ONE FOR MANY: AN INSTAGRAM INSPIRED BLACK-BOX ADVERSARIAL ATTACK

Anonymous authors
Paper under double-blind review

ABSTRACT

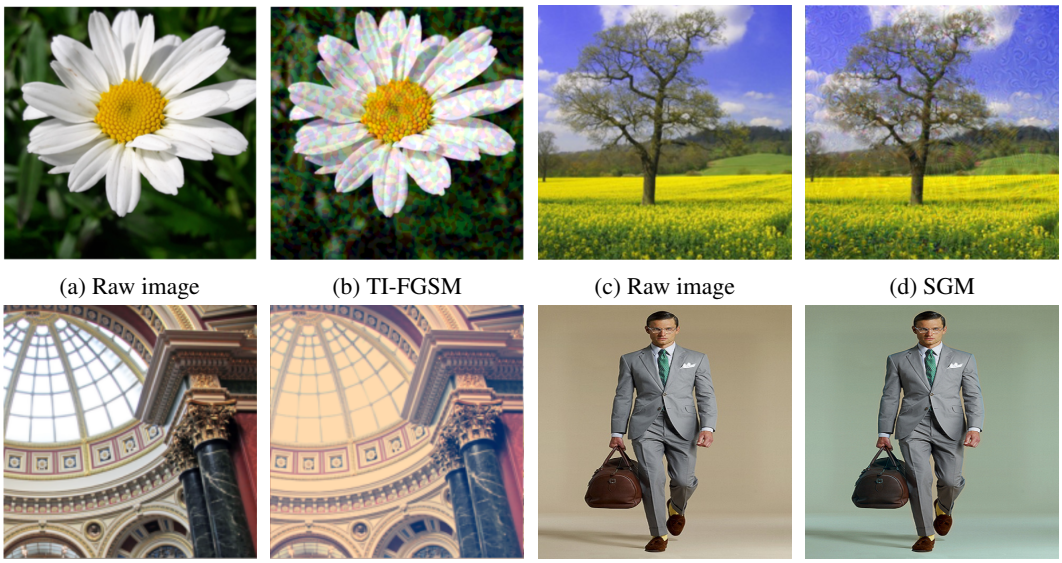
It is well known that deep learning models are susceptible to adversarial attacks. To produce more robust and effective attacks, we propose a nested evolutionary algorithm able to produce multi-network (decision-based) black-box adversarial attacks based on Instagram inspired image filters. Due to the multi-network training, the system reaches a high transferability rate of attacks and, due to the composition of image filters, it is able to bypass standard detection mechanisms. Moreover, this kind of attack is semantically robust: our filter composition cannot be distinguished from any other filter composition used extensively every day to enhance images; this raises new security issues and challenges for real-world systems. Experimental results demonstrate that the method is also effective against ensemble-adversarially trained models and it has a low cost in terms of queries to the victim model.

1 INTRODUCTION

It is well known that deep learning models are susceptible to adversarial attacks and many recent researches in the field have been devoted to produce ever more reliable and effective attacks. Attack reliability is strictly connected to its applicability in real-world scenarios and to its ability to bypass potential defense mechanisms; this is why the hard-label black-box setting (also called decision-based attack) and the transferability property of white box attacks gained increasing attention. Several techniques have been proposed to increase the transferability of both black-box and white-box attacks (Cheng et al. (2020); Wu et al. (2020); Brendel et al. (2018) , among the most popular). One of the most commonly adopted technique is to craft the attacks by using an ensemble of multiple models as proposed by Liu et al. (2017). Moreover, these studies also led to the discovery of surprising properties of Deep Neural Networks exposing important weakness about network architectures, as is the case of skip connections (Wu et al., 2020).

