Published in: 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-6, doi: 10.1109/IJCNN52387.2021.9534138

Studying the Transferability of Non-Targeted Adversarial Attacks

Enrique Álvarez
dept. Computer Science and A.I.
University of Alicante
Alicante, Spain
enrique.alvarez@ua.es

Rafael Álvarez
dept. Computer Science and A.I.
University of Alicante
Alicante, Spain
ORCID: 0000-0002-8254-6255

Miguel Cazorla
dept. Computer Science and A.I.
University of Alicante
Alicante, Spain
ORCID: 0000-0001-6805-3633

***Abstract*—There is no doubt that the use of machine learning is increasing every day. Its applications include self-driving cars, malware detection, recommendation systems and many other fields. Although the broad scope of this technology highlights the importance of its reliability, it has been shown that machine learning models can be vulnerable to adversarial attacks. In this paper, we study a property of these attacks called transferability across different architectures and models, measuring how these attacks transfer based on a specific number of parameters among three adversarial attacks: Fast Gradient Sign Method, Projected Gradient Descent and HopSkipJumpAttack.**

***Index Terms*—Deep Learning, Adversarial Attacks, Convolutional Neural Networks**

I. Introduction

Deep Neural Networks have revolutionized the field of artificial intelligence and its application to numerous tasks such as speech recognition [1], natural language processing [2], malware detection [3] or image classification [4], among others. Hardware and software have evolved significantly, accelerating AI research, and allowing to train deeper networks. Also, new architectures, such as VGG [5], GoogLeNet [6], ResNet [7] or EfficienNet [8], have achieved great success in large-scale image classification. Despite the great success of these architectures, it has been shown that there are small, non-randomly selected imperceptible perturbations that can cause the misclassification of input examples and, even worse, these perturbations can be generalized across different models. Szegedy et al. [9] discovered in 2013 that some machine learning models are vulnerable to these perturbations called *adversarial examples*. In this paper, we study the generalization of these perturbations, called the *transferability* property in the literature, performing non-targeted attacks, such as Fast Gradient Method [10], Projected Gradient Descent [11] and HopSkipJumpAttack [12], on different models. We then measure the classification rate across the generated images and models with the aim of observing how transferability behaves across deep neural networks.

The rest of the paper is structured as follows. First, an introduction to adversarial attacks and their types can be found in Section II. Then, Section III, describes the study conducted in this research. Next, we discuss the results we have obtained in Section IV. Finally, Section V states some conclusions and future research directions.

II. Adversarial attacks

Machine learning models represent a mapping function of an input x to an output y in the form of $F\colon X \to Y$ and define a solution space that is determined by the training dataset. In the context of image classification, we can define an adversarial example as:

$$\tilde{x} = x + \eta$$

where x is the original input and η is the generated perturbation. Once the adversarial example is created, it can be used to deceive the model:

$$f(x) \neq f(\tilde{x}) \quad \Rightarrow \quad \left(Y \neq \tilde{Y}\right)$$

Since Szegedy et al. [9] described the Box-constrained L-BFGS optimization problem as a technique to craft adversarial examples, new attacks have been developed over the years like Fast Gradient Method [10], DeepFool [13], Jacobian Saliency Maps [14], Carlini and Wagner [15], Pixel Attack [16], Projected Gradient Descent [11], HopSkipJump [12], and many others.

