Geoinformatica (2022) 26:347–377
https://doi.org/10.1007/s10707-020-00418-7

AP-GAN: Adversarial patch attack on content-based image retrieval systems

Guoping Zhao[1] · Mingyu Zhang[1] · Jiajun Liu[1,2] · Yaxian Li[1] · Ji-Rong Wen[1,2]

Received: 14 May 2020 / Revised: 24 June 2020 / Accepted: 10 July 2020 /
Published online: 2 August 2020

Abstract
Key Smart City applications such as traffic management and public security rely heavily on the intelligent processing of video and image data, often in the form of visual retrieval tasks, such as person Re-IDentification (ReID) and vehicle re-identification. For these tasks, Deep Neural Networks (DNNs) have been the dominant solution for the past decade, for their remarkable ability in learning discriminative features from images to boost retrieval performance. However, it is been discovered that DNNs are broadly vulnerable to maliciously constructed adversarial examples. By adding small perturbations to a query image, the returned retrieval results will be completely dissimilar from the query image. This poses serious challenges to vital systems in Smart City applications that depend on the DNN-based visual retrieval technology, as in the physical world, simple camouflage can be added on the subject (a few patches on the body or car), and turn the subject completely untrackable by person or vehicle Re-ID systems. To demonstrate the potential of such threats, this paper proposes a novel adversarial patch generative adversarial network (AP-GAN) to generate adversarial patches instead of modifying the entire image, which also causes the DNNs-based image retrieval models to return incorrect results. AP-GAN is trained in an unsupervised way that requires only a small amount of unlabeled data for training. Once trained, it produces query-specific perturbations for query images to form adversarial queries. Extensive experiments show that the AP-GAN achieves excellent attacking performance with various application scenarios that are based on deep features, including image retrieval, person ReID and vehicle ReID. The results of this study provide a warning that when deploying a DNNs-based image retrieval system, its security and robustness needs to be thoroughly considered.

Keywords Adversarial attack · Adversarial patch · Image retrieval · GAN

✉ Jiajun Liu
jiajunliu@ruc.edu.cn

[1] School of Information, Renmin University of China, Beijing, 100872, China

[2] Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China

1 Introduction

A major goal of Smart City technologies is to connect, protect, and enhance the lives of citizens with advanced information technology. The research and application of smart cities are very diverse highly diverse, including smart transport [35], smart health [11, 45], and smart security [53], etc. In recent years, Deep Learning has driven the development and application of smart city systems greatly. Benefiting from their strong feature extraction capabilities, Deep Neural Networks (DNNs) have been widely applied in numerous smart city systems and achieved state-of-the-art performance in a variety of applications such as image classification [37], object detection [32], video surveillance [39, 44] and autonomous driving [4]. Many of these applications, such as smart traffic control or security surveillance are fundamentally relying on Content-based image retrieval (CBIR). For example, person Re-Identification (ReID), vehicle ReID and face search, which are widely deployed in the urban video surveillance systems, are all derived from image retrieval algorithms. These systems are used to retrieve the images of a person or a vehicle, recorded by different surveillance cameras.

However, the safety and robustness of DNNs have received increasing attention of many researchers as various investigations report that DNNs are prone to maliciously constructed adversarial examples [9]. Adversarial example as an attack method to DNNs was originally introduced in [40]. It found that deep image classification models will output a false result with high confidence for input images with well-designed indistinguishable disturbances. Subsequently, adversarial examples were found in the models for various tasks, such as object detection [55], semantic segmentation [50], image caption [51], etc. DNN's vulnerability to adversarial examples has become one of the major risks for applying DNNs in critical environments, such as Smart City systems and Automated Driving Systems (ADS). In this work, we studied the potential security risks of these visual retrieval systems in smart cities. It warns that when deploying a DNNs-based image retrieval system, it is necessary to thoroughly consider its security and robustness.