Besides the categorization in white-box and black-box methods, attacks can be classified as *restricted* or *unrestricted*, considering the amount of modifications they apply to the images in order to fool the systems.
Restricted attacks: they generally use a L_p-norm distance to bound the modifications. The attacks are crafted with the aim of minimizing the differences between the original image and the adversarial one, even if it means having visible (more or less) artifacts. In Figures 1b and 1d, the attacks produced by Dong et al. (2019) and Wu et al. (2020), two of the most recent state-of-the-art adversarial methods, are shown. In both the cases the generated artifacts are clearly visible.
Unrestricted attacks: they employ large and visible perturbations while keeping the images realistic, natural looking and non-suspicious. The idea is to obtain images that can admit great differences from the original one but, beyond a direct comparison with the original one they cannot be distinguished from any other real (maybe filtered) image. In Figures 1f and 1h, the attacks produced by ACE-Ins (Zhao et al., 2020) and Colorfool (Shahin Shamsabadi et al., 2020) are shown. The differences between the original image and the adversarial one are evident but, if the modifications are good enough, looking just at the adversarial one we might not be able to say that it is the attacking one.

Another aspect that has to be taken into account for real-world attacks is the amount of queries to the victim model that are necessary to craft effective attacks. All the systems proposed in literature need a huge amount of queries and, also in the case of systems built to work with limited access to



(a) Raw image (b) TI-FGSM (c) Raw image (d) SGM

(e) Raw image (f) ACE-Ins (g) Raw image (h) Colorfool

Figure 1: Examples of adversarial samples generated by TI-FGSM (Dong et al., 2019), SGM (Wu et al., 2020), ACE-Ins (Zhao et al., 2020), and Colorfool (Shahin Shamsabadi et al., 2020).

the victim model, some thousands of queries are needed to produce reliable attacks. For example, the state-of-the-art Sign-OPT and QEBA attacks, very efficient in terms of number of queries also in the hard-label black box setting, requires respectively 30k and 20k accesses to the victim model to reach significant success rates (Cheng et al., 2020; Li et al., 2020) against ResNet50 and ResNet18 on the ImageNet dataset.

In this paper we propose a system to craft reliable, effective and highly transferable attacks by composing image filters. This kind of attack offers multiple benefits: *(i)* it is naturally robust to detecting methods able to find noise, injected patterns and irregularities in the high frequencies image (Moosavi-Dezfooli et al., 2018; Liao et al., 2018). *(ii)* it is naturally robust to masking-gradients (and their numerical estimate) defending methods. *(iii)* it produces natural looking artifacts-free images. *(iv)* it produces adversarial attacks that cannot be distinguished from any other filtered images produced extensively every day, especially on social media platforms.

The system works in the pure hard-label black box setting and implements a multi-network approach to increase the attack transferability. It can reach, with artifact-free images, a transferability rate higher than 70% in the case of unsecured networks and, even more interestingly, higher than 60% in the case of adversarially trained network.

Moreover, the system requires a very low number of queries to find an attack, around 640 in the average case, and it can be limited by construction adjusting the algorithm parameters. In the experiments presented here the maximum number of allowed queries is 1610.

Our contribution can be summarized as follows:

- we propose the AGV-multinetwork attack, a system able to craft powerful adversarial attacks capable of attacking even secured adversarially trained models by composing Instagram inspired image filters
- we empirically demonstrate the high transferability of these attacks and we compare our results with other state-of-the-art systems.
- we empirically demonstrate the efficiency of our system in terms of queries to the victim model and we compare our results with other state-of-the-art systems.

2 BACKGROUND

2.1 ADVERSARIAL MACHINE LEARNING

Given an input image $x \in X \subset \mathbb{R}^d$ and its corresponding label y, let F be a neural network classifier that (correctly) predicts the class label for the input image $x : F(x) = y$. An adversarial attack attempts to modify the input image x by adding a perturbation δ into an adversarial image $x^* = x + \delta$ such that the classifier is mislead into making a wrong prediction, i.e. $F(x^*) \neq F(x)$.

If we consider the type of the applied perturbation δ, the attacks can be classified as restricted or unrestricted. In the restricted case, the modifications applied to the original image are usually small and bounded by a L_p-norm distance measure, forcing the adversarial image x^* to be as close as possible to the original one. On the contrary, unrestricted attacks use large perturbations without L_p-bounded constraints that manipulate the image in order to create photo-realistic adversarial examples. In this case the objective is not to limit the modifications on pixels but limit the human perception that a modification has been applied (Shahin Shamsabadi et al., 2020; Zhao et al.; Wang et al., 2021).