Adversarial attacks can be categorized in several ways, considering knowledge of the targeted model (*white-box* or *black-box*), misclassification precision (*targeted* or *non-targeted*), and input domain (*digital* or *physical*):

- **White-box**: assumes that the adversary has partial or complete knowledge of the targeted model including weights, activation functions, architecture and hyper-parameters. Commonly, these attacks are based on the model gradients [10], [11], [13]–[16].
- **Black-box**: in this case, the adversary only has access to the output of the targeted model and, usually, with certain querying limits. Generally, these attacks are evaluated based on the number of model requests [12], [17], [18].
- **Non-targeted**: a non-targeted attack generates adversarial examples without targeting a specific model output class, the main objective being that the predicted label does not correspond to that of the original input. Usually, these attacks are easier to perform due to the high dimensional space of the possible classes.
- **Targeted**: the aim of these attacks is to force the model to predict a specific output label that does not correspond to

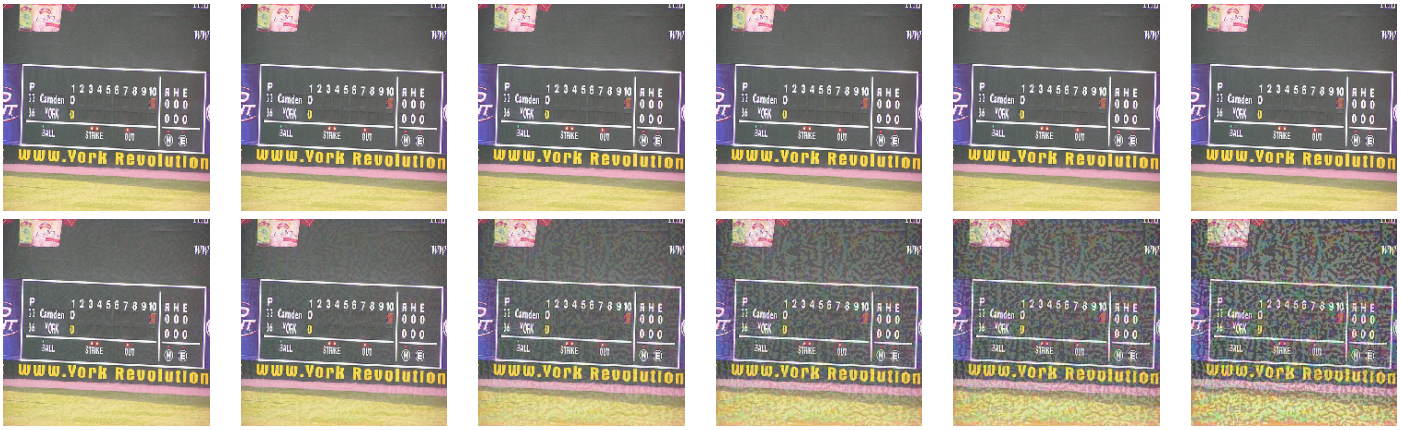


Fig. 1. Fast Gradient Sign Method attack with a progression of *epsilons*.

the original input, i.e. the original input image represents an apple and an attacker generates an adversarial image that forces the model to precisely predict a rifle and not simply cause a generic misclassification. Most of the mentioned attacks have targeted versions also.

- **Digital**: the result of these attacks remains in the digital domain. The adversarial images generated are sent directly to the model as digital input.
- **Physical**: these attacks use physical modifications to fool the target models, e.g. Sharif et al. [19] printed a pair of eyeglass frames to force misclassifications on a state-of-the-art face-recognition algorithm.

Black-box techniques pose, perhaps, a greater threat because they have been shown to be feasible in real world scenarios. Many of these techniques take advantage of transferable examples to perform the attacks. A good example would be an attack targeting an autonomous car guidance system in which the adversary has no information regarding the classification model but is capable of crafting an image that looks like a stop sign to human eyes while being identified as a speed limit sign by the classifier. Goodfellow [10] argued that the reason why transferability works is because the perturbations are aligned with the weight vector. However, Lui & Chen demonstrated that this hypothesis was not valid in the case of large datasets like Imagenet [20] and showed that "*the gradient directions of different models are orthogonal to each other*" [21].

III. Study description

Our study tries to shed more light on the transferability property between different prediction models. We have selected seven models with different architectures, randomly chosen one hundred images from the Imagenet dataset (all of them correctly classified by the selected models) and tested three different attacks, being two of them white-box and the remaining one black-box.

