arXiv:2107.09225v1 [cs.CV] 20 Jul 2021

# Discriminator-Free Generative Adversarial Attack

Shaohao Lu
School of Computer Science and Engineering
Guangzhou, China
lushh7@mail2.sysu.edu.cn

Yuqiao Xian
School of Computer Science and Engineering
Guangzhou, China
xianyq3@mail2.sysu.edu.cn

Ke Yan
Tencent Youtu Lab
Shanghai, China
kerwinyan@tencent.com

Yi Hu
Tencent Youtu Lab
Shanghai, China
ferdinandhu@tencent.com

Xing Sun
Tencent Youtu Lab
Shanghai, China
winfred.sun@gmail.com

Xiaowei Guo
Tencent Youtu Lab
Shanghai, China
scorpioguo@tencent.com

Feiyue Huang
Tencent Youtu Lab
Shanghai, China
garyhuang@tencent.com

Wei-Shi Zheng
School of Computer Science and Engineering
Guangzhou, China
wszheng@ieee.org

(a) AdvGAN (A method w/ discriminator.)

(b) Ours (A method w/o discriminator.)

**Figure 1: Both the images above can make ResNet18 [11] fail without changing in appearance significantly. The one generated by our method gets better visual quality without discriminator, while the other one generated by AdvGAN [35] has conspicuous circular noises.**

## ABSTRACT

The Deep Neural Networks are vulnerable to **adversarial examples** (Figure 1), making the DNNs-based systems collapsed by adding the inconspicuous perturbations to the images. Most of the existing works for adversarial attack are gradient-based and suffer from the latency efficiencies and the load on GPU memory. The generative-based adversarial attacks can get rid of this limitation, and some relative works propose the approaches based on GAN. However, suffering from the difficulty of the convergence of training a GAN, the adversarial examples have either bad attack ability or bad visual quality. In this work, we find that the discriminator could be not necessary for generative-based adversarial attack, and propose the **Symmetric Saliency-based Auto-Encoder (SSAE)** to generate the perturbations, which is composed of the saliency map module and the angle-norm disentanglement of the features module. The advantage of our proposed method lies in that it is not depending on discriminator, and uses the generative saliency map

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*MM '21, October 20–24, 2021, Virtual Event, China*
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475290

to pay more attention to label-relevant regions. The extensive experiments among the various tasks, datasets, and models demonstrate that the adversarial examples generated by SSAE not only make the widely-used models collapse, but also achieves good visual quality. The code is available at https://github.com/BravoLu/SSAE.


## CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; **Object identification**.

## KEYWORDS

adversarial attack, image classification, person re-identification

**ACM Reference Format:**
Shaohao Lu, Yuqiao Xian, Ke Yan, Yi Hu, Xing Sun, Xiaowei Guo, Feiyue Huang, and Wei-Shi Zheng. 2021. Discriminator-Free Generative Adversarial Attack. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475290


## 1 INTRODUCTION

The Deep Neural Networks(DNNs) play an increasingly important role on a wide variety of computer vision tasks, e.g, object classification, object detection, and image retrieval. Despite the breakthroughs brought by the DNNs for these tasks, the DNNs are found to be vulnerable to the adversarial examples [2, 5, 20, 21, 28], which brings the severe security risk in the safety-critical applications. Szegedy et.al [28] firstly proposed that the adversarial examples can make the classifier based on the DNNs collapsed, which is indistinguishable to the human visual systems. The lack of robustness of the DNNs to the adversarial examples has raised serious concerns for the security of the DNN-based systems closely related to human safety, e.g., the autonomous driving system.

Prevalent adversarial attack methods [2, 5, 21] are gradient-based, which have been proven for their effectiveness in attacking the models for classification. Su et.al [25] successfully attacks the images by modifying just one pixel. However, the limitation of these methods comes from its latency efficiencies, specifically, it needs multiple times of the gradient back-propagation to obtain the adversarial examples. To tackle this limitation, Xiao et.al [36] proposed a method based on GAN called AdvGAN, which can instantly produce adversarial examples. However, the GAN-based adversarial network always suffers from the convergence because of the min-max game between the generator and the discriminator [1], and always lead to either the perceptible perturbation or bad attack ability. Training a effective GAN needs strict training strategy. To address the aforementioned issues, we propose a discriminator-free network: Symmetric Saliency-Based Auto-Encoder, to generate the more effective and inconspicuous perturbations.

Different from the existing attack methods which target at confusing the decision boundary, our method attacks the target model on the feature space, which guarantees its transferability among various tasks. Specifically, we propose angle-norm disentanglement to disentangle the cosine angle and norm between the features of raw images and those of the perturbed images, and push away their angles to make the systems misidentify the perturbed images. In addition, to improve the inconspicuousness of the adversarial examples, we propose a saliency map module to restrict the size of the target region, and weaken the perturbations in label-irrelevant regions of images (e.g., the background, the label-irrelevant objects), while making the model pay more attention to the label-relevant areas (i.e., the main object).

To validate the effectiveness of the proposed method, we conduct extensive experiments on two tasks: (1) Image classification and (2) image retrieval (person re-identification). The experimental results validate the superiority of our method, ranging from the attack ability, the visual quality, and the transferability in the semi-black box (transfer) manner.

In summary, this work proposes a novel adversarial attack method for object recognition. The contributions include the following:

- We consider that attacking the corresponding systems without discriminator is more simplified but comparably effective for adversarial attack. We consider that only the magnitude of the perturbation is tiny enough, the adversarial examples can deceive the human eyes. It is not necessary that whether the discriminator can distinguish the adversarial examples or not. We conduct lots of analysis and experiments for this perspective.
- We propose a novel discriminator-free generative adversarial attack network called SSAE. The saliency map module of it makes the perturbations more deceptive while keeping the attack ability.
- The angle-norm disentanglement of SSAE is proposed to attack the target model by pushing away the angles between the raw instances and the perturbed instances, which enables our method to be easily applied in various tasks with limited modification. We successfully attack the image classification systems and image retrieval systems, and conduct lots of experiments about transferablity.