2.2 IMAGE FILTERS

Photo editing has become a common practice in many business areas and particularly on social media sites where image enhancement and manipulation are extensively and excessively used. Inspired by Instagram, which offers tools to seamlessly modify images, we propose to combine multiple image filters in order to create custom adversarial image transformations. This approach provides plenty styles options, ranging from subtle and warm looks to more dramatic and vivid colors effects.

Therefore, we implemented ten of the most popular Instagram filters using Python3 and the Pillow, OpenCV and Numpy libraries: *Clarendon, Juno, Reyes, Gingham, Lark, Hudson, Slumber, Stinson, Rise,* and *Perpetua*. Each filter has distinct characteristics and effects given by different level of contrast, saturation, brightness, shadows, etc. For instance, *Clarendon* brightens and highlights a photo, *Juno* adds saturation and warmth making the colors more intense, *Rise* gives a warm glow by mixing a radial gradient with a light sepia tone, while *Hudson* bumps up the blues giving a more colder feel.

Each filter is parameterized by two parameters that have to be optimized by the algorithm: *intensity* α and *strength* s. The role of the parameter α is to alter the intensity of each basic component inside each filter implementation such as brightness, contrast, saturation, edge enhancement, gamma correction and many more. The parameter s is used to control the strength of the filter application and it is defined as the convex interpolation between the original image x and the transformed image x^*:

$$strength(x, x^*, s) = (1.0 - s) \cdot x + s \cdot x^* \tag{1}$$

thus, if $s = 0$ the output image of the filter is the original image, while with $s = 1$ the filter returns the modified image x^*.

3 APPROACH AND ALGORITHM

3.1 PROBLEM DEFINITION

Given a set $S = \{f_1, f_2, \ldots, f_m\}$ of Instagram inspired image filters as described in Section 2.2, and a clean image x, we want to find a sequence of n parametrized filters $\{f_{k_1}(\alpha_1, s_1), \ldots, f_{k_n}(\alpha_n, s_n)\}$ able to produce an adversarial attack against a classifier model F starting from the image x, that is

$$\begin{cases} F(x) \neq F(x^*) \\ x^* = f_{k_n}(\alpha_n, s_n)(\ldots f_{k_2}(\alpha_2, s_2)(f_{k_1}(\alpha_1, s_1)(x))) \end{cases} \tag{2}$$

3.2 APPROACH

The algorithm used to optimize the sequence of filters and their parameters is inspired by Baia et al. (2021). In our proposal the universal approach proposed by Baia et al. has been transformed into

a *multi-network per-image* approach, where the attack is crafted and optimized for just one image with respect to more target models.

Since the algorithm uses an evolutionary approach, the fitness function used to guide the search can have any (also non differentiable) form, and we decide to use it to implement a sort of generalization ability induced by attacking more models simultaneously. This is inspired by Liu et al. (2017) where the authors suggested that attacking an ensemble of multiple networks simultaneously can generate much stronger adversarial examples. Instead of attacking an ensemble network we decided to exploit the optmization ability of the evolutionary approach and try to simoultaneously attack all the (or at least k) reference networks. Therefore, our goal is to find *one* adversarial perturbation per-image that can fool *many* deep learning models. This objective is much harder to achieve than attacking a single model but it guarantees complete (100%) transferability of the attack towards the reference networks.

The proposed multi-network approach has the objective to obtain adversarial images with better transferability avoiding the natural overfitting trap we can fall into when the attacks are crafted by using a unique reference model.

3.3 Algorithm

The optimization method consists of two evolutionary nested algorithms: the outer algorithm, using a GA approach, in charge of finding the sequence of filters to use, and the inner algorithm, based on ES, used to choose the filter parameter values. The general structure of the multi-network algorithm is shown in Algorithm 1.

3.3.1 Outer Algorithm

The outer optimization step is performed by a genetic algorithm: a population of N candidate perturbations is iteratively evolved towards better solutions. In order to breed a new generation, population members are randomly selected and the crossover and mutation operations are performed. The quality of the candidates is evaluated based on their fitness values and, at the end of each iteration, the N best individuals are chosen for the next generation.