Among the architectures selected for the study, there are five types of families:

- **Residual Networks (ResNet)**: deep residual networks address the degradation problem (regarding training accuracy), that occurs when network depth is increased, through the addition of identity mappings (a new neural layer called Residual Block). Basically, these are connection layers that sum the output of previous layers, feeding the result into the following layers.
- **Very Deep Convolutional Networks (VGG)**: VGG networks use small convolution filters, 16-19 layers, small stride for the first convolution layer and other improvements in order to increase accuracy [5].
- **Inverted Residuals (MobileNetV2)**: specifically designed for mobile devices or those with limited computational capability, they reduce the number of operations and memory required [22].
- **EfficientNet**: Mingxing Tan [8] proposed a new technique to scale up neural networks using resolution, width and depth dimensions and scaling each of them with a constant ratio. Based on this work, new EfficienNet baseline models were created.
- **Densely Connected Conv. Networks (DenseNet)**: these networks connect each layer to every other layer so that the model ends up with $L(L+1)/2$ direct connections between them. The goal of this design is to reduce the number of parameters, alleviate the vanishing-gradient problem, encourage feature reuse, and strengthen feature propagation [23].

The architectures under study are VGG16, VGG19, EfficientB0, ResNet50, ResNet152V2, MobileNetV2, and DenseNet201 from the Keras[1] API. These are loaded with default Imagenet pre-trained weights with an average of 75% in Top-1 accuracy.

The chosen attacks are Fast Gradient Sign Method [10], Projected Gradient Descent [11], and HopSkipJump [12]. To carry out the attacks we have used the Adversarial Robustness Toolbox (ART)[2] tool written in python, that implements support for many types of attacks and defenses [24]. We briefly describe these attacks in the following sections.

[1]https://keras.io/api/applications/
[2]https://github.com/Trusted-AI/adversarial-robustness-toolbox

TABLE I
FAST GRADIENT SIGN METHOD ATTACK 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%

A. Fast Gradient Sign Method

The fast gradient sign method (FGSM) [10] is a single-step white-box attack that uses the gradients of the loss function with respect to the input image to determine the direction in which to modify the pixels:

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

where ϵ is the perturbation multiplier (intended to be small), θ the parameters of the model and J the loss function.

B. Projected Gradient Descent

Projected Gradient Descent (PGD) [11] is an improvement of the Basic Iterative Method (BIM) proposed in [25], with the only change of starting from a random point within the ϵ norm ball [26].

At the same time, BIM is an iterative extension of FGSM where, in each iteration, a small step size is applied, and the intermediate pixels are clipped to guarantee that they are in the ϵ-neighborhood 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. HopSkipJumpAttack

HopSkipJumpAttack [12] is a query-efficient decision-based (using the predicted labels of the targeted model) black-box attack. It is based on the Boundary Attack [27], in which the gradient direction, step-size and boundary search are calculated for each iteration via binary search.

IV. RESULTS AND DISCUSSION

The attacks implemented in the ART library accept multiple configuration parameters. In the case of FGSM, we have chosen an arbitrary scale of *eps* values (with the epsilon parameter we decide the magnitude of the perturbation of the final adversarial example) that include the values $(0.1, 0.2, 0.3, 0.5, 0.8, 1, 3, 5, 10, 15, 20, 30)$ and combined them with L_∞, L_1, L_2 *norms*, which represent the perturbation constraint type; the rest of the parameters were left as default.

The source images are pre-processed before the attack and in some cases we scaled the *eps* values accordingly to the input pixel range of the target network, e.g. DenseNet201 pixel range is $x/255$ and the *eps* values used to the attack were $(0.0003, 0.0007, 0.0011)$ for the first three values.

The effect of progressively increasing *eps* values on the original image can be seen in Fig. 1. Therefore, for each of the seven models, 100 images of each combination of *eps* and *norm* have been generated, making a total of 3600 images for each classifier; then, we have classified all generated images through all models and have measured the ratio of successful classifications.

