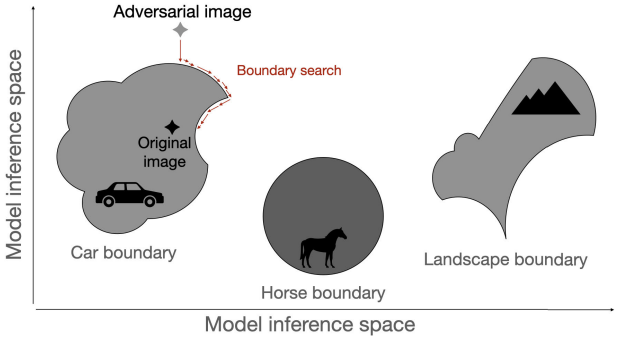

FIGURE 2. Boundary search.

D. GeoDA

Geometric Decision-based Attack [31] is a query-efficient and iterative framework for crafting adversarial attacks. GeoDA is based on the geometric properties of contemporary deep neural networks, in which the decision boundary has a low mean curvature in the vicinity of data samples (see [32]). The key to this attack is to use a hyperplane in the vicinity of a point x to locate a local normal vector (gradient estimation) to the decision boundary. The attack aims to address the following optimization problem:

$$\min_{v} ||v||_p$$
$$s.t.\ w^T(x+v) - w^T x_B = 0$$

where x_B is the boundary point and w is the normal vector to the decision boundary.

E. SurFree

The SurFree attack [22] aims to reduce query budget and accelerate distortion decay relative to HSJA, QEBA [33] and GeoDA by focusing on trials along the decision boundary in different directions rather than gradient surrogate estimations. Therefore, SurFree does not use any information substitution to proceed. In addition, the attack uses the DCT transform to restrict the perturbations to a low-dimensional subspace in order to generate fewer queries.

Given a classifier $f(x) : [0, 1]^D \rightarrow \mathbb{R}^C$ where the output is defined as $cl(x) := \arg\max_k f_k(x)$, the attack will succeed if the adversarial x_a is close to the original input x_0 and $cl(x_a) \neq cl(x_0)$ in the non-targeted setup. Thereby, an output region is defined as $O = \{x \in \mathbb{R}^D : cl(x) \neq cl(x_0)\}$. This gives an optimal output:

$$x_a^* = arg\min_{x \in O} ||x - x_0||$$

The authors assume that, given a point $y \in O$, a line search can be used to find another point $x_b \in [x_0, y]$ that lies on the classifier boundary denoted by ∂O. This concept is based on the work by Fawzi et al. [32], which demonstrated that the boundary of deep neural networks is a smooth, low-curvature surface, justifying the use of a hyperplane to approximate the boundary around the point. We refer to the original work for more details about the attack.

IV. STUDY DESCRIPTION

Using three top-1 label black-box and two white-box attacks, this study attempts to quantify the transferability between various deep neural network architectures. We generate adversarial images with multiple configuration parameters using two white-box algorithms, Fast Gradient Sign Method (FGSM, see [4]) and Projected Gradient Descent (PGD, see [17]), and three black-box algorithms, HopSkipJumpAttack [19], GeoDA [31], and SurFree [22]. One hundred images were randomly selected from the Imagenet [34] database for the attacks (all images were correctly classified by the selected models).

For each attack, we generate 100 adversarial images for each configuration parameter combination. These parameters vary depending on the type of attack. FGSM, PGD, and GeoDA rely on epsilon values as their primary configuration source. SurFree and HSJA, on the other hand, take the number of iterations.

For FGSM and PGD attacks, ART toolbox implementations were utilized. Based on previous experiments, an arbitrary progressive scale of epsilon values and three distance

FIGURE 3. Fast Gradient Sign Method attack with a progression of *epsilons*.

norms were selected as parameters:

$$\epsilon : (0.1, 0.2, 0.3, 0.5, 0.8, 1, 3, 5, 10, 15, 20, 30)$$
$$norms : (L_\infty, L_1, L_2)$$

The epsilon values determine the magnitude of the perturbation on the adversarial image generated, while the norm values establish the attack bound constraint. The effect of applying different epsilon values is illustrated in Fig. 3.

The combination of these parameters allowed us to generate 3,600 adversarial images for each classifier, which we then used to evaluate the transferability between models. All images in our dataset were preprocessed prior to the attack according to the input, and the epsilon values were adapted to the pixel range of the target network; e.g., DenseNet201's pixel range is $x/255$ and the epsilon values used in the attack for the first three values were (0.0003, 0.0007, 0.0011).

In the case of black box attacks, GeoDA and SurFree were not implemented in the ART library at the time this research was conducted; therefore, we used Foolbox to generate adversarial images for GeoDA and the original code for the SurFree study.[5] Although both libraries offer HSJA, we chose the ART implementation. For the GeoDA attack, we specified (1, 3, 5, 10, 25, 50) as the list of ϵ, *sub* as the search space, and L_∞ as the distance norm; the remaining parameters were left at their default values. In the case of the SurFree and HSJA attacks, we used a maximum-query list with values: 500, 1000, 2500, 5000, 10000, 25000, and 50000. This parameter limits the number of attempts made by the algorithms to deceive the target models. 5600 images were generated in total for the black-box and white-box attacks. As stated previously, adversarial images are evaluated through each model to determine their transferability across networks.