Early methods on adversarial examples need to manipulate each pixel in the digital image [40], but this is not viable for attacks in the physical world. In real-life Smart City applications, the image is usually directly captured by the camera, and can not be modified for each pixel. However, an adversarial patch [3, 7, 19, 41] can serve as an alternative method to generate adversarial examples and be applied in the physical world (e.g. printed as stickers on body of subject), as an agent to carry to attack against the retrieval system. Inspired by recent advances in adversarial patch generation, we further propose a novel method to generate adversarial patches, which aims at attacking image retrieval systems and its derived applications: person ReID and vehicle ReID. Our method produces **A**dversarial **P**atch with **G**enerative **A**dversarial **N**etworks, and is called AP-GAN. AP-GAN aims at attacking the image retrieval systems by learning to generate the adversarial patches, when 'sticked' on the objects (e.g., buildings, persons, vehicles) regions in the image, the image retrieval systems return dissimilar images. In Fig. 1, we show the process of generating an adversarial patch and the example of adversarial patch attack.

Attacking an image retrieval system is very challenging. For the image retrieval system, the image is represented by the feature maps generated by convolutional layers, and the similarity between two images is determined by the Euclidean distance or cosine similarity of their feature vectors. The distance between feature vectors is usually invariant and robust to minor local changes. Based on this observation, Naveed et al. [1] proposed a nearest neighbors defense strategy, to defense adversarial images via searching nearest-neighbor

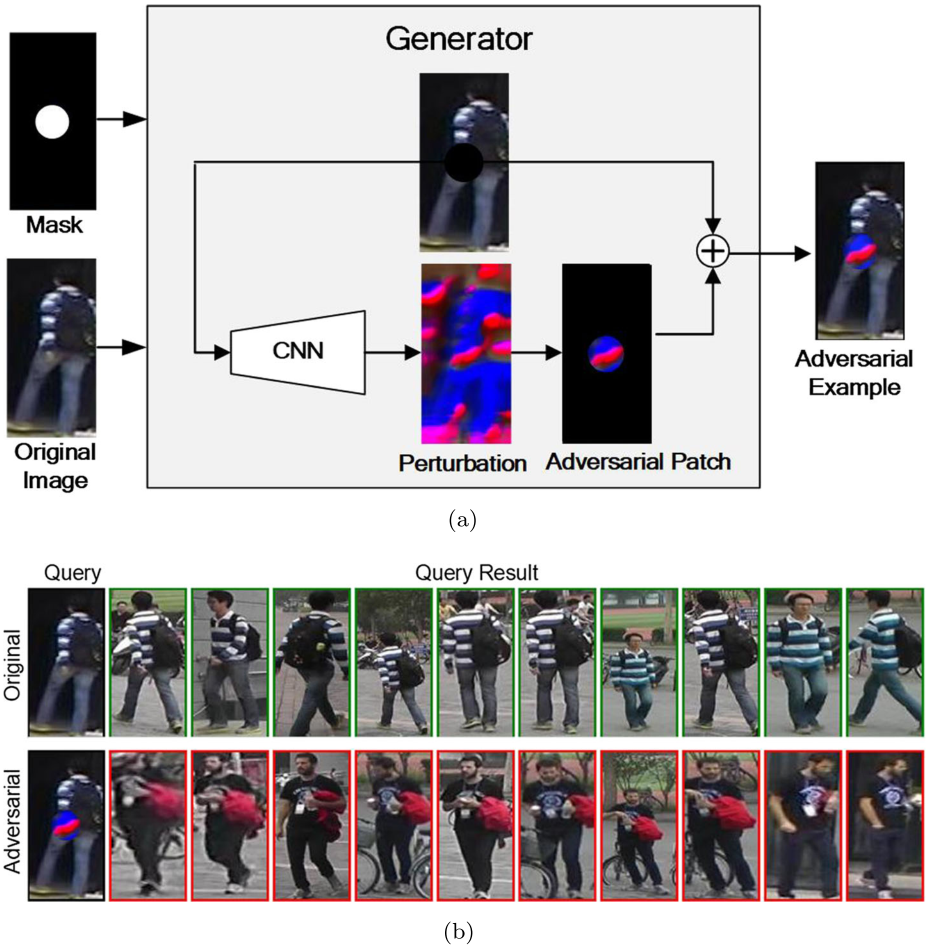


Fig. 1 The process of generating an adversarial patch by AP-GAN (top), and the example of adversarial patch attack on person ReID system (bottom). The adversarial example is generated by placing the adversarial patches on the original image. The person ReID system returns visually dissimilar retrieval results as a result of the attack. The green and red borders on returned images denote correct and incorrect results, respectively

images from a large-scale image database. It is further proved that for image retrieval, minor perturbations do not significantly affect the relative similarities or ranking results of the candidates to the image. Furthermore, the image retrieval datasets lack well-defined labels. To practically resolve these problems, on one side, we propose a metric learning-based loss function to break the similarity relationship in the image retrieval system. On the other side, we train the AP-GAN in a self-supervised way, i.e., only a small amount of unlabeled data for training. Once trained, the AP-GAN can generate adversarial patches without accessing the attacked target model any more. Therefore, AP-GAN is a semi-whitebox [49] attack approach.