## 2 RELATED WORK

**Adversarial Attack.** According to the goal of the adversarial attack, it can be classified into two categories: the *targeted adversarial attack* and the *non-targeted adversarial attack*. The targeted attack requires the target model to recognize the adversarial attack as the specific category, while the non-targeted attack only focuses on whether the adversarial examples deceive the target model or not. According to the degree of acquisition to the target model, adversarial attack also can be classified into another two categories: the *white-box adversarial attack* and the *black-box adversarial attack*. In the white-box adversarial attack, we have complete access to the target model, including the architecture of the neural network and the weights of the neural network. With this information about the target model, we can use a gradient-based method to obtain adversarial examples. The most existing adversarial attack methods are in a white-box manner. The black-box attack is a more challenging task, where we only have the output of the target model. Our method is mainly in the non-target and the white-box manner, and we also test its performance in the black-box (transfer) manner.

**Adversarial Attack for Image Classification.** Most existing adversarial attack methods [2, 5, 19, 21] are proposed for the image classification task. Szegedy [28] firstly proposed the adversarial

examples by formulating the following minimization problem.

$$\arg\min_{\mathcal{P}} f(I+\mathcal{P}) \neq f(I) \quad s.t.\ (I+\mathcal{P}) \in D. \tag{1}$$

The input example $I$ will be misclassified after being perturbed by the perturbation $\mathcal{P}$, and $I+\mathcal{P}$ are constrained in the domain $D$. For keeping the inconspicuousness of adversarial example, the $\mathcal{P}$ is expected to be minimized. The *L-BFGS* [17] are used to solve this equation for non-convex models like DNN. Goodfellow et.al. [9] proposed the *FGSM*, a more efficient solution to solve Eq (1). In addition, the *PGD* [19], the *C&W* [2] and the *JSMA* [21] also have achieved great success on the adversarial attacks on image classification. All the aforementioned methods are effective to obtain good adversarial examples but suffer from their expensive computation. For each input image, these approaches search for the optimal perturbation $\mathcal{P}$ with the gradients, which is a time-consuming process. The GAN-based method [36] is proposed to alleviate this limitation, which only needs forward inference once instead of iterative computations. However, the limitation of GAN-based methods lies in training a good generative adversarial network.

**Adversarial Attack for Image Retrieval.** Adversarial Attack for Image Retrieval is a more challenging task because, in the image retrieval task, a query image searches through a database on visual features in the absence of the class labels. Recently, some works [14, 18, 30, 40] target in the adversarial attack for image retrieval, all of them are gradient-based approaches. Wang et.al [32] firstly proposed a GAN-based adversarial attack approach for person re-identification, a specific task of image retrieval. It has a high attack success rate for ReID systems, but poor visual quality. We assume that suffering from the min-max game between generator and discriminator, the perturbations generated by GAN are not stable, thus leading to poor visual quality.

# 3 SYMMETRIC SALIENCY-BASED AUTO-ENCODER

## 3.1 Overview

Figure 2 illustrates the proposed Symmetric Saliency-Based Auto-Encoder framework for adversarial attack. Compared with the GAN-based adversarial attack methods, which are always limited by its convergence difficulty because of the min-max games between the generator and the discriminator, our method based on auto-encoder can obtain adversarial examples with both more good attack ability and visual quality. The symmetric saliency-based auto-encoder consists of an encoder $\mathcal{E}$ and two decoders. Specifically, the two decoders are symmetric: (1) The noise decoder $\mathcal{G}_{\mathcal{N}}$ and (2) the saliency map decoder $\mathcal{G}_{\mathcal{M}}$. The original image $I$ is encoded as the latent vectors by the encoder. The latent vectors are then decoded by noise decoder $\mathcal{G}_{\mathcal{N}}$ to generate noise $\mathcal{N}$, and decoded by saliency map decoder $\mathcal{G}_{\mathcal{M}}$ to generate the mask $\mathcal{M}$, respectively. The $\mathcal{M}$ is designed for alleviating the magnitude of the perturbation in the label-irrelevant region for better visual quality. We denote the raw instance as $I$, the perturbed instance as $\widetilde{I}$, and the features before the fully connected layers of the target model as $\mathcal{T}(\cdot)$. For making the target models misidentify the adversarial examples, we conduct the angle-norm disentanglement which pushes away the cosine angles between $\mathcal{T}(I)$ and $\mathcal{T}(\widetilde{I})$ for mismatching while keeping the value of their norms changeless.

## 3.2 The Discriminator-Free Adversarial Attack

Most existing generative-based methods of adversarial attack are of a similar GAN-based architecture, which consists of a generator and a discriminator. The discriminator $\mathcal{D}$ are supposed to distinguish the adversarial ones or the original ones, to keep the visual inconspicuousness of perturbed instances. However, we consider that the discriminator may be redundant in the adversarial attack as long as the perturbations are tiny enough. The min-max games between generator and discriminator only guarantee the indistinguishability between original ones and perturbed ones in the discriminative space rather than human visual systems.

The min-max games make the algorithm hard to converge, which needs strict training strategy, thus usually leading to either bad visual quality or bad attack ability. Based on the above assumptions, we propose a novel network to get rid of the difficulties of training a GAN, which is simplified but more effective. It is a discriminator-free and generative-based adversarial attack method. The overview of symmetric saliency-based auto-encoder is illustrated in Figure 2. The network consists of three modules: (1) The encoder (2) the perturbation decoder and (3) the saliency map decoder.

**The encoder** $\mathcal{E}$ is lightweight, which is composed of one $7\times7$ convolution, two $3\times3$ convolutions and six *ResBlocks* [11].

**The perturbation decoder** $\mathcal{G}_{\mathcal{N}}$ consists of two $3\times3$ transposed convolutions and one $7\times7$ transposed convolution. The dimension of the output is $3\times H\times W$, the same as that of the input. However, the raw perturbations $\mathcal{N}_0$ may be too large to deceive the human visual system, so we make an extra constraint for its value:

$$\mathcal{N}(I) = min(|\mathcal{G}_{\mathcal{N}}(I)|, \delta), \tag{2}$$

we can manually set the $\delta$ according to our needs: the larger $\delta$ means the higher attacking performance while the smaller one means the better visual quality. In our experiments, we set $\delta = 0.1$.