A. Fast Gradient Sign Method

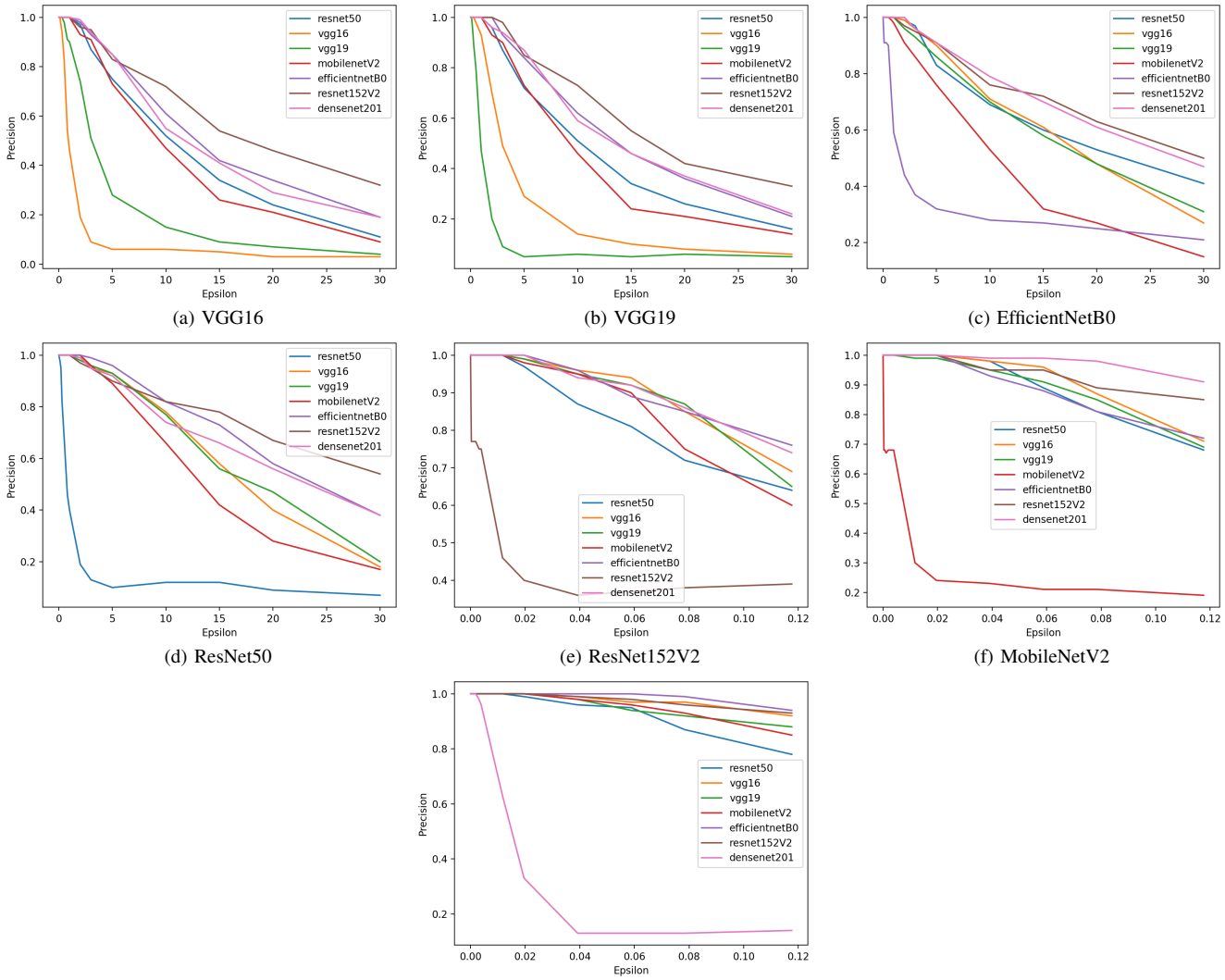
We have observed that all families are vulnerable to the FGSM attack but some more than others. For the VGG16 network, the attack starts to be effective with a very low *eps* of around 0.3, dropping the precision of correct classifications from 100% to approximately 50%, while the VGG19 network holds up to 0.5 *eps* and drops to 60% *eps* with VGG16's adversarial examples. The transferability between VGG models is high as we can see in Fig. 2a and 2b, when the value *eps* reaches 5 the precision of the networks drops to 0.3 approximately. This attack transfers to the other networks with a similar slope, being the MobileNetV2 network the most affected.