V. RESULTS AND DISCUSSION

A. FAST GRADIENT SIGN METHOD

Despite being the simplest attack, it appears that all of the selected networks in this study are susceptible to it, albeit some more so than others. As seen in Fig. 4, the VGG family is extremely vulnerable to attack. With decreasing epsilon values, the VGG16 network classification rate decreases significantly; with 0.3, its precision falls to 50%, and with 0.8, it falls to approximately 10%. Even though VGG19 has more layers than VGG16, its behavior is comparable. In terms of transferability, the results indicate that adversarial examples generated by the attack transfer to each other with a high degree of success, although this becomes more apparent as epsilon approaches 5. To achieve success, higher values of epsilon are required to have a significant impact on the classification rate for the other networks, which have a less steep slope.

The ResNet50 and ResNet152V2 networks, like VGG networks, were highly sensitive to low epsilon values; their precision drops to about 10% with a value of 0.30. The results indicate that a high value of epsilon is required for successful transfer; the slopes of ResNet50 are comparable to those of VGG16 or VGG19, but in the case of ResNet152V2, more distortion is required for a successful transfer.

The transferability slopes of the EfficientNetB0 network are comparable to those of the ResNet and VGG families; however, it is less effective in FGSM for low values of epsilon. Beginning with a value of 5, the attack success stabilizes and remains nearly constant.

As can be seen in the respective graphs, MobileNetV2 and DenseNet201 are highly sensitive to FGSM. With an epsilon of 0.02, MobileNetV2 precision decreases rapidly to 70% and then to 25%, while DenseNet201 precision decreases to 30%. However, for both networks, adversarial samples generated at low values of epsilon do not transfer well, so higher values are required to improve transferability.

In conclusion, FGSM is quite effective on every network tested, even with low image distortion. The transferability property is present in all instances, but its magnitude is proportional to the degree of distortion in the adversarial image and the depth of the targeted network. Adversarial samples generated for deep networks, such as DenseNet201 or ResNet152V2, are typically not transferable. This may be

