ON THE TRANSFERABILITY OF ADVERSARIAL EXAMPLES AGAINST CNN-BASED IMAGE FORENSICS

*M.Barni, K. Kallas, E. Nowroozi, B.Tondi

Department of Information Engineering and Mathematics, University of Siena, Italy.

ABSTRACT

Recent studies have shown that Convolutional Neural Networks (CNN) are relatively easy to attack through the generation of so called adversarial examples. Such vulnerability also affects CNN-based image forensic tools. Research in deep learning has shown that adversarial examples exhibit a certain degree of transferability, i.e., they maintain part of their effectiveness even against CNN models other than the one targeted by the attack. This is a very strong property undermining the usability of CNN's in security-oriented applications. In this paper, we investigate if attack transferability also holds in image forensics applications. With specific reference to the case of manipulation detection, we analyse the results of several experiments considering different sources of mismatch between the CNN used to build the adversarial examples and the one adopted by the forensic analyst. The analysis ranges from cases in which the mismatch involves only the training dataset, to cases in which the attacker and the forensic analyst adopt different architectures. The results of our experiments show that, in the majority of the cases, the attacks are not transferable, thus easing the design of proper countermeasures at least when the attacker does not have a perfect knowledge of the target detector.

Index Terms— Adversarial multimedia forensics, adversarial machine learning, adversarial examples, attack transferability, image forensics.

1. INTRODUCTION

Convolutional Neural Networks (CNN) are increasingly used in image forensic applications due to their superior accuracy in detecting a wide number of image manipulations, including multiple JPEG compression [1, 2], median filtering [3], resizing [4], contrast manipulation [5]. Good performance of CNNs have also been reported for image source attribution, i.e., to identify the model of the camera which acquired a certain image [6–8]. Despite the good performance they achieve, the use of CNNs in security-oriented applications, like image forensics, is hindered by the easiness with which adversarial examples can be built [9–11]. As a matter of fact, an attacker who has access to the internal details of the CNN used for a certain image recognition task can easily build an attacked image which is visually indistiguishable from the original one, but is misclassified by the CNN. Such a problem is currently the subject of an intense research activity, yet no satisfactory solution has been found yet (see [12] for a recent survey on this topic). The problem is worsened by the observation that adversarial attacks are often transferrable from the target network to other networks designed for the same task [13]. This means that even in a Limited Knowledge (LK) scenario, wherein the attacker has only partial information about the to-be-attacked network, he can attack a surrogate network mimicking the target one and the attack will be effective also on the target network with good probability. Such a property opens the way towards very powerful attacks that can be used in real applications wherein the attacker does not have full access to the attacked system [13].

Following some recent researches, showing that CNN-based image forensics tools are also endangered by the existence of adversarial examples [14–16], the goal of this paper is to investigate if and to which extent the transferability of adversarial examples holds in image forensics applications. The answer to this question is of primary importance, since attack transferability would greatly complicate the development of anti-counter-forensics measures. In fact, even denying to the attacker a full access to the forensic tools would not guarantee that the forger can not mislead the forensic analysis. To the best of our knowledge, the only previous works partially addressing this problem are [15] and [17]. In particular, [15] reports some tests aiming at assessing the transferability of adversarial examples targeting various CNN-based camera model identification systems. According to [15], in a camera model identification scenario, attacks are only partially transferable, since the transferred attack succeed in no more than 40% of the cases (often much less). In [17], the transferability between different network models is assessed by considering a case of attack carried out by the FGSM [18]. The analyses in [15] and [17] are very preliminary, hence calling for new tests addressing different sources of mismatch between the attacked network and the targeted one, different forensics scenarios, and the impact that the attack strength has on the transferability of the attacks. In this paper, we make some steps in this direction. We consider two forensic tasks boiling down to a binary detection problem, namely, median filtering and image resizing detection. We analyse separately the effect of training data mismatch and network architecture mismatch on the transferability of the attacks, by considering two different attack methodologies, namely the Jacobian-based Saliency Map Attack (JSMA) [11] and the Iterative Fast Gradient Sign Method (I-FGSM) [19] (i.e. the refined iterative version of the original FGSM attack [18]), and evaluate the transferability of the attacks also in the presence of double-to-integer rounding, which is a necessary step to bring back the attacked image into the integer domain. As we will see, our experiments cast serious doubts on the transferability of adversarial attacks in image forensic applications, thus opening the way to the development of proper countermeasures.