Initial population: it is generated by randomly selecting l filters from the set S of available filters and their parameters are initialized with default values equal to 1.

Crossover: a standard one-point crossover is used to generate new off-springs from randomly selected members. Each child is guaranteed to inherit some genetic information from both parents, including the optimized parameters.

Mutation: it is applied by substituting a filter with another one based on a mutation probability. The substituent filter is initialized with random parameter values.

Selection: at the end of each iteration, we choose the N best individuals from the set of 2N candidates (parents and offsprings) according to their fitness values. This process is repeated until the algorithm exhausts the allowed number of generations.

3.3.2 Inner Algorithm

This algorithm is devoted to the parameter optimization and is achieved by using $(1, \lambda)$ evolution strategy with $\lambda = 5$. ES iteratively updates a search distribution by following the natural gradient towards higher expected fitness. In our case, for each list of parameters we compute a batch of λ samples by perturbing the original individual. A gradient towards a better solution is estimated using the fitness values of the λ samples. This gradient is then used to update the original individual. This process is repeated until the algorithm reaches the allowed number of iterations.

3.4 Evaluation

We define the fitness function based on the attack success rate (*ASR*) since the objective of the algorithm is to fool image classification systems.

In the case of multi-networks attacks we average the attack success rate values computed on multiple networks simultaneously. Specifically, given K F_i neural networks models, a I legitimate image,

and the corresponding perturbed image I^* obtained by applying the sequence of parametrized filters we want to evaluate, we defined the multi-network fitness function as:

$$\mathcal{F}_{\mathcal{K}}(I, I^*) = \frac{1}{K} \sum_{k=1}^{K} ASR_k(I, I^*), \tag{3}$$

where ASR_k represents the attack success rate calculated by querying the k-th target model: it returns 1 if $F_k(I) \neq F_k(I^*)$, 0 otherwise with respect to the Top-1 prediction.

3.4.1 Allowed Queries

The algorithm works with limited accesses to the victim model. In our algorithm the victim model is called every time we have to compute the fitness function. Besides the explicit calls to the evaluation function made in the initialization step and in the last step of each outer iteration, we have to consider the $E_{in} \times N_{in}$ calls made by the inner optimization phase (filters' parameters optimization) for each element in the outer population. The maximum number of allowed queries can be easily computed as:

$$Q_{MAX} = N_{out} + E_{out} \times (N_{out} \times E_{in} \times N_{in} + N_{out}) \tag{4}$$

Algorithm 1: AGV-multi-network algorithm

Input: Image I, K classification models, outer population size N_{out}, inner population size N_{in},number of outer generations E_{out}, number of inner generations E_{in}, obj the number of target models we want to attack.
Initialize population P of N_{out} individuals;
foreach $p \in P$ **do**
  Evaluate the fitness of p by $\mathcal{F}_{\mathcal{K}}$;
end
e=0 ;
while *($e < E_{out}$ **and** $\mathcal{F}_{\mathcal{K}} \cdot K < obj$)* **do**
  Offsprings = $\{\emptyset\}$;
  for $i = 1$ **to** N_{out} **do**
    Select randomly two parents p_1,p_2 from P ;
    y_i = crossover(p_1, p_2);
    $\overline{y}_i$ = mutation(y_i) ;
    n_i = ES_optimizer($\overline{y}_i, E_{in}, N_{in}$) ;
    Offsprings $\leftarrow$ (n_i) ;
  end
  foreach $o \in$ Offsprings **do**
    Evaluate the fitness of o by $\mathcal{F}_{\mathcal{K}}$;
  end
  P = selection(P, Offsprings) ;
  $e = e + 1$;
end
return: best filter sequence for image I;

4 Experiments

4.1 Experimental setup

Networks: To validate the proposed method we employed five state-of-the-art neural networks for image classification as reference models to find the adversarial filter sequences, namely ResNet50, VGG19, DenseNet201, MobileNet and NasNetMobile. To evaluate the effectiveness of the transferability we use as hold-out models two normally trained networks, i.e. InceptionV3 and InceptionV4, and one adversarially trained network, i.e InceptionResNetV2-ens-adv (Tramèr et al., 2018).