[5]https://github.com/t-maho/SurFree

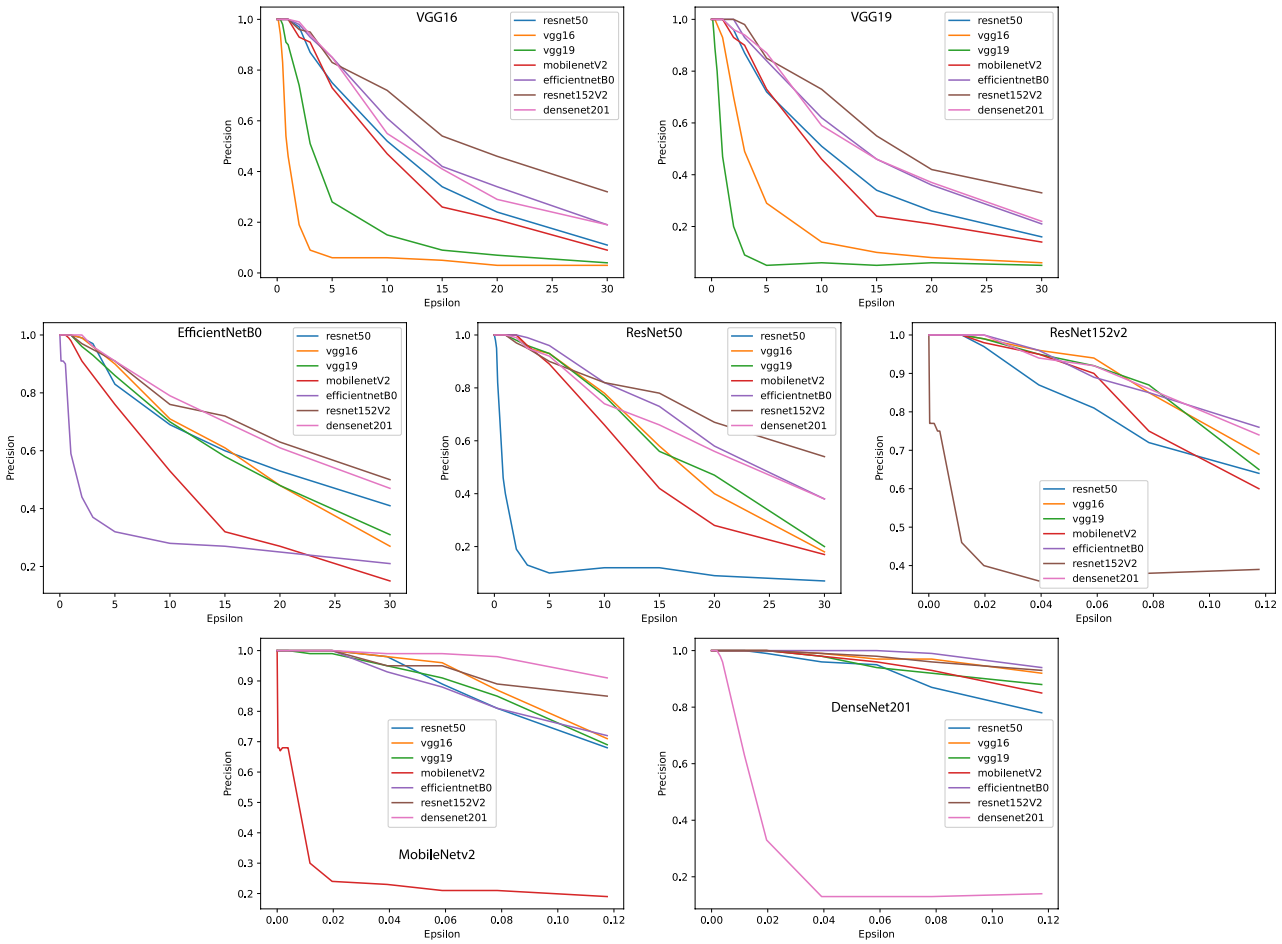

FIGURE 4. FGSM transferability.

TABLE 1. FGSM transferability accuracy.

	VGG16	VGG19	ResNet50	ResNet152V2	MobileNetV2	EfficientNetB0	DenseNet201
VGG16	0.06	0.15	0.52	0.72	0.47	0.61	0.55
VGG19	0.14	0.06	0.51	0.73	0.46	0.62	0.59
ResNet50	0.78	0.77	0.12	0.82	0.66	0.82	0.74
ResNet152V2	0.69	0.65	0.64	0.39	0.60	0.75	0.74
MobileNetV2	0.71	0.69	0.68	0.85	0.19	0.72	0.91
EfficientNetB0	0.71	0.70	0.69	0.76	0.53	0.28	0.79
DenseNet201	0.97	0.94	0.95	0.98	0.96	1.00	0.13

due to the fact that the gradients of the cost function make the outcomes more specific.

In addition, the input image and epsilon values are adapted to a specific pixel range, which can affect the transferability of the network: VGG, EfficientNet, and ResNet50 have an input range of $[0, 255]$, while MobileNetV2 and ResNet152V2 have input ranges of $[x/127.5]$, and $[x/255]$ for DenseNet201. This result conforms to the L_∞ norm; the attack did not meet the L_1, L_2 norms, possibly because the default parameters were insufficient.

Table 1 illustrates how adversarial images generated with FGSM are transferred between networks using an epsilon value of 10 for networks with an input pixel range of [0, 255], $\epsilon = 0.0392$ for a range of $[x/127.5]$, and $\epsilon = 0.0784$ for a range of $[x/255]$. ResNet50 adversarial samples, for instance, reduce VGG16 accuracy to 0.78 and DensetNet images to 0.97.

B. PROJECTED GRADIENT DESCENT

As was the case with FGSM, the PGD attack was unsuccessful for L_1, L_2 norms, so we discarded the images generated for these norms and focused on the L_∞ norm using the same epsilon values and 1200 iterations. With this configuration, the attack success is close to 100% for all networks with low epsilon values. However, almost none of the resulting adversarial samples transferred to other networks.