## 3.3 The Saliency Map for Effective Attack

**The saliency map decoder** $\mathcal{G}_{\mathcal{M}}$ has a similar architecture with the perturbation decoder except the last transposed convolution layer. It is obvious that some regions of the images are critical for the DNN to recognize (i.e., the label-relevant objects), while some regions of the images are less important (e.g., background). Adding the perturbation to label-relevant regions of images leads to a more effective attack. A simple idea is that we can use Mask-RCNN [10] to get the corresponding foreground masks. However, there are some defects of directly applying object segmentation algorithms for generating masks: (1) The obvious boundary of the attack region sometimes leads to the worse visual quality, which makes the perturbation more obvious. (2) The foreground mask can not reflect the relative importance of a specific region of the foreground. To tackle this problem, we propose a saliency map decoder to learn the relative importance of the region by itself. The dimension of the output of $\mathcal{G}_{\mathcal{M}}$ is $1\times W\times H$. We obtain the

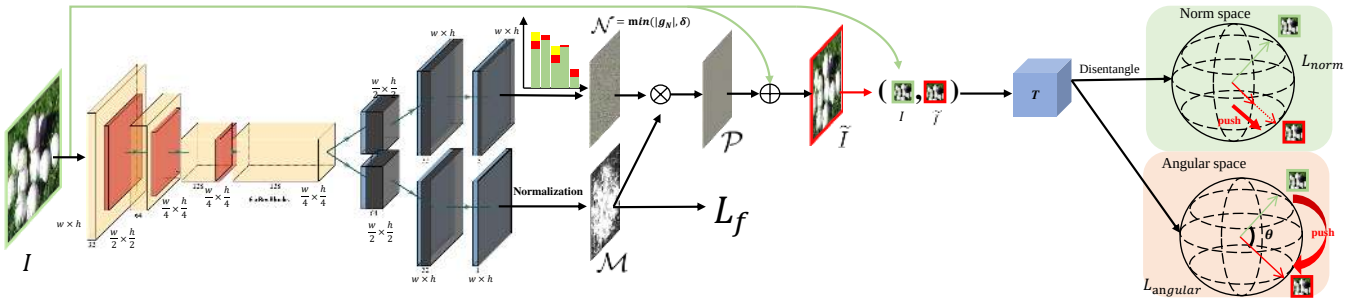



**Figure 2: The overview of the Symmetric Saliency-based Auto-Encoder for adversarial attack. The original images $I$ are feed forward the auto-encoder to generate the perturbations $\mathcal{N}_0$ and saliency map $\mathcal{M}$. The raw image and the perturbed image pairs $(I, \widetilde{I})$ are feed into the target model $\mathcal{T}$, and disentangled with the angles and the norms. The angle-norm disentanglement pushes away the angles between $I$ and $\widetilde{I}$ to collapse the target model while keeps the norms changeless for better visual quality.**

normalized matrix $\mathcal{M}$ by:

$$\mathcal{M}(I^{r,c}) = \frac{\mathcal{G}_{\mathcal{M}}(I^{r,c}) - \min_{r,c}(\mathcal{G}_{\mathcal{M}}(I^{r,c}))}{\max_{r,c}(\mathcal{G}_{\mathcal{M}}(I^{r,c}) - \min_{r,c}(\mathcal{G}_{\mathcal{M}}(I^{r,c})))}, \tag{3}$$

such that the value of $\mathcal{M}(I^{r,c})$ can represent the importance of the point in location $(r, c)$ of the image $I$. In order to make the attacked regions more concentrated, we use the Frobenius norm to optimize the parameters of the saliency map decoder, which makes the saliency map more concentrated. The formula for it is as follows:

$$\mathcal{L}_f = \sum_{i=1}^{N} \sqrt{tr(\mathcal{M}(I_i)^T \mathcal{M}(I_i))}, \tag{4}$$

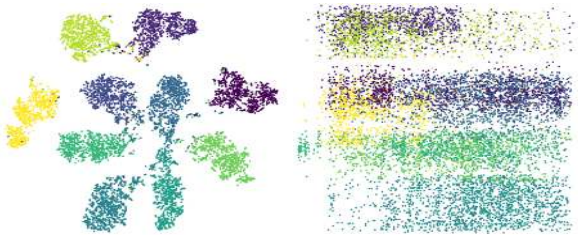
where the $tr(\cdot)$ is the trace function, and the $N$ is the number of images. The Eq.(4) is out of any label supervision, so the label-relevant regions can be well extracted only with other objectives like Eq.(6).

## 3.4 The Angle-Norm Disentanglement for Mis-matching

Chen et.al [3] revealed the gap between the human visual system and the DNN models, and found that the human selection frequency is highly correlated to the angle of the features. Chen et.al [4] proposes norm-aware embedding for efficient person search. We visualize the distribution of the CNN features of all the images in the test set of *cifar-10* with t-SNE [31] in Figure 3 for validation, and we can find that the distribution of the CNN features of *cifar-10* highly depends on the angle. Inspired by the aforementioned facts, we disentangle the norm and angles from the original instance and perturbed instance respectively, as shown in the right part of Figure 2.

**Angle**. For effectively attacking the target models, we focus on applying an attack on the angle between perturbed instance and the original instance. We attempt to push away the cosine angle between the CNN feature of the original instance and the perturbed instance using the following objective formulas:

$$\mathcal{P}(I_i) = \mathcal{N}(I_i) \odot \mathcal{M}(I_i), \tag{5}$$



**(a) The original distribution of the CNN features.** **(b) The perturbed distribution of the CNN features.**

**Figure 3: The t-SNE visualization of the CNN features of *cifar-10*. The distribution totally changes after pushing way their angles.**

$$\mathcal{L}_{angular} = \sum_{i=1}^{N}(1 + \frac{\mathcal{T}(I_i) \cdot \mathcal{T}(I_i + \mathcal{P}(I_i))}{max(||\mathcal{T}(I_i)|| \cdot ||\mathcal{T}(I_i + \mathcal{P}(I_i))||, \epsilon)}), \tag{6}$$

where the $\mathcal{P}$ is the perturbation and the $\epsilon$ is set as $1e-12$ for numeric stability.