The rest of this paper is organised as follows. In Sect. 2, we describe the methodology used for our experiments, including: i) the description of the algorithms used to generate the adversarial

This work has been partially supported by a research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

* The list of authors is provided in alphabetic order.

examples; ii) the description of the CNN architectures targeted by the attacks; iii) the description of the experimental campaign, iv) the datasets used for training and testing the CNNs. The results of the experiments are presented in Sect. 3, together with a discussion of our main findings. Finally, in Sect. 4, we present a roadmap for future research.

2. METHODOLOGY

In order to evaluate the factors that influence the transferability of adversarial attacks against CNN-based detection of image processing operators, we considered two different kinds of attacks, two detection tasks solved by relying on two different networks, and three sources of mismatch between the network used to create the adversarial attack (hereafter referred to as Source Network - SN) and the one the attack should be transferred to (hereafter referred to as Target Network - TN). In particular we considered the cases of two different networks trained on the same dataset and the case of a single network trained on different datasets. With reference to the terminology established in [13], we refer to the first type of transferability as *cross-model transferability* and to the second as *cross-training transferability*. We also considered the case of two different networks trained on different datasets (*cross-model-and-training transferability*). The combination of the above factors resulted in an extensive campaign of experiments whose results will be discussed in Sect. 3.

2.1. Attacks

In our experiments, the adversarial examples were built by relying on the FGSM algorithm, originally proposed in [18], and the JSMA [11]. The Foolbox toolbox [20] was used to implement them.

For the FGSM, as we said, we considered the refined iterative version (I-FGSM) described in [19]. In its original implementation, FGSM obtains an adversarial perturbation in a computationally efficient way by computing the gradient of the output with respect to the input image and considering its sign multiplied by a strength factor. The I-FGSM algorithm is a multi-step variant of FGSM; for a given attack strength ε, the algorithm is applied iteratively until an adversarial image can be produced (that is, an image which is misclassified by the network), for a maximum number of steps S. Several values of ε are considered, i.e. $\varepsilon \in E$; the value which minimizes the distortion of the final attacked image with respect to the original one is eventually selected as best strength, for the given maximum number of iterations of the algorithm S. In the foolbox implementation of I-FGSM, ε corresponds to the normalized strength factor. Then, at each iteration $i+1$, the image is updated as follows: $X_{i+1} = X_i + \varepsilon(\max(X_i) - \min(X_i)) \cdot \text{sign}(\nabla_X J_\theta(X_i, y))$, where $J_\theta(X, y)$ is the cross-entropy cost function with parameters θ, and y is the ground truth label of X.

The JSMA algorithm, proposed by Papernot et al. [11], consists of a greedy iterative procedure which relies on forward propagation to compute, at each iteration, a saliency map, indicating the pixels that contribute most to the classification. The pixels are then modified based on this map by a relative amount θ, $\theta < 1$ (θ is relative to the range of the values of the image, the pixel modification being $\theta \cdot (\max(X_i) - \min(X_i))$). A constrain is put on the maximum number of times T the same pixel can be modified. We do not limit the maximum number of iterations. Then, the procedure ends when the attacker succeeds or the pixels are modified by a too large amount (i.e., the number of modifications reaches the maximum prescribed number for all pixels) [11].

Both the I-FGSM and the JSMA algorithms produce a real-valued attacked image. While in some cases we can assume that the attacked image is used as is, in most applications image pixels must be mapped back into the integer domain before being fed to the CNN. This may result in a loss of effectiveness of the attack, since some of the subtle changes introduced by the attack are deleted when pixels are rounded (or truncated) to integer values.

2.2. Datasets

In order to evaluate the transferability of the attacks when the SN and the TN are trained on different datasets, we considered the RAISE (R) [21] dataset and the VISION (V) dataset [22].

For our experiments, about 2000 uncompressed, camera-native, images (.tiff) were taken from the RAISE dataset, with size 4288×2848. These images are camera-native images coming from three different cameras. The same number of images were taken from the VISION dataset. This dataset consists of native images acquired by smartphones/tablets belonging to several brands. To get similar resolution images for the two datasets, we only selected the devices for which the resolution was not very different from that of the images from RAISE. Specifically, the sizes of the images we considered ranges from a minimum of 2336×4160 up to 3480×4640. The images from the VISION dataset are in JPEG format. In order to reduce the possible impact of compression artefact, we selected images only from the high-quality devices, for which the JPEG Quality Factor is larger than 97. The images from both R and V datasets were split into training (and validation) set and test set, and then processed to produce the images for the manipulated class, namely, median and resizing. For all our tests we considered one-channel images, then all the images from R and V were converted to gray-scale.