The EfficientNetB0 network is vulnerable to low values of *eps*. However, it stabilizes starting from a value of 5. The generated examples are transferred with similar success to VGG (see Fig. 2c), being MobileNetV2 the most affected, as before.

In the ResNet family, the ResNet50 network appears to be very sensible as well to low *eps* values and drops to 10% precision with a value of approximately 0.3 (see Fig. 2d). The generated images transfer well to the other networks, although for larger *eps* values (see Fig. 2e). The attack for ResNet152V2 succeeds with similar results as ResNet50. In this case, the transferability between these two networks is low, being possible that the different pixel range of both models is the reason for these results.

MobileNetV2 precision drops to 70% with very low values and to 25% with an *eps* of 0.02; however, the transferability is not as pronounced (see Fig. 2f). DenseNet201 precision drops to 30% with an *eps* of 0.02 and the transferability is similar to MobileNetV2 (see Fig. 2g).

The transferability for the FGSM attack is more significant in cases where the *eps* values have not been adapted to the input pixel range of the network. The VGG family,



(a) VGG16 (b) VGG19 (c) EfficientNetB0

(d) ResNet50 (e) ResNet152V2 (f) MobileNetV2

(g) DenseNet201

Fig. 2. Fast Gradient Sign Method attack transferability

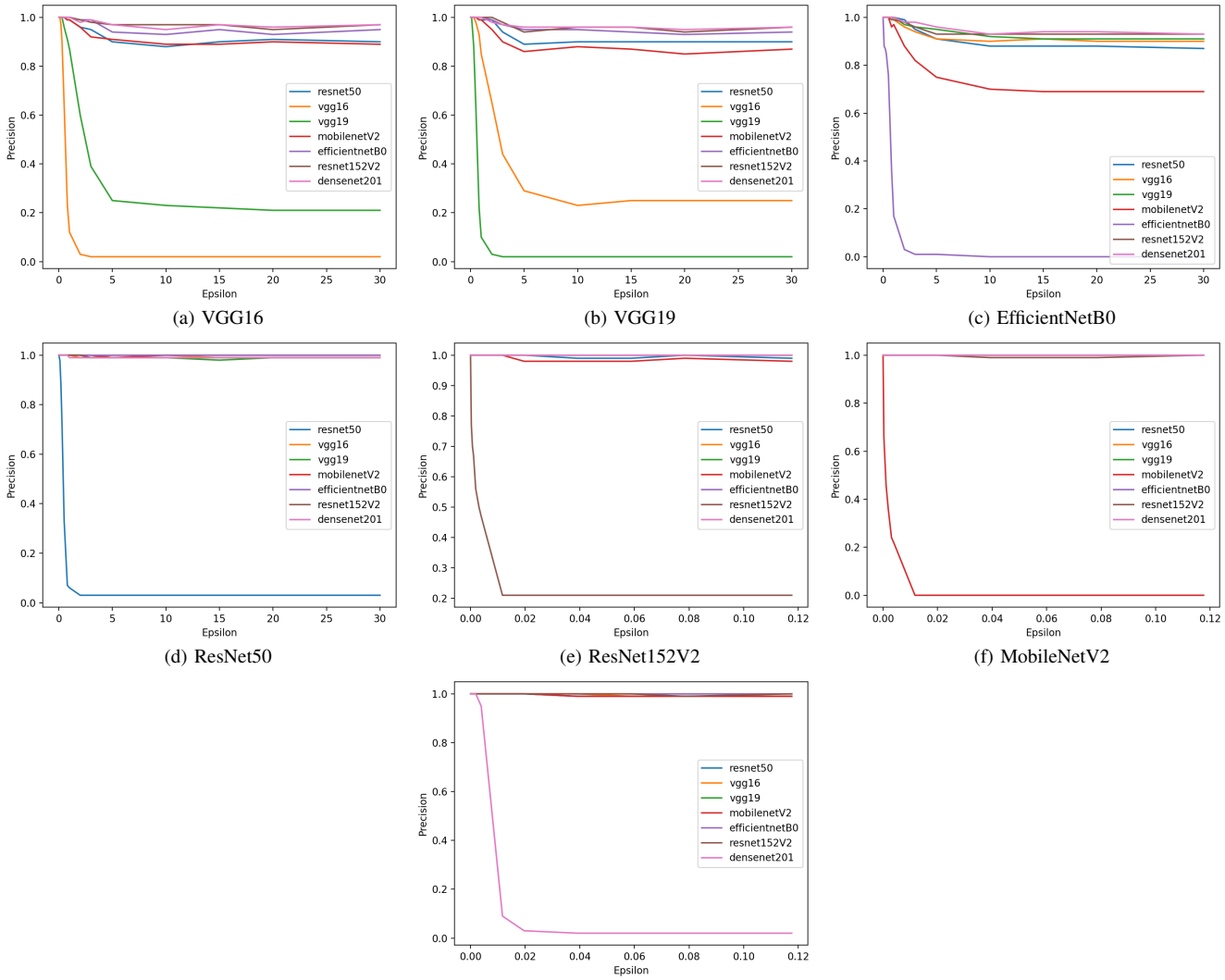