**Norm**. We assume that the less we change the features, the more deceptive the perturbed instances are. So we keep the value of the norm changeless as much as possible for better visual consistency using the following objective formulas:

$$\mathcal{L}_{norm} = \sum_{i=1}^{N}(||\mathcal{T}(I_i)|| - ||\mathcal{T}(I_i + \mathcal{P}(I_i))||)^2. \tag{7}$$

The overall loss objective is:

$$\mathcal{L} = \mathcal{L}_{angular} + \alpha(\mathcal{L}_{norm} + \mathcal{L}_f), \tag{8}$$

where $\alpha$ is the hyperparameter that controls the relative importance of the $\mathcal{L}_{norm}$ and the $\mathcal{L}_f$. Finally, we can obtain the perturbed instance by:

$$\widetilde{I} = tensor2img(I + \mathcal{P}(I)), \tag{9}$$

where the $tensor2img(\cdot)$ is the inverse transformation of normalization.

**Table 1: The performance of attacking the classification networks. Our method achieves comparable results on both the attack ability and the visual quality. The AdvGAN does not work in ResNet18, EfficientNet-B0, and MobileNetv2 at all for *imagenette* (red). We apply 20*log to psnr. ResNet18:[11], EfficientNet-B0:[29], GoogLeNet:[27], MobileNetv2[23], DenseNet121[13].**

(a) ***cifar10***

| **Models** | **Accuracy(%)** | | | | | **ssim/ms-ssim/psnr** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | before | PGD | FGSM | AdvGAN | Ours | PGD | FGSM | AdvGAN | Ours |
| ResNet18 | 95.4 | 28.0 | 49.1 | 20.2 | **8.2** | 1.0/1.0/97.0 | 0.96/1.0/82.1 | 0.95/1.0/69.7 | 0.96/1.0/82.2 |
| EfficientNet-B0 | 91.2 | 23.8 | 10.2 | 20.5 | **11.4** | 1.0/1.0/96.9 | 0.96/1.0/82.1 | 0.94/0.99/69.7 | 0.96/1.0/82.4 |
| GoogLeNet | 95.2 | **12.7** | 26.3 | 43.8 | 19.8 | 1.0/1.0/97.0 | 0.96/1.0/82.1 | 0.95/1.0/69.3 | 0.97/1.0/82.5 |
| MobileNetv2 | 97.0 | 26.6 | 18.6 | 52.7 | **4.1** | 1.0/1.0/97.0 | 0.96/1.0/82.1 | 0.95/1.0/69.3 | 0.96/1.0/82.4 |
| DenseNet121 | 95.6 | 21.1 | 42.4 | 19.1 | **7.1** | 1.0/1.0/97.0 | 0.96/1.0/82.1 | 0.95/1.0/69.2 | 0.97/1.0/82.4 |

(b) ***imagenette***

| **Model** | **Accuracy(%)** | | | | | **ssim/ms-ssim/psnr** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | before | PGD | FGSM | AdvGAN | Ours | PGD | FGSM | AdvGAN | Ours |
| ResNet18 | 94.0 | 78.4 | **8.9** | 94.0 | 11.1 | 1.0/1.0/114.1 | 0.87/0.98/82.1 | 1.0/1.0/149.7 | 0.95/1.0/86.2 |
| EfficientNet-B0 | 91.1 | 77.2 | **7.2** | 91.1 | 8.5 | 1.0/1.0/114.1 | 0.87/0.87/82.1 | 1.0/1.0/157.3 | 0.94/0.99/85.4 |
| GoogLeNet | 90.4 | 63.6 | 14.1 | 16.0 | **12.1** | 1.0/1.0/114.0 | 0.87/0.98/82.1 | 0.91/0.99/84.0 | 0.94/1.0/86.4 |
| MobileNetv2 | 93.7 | 49.0 | **8.5** | 93.7 | 10.7 | 1.0/1.0/114.0 | 0.87/0.98/82.1 | 1.0/1.0/156.0 | 0.95/1.0/86.7 |
| DenseNet121 | 86.3 | 70.4 | 22.5 | **10.8** | 19.2 | 1.0/1.0/114.0 | 0.88/0.97/82.1 | 0.90/0.99/83.8 | 0.97/1.0/89.5 |

(c) ***caltech101***

| **Models** | **Accuracy(%)** | | | | | **ssim/ms-ssim/psnr** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | before | PGD | FGSM | AdvGAN | Ours | PGD | FGSM | AdvGAN | Ours |
| ResNet18 | 72.1 | 64.5 | **14.1** | 14.3 | 14.8 | 0.99/0.99/113.9 | 0.86/0.97/82.1 | 0.64/0.96/74.6 | 0.86/0.98/82.3 |
| EfficientNet-B0 | 76.1 | 71.5 | **10.2** | 31.2 | 14.4 | 0.99/0.99/113.9 | 0.86/0.97/82.1 | 0.64/0.96/74.5 | 0.87/0.97/82.3 |
| GoogLeNet | 57.9 | 36.2 | 11.6 | 14.9 | **6.7** | 0.99/0.99/113.9 | 0.86/0.98/82.1 | 0.61/0.95/74.3 | 0.88/0.98/82.7 |
| MobileNetv2 | 69.4 | 47.3 | 10.0 | 27.4 | **3.1** | 0.99/0.99/113.9 | 0.86/0.98/82.1 | 0.63/0.95/74.7 | 0.87/0.97/82.5 |
| DenseNet121 | 54.9 | 47.3 | 15.6 | 9.5 | **2.8** | 0.99/0.99/113.9 | 0.86/0.98/82.1 | 0.68/0.96/74.0 | 0.86/0.99/82.3 |

**Table 2: The performance of attacking the various ReID systems. Compared to the GAN-based method DMR[32], our method outperforms in all the three classical metrics that applied in the fully reference of image quality assessment. We apply 20*log to psnr. IDE:[39], Mudeep:[22], AlignedReID:[37], PCB:[26], HACNN:[16], CamStyle-Era:[42], HHL:[41], SPGAN:[6].**