2.3. Networks

In our experiments, we considered two different detection tasks, namely the detection of image resizing (downsampling, by a 0.8 factor) and median filtering (by a 5×5 window). To cope with them, we built several networks generally indicated as $N^{\text{tr}}_{\text{ar}}(\text{task})$, where "ar" indicates the architecture of the network, "tr" $\in \{\text{R, V}\}$ the dataset used for training and "task" $\in \{\text{med, res}\}$ the detection task ("med" indicating median filtering and "res" resizing).

With regard to the architectures, we considered the network in [23] (recently extended in [4]), hereafter referred to as BSnet ("ar" = BS), and the one in [5], hereafter denoted as GCnet ("ar" = GC). BSnet, originally proposed for image manipulation detection and classification, consists of 3 convolutional layers, 3 max-pooling layers and 3 fully-connected layers. Residual-based features are extracted by constraining the filters of the first layer (with 5×5 receptive field), by enforcing a high-pass nature of the filters (see [4] for more details). For the second and third convolutional layers the filter size is set to 7×7 and 5×5 respectively, and the stride is set to 2. For the max-pooling, a kernel size 3×3 is used with stride 2. GCnet was originally proposed to detect generic contrast adjustment operators. With respect to BSnet, GCnet is significantly deeper, consisting of 9 convolutional layers. The network has only 2 max-pooling layers and one fully-connected layer. A kernel size of 3×3 and stride 1 was used for all the convolutional layers. Max-pooling is applied with kernel size 2×2 and stride 2. The number of parameters is then reduced by halving the number of feature maps in the final convolutional layer, and considering just one fully-connected layer.

Then, we built 6 networks, indexed as: $N^{\text{tr}}_{\text{BS}}(\text{task})$, "tr" $\in \{\text{R, V}\}$, "task" $\in \{\text{med, res}\}$, and $N^{\text{R}}_{\text{GC}}(\text{task})$, "task" $\in \{\text{med, res}\}$.

2.4. Experiments

The experimental campaign was designed in such a way to highlight attack transferability in a wide variety of settings. Experiments have been split into three categories according to the type of mismatch between the SN and the TN. We started studying cross-training transferability, according to which SN and TN share the same architecture, but are trained on different datasets. Then we passed to analyse cross-model transferability, in which different network architectures are trained on the same dataset. Eventually, we passed to cross-model-and-training transferability according to which the SN and the TN share neither the architecture nor the training data. All the tests have been repeated for both resizing and median filtering detection. For sake of simplicity we did not consider all possible combinations, however, the amount of experiments we carried out is sufficient to draw a number of significant conclusions. In particular, the experiments for the cross-training transferability are carried out by considering only BSnet as the SN, trained on R, i.e., $SN = N_{\text{BS}}^{\text{R}}$ (in this case $TN = N_{\text{BS}}^{\text{V}}$), and on V, i.e., $SN = N_{\text{BS}}^{\text{V}}$ ($TN = N_{\text{BS}}^{\text{R}}$). For the experiments on the cross-model transferability, BSnet is taken as SN and GCnet as TN, both trained on R, i.e., $SN = N_{\text{BS}}^{\text{R}}$ and $TN = N_{\text{GC}}^{\text{R}}$. Finally, for the cross-model-and-training case, we set $SN = N_{\text{BS}}^{\text{V}}$ and $TN = N_{\text{GC}}^{\text{R}}$.

With regard to the attacks, for the I-FGSM attack, the number of steps S is fixed to 10 (default). The best strength is searched in the range $E = [0 : \varepsilon_s : 0.1]$, where ε_s is the search step size, which then also corresponds to the minimum normalized strength considered. Setting a larger ε_s generally corresponds to consider a stronger attack. In our experiments, we considered $\varepsilon_s = 0.001$ and 0.01, for which the average PSNR remains above 40 dB. For the JSMA, T is set to 7. The relative modification per pixel θ is set to 0.01 and 0.1, the second case corresponding to a stronger attack. We did not consider θ values larger than 0.1, since above this value the maximum pixel distortion introduced by the attack starts becoming too large (> 70). Eventually, we repeated all the experiments by rounding the output of the attack to integer values.