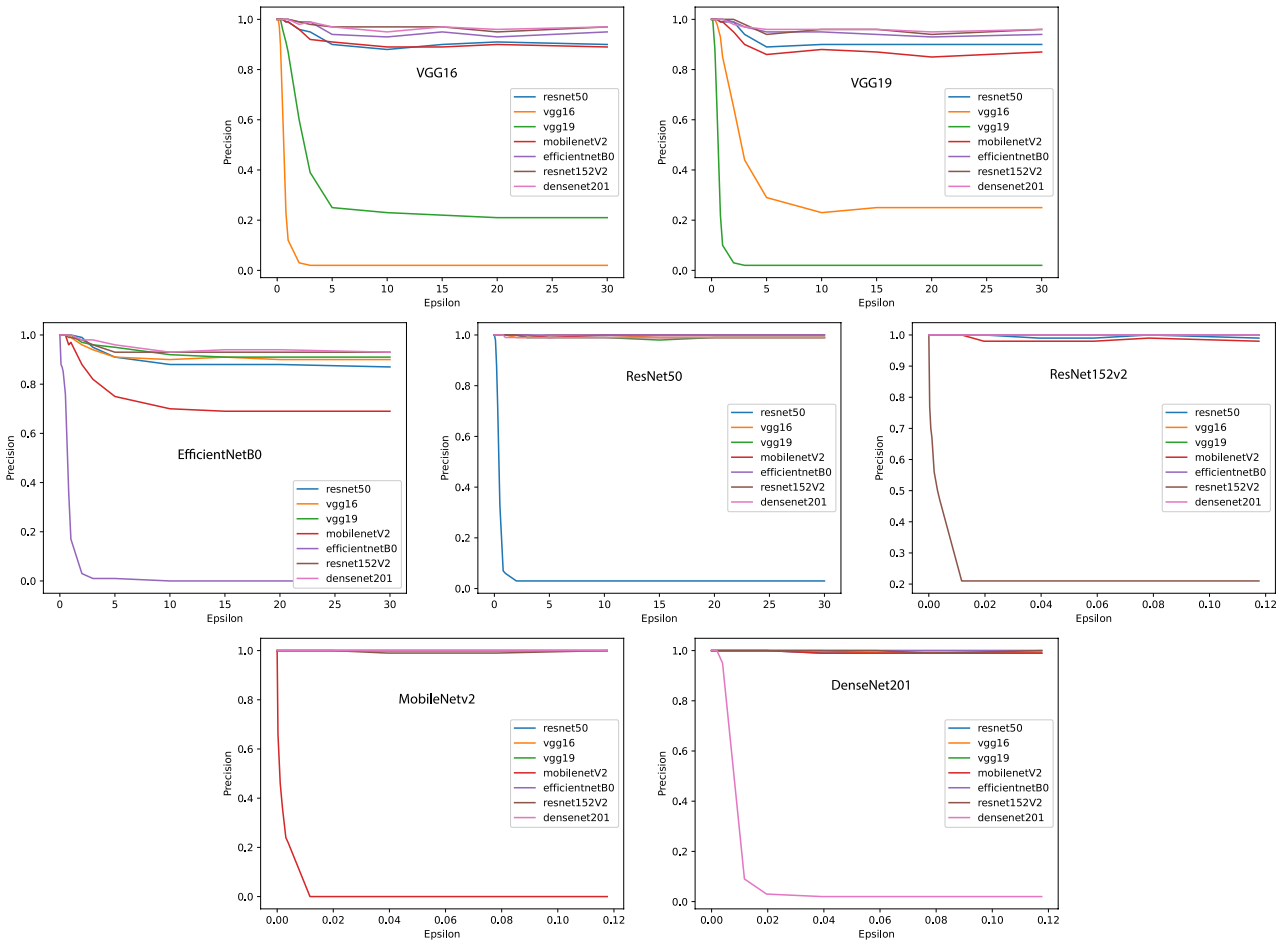

FIGURE 5. PGD attack transferability.

As shown in Fig. 5, ResNet50, ResNet152V4, MobileNetV2, and DenseNet201 images did not result in misclassifications for the other networks.

In this case, only the VGG family presents sensitivity to transferability (the slopes are similar to the FGSM method results). The EfficientNetB0 attack demonstrates a modest degree of transferability with MobileNetV2, reducing its precision to approximately 70%.

Overall, the attack was successful, but its transferability was limited. This may be due to the fact that PGD perturbations are tailored more towards the decision boundary of the target network.

C. GeoDA

This attack was not implemented in ART at the time of this study, so adversarial images were generated using Foolbox. We use *sub* (low frequency sub-space) as the search space, L_∞ as the norm, and a list of epsilon values $(1, 3, 5, 10, 25, 50)$ for the configuration parameters. As shown in Fig. 7, the attack begins to be effective with high values of epsilon on all networks, but the accuracy decreases significantly after the value of 10 is reached. With low values such as those utilized in the FGSM attack, the attack was unable to generate images capable of deceiving the target models. The attack performance for a range of epsilon values is displayed in Table 2.

TABLE 2. GeoDA attack accuracy.

	1	3	5	10	25	50
VGG16	0.04	0.14	0.31	0.62	0.97	0.99
VGG19	0.03	0.09	0.31	0.68	0.98	1.00
ResNet50	0.00	0.02	0.12	0.46	0.89	0.97
ResNet152	0.00	0.20	0.07	0.39	0.90	0.97
MobileNetV2	0.02	0.11	0.25	0.56	0.93	1.00
DenseNet201	0.00	0.02	0.12	0.48	0.92	0.97
GoogleNet	0.01	0.05	0.12	0.40	0.82	0.98

Regarding transferability, it appears in all cases and roughly corresponds to when an attack begins to be successful. Again, VGG families transfer better than others; in other cases, a high epsilon value is required for transferability to emerge.

Fig. 6 depicts the outcome of the GeoDA attack. The success of the attack can be correlated with the precision of the model and the transferability of the generated images;

FIGURE 6. GeoDA attack with a progression of *epsilons*.