(a) ***Market1501***

| **Methods** | | **Rank-1 (%)** | | | | | **mAP (%)** | | | | | **ssim/ms-ssim/psnr** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Before | GAP | PGD | DMR | Ours | Before | GAP | PGD | DMR | Ours | DMR | Ours |
| Backbone | IDE(ResNet50) | 83.1 | 5.0 | 4.5 | 3.7 | **2.1** | 63.3 | 5.0 | 4.6 | 4.4 | **2.6** | 0.69/0.89/73.0 | **0.92/0.98/83.1** |
| | Mudeep(Inception-V3) | 73.0 | 3.5 | 2.6 | **1.7** | 8.4 | 49.9 | 2.8 | 2.0 | **1.8** | 5.7 | 0.62/0.90/73.2 | **0.97/0.99/82.3** |
| Part-Aligned | AlignedReID | 91.8 | 10.1 | 10.2 | 1.4 | **0.9** | 79.1 | 9.7 | 8.9 | 2.3 | **2.2** | 0.56/0.90/73.0 | **0.93/0.98/83.7** |
| | PCB | 88.6 | 6.8 | 6.1 | **5.0** | 9.2 | 70.7 | 5.6 | 4.8 | **4.3** | 6.0 | 0.78/0.91/72.9 | **0.94/0.97/81.6** |
| | HACNN | 90.6 | 2.3 | 6.1 | **0.9** | 5.5 | 75.3 | 3.0 | 5.3 | **1.5** | 5.1 | 0.72/0.95/73.0 | **0.95/0.99/81.5** |
| Data Augmentation | CamStyle-Era (IDE) | 86.6 | 6.9 | 15.4 | 3.9 | **2.7** | 70.8 | 6.3 | 12.6 | 4.2 | **3.3** | 0.61/0.90/73.1 | **0.93/0.98/82.8** |
| | HHL(IDE) | 82.3 | 5.0 | 5.7 | 3.6 | **1.5** | 64.3 | 5.4 | 5.5 | 4.1 | **2.5** | 0.71/0.91/72.9 | **0.93/0.98/83.3** |
| | SPGAN(IDE) | 84.3 | 8.8 | 10.1 | **1.5** | 6.2 | 66.6 | 8.0 | 8.6 | **1.6** | 5.1 | 0.71/0.93/72.9 | **0.94/0.98/82.6** |

(b) ***cuhk03***

| **Methods** | | **Rank-1 (%)** | | | | | **mAP (%)** | | | | | **ssim/ms-ssim/psnr** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Before | GAP | PGD | DMR | Ours | Before | GAP | PGD | DMR | Our | DMR | Ours |
| Backbone | IDE(ResNet50) | 24.9 | 0.9 | 0.8 | **0.4** | 0.6 | 24.5 | 2.0 | 6.7 | **0.7** | 0.9 | 0.78/0.90/73.4 | **0.96/0.98/84.9** |
| | Mudeep | 32.1 | 1.1 | 0.4 | 0.1 | 0.6 | 30.1 | 3.7 | 1.0 | **0.5** | **0.5** | 0.75/0.96/73.8 | **0.96/0.98/83.2** |
| Part-Aligned | AlignedReID | 61.5 | 2.1 | 1.4 | **1.4** | 2.4 | 59.6 | 4.6 | 2.2 | 3.7 | **2.7** | 0.68/0.92/73.2 | **0.97/0.98/85.0** |
| | PCB | 50.6 | 0.9 | 0.5 | **0.2** | 2.6 | 48.6 | 4.5 | 2.1 | **1.3** | 2.4 | 0.73/0.86/73.0 | **0.93/0.96/81.5** |
| | HACNN | 48.0 | 0.9 | 0.4 | **0.1** | 2.6 | 47.6 | 2.4 | 0.9 | **0.3** | 2.8 | 0.92/0.93/73.1 | **0.96/0.98/81.9** |

**Table 3: The time consumption and the GPU memory consumption of the generative-based attack methods and the gradient-based methods. The experimental results are conducted on the imagenette, GPU GTX1080, pytorch and $batchsize = 1$.**

| **Category** | **Methods** | **Speed (FPS)** | | | | | **GPU Memory (MiB)** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ResNet18 | EfficientNet-B0 | GoogleNet | MobileNetV2 | DenseNet121 | ResNet18 | EfficientNet-B0 | GoogleNet | MobileNetV2 | DenseNet121 |
| generative | AdvGAN | 60.9 | 42.4 | 16.1 | 36.3 | 16.0 | 741 | 675 | 1043 | 753 | 1251 |
| | Ours | 27.1 | 21.4 | 12.4 | 20.0 | 11.7 | 773 | 703 | 1039 | 795 | 1283 |
| gradient | PGD | 0.43 | 0.34 | 0.10 | 0.25 | 0.08 | 1437 | 1561 | 5167 | 2559 | 6461 |
| | FGSM | 11.53 | 7.7 | 2.46 | 6.60 | 1.56 | 1579 | 1435 | 5165 | 2587 | 6559 |

# 4 EXPERIMENTS

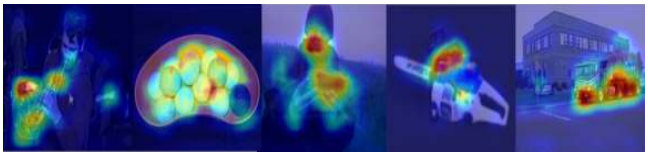
## 4.1 Attack On Image Classification

The most existing adversarial attacks on the image classification are performed on *cifar-10*. However, the images of *cifar-10* are too small to be aware of the perturbations. For better evaluating the inconspicuousness of perturbation, we additionally perform our attack on two more datasets. One is a dataset provided by *fast.ai*: *imagenette* [12], which is a subset of ImageNet, composed of 10 classes, and the other is dataset *caltech101* [8]. We randomly choose 20 images as the test set for *caltech101*. To fully evaluate the effect of the proposed attack, we select five representative classification networks to attack, including ResNet18 [11], EfficientNet-B0 [29], GoogLeNet [27], MobileNetV2 [23] and DenseNet121 [13]. For quantitatively evaluating the overall inconspicuousness of our attack, we refer to the three classical metrics that widely used in thse full reference of Image Quality Assessment (IQA): the structural similarity index measure (**ssim**) [33], the multi-scale structural similarity (**ms-ssim**) [34] and the peak-signal-to-noise ratio (**psnr**) for references. The experimental results are illustrated on Table 1.