3. RESULTS AND DISCUSSION

In this section we discuss the results of the experiments we have carried out. For sake of brevity, we will focus on the floating point version of the attacks, being this case more favorable to the attacker, and we will briefly touch upon the integer-valued case at the end.

To build our models N_{BS}^{R} and N_{BS}^{V} (for both detection tasks), we considered 200.000 patches for training (and validation) and 10000 for testing, per class. In order to use all the images in the datasets R and V, a maximum number of 100 patches is selected (randomly) for each image. A number of 30 training epochs was considered (as in [23]). For the deeper models N_{GC}^{R} for both the "med" and "res" task, we used 10^6 patches for training, 10^5 for validation, and $5 * 10^4$ for testing. To reach these numbers, all the image patches were selected from all the images. By following [5], the number of training epochs is set to 3. The input patch size is set to 128×128. For training both BSnet and GCnet, the Adam solver is used with learning rate 10^{-4} and momentum 0.99. The batch size for training is set to 32 images, the test batch size to 100. The accuracies achieved by the BSnet in absence of attacks in the various cases are: 98.1% for $N_{\text{BS}}^{\text{R}}(\text{med})$, 99.5% for $N_{\text{BS}}^{\text{V}}(\text{med})$, 97.5% for $N_{\text{BS}}^{\text{R}}(\text{res})$, 96.6% for $N_{\text{BS}}^{\text{V}}(\text{res})$. With regard to GCnet, it got the following accuracies: 98.4% for $N_{\text{GC}}^{\text{R}}(\text{med})$ and 98.5% for $N_{\text{GC}}^{\text{R}}(\text{res})$.

In the next section, we discuss the performance of the models in the presence of attacks, in the matched and mismatched cases. In counter-forensic applications, it is reasonable to assume that the attack is only in one direction, since the attacker wants to pass off a manipulated image as an original one, i.e. aims at causing a false negative error. Therefore, in our experiments, we only attack images from the manipulated class. In all the cases, the performance of the attack are assessed on 500 patches, obtained by attacking a subset of the patches from the corresponding test set in each case. Obviously, we attack only images for which the classification of the network is correct. An attack is declared successful when it is able to switch the network decision, i.e., when the manipulated image is labeled as original after the attack (the output soft score for the original class becomes larger than 0.5).

3.1. Cross-training transferability

As detailed in Sect. 2.4, these experiments were carried out by considering only the BS architecture. The results we got are reported in Table 1. For each case, the table reports the accuracy on the manipulated class for both SN and TN without attacks. For each attack type, the PSNR, L_1 distortion and maximum absolute distortion are reported, averaged on all the images successfully attacked in the matched case (i.e., successfully fooling SN). The attack success rate with respect to SN and TN is reported in the last two columns. As we can see, the attacks are generally non-transferable and the images attacked using SN are not able to deceive the TN. More specifically, with the FGSM attack, the adversarial examples can be transferred in a significant number of cases only when the larger strength is considered ($\varepsilon_s = 0.01$) and the SN corresponds to $N_{\text{BS}}^{\text{R}}(\text{res})$ and $N_{\text{BS}}^{\text{R}}(\text{med})$ (attack success rate 0.692 and 0.845 respectively) and to $N_{\text{BS}}^{\text{V}}(\text{res})$ (attack success rate 0.941). For the JSMA case, the attack can be transferred only when SN is $N_{\text{BS}}^{\text{R}}(\text{res})$ and strong attack with $\theta = 0.1$ is considered, with success rate 0.782. Furthermore, we observe that the JSMA is never transferable when the VISION dataset is used to train the SN. It is also interesting to observe that, for a given detection task, the transferability is not symmetric with respect to the datasets used for training. This suggests that, in forensic applications, the features learned by the network may also be affected in some way and up to some extent by the underlying dataset. This point deserves further investigation as a future work.

3.2. Cross-model transferability

In this case, the experiments were carried out by considering only the R dataset and using the BS architecture for the SN. The results we have got are reported in Table 2. The experiments show the lack of transferability with respect to a mismatch in the network model. The only exception is for the "med" case, in which case the stronger attack (with $\varepsilon_s = 0.01$) is transferable 82.5% of the times. However, it is worth stressing that, when the FGSM is applied with such a strength, although the PSNR is not very low (40.03 dB), the average L_1 distortion is around 2.5 (a similar value is attained by the maximum absolute distortion). With such values of L_1, the visual quality of the FGSM attacked images is impaired and peculiar visual artifacts appears, especially in relatively uniform image patches.