Different architectures were chosen in order to explore the generalization capability of the algorithm given that these models are fundamental components of other deep learning models used for object

detection and image segmentation. Besides, MobileNet and NasNetMobile are architectures often used in ML-powered mobile apps thanks to their low-latency and lightweight nature which makes them an interesting case study. These models were trained on the large visual ImageNet training dataset (Deng et al., 2009) and they are publicly available.

Multi-network setup: In the multi-network configuration we simultaneously attack multiple deep learning models. More specifically, we study three multi-network setups:

- **5-NET**: includes MobileNet, ResNet50, VGG19, DenseNet201, NasNetMobile
- **4-NET**: includes MobileNet, ResNet50, VGG19, DenseNet201
- **3-NET**: includes MobileNet, ResNet50, VGG19

Dataset: We performed the experiments on a subset of 1000 images from the ImageNet validation dataset. The images were preprocessed according to the requirements of each pre-trained neural network.

Implementations details. The hyperparameters of the outer algorithm were set as follows: $N_{out} = 10$, $E_{out} = 10$ and mutation probability $\rho = 0.5$; The hyperparameters of the inner algorithm were set as follows: $N_{in} = 5$, $E_{in} = 3$, initial learning rate = 0.1 and decay rate = 0.75. These values are the result of a preliminary experimental phase conducted to find a good trade-off between performance and computational time. They allow a maximum of 1610 queries to the victim model we want to attack.

4.2 Attacks and transferability

In this section, we analyze the multi-network scenario. According to (Liu et al., 2017), if an adversarial instance remains adversarial for multiple networks then it is more likely to transfer to other unseen networks as well. Therefore, we leverage the capabilities of the AGV single-model by (Baia et al., 2021) algorithm to attack multiple neural networks simultaneously in order to improve the adversarial robustness and generate adversarial examples with high transferability rates.

We examine the three different multi-network configurations 5-NET, 4-NET and 3-NET described before with filter sequences of length 3, 4, and 5. In this case, we consider a combination of filters to be adversarial if it is capable of misleading all models included in the multi-network setup with respect to the Top-1 predicted class. This is the strongest requirement because we ask the system to find an adversarial filter composition that is able to fool all the networks in the multi-network configuration. This is different from attacking an ensemble network: we guarantee that these images (when they are found) surely will fool all the networks used, while this is not guaranteed by the ensemble training. For example Xie et al. (2019) showed in their experiments for DI^2-FGSM and M-DI^2-FGSM that the attacks trained with the ensemble reach the 90% ASR just in few cases. Results are reported in Table 1. As expected, attacking multiple networks is a more difficult task than attacking single models. However, since the multi-network adversarial examples are guaranteed to work across all built-in models this reduces the need to run the algorithm for each single network individually. Unsurprisingly, there is a correlation between the number of filters and the ASR: longer sequences produce more visible and large modifications that enables to guide an image towards an adversarial sample more easily.

We also computed the transferability using the three test models, namely InceptionV3, InceptionV4, and the adversarially trained InceptionResnetV2-ens-adv, for both multiple and single model attacks. The results are shown in Table 2. In this case, we can notice that the multi-network based attacks achieve higher transferability rates compared to the single-model setup.

Moreover, our attacks can also mislead InceptionResNetV2-ens-adv on more than 60% of the adversarial samples. This shows that even ensemble-adversarially trained models are susceptible to unrestricted filter-based adversarial perturbations. This indicates the potential of the proposed method whose goal is to transform image filters into malicious applications. It raises new security issues and challenges for real-world system since it is particularly difficult to distinguish between benign and malignant filters.

Comparisons with similar attack models.