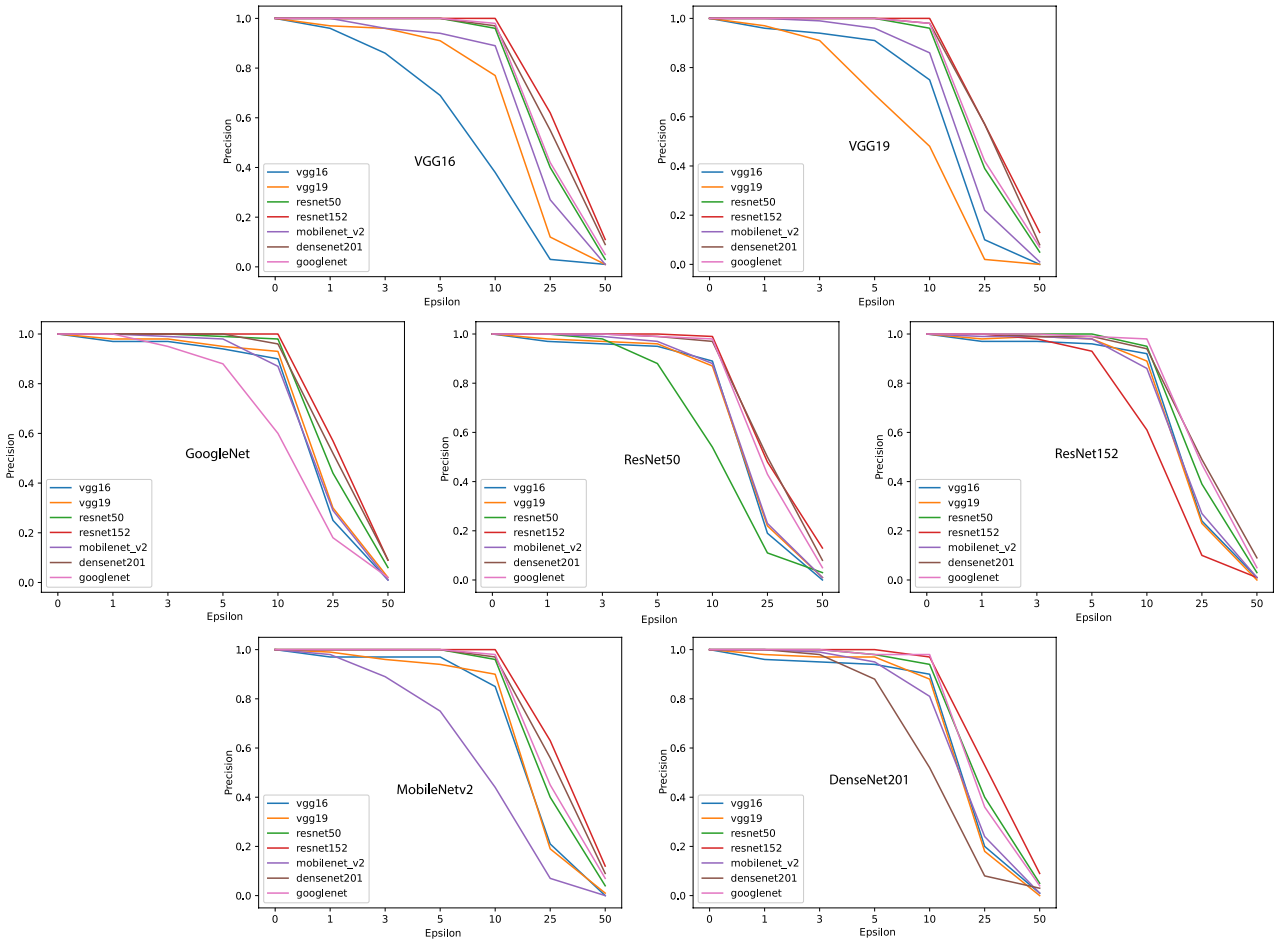

FIGURE 7. GeoDA transferability.

the latter two correspond to epsilon values of 25 and 50. These perturbations can transfer to other networks, but they are highly suspicious (easily detectable) and unreliable in a real-world scenario.

D. SurFree

As previously stated, we utilized the original authors' code to execute this attack. Instead of directly using epsilon values, we set the algorithm to *auto mode*, where the epsilon value is automatically calculated to be as small as possible in each generated image. The attack uses the *max_queries* parameter as a threshold to limit the number of evaluations that can be performed on the target network, with a high number of queries increasing the likelihood of generating an adversarial image with low perturbation and close proximity to the original. We carried out the attack using a range of maximum queries (500, 1000, 2500, 5000, 10000, 25000, 50000) and left the remaining parameters at their default values.

The attack is effective against all networks and is able to deceive them between 500 and 50000 queries. Fig. 8 depicts the outcome of the attack, in which all 100 generated images successfully misled the targets.

However, transferability is only apparent in images generated with few iterations. In the VGG family of networks and MobileNetV2, transferability is limited. Resnet152 and