EfficientNetB0 and ResNet50 inputs are in $[0, 255]$ range, $x/255$ for DenseNet201 and $x/127.5$ for MobileNetV2 and ResNet152V2. We can hypothesize that the generated adversarial examples from networks with an input pixel range of $[0, 255]$ are more transferable due to the *eps* values being larger and, therefore, causing steeper gradients. The attacks performed with L_1, L_2 norms have not been successful, possibly because they represent more restricted boundaries.

Table I shows the accuracy percentages of the models for images generated by the FGSM attack with an *eps* of 10 for EfficientNetB0, VGG16/19 and ResNet50 networks. For DenseNet201, a scaled *eps* of 0.0392 and 0.0784 for ResNet152V2 and MobileNetV2 networks. As an example, in the first row we can see how the attack reduces the accuracy of VGG16 to 0.06% and how the adversary examples transfer to the other networks, being ResNet152V2 the least affected with a 0.72% of accuracy.

B. Projected Gradient Descent

For the PGD method we only perform the attack for the L_∞ norm with the same values of *eps* used in FGSM and 1200 iterations. In this case, we have observed that the transferability between the two networks of the VGG family is similar as with the FGSM attack, while the others networks have barely been affected, see Fig. 3a and 3b. For the EfficientNetB0 (see Fig. 3c) the attack succeeds with all 100 images at an *eps* of 0.5 and is only transferable to MobileNetV2, reducing its precision to 60%. In the case of the ResNet50, ResNet152V2, MobileNetV2, and DenseNet201 networks transferability is insignificant (see Fig. 3d, 3e, 3f, and 3g).

In general, the attack was successful in all networks but transferability was lower than in the case of the FGSM attack. This may be because the perturbations obtained with the PGD attack are more adjusted to the decision boundary of target network.