Table 1: Attack Success Rate (ASR)

	Multi-network			**Single-network**		
Setup	**5-NET**	**4-NET**	**3-NET**	**MobileNet**	**ResNet50**	**VGG19**
3filters	44%	50 %	52%	64%	74%	70%
4filters	58%	50%	70%	74%	88%	82%
5filters	68%	66%	76%	78%	96%	86%

Table 2: Transferability rates

	Trained on								
	5-NET			**4-NET**			**3-NET**		
Tested on	3f	4f	5f	3f	4f	5f	3f	4f	5f
InceptionV3	0.636	**0.655**	0.617	0.600	0.517	0.575	0.500	0.457	0.578
InceptionV4	0.545	**0.793**	0.705	0.600	0.551	0.757	0.538	0.657	0.657
IncResV2-ens-adv	0.545	**0.655**	0.558	0.480	0.517	0.636	0.269	0.514	0.447
	Trained on								
	MobileNet			**ResNet50**			**VGG19**		
Tested on	3f	4f	5f	3f	4f	5f	3f	4f	5f
InceptionV3	0.375	0.378	**0.410**	0.216	**0.340**	0.312	0.257	**0.365**	0.348
InceptionV4	0.437	0.540	**0.564**	0.405	0.477	**0.541**	0.342	0.439	**0.441**
IncResV2-ens-adv	0.250	0.297	**0.359**	0.216	0.250	**0.270**	**0.285**	0.243	0.279

We also compare our algorithm with other existing adversarial attack methods that have similar colorization and filtering characteristics.

SemanticAdv (Hosseini & Poovendran, 2018) randomly changes the hue and saturation values of an image while not affecting the shape of objects.
Colorfool (Shahin Shamsabadi et al., 2020) uses image semantics to modify the colors by employing priors on color perception trying. It tackles the limitations of SemanticAdv which has been found to produce unnatural looking images.
Edgefool (Shamsabadi et al.) and **FilterFool** (Shahin Shamsabadi et al., 2021) generate adversarial examples by applying generic image-enhancement filters obtained by means of neural networks. For example, FilterFool-GC tries to create a perturbation that imitates the gamma correction filter. It is important to note that these methods do not work in a pure hard-label black-box setting because they use the logits information that in general are not exposed and visible to the final user.
ACE (Zhao et al., 2020), **ReColorAdv** (Laidlaw & Feizi, 2019), and **cAdv** (Bhattad et al., 2019) methods use gradient information in order to optimize a color transformation.
In particular, we focus on **ACE-Ins1** and **ACE-Ins2** which employ two Instagram filters to guide the color optimization towards specific styles of enhancement, namely Nashville and Toaster. Adversarial samples generated by the above mentioned methods are reported in Appendix A.

In Table 3 we report the Attack Success Rates obtained by crafting adversarial samples on the reference model (ASR-ref) and the average Transferability Rate (Avg-TR) with respect to different testing models.

Given that the above mentioned techniques were tested on different sets of networks, we decided to report the average value of the published results in order to have an overall panorama of their effectiveness. We chose as reference model ResNet50 since data related to it was available in almost all cited works, with the exception of ReColorAdv where we report the figures regarding InceptionV3.

The experimental results show that our algorithm achieves an attack success rate and transferability rate comparable, and in some cases even better, to these techniques but using less information (top1-label vs logits).

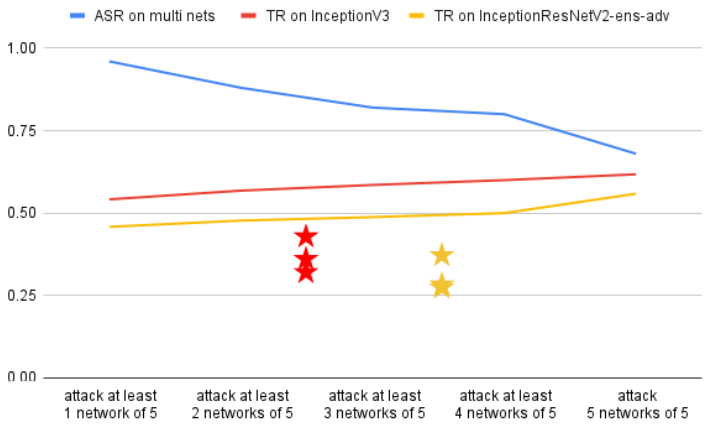


Figure 2: Transfer rate variability depending on the degree of difficulty imposed on the attack. Attacking all 5 reference models at once results in more transferable attacks. The stars indicate the transferability rates of single-model crafted attacks.