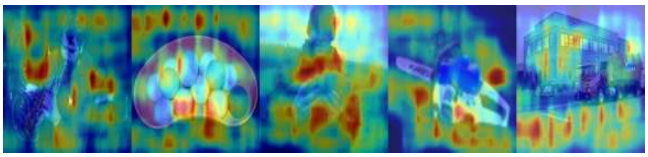
**Attack ability.** The results show that our attack can make all the tested classifiers unreliable, for example, our attack makes the ResNet18 obtain 87.2%/82.9% accuracy degradation on the *cifar-10* and the *imagenette*, respectively. Under the same setting ($\delta = 0.1$), the PGD obtains the best visual quality while the worst attack ability. Specifically, the AdvGAN does not work in lightweight networks (ResNet18, EfficientNet-B0, and MobileNetv2) in *imagenette* at all. To some extent, it validates that our assumption: the GAN-based adversarial attacks are not stable.

**Visual quality.** A typical adversarial sample of the *imagenette* generated by our method are illustrated in Figure 1. More adversarial examples of *caltech101* generated by our method are provided in supplementary for reference. We find that the predicted categories of the models are changed though the visual performance changes little. We also use the Grad-cam [24] to visualize the label-relevant regions before and after the attack in Figure 4. The results show that the label-relevant regions change totally after being attacked by our proposed method.

**(a) The original images.**



**(b) The heat maps before attack.**



**(c) (c) The heat maps after attack.**

**Figure 4: The heat maps generated with Grad-cam[24] are totally changed after attacked by our proposed method. (*imagenette*)**

## 4.2 Attack On Image Retrieval

We also validate the effectiveness of our model on Person Re-ID, a fine-grained image retrieval task aiming at matching pedestrians across non-overlapping cameras. Wang et.al proposed a GAN-based adversarial attacks (denoted as **DMR** [32] in the following) on person re-identification, but we found that the perturbations of DMR are easy to detect. One possible reason might be the instability of the GAN-based method. To have a fair comparison with the DMR, we conduct the corresponding experiments with the same protocols as the DMR: the same target models with the same weights, and the same datasets. According to the protocols in the DMR, we attack the model from three aspects: the backbones, the part-aligned model and the data augmentation on the two widely-used datasets: The *Market1501* [38] and the *cuhk03* [15], to validate the robustness of our method. The results are illustrated in Table 2.

**Attacking ReID Systems with Different Backbones.** We firstly attack the models with the different backbones: The ResNet50 and the Inception-V3. Both the rank-1 and mAP accuracy drop sharply.

**Attacking the Part-based ReID Systems.** Many efficient ReID systems are based on partial alignment. However, the results tell that our attacks also can collapse these systems. For instance, the AlignedReID model gets about 77% degradation in mAP after being attacked, and the PCB can not resist our attack, either.

**Attacking Augmented ReID Systems.** Data augmentation is a widely-used technique to increase the generalization performance of the DNN. We examine the efficiency of our method against the augmentation-based systems. For instance, the CamStype-Era and SPGAN used the GAN as the data augmentation, completely fail after the attack of our method.

**Visual Quality.** From Table 2, we can find that the DMR can also successfully attack the corresponding models and outperforms our method in some cases. However, the most important advantage of our method lies in the visual quality of the adversarial examples. We get this conclusion from two aspects: (1) the image quality assessment measures ssim/ms-ssim/psnr of our method are much higher than those of the DMR. (2) We visualize the adversarial examples in Figure 5 and find that the perturbations of the DMR are much more obvious than ours.

## 4.3 Transferability

The transferability is an important metric for adversarial attack because in the real-world scene we always have no knowledge about the target model. One of the benefits of generative-based attacks comes from their transferability. We evaluate the transferability of our method from four aspects: (1) cross models, (2) cross datasets (3) cross both models and datasets and (4) cross tasks.

**Cross Models.** We convey the cross-model experiments in classification tasks. We train the SSAE in attacking ResNet18 and test its

(a) The original images.

(b) The adversarial examples of DMR[32]

(c) The adversarial examples of Ours.

**Figure 5: The adversarial examples of ours are obviously more deceptive. (*Market1501*)**

performances on attacking other models, including the EfficientNet-B0, the MobileNetv2, the DenseNet121 and the GoogLeNet, and obtain the results in Table 4a. We can find that our attack gets good attack ability in the MobileNetv2 and the GoogLeNet, whose accuracy drop about 76.0% and 58.7%, respectively. When attacking the EfficientNet-B0 and the DenseNet, the classification systems can retain about 50% accuracy, and the attack ability is relatively low. However, in real-world applications, we think 50% accuracy already means that the systems break down.

**Cross Datasets.** We convey the cross-dataset experiments of image retrieval task in Table 4b. We use the Mudeep and the PCB as our target models. All the mAP and Rank-1 have certain degradation, which means our method can work in a cross-dataset manner.

**Cross both Datasets and Models**. We convey the cross-dataset and cross-model experiments in the person re-identification task and obtain the results in Table 4c. We find that the attack ability on the cross-dataset and cross-model setting are even better than the cross-dataset setting only. The cross models and the cross datasets setting can be viewed as semi-black box and the cross dataset and corss-model can be viewed as black-box settings. That is to say, The experimental results validate the effectiveness of SSAE in a black-box manner on the one hand.

**Cross Tasks**. Actually, the cross-task setting is not practical in application, because we always know the task of a real-world system in advance. We test this manner for further validating that our method can be easily transferred to another task. We use the SSAE that trained in the image classification task to perturb the instances in the person re-identification task, vice versa, and obtain the results in Table 4d. The results validate that our attack is transferable even in the cross-tasks scene.