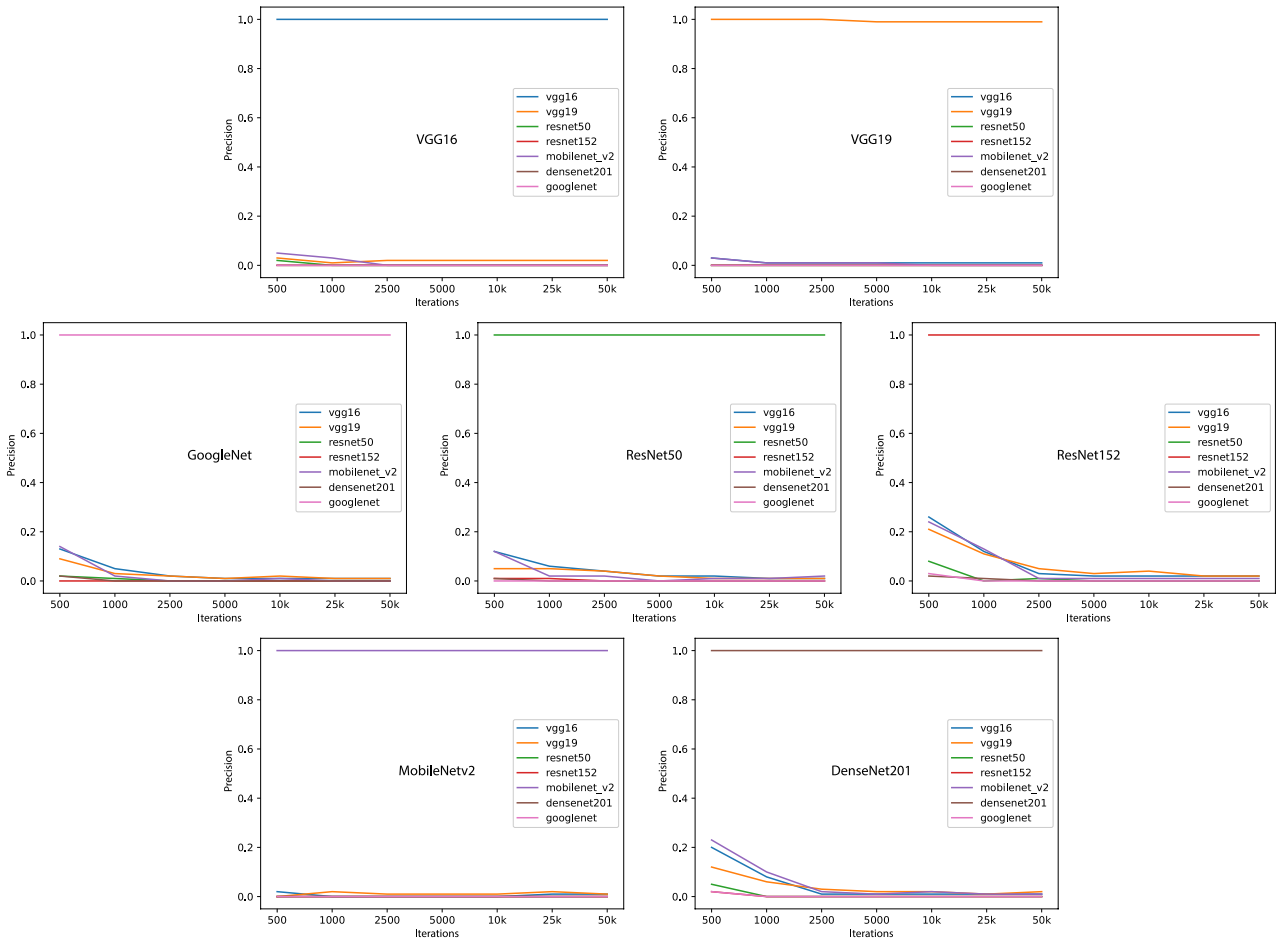

FIGURE 8. SurFree transferability.

FIGURE 9. SurFree attack iterations.

Densenet201 adversarial samples are more transferable, but after 2500 iterations, this property diminishes.

In Fig. 9, we display the adversarial samples generated between 500 and 50000 queries. Since the query budget of the attack is low (500 or 1000), the initial adversarial samples are more distorted; with a larger evaluation budget, however, the attack can generate adversarial samples that are closer to the original image.

E. HOP SKIP JUMP

The HopSkipJump attack was run with the same number of evaluations as the SurFree attack and the *max_iter* parameter set to 5000; all other configuration parameters were left at their default values. The attack had a variety of outcomes and did not behave as expected, as illustrated in Fig. 10; we expected better results and greater consistency as the number of evaluations increased.

The attack's success rate on the VGG16 and VGG19 networks is approximately 0.4 and 0.3, respectively. The attack is almost entirely successful in the case of EfficientNetB0, resulting in a 0.98 reduction in precision. In the case of the ResNet family, the attack only reaches 0.10 for ResNet50 and 0.45 for ResNet152V2, while MobileNetV2 achieved a maximum of 0.50 to 0.38 with 5000 and