To evaluate the performance of our proposed approach, we report results on three tasks: content-based image retrieval, person ReID and Vehicle ReID. The results show the effectiveness of our method on several widely used benchmark datasets. In addition, we conduct a series of ablation experiments to analyze the impact of each component in our architecture on the performance. The adversarial patches generated by AP-GAN can consistently attack the image retrieval system successfully, signalling an imperative need for the research community to improve the security and the robustness of DNNs-basd image retrieval systems. In summary, our main contributions are four-fold:

- We propose an efficient GAN-based attack framework called AP-GAN for generating region-restricted adversarial patches which can be physically printed out and carried to attack image retrieval systems.
- AP-GAN is trained in a self-supervised way and requires only a small amount of unlabeled images for training. Once trained, it can instantly generate visually natural adversarial patches for any input.
- We evaluate AP-GAN on three tasks: content-based image retrieval, person ReID and Vehicle ReID. Empirical results show high effectiveness of Ap-GAN in all tasks.
- Our method exposes potential security risks in the smart city systems, prompting us to consider its robustness when using the DNNs-based image retrieval models and enhance the defense capabilities against malicious attacks such as against samples.

The rest of the paper is organized as follows. Relevant literature is surveyed in Section 2. Our network structure and learning details are presented in Section 3. Section 4 reports comprehensive experimental results and analysis. Finally, we conclude our work in Section 5.

2 Related work

2.1 Smart city and urban computing

Smart city and urban computing aim to enhance both human life and urban environment smartly through fusing the computing science with traditional fields in the context of urban spaces [58]. Most studies on urban computing and smart city focus on managing and analyzing urban data generated by a diversity of sources in urban spaces. The urban data is usually massive, heterogeneous, and spatio-temporal [5]. In recent years, surveillance cameras are widely deployed in urban areas. The applications based on video and image data have become an important part of smart city systems. The range of these applications is very diverse, including face recognition [33], crowd flows prediction [54], person ReID [57], vehicle ReID [20], etc. These applications are critical to citizens' lives and city operations, so the security risks they may be exposed to need to be thoroughly considered.

2.2 Image retrieval

Image retrieval aims to search from a large scale image database for similar images as the query.

It has been widely used in real-life Smart City applications. Person ReID and vehicle ReID are the two most essential derived application of image retrieval, most of the state-of-the-art methods are based on DNNs. ReID methods aim at searching in the gallery, captured from non-overlapping cameras, for images containing the same person/vehicle with the query image. The latest DNN-based image retrieval methods represent images by aggregating the deep features extracted from a pre-trained or a fine-tuned CNN model. The similarity between two images is then directly measured by the euclidean distance or cosine similarity of two image representations in the deep feature space.

An intuitive way to produce the aggregated feature is sum-pooling (SPoC [2]) or max-pooling (MAC [42]) the feature maps output from convolutional layers. CroW [14] applies both cross-dimensional and spatial weighting before sum-pooling to create powerful image representations. There are methods that further improve the performance of image retrieval by fine-tuning the backbone networks (e.g., VGG, ResNet) on task-related datasets. Filip et al. proposed a trainable Generalized-Mean (GeM) pooling layer in [30], which has been shown to outperform previous state-of-the-art approaches.

Person ReID and vehicle ReID, as derived tasks from image retrieval, have attracted a substantial amount of research efforts for their common usage in Smart City applications. Feature representation learning methods normally consider a ReID model as a multi-class classification problem by treating each identity as a distinct class. These methods include: global feather representation learning [57], local feature representation learning [39] and hybrid representation learning [44]. Other methods treat ReID as deep metric learning paradigm, they aim at constructing a feature space by pulling similar images closer while pushing dissimilar images further in the target feature space, by designing different batch sampling strategies and metric loss functions. The most commonly used loss functions include contrastive loss [47], triplet loss [6] and their variants. Some methods study the strategy for informative batch sample mining, such as hard negative mining [36