**Table 4: The experimental results about the transferability of the symmetric saliency-based auto-encoder.**

**(a) Cross Models. The model are trained based on ResNet18 on *imagenette*. The test dataset is *imagenette*. All the accuracy drop sharply in cross model manner.**

| Tagret Model | Accuracy (%) | Degradation (%) |
|---|---|---|
| ResNet18→EfficientNet-B0 | 48.9 | 42.3↓ |
| ResNet18→MobileNetv2 | 17.6 | 76.0↓ |
| ResNet18→DenseNet121 | 49.0 | 37.2↓ |
| ResNet18→GoogleNet | 31.7 | 58.6↓ |

**(b) Cross Datasets. The first two rows are based on Mudeep model, the last two rows are based on PCB model.**

| Target Model | Dataset | mAP (%) | R-1 (%) |
|---|---|---|---|
| Mudeep | *Market1501→cuhk03* | 15.8(14.3↓) | 26.3(5.8↓) |
| | *cuhk03→Market1501* | 3.1(46.8↓) | 3.4(69.6↓) |
| PCB | *Market1501→cuhk03* | 14.5(34.1↓) | 18.7(31.9↓) |
| | *cuhk03→Market1501* | 20.0(50.7↓) | 35.1(53.5↓) |

**(c) Cross both Datasets and Models. The left part of the → is the target model and dataset in the training phase and the right part of the → is the target model or dataset in the test phase.**

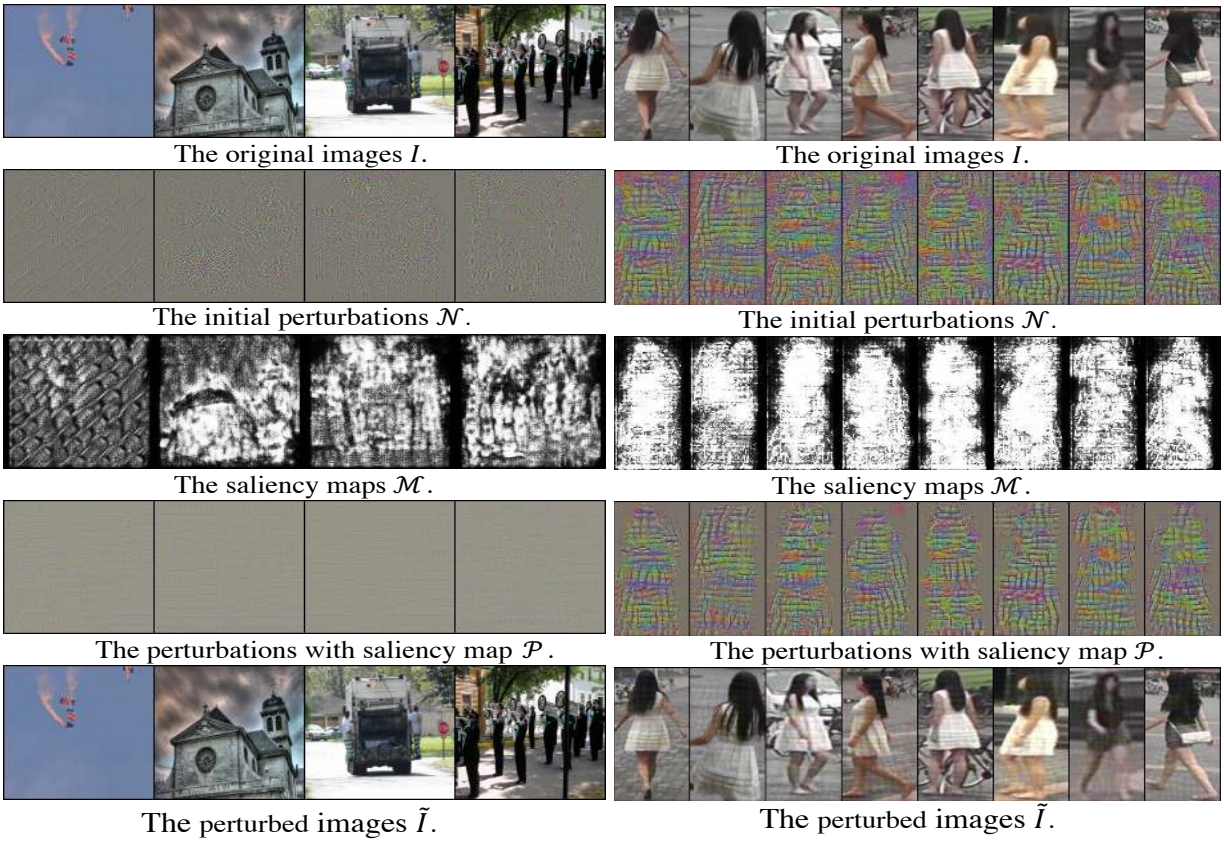
| Target Model | Dataset | mAP (%) | R-1 (%) |
|---|---|---|---|
| PCB→Mudeep | *Market1501→cuhk03* | 12.58 | 19.69 |
| Mudeep→PCB | *cuhk03→Market1501* | 1.13 | 0.98 |
| HACNN→Aligned | *Market1501→cuhk03* | 7.81 | 7.98 |
| Aligned→HACNN | *cuhk03→Market1501* | 35.52 | 45.43 |

**(d) Cross Tasks. The classification task are based on model ResNet18, and on dataset imagenette. The retrieval task are based on model IDE, and on dataset Market1501.**

| Task | Accuracy (%) | mAP (%) |
|---|---|---|
| Classification→Retrieval | - | 63.30→38.57 |
| Retrieval→Classification | 93.96→43.11 | - |

## 4.4 Consumption of Resources

We have mentioned that the generative-based methods are resource-saving compared to the gradient-based method. To further illustrate the benefits of the generative-based attack method, we conduct the corresponding experiments on the speed (FPS) and the load on GPU memory (MiB). We test the speed and the load on GPU memory on PyTorch, one GTX1080 with batch size as 1. We use a public library, AdverTorch [7], to reproduce the performance of PGD and FGSM. Table 3 shows that the generative-based methods are both more time-saving and GPU-memory saving comparing with the gradient-based methods (i.e., FGSM, PGD). The PGD needs multiple times of gradient backpropagation to calculate the perturbations, thus leading to poor performance of speed. Although the FGSM only needs one time of gradient backpropagation, its speed depends on the architecture of the target model, while our proposed method is based on a lightweight auto-encoder with only forward inference once. Both the PGD and the FSGM calculate and save the gradients, thus the load on GPU memory of them is much higher than generative-based methods.




**(a) Image Classification (ResNet18, imagenette)** **(b) Image Retrieval (AlignedReID, Market1501)**

**Figure 6: The visualization of perturbation maps and saliency maps. The white regions in saliency maps mean high label-relevant, while the black regions mean low label-relevant. The perturbations with saliency map are obviously more deceptive. (Best viewed with zoom in.)**

## 4.5 Implementation Details

**The setting of training target models.** All the classification models in the experiments are trained by ourselves. All the training settings are the same amount three dataset, except the input size of *cifar-10* is $32 \times 32$ instead of $224 \times 224$. All the target models in person re-id are provided by Wang et.al [32].