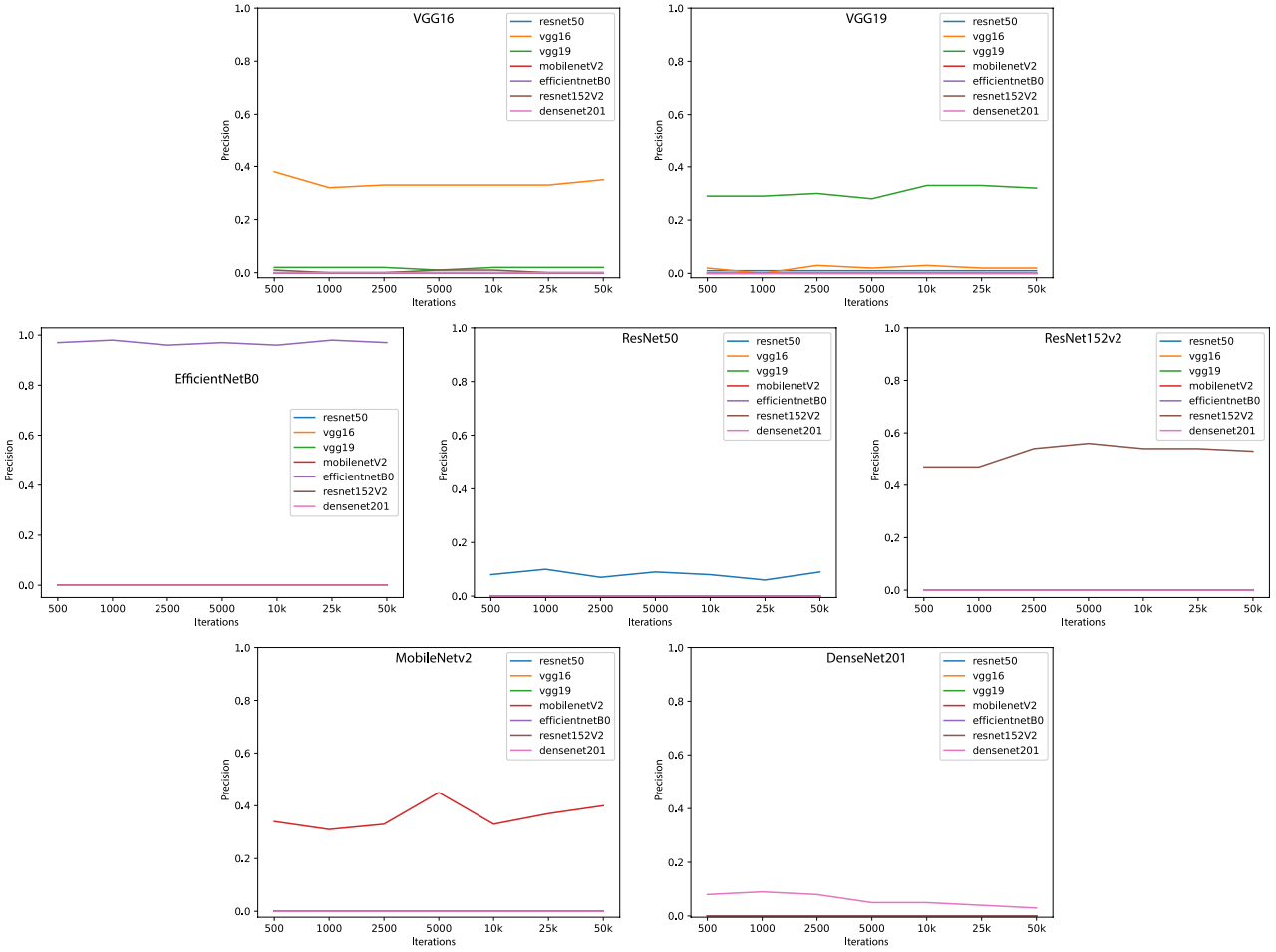

FIGURE 10. HSJA transferability.

10000 evaluations. With a high number of evaluations, the attack for DenseNet201 reaches 0.10 and drops to nearly 0.0. In terms of transferability, none of the experiments produced images that could be transferred.

F. ATTACK FEATURE ANALYSIS

The attacks analyzed in this research were quite effective at deceiving the targeted networks, but the resulting images did not always transfer properly. Fig. 11 depicts the features, or pixels, employed by the attacks to generate adversarial images. The contrast of the images has been adjusted for clarity so that we can see the shape of the features more clearly. Each row represents one of the five attacks: FGSM, PGD, HSJA, GeoDA, and Surfree.

Regardless of the targeted network, all attacks appear to use similar features to achieve misclassification. The features used in FGSM are more distorted than in PGD because it is a one-shot attack, while GeoDa and SurFree generate images with similar pixels. In addition, the features used in the attacks are quite similar across networks. Taking this into consideration, we might wonder why transferability does not occur in an HSJA attack, for example.

The attack logic is to push the adversarial sample towards an incorrect label while avoiding deviating too far from the original image, which is limited by a norm restriction, and it appears to change the most significant pixels of the image that the targeted network uses to identify the label, with all networks appearing to have learned the same patterns. It is possible that the boundaries between labels learned by the models are brittle and that adversarial attacks exploit this to modify pixels. However, these changes did not affect other networks in some cases.

We discovered that in FGSM with a relatively high epsilon value, images begin to transfer on all networks since the pixels are very perturbed, and there is a chance that the images will fall on incorrect labels in the other networks. The images will not transfer if the attack is more effective and achieves a lower distortion because the pixels are adjusted more precisely to the learning boundaries of the targeted network, and this change is unlikely to be sufficient to correct the incorrect labeling of the other models.