The fact that the lack of transferability is even stronger in the "res" case than in the "med" case can be probably justified by the ease of the median filtering detection task (even because the median filtering is performed with a rather large window size), compared to the resize. Therefore, we might expect that in the case of "med" similar peculiar features are learned by the shallow and deeper network, hence facilitating the transferability of the attacks.

3.3. Cross-model-and-training transferability

In this case, the experiments were carried out by considering the BS architecture trained on the V dataset as the SN, and the GC ar-

Table 1. Experimental results for Cross Training. Transferable attacks are highlighted in bold.

SN	TN	Accuracy	Attack type	PSNR	L1 dist	max. dist	Attk succ rate (SN)	Attk succ rate (TN)
N^{R}_{BS}(res)	N^{V}_{BS}(res)	SN= 97.60%, TN= 96.00%	I-FGSM, $\varepsilon_s = 0.01$	40.02	2.53	2.55	**1.000**	**0.692**
N^{R}_{BS}(res)	N^{V}_{BS}(res)	SN=97.60%, TN= 96.00%	I-FGSM, $\varepsilon_s = 0.001$	58.46	0.26	0.27	1.000	0.0491
N^{R}_{BS}(res)	N^{V}_{BS}(res)	SN= 97.60%, TN= 96.00%	JSMA, $\theta = 0.1$	46.04	0.07	58.32	**1.000**	**0.782**
N^{R}_{BS}(res)	N^{V}_{BS}(res)	SN= 97.60%, TN= 96.00%	JSMA, $\theta = 0.01$	54.99	0.04	15.09	0.991	0.115
N^{V}_{BS}(res)	N^{R}_{BS}(res)	SN= 97.80%, TN= 99.60%	I-FGSM, $\varepsilon_s = 0.01$	40.03	2.53	2.55	1.000	0.002
N^{V}_{BS}(res)	N^{R}_{BS}(res)	SN= 97.80%, TN= 99.60%	I-FGSM, $\varepsilon_s = 0.001$	59.64	0.26	0.27	1.000	0.000
N^{V}_{BS}(res)	N^{R}_{BS}(res)	SN= 97.80%, TN= 99.60%	JSMA, $\theta = 0.1$	50.55	0.01	69.42	0.989	0.000
N^{V}_{BS}(res)	N^{R}_{BS}(res)	SN= 97.80%, TN= 99.60%	JSMA, $\theta = 0.01$	57.78	0.01	17.06	0.979	0.000
N^{R}_{BS}(med)	N^{V}_{BS}(med)	SN= 98.20%, TN= 100%	I-FGSM, $\varepsilon_s = 0.01$	40.03	2.53	2.55	**1.000**	**0.845**
N^{R}_{BS}(med)	N^{V}_{BS}(med)	SN= 98.20%, TN= 100%	I-FGSM, $\varepsilon_s = 0.001$	59.67	0.26	0.27	1.000	0.045
N^{R}_{BS}(med)	N^{V}_{BS}(med)	SN= 98.20%, TN= 100%	JSMA, $\theta = 0.1$	49.64	0.03	38.11	1.000	0.012
N^{R}_{BS}(med)	N^{V}_{BS}(med)	SN= 98.20%, TN= 100%	JSMA, $\theta = 0.01$	58.47	0.02	14.05	0.984	0.002
N^{V}_{BS}(med)	N^{R}_{BS}(med)	SN= 100%, TN= 99.20%	I-FGSM, $\varepsilon_s = 0.01$	40.04	2.53	2.55	**1.000**	**0.941**
N^{V}_{BS}(med)	N^{R}_{BS}(med)	SN= 100%, TN= 99.20%	I-FGSM, $\varepsilon_s = 0.001$	59.94	0.25	0.25	1.000	0.077
N^{V}_{BS}(med)	N^{R}_{BS}(med)	SN= 100%, TN= 99.20%	JSMA, $\theta = 0.1$	49.55	0.03	32.09	1.000	0.010
N^{V}_{BS}(med)	N^{R}_{BS}(med)	SN= 100%, TN= 99.20%	JSMA, $\theta = 0.01$	58.13	0.01	14.08	0.988	0.008

Table 2. Experimental results for Cross Model. Transferable attacks are highlighted in bold.