Table 3: Comparison with similar colorization and filtering techniques

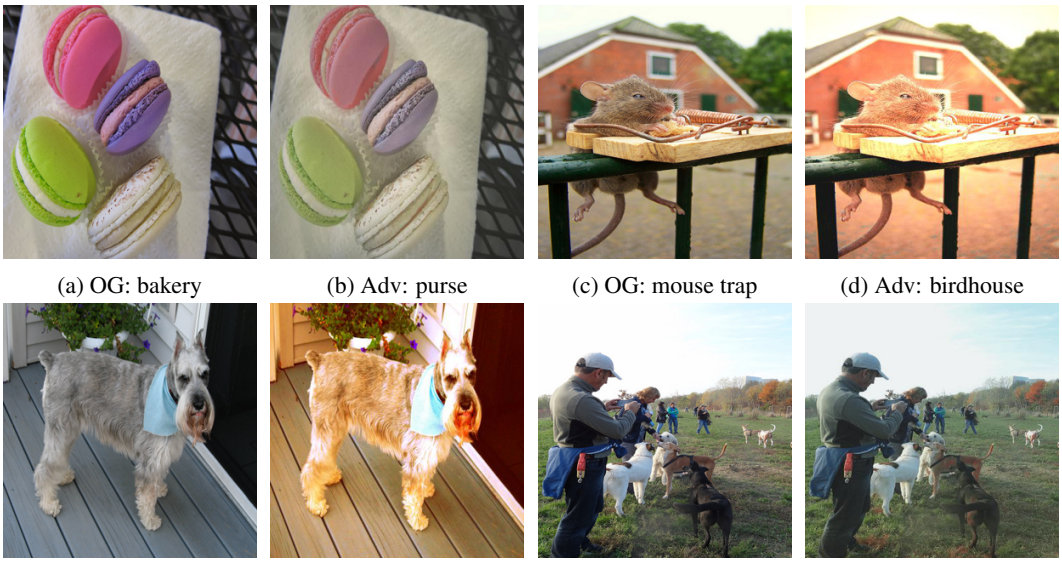
	AGV-5-NET 5f	**AGV-single 5f**	**SemanticAdv**	**Colorfool**	**Edgefool**
ASR-ref	68%	96%	88 %	91.7%	98.1%
avg-TR	62.75 %	37.5%	65.45%	47.00%	43.45%
	Filterfool GC	**ACE-Ins1**	**ACE-Ins2**	**ReColorAdv**	**cAdv**
ASR-ref	99.9%	99.3%	99.4 %	89.2%	99.83%
avg-TR	34.7%	24.24%	26.31%	17.79%	29.78%

From an ASR point of view, the AGV-single method performs better than SemanticAdv and Colorfool but it falls behind on the transfer rate, while AGV-multi-network outperform almost all other systems on the hold-out models. Given the random nature of its approach, the SemanticAdv generates suspicious perturbations that lead to excessively transformed images (people having blue skin or green horses) which can induce misclassification in unseen models more easily. Even though Colorfool tries to limit the adversarial perturbations by using image segmentation to modify the colors within specific ranges, it often produces noticeable, artificial colorization effects. This is mostly due to the inaccuracy of the image segmentation techniques that poorly identifies the objects resulting in unnatural colorization patterns, such as skies with pink clouds or dogs half white, half blue. Despite accessing only the final class-label decision of a model, AGV reaches a higher transferability rate than many other methods relying on richer information for the attack process, such as FilterFool, ACE, ReColorAdv and cAdv. Moreover, we have a great diversity in terms of filter styles, ranging from more bright and colourful to more soft and subtle transformations which are frequently found on many social media platforms, like Instagram. Examples that sucessfully transfer to the InceptionResNetV2-ens-adv are illustrated in Figure 3 and 4 in Appendix A.

5 Conclusions and Future Works

In this work we present the algorithm AGV-multi-network for the optimization of Instagram-style image filters in a multi-network setting able to fool classification systems with a low cost in term of queries and a high transferability rate.

The algorithm was experimentally tested on a subset of the Imagenet validation set. From the testing phase we can conclude that: (*i*) the algorithm shows very good transferability rate also in the case of adversarial trained network and with respect to other method producing images with evident artifacts; (*ii*) the algorithm is able to craft the attacks using a very limited number of queries.