(a) VGG16 (b) VGG19 (c) EfficientNetB0

(d) ResNet50 (e) ResNet152V2 (f) MobileNetV2

(g) DenseNet201

Fig. 3. Projected Gradient Descent attack transferability

C. *HopSkipJumpAttack*

Finally, the HopSkipJump attack was performed using 30 maximum iterations and 20000 evaluations (see Fig. 4). The attack almost succeeds completely for the EfficientB0 and ResNet152V2 networks, decreasing precision to 0.01 and 0.10, respectively. For the MobileNetV2, VGG16 and VGG16 networks the results were worse than in the previous case, but still considerably effective with precisions of 0.28, 0.4, and 0.48, respectively. ResNet50 and DenseNet201 were the most resistant networks to the HopSkipJump attack with these parameters. Despite the effectiveness of the attack, transferability was negligible in all cases.

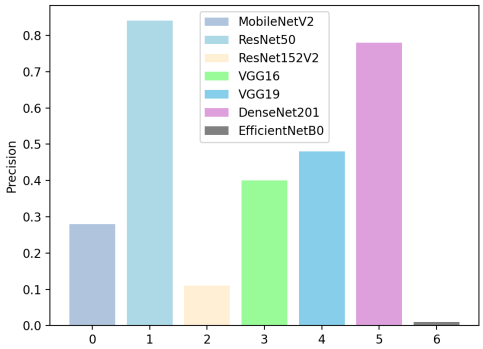


Fig. 4. HopSkipJump attack success

V. Conclusions

In this work, we show that even modern state-of-the-art networks with many layers and parameters are vulnerable to a simple one-step attack, and that the transferability property works well between similar architectures and different classification models in some cases. For the deeper networks, only the images created with high values of *eps* have been transferred between networks, but with the drawback that the images are easily detectable and very different to the originals.

The adversarial images generated with the Fast Gradient Sign Method and Projective Gradient Descent attacks for the VGG networks transfer well within the same family, but not in the case of the other networks. We observed that transferability depends both on the *eps* scale and the pixel input range of the networks: the *eps* values are higher for networks with a pixel range of $x/255$, and the perturbations are more obvious on the generated images.

Regarding the HopSkipJump attack, the transferability was practically non-existent. This black-box method is not reliant on the gradient, this could be a possible explanation for the poor transferability among networks in this case. Future research will involve more black-box attacks and further transferability studies.

Acknowledgment

Experiments were made possible by a generous hardware donation from NVIDIA. Research partially supported by the Spanish Government under project grant RTI2018-097263-B-I00 (ACTIS).

References

[1] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Edeep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process*, 2012.

[2] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with task learning. in proceedings of the 25th international conference on machine learning," *ACM*, 2008.

[3] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," *ACM*, 2014.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, 1989.

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

[6] C. Szegedy, W. Liu, Y. Ji, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv:1409.4842*, 2014.

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

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

[9] C. Szegedy, W. Zaremba, I. Sutskever, and J. Bruna, "Intriguing properties of neural networks," *arXiv*, 2013.

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

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv:1706.06083*, 2017.

[12] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," *arXiv:1904.02144*, 2019.

[13] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *arXiv:1511.04599*, 2015.

[14] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *arXiv:1511.07528*, 2015.

[15] D. W. N. Carlini, "Towards evaluating the robustness of neural network," *arXiv:1608.04644*, 2016.

[16] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," *arXiv:1710.08864*, 2017.

[17] C. Guo, J. Gardner, Y. You, A. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," *arXiv:1905.07121*, 2019.

[18] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," *arXiv:1912.00049*, 2019.

[19] 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," *CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications*, 2016.

[20] 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," *arXiv:1409.0575*, 2014.

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

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) arXiv:1801.04381*, 2018.

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

[24] M. Nicolae, M. Sinn, M. 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," *arXiv:1807.01069*, 2018.

[25] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv:1607.02533*, 2016.

[26] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe, "Adversarial attacks and defences competition," *arXiv:1804.00097*, 2018.

[27] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv:1712.04248 [stat.ML]*, 2017.