FIGURE 11. Adversarial attack features.

VI. CONCLUSION

The primary objective of this research was to better comprehend how transferability behaves with different types of attacks and state-of-the-art networks, as well as how dangerous it can be in real-world situations. Transferability is an important topic in the literature on adversarial attacks because it can be used to deceive AI systems or, worse, to cause malfunctions that endanger human lives. Our findings indicate that the attacks are quite effective against the targeted networks, although, in some instances, a higher amount of image distortion was required to increase their efficacy.

Transferability is more prevalent when target networks have comparable architectures, but also when a greater distance to the original image is established via the epsilon parameter. With a high value of perturbation, however, the images become more suspicious to the human eye and easier to detect, as demonstrated by the obtained results. Using transferability to achieve certain attacks in a real-world scenario can be challenging, as the attacker typically lacks sufficient information to use white-box attacks, and black-box attacks, as we have observed, do not transfer at all. This may be the norm for black-box attacks, but additional testing is required to confirm this hypothesis.

In this paper, we examine a small subset of attacks and network families that are restricted to image recognition tasks, giving us a limited perspective on transferability and the adversarial problem. Although the fundamental problem remains the same, transferability may behave differently in different contexts.

Regarding future work, the study of transferability in a physical world (using attacks as [25], [26], [35]) can be useful in determining the extent to which an attacker could exploit transferability in this context.

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.

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

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

[5] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*.

[6] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," 2017, *arXiv:1704.03453*.

[7] D. Petrov and T. M. Hospedales, "Measuring the transferability of adversarial examples," 2019, *arXiv:1907.06291*.

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

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[11] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.

[12] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, *arXiv:1608.06993*.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
[14] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," 2015, *arXiv:1511.04599*.
[15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," 2015, *arXiv:1511.07528*.
[16] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," 2016, *arXiv:1608.04644*.
[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
[18] J. Su, D. Vasconcellos Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," 2017, *arXiv:1710.08864*.
[19] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," 2019, *arXiv:1904.02144*.
[20] C. Guo, J. R. Gardner, Y. You, A. Gordon Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," 2019, *arXiv:1905.07121*.
[21] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," 2019, *arXiv:1912.00049*.
[22] T. Maho, T. Furon, and E. Le Merrer, "SurFree: A fast surrogate-free black-box attack," 2020, *arXiv:2011.12807*.
[23] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," 2016, *arXiv:1602.02697*.
[24] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," 2017, *arXiv:1708.03999*.
[25] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1528–1540.
[26] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning models," 2017, *arXiv:1707.08945*.
[27] M.-I. Nicolae, M. Sinn, M. Ngoc Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.0.0," 2018, *arXiv:1807.01069*.
[28] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, *arXiv:1607.02533*.
[29] A. Kurakin et al., "Adversarial attacks and defences competition," 2018, *arXiv:1804.00097*.
[30] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2017, *arXiv:1712.04248*.
[31] A. Rahmati, S.-M. Moosavi-Dezfooli, P. Frossard, and H. Dai, "GeoDA: A geometric framework for black-box adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2020, pp. 8446–8455.
[32] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: From adversarial to random noise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1632–1640.
[33] H. Li, X. Xu, X. Zhang, S. Yang, and B. Li, "QEBA: Query-efficient boundary-based blackbox attack," 2020, *arXiv:2005.14137*.
[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," 2014, *arXiv:1409.0575*.
[35] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2017, *arXiv:1712.09665*.

ENRIQUE ÁLVAREZ received the bachelor's degree in computer science, the master's degree in the development of software for mobile devices, and the master's degree in cybersecurity from the University of Alicante, in 2012, 2015, and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Artificial Intelligence. In addition to his academic achievements, he has a proven track record of success in software development in international companies.

RAFAEL ÁLVAREZ received the bachelor's and masters' degrees in computer science, in 2001, and the Ph.D. degree in computer science, in 2005. He is currently an Associate Professor with the Department of Computer Science and Artificial Intelligence, University of Alicante. He is also a member of the Computational Security and Cryptology Research Group. His research interests include security, cryptography, machine learning, and their applications in computer science. He has participated in numerous international conferences and he has been published in prestigious journals. He received the Extraordinary Doctorate Award, in 2009.

MIGUEL CAZORLA (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the University of Alicante, in 2000.

He was a Computer Engineer with the University of Alicante, in 1995. In 1995, he started as an Assistant Professor with the University of Alicante, where he has been a Full Professor, since 2017. He has published more than 70 papers indexed in JCR (with more than 20 in Q1) and more than 100 publications in national and international conferences. He has supervised 19 Ph.D. theses and he is a principal investigator in several national projects (CICYT, Challenges), and having completed multiple transfer contracts with the industry. He is a member of different program committees of national and international conferences. His research interest includes computer vision. From the beginning, he applied these skills to try to solve robotic tasks. In recent years, he has diversified his lines to apply deep learning techniques to different areas (medical image, object recognition, depth estimation, and identification of traffic objects). All his research in recent years has focused on social robotics, that is, applying these techniques to help dependent persons.