(a) OG: bakery (b) Adv: purse (c) OG: mouse trap (d) Adv: birdhouse

(e) OG: Std. Schnauzer (f) Adv: Mini. Schnauzer (g) OG: Labrador (h) Adv: Doberman

Figure 3: Examples of clean images and their corresponding adversarial example. The adversarial labels are obtained by transferring the modified images to the InceptionResNetV2-ens-adv model.

There is a big room for improvements that are under study. In particular we aim to exploit the multi-objective fitness function of the original AGV algorithm in order to craft targeted deceptions, move the right classification even further away in the modified class distribution, apply automatic measure for image quality assessment and optimize the attacks also to avoid detection methods.

REFERENCES

Alina Elena Baia, Gabriele Di Bari, and Valentina Poggioni. Effective universal unrestricted adversarial attacks using a moe approach. In *In Proc of EvoAPPS 2021*, 2021.

Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A. Forsyth. Big but imperceptible adversarial perturbations via semantic manipulation. *CoRR*, abs/1904.06347, 2019.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4312–4321, 2019.

Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. *CoRR*, abs/1804.00499, 2018. URL `http://arxiv.org/abs/1804.00499`.

Cassidy Laidlaw and S. Feizi. Functional adversarial attacks. In *NeurIPS*, 2019.

Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1218–1227, 2020. doi: 10.1109/CVPR42600.2020.00130.

Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. Divide, denoise, and defend against adversarial attacks. *arXiv preprint arXiv:1802.06806*, 2018.

Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. *In Proc of CVPR 2020*, 2020.

Ali Shahin Shamsabadi, Changjae Oh, and Andrea Cavallaro. Semantically adversarial learnable filters. *IEEE Transactions on Image Processing*, 30:8075–8087, 2021. doi: 10.1109/TIP.2021.3112290.

Ali Shahin Shamsabadi, Changjae Oh, and Andrea Cavallaro. Edgefool: an adversarial image enhancement filter. *In Proc of ICASSP 2020*.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick Drew McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Yajie Wang, Shangbo Wu, Wenyi Jiang, Shengang Hao, Yu-an Tan, and Quanxin Zhang. Demiguise attack: Crafting invisible semantic adversarial perturbations with perceptual similarity. *arXiv preprint arXiv:2107.01396*, 2021.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *International Conference on Learning Representations*, 2020.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019.

Zhengyu Zhao, Zhuoran Liu, and M. Larson. Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter. *In Proc. of BMVC 2020*.

Zhengyu Zhao, Zhuoran Liu, and Martha A. Larson. Adversarial robustness against image color transformation within parametric filter space. *CoRR*, abs/2011.06690, 2020.

A Appendix

We report different adversarial examples generated by state-of-the-art methods and similar systems.

(a) OG: Std. schnauzer (b) OG: Mini. schnauzer (c) OG: Mini. schnauzer (d) Adv: Mini. schnauzer

Figure 4: Examples of different adversarial perturbations styles crafted on the same image by AGV-multi-network with different configurations.

(a) Adv. samples crafted by SGM on ResNet-152

(b) Zoom on the adv sample

(c) Adv. samples crafted by SGM on DenseNet-201

(d) Zoom on the adv sample

Figure 5: Advesarial samples crafted by SGM (Wu et al., 2020) method. The adversarial perturbations creates visible artifacts in the image.

(a) Adv. samples crafted by AGV-multi-network

(b) Zoom on the adv sample

(c) Adv. samples crafted by AGV-multi-network

(d) Zoom on the adv sample

Figure 6: Examples of adversarial samples generated by AGV-multi-network. The adversarial perturbation does not create artificial patterns in the image.

(a) Clean imange (b) SemanticAdv (c) Colorfool

(d) Clean image (e) Colorfool

(f) Clean image (g) Edgefool

(h) Clean image (i) Filterfool

Figure 7: Adversarial examples generated with other filter-based and colorization systems.

(a) ACE-Ins with color style guidance from Instagram filters

(b) ReColorAdv adversarial examples

(c) cAdv adversarial examples

Figure 8: Adversarial examples generated with other filter-based and colorization systems.