**The setting of training the auto-encoder.** We train SSAE with $\mathcal{L}_{angular}$ in Eq.(6) only for the first 20 epochs and then with the $\mathcal{L}$ in Eq.(8) for another 20 epochs. We train $\mathcal{L}_{angular}$ alone at the beginning because the loss function $L_f$ tends to dominate the optimization if we train loss $L$ directly. The batch size in our experiments is 16 for both image classification and person re-identification. We use Adam as the optimizer and set the learning rate as 1e-5. The hyperparameter $\alpha$ in Eq.(8) is set as 0.0001 in our experiments.

## 4.6 Effect of Saliency map

Although the saliency map module does not use any label supervision directly, it can make the model pay more attention to the label-relevant region with angle-norm disentanglement module together. We randomly choose some examples, and visualize their corresponding perturbations, the saliency map and adversarial examples, respectively in Figure 6. From the saliency maps $\mathcal{M}$ in the third row of Figure 6, we find that the values of the label-relevant regions are closed to 1, while the values of the regions of the label-irrelevant regions are closed to 0. The advantage compared to the binary mask (e.g., Mask-RCNN) is that the saliency map can reflect the relative importance of different regions and make the perturbation smoother. The visual results are in line with the original design intention of the saliency map module. And the saliency map module not only works on the images with one object but also can be extended to the images with multiple objects: the fourth image in **??** has numerous objects and the corresponding saliency map can reflect it. Applying the saliency maps to the initial perturbations makes them more inconspicuous.

## 5 CONCLUSION

This work proposes a discriminator-free generative-based adversarial attack method. To improve the visual quality, a saliency map module is used to limit the added perturbations within a sub-region of the whole image. We disentangle the norm and the angle of the CNN features, push away the angles between raw instances and perturb instances to make the CNN model failed. The abundant experiments validate the effectiveness of our proposed attack, even in the transferable manner.

# REFERENCES


[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017).

[2] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp). IEEE, 39–57.

[3] Beidi Chen, Weiyang Liu Animesh Garg, Zhiding Yu, Anshumali Shrivastava, Jan Kautz, and Anima Anandkumar. 2019. Angular visual hardness. arXiv preprint arXiv:1912.02279 (2019).

[4] Di Chen, Shanshan Zhang, Jian Yang, and Bernt Schiele. 2020. Norm-Aware Embedding for Efficient Person Search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 12615–12624.

[5] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2018. Ead: elastic-net attacks to deep neural networks via adversarial examples. In Thirty-second AAAI conference on artificial intelligence.

[6] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. 2018. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In Proceedings of the IEEE conference on computer vision and pattern recognition. 994–1003.

[7] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. 2019. AdverTorch v0.1: An Adversarial Robustness Toolbox based on PyTorch. arXiv preprint arXiv:1902.07623 (2019).

[8] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In 2004 conference on computer vision and pattern recognition workshop. IEEE, 178–178.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014).

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision. 2961–2969.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 770–778.

[12] Jeremy Howard. [n.d.]. Imagewang. https://github.com/fastai/imagenette/

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4700–4708.

[14] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. 2019. Universal perturbation attack against image retrieval. In Proceedings of the IEEE International Conference on Computer Vision. 4899–4908.

[15] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. 2014. DeepReID: Deep Filter Pairing Neural Network for Person Re-identification. In CVPR.

[16] Wei Li, Xiatian Zhu, and Shaogang Gong. 2018. Harmonious attention network for person re-identification. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2285–2294.

[17] Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. Mathematical programming 45, 1-3 (1989), 503–528.

[18] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. 2019. Who's Afraid of Adversarial Queries? The Impact of Image Modifications on Content-based Image Retrieval. In Proceedings of the 2019 on International Conference on Multimedia Retrieval. 306–314.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In International Conference on Learning Representations. https://openreview.net/forum?id=rJzIBfZAb

[20] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 (2016).

[21] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 372–387.

[22] Xuelin Qian, Yanwei Fu, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. 2017. Multi-scale deep learning architectures for person re-identification. In Proceedings of the IEEE International Conference on Computer Vision. 5399–5408.

[23] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4510–4520.

[24] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision. 618–626.

[25] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation 23, 5 (2019), 828–841.

[26] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. 2018. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In Proceedings of the European conference on computer vision (ECCV). 480–496.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1–9.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013).

[29] Mingxing Tan and Quoc V Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019).

[30] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. 2019. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In Proceedings of the IEEE International Conference on Computer Vision. 5037–5046.

[31] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).

[32] Hongjun Wang, Guangrun Wang, Ya Li, Dongyu Zhang, and Liang Lin. 2020. Transferable, Controllable, and Inconspicuous Adversarial Attacks on Person Re-identification With Deep Mis-Ranking. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 4 (2004), 600–612.

[34] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol. 2. Ieee, 1398–1402.

[35] Chaowei Xiao, Bo Li, Jun yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating Adversarial Examples with Adversarial Networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, 3905–3911. https://doi.org/10.24963/ijcai.2018/543

[36] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610 (2018).

[37] Xuan Zhang, Hao Luo, Xing Fan, Weilai Xiang, Yixiao Sun, Qiqi Xiao, Wei Jiang, Chi Zhang, and Jian Sun. 2017. Alignedreid: Surpassing human-level performance in person re-identification. arXiv preprint arXiv:1711.08184 (2017).

[38] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. 2015. Scalable person re-identification: A benchmark. In Proceedings of the IEEE international conference on computer vision. 1116–1124.

[39] Liang Zheng, Yi Yang, and Alexander G Hauptmann. 2016. Person re-identification: Past, present and future. arXiv preprint arXiv:1610.02984 (2016).

[40] Zhedong Zheng, Liang Zheng, Zhilan Hu, and Yi Yang. 2018. Open set adversarial examples. arXiv preprint arXiv:1809.02681 (2018).

[41] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. 2018. Generalizing a person retrieval model hetero-and homogeneously. In Proceedings of the European Conference on Computer Vision (ECCV). 172–188.

[42] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. 2018. Camera style adaptation for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5157–5166.