SN	TN	Accuracy	Attack type	PSNR	L1 dist	max. dist	Attk succ rate (SN)	Attk succ rate (TN)
N^{R}_{BS}(res)	N^{R}_{GC}(res)	SN= 97.60%, TN= 98.20%	I-FGSM, $\varepsilon_s = 0.01$	40.02	2.53	2.55	1.000	0.002
N^{R}_{BS}(res)	N^{R}_{GC}(res)	SN= 97.60%, TN= 98.20%	I-FGSM, $\varepsilon_s = 0.001$	58.48	0.31	0.33	1.000	0.002
N^{R}_{BS}(res)	N^{R}_{GC}(res)	SN= 97.60%, TN= 98.20%	JSMA, $\theta = 0.1$	46.09	0.07	57.88	1.000	0.016
N^{R}_{BS}(res)	N^{R}_{GC}(res)	SN= 97.60%, TN= 98.20%	JSMA, $\theta = 0.01$	54.98	0.04	15.14	0.992	0.006
N^{R}_{BS}(med)	N^{R}_{GC}(med)	SN= 98.20%, TN= 100%	I-FGSM, $\varepsilon_s = 0.01$	40.03	2.53	2.55	**1.000**	**0.825**
N^{R}_{BS}(med)	N^{R}_{GC}(med)	SN= 98.20%, TN= 100%	I-FGSM, $\varepsilon_s = 0.001$	59.67	0.26	0.27	1.000	0.181
N^{R}_{BS}(med)	N^{R}_{GC}(med)	SN= 98.20%, TN= 100%	JSMA, $\theta = 0.1$	49.64	0.03	38.11	1.000	0.010
N^{R}_{BS}(med)	N^{R}_{GC}(med)	SN= 98.20%, TN= 100%	JSMA, $\theta = 0.01$	58.47	0.02	14.05	0.984	0.016

Table 3. Experimental results for Cross Training and Model. Transferable attacks are highlighted in bold.

SN	TN	Accuracy	Attack type	PSNR	L1 dist	max. dist	Attk succ rate (SN)	Attk succ rate (TN)
N^{V}_{BS}(res)	N^{R}_{GC}(res)	SN= 99.20%, TN= 99.60%	I-FGSM, $\varepsilon_s = 0.01$	40.03	2.53	2.55	1.000	0.004
N^{V}_{BS}(res)	N^{R}_{GC}(res)	SN= 99.20%, TN= 99.60%	I-FGSM, $\varepsilon_s = 0.001$	59.57	0.27	0.27	1.000	0.002
N^{V}_{BS}(res)	N^{R}_{GC}(res)	SN= 99.20%, TN= 99.60%	JSMA, $\theta = 0.1$	50.20	0.02	70.87	1.000	0.000
N^{V}_{BS}(res)	N^{R}_{GC}(res)	SN= 99.20%, TN= 99.60%	JSMA, $\theta = 0.01$	57.40	0.01	17.16	0.992	0.000
N^{V}_{BS}(med)	N^{R}_{GC}(med)	SN= 100%, TN= 100%	I-FGSM, $\varepsilon_s = 0.01$	40.04	2.53	2.55	**1.000**	**0.796**
N^{V}_{BS}(med)	N^{R}_{GC}(med)	SN= 100%, TN= 100%	I-FGSM, $\varepsilon_s = 0.001$	59.91	0.25	0.26	1.000	0.008
N^{V}_{BS}(med)	N^{R}_{GC}(med)	SN= 100%, TN= 100%	JSMA, $\theta = 0.1$	49.56	0.03	31.83	1.000	0.008
N^{V}_{BS}(med)	N^{R}_{GC}(med)	SN= 100%, TN= 100%	JSMA, $\theta = 0.01$	58.06	0.01	14.18	0.990	0.012

chitecture trained on the R dataset as the TN. Similar results can be obtained by combining architecture and dataset in the other way round. The results we have got are reported in Table 3. Quite expectedly, the table shows that the transferability of the attacks in this case decreases further and the attack success rate is below 0.01 in all the cases but for the case of FGSM with $\varepsilon_s = 0.01$, for which a success rate of 0.796 can still be achieved.

As a general behavior, according to our tests, for all the three types of mismatch considered, attacks obtained by JSMA are less transferable than those produced by FGSM. A possible motivation can be the following: since very few pixels are modified by JSMA, it tends to overfit more the attacked model. Related to this, with JSMA, the average output scores returned by SN on the successfully attacked samples are very close to 0.5 (in the range $[0.5, 0.6]$), while with FGSM they are always much larger than 0.5 (often > 0.9). Lastly, we repeated all the experiments by rounding the pixel values of the attacked images to integers. According to the results we have got, integer rounding does not have a big impact on the transferability of the attacks. Rather it influences the effectiveness of the attack on the SN itself, as already reported in several studies, e.g. [15, 24].

4. CONCLUDING REMARKS

We investigated the transferability of adversarial examples in an image forensics scenario. By focusing on two manipulation detection tasks, we run tests by considering two well known attack methodologies and several sources of mismatch. Our tests show that adversarial examples are generally non-transferable, in contrast to what happens in typical pattern recognition applications. This states an important result, since the lack of transferability can be exploited by the forensic analyst to make the attack more difficult. For instance, a LK scenario can be enforced in some way to combat adversarial examples, as done with the approaches based on standard ML. Even if our results clearly show that adversarial examples can not be easily transferred from one network to another, further tests are needed before we can draw some final conclusions. First of all, more detection tasks should be considered, together with different sources of mismatch between the SN and the TN. As an example, we may wonder if a mismatch in the training procedure is enough to prevent transferability. Also, the reason why image-forensic networks are less prone to attack transfers should be understood. On the attacker's hand, further research is needed to understand if and how the transferability can be improved by increasing the attack strength to enter more inside the region.

5. REFERENCES

[1] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security*, vol. 2016, no. 1, 2016.

[2] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro, "Aligned and non-aligned double JPEG detection using convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 153–163, 2017.

[3] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, Nov 2015.

[4] B. Bayar and M. Stamm, "Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, 2018.

[5] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "CNN-based detection of generic contrast adjustment with JPEG post-processing," in *ICIP 2018, IEEE International Conference on Image Processing*, Athens, Greece, 2018.

[6] L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E.J. Delp, and S. Tubaro, "First steps toward camera identification with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 259–263, 2017.

[7] D. Freire-Obregon, F. Narducci, S. Barra, and M. Castrillon-Santana, "Deep learning for source camera identification on mobile devices," *Pattern Recognit. Lett.*, 2018.

[8] B. Bayar and M. C. Stamm, "Towards open set camera model identification using a deep learning framework," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 2007–2011.

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

[10] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *arXiv preprint arXiv:1608.04644, 2016*.

[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, March 2016, pp. 372–387.

[12] N. Akhtar and M. Ajmal, "Threat of adversarial attacks on deep learning in computer vision: a survey," *IEEE Access*, vol. 2018, no. 6, pp. 14410–14430, 2018.

[13] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[14] D. Guera, Y. Wang, L. Bondi, P. Bestagini, S. Tubaro, and E. J. Delp, "A counter-forensic method for CNN-based camera model identification," in *IEEE Computer Vision and Pattern Recognition Workshops,*, July 2017, pp. 1840–1847.

[15] F. Marra, D. Gragnaniello, and L. Verdoliva, "On the vulnerability of deep learning to adversarial attacks for camera model identification," *Signal Processing: Image Communication*, vol. 65, pp. 240–248, July, 2018.

[16] Mauro Barni, Matthew C Stamm, and Benedetta Tondi, "Adversarial multimedia forensics: Overview and challenges ahead," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 962–966.

[17] Diego Gragnaniello, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva, "Analysis of adversarial attacks against cnn-based image forgery detectors," *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 967–971, 2018.

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

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

[20] J. Rauber, W. Brendel, and M. Bethge, "Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models," *arXiv preprint arXiv:1707.04131*, 2017.

[21] DT. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*, New York, NY, USA, 2015, MMSys '15, pp. 219–224, ACM.

[22] D. Shullani, M. Fontani, M. Iuliani, O.A. Shaya, and A. Piva, "VISION: a video and image dataset for source identification," *EURASIP Journal on Information Security*, pp. 1–16, 2017.

[23] B. Bayar and M. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *ACM Workshop on Info. Hiding & Multimedia Security*, 2016, pp. 5–10.

[24] B. Tondi, "Pixel-domain adversarial examples against CNN-based manipulation detectors," *Electronics Letters*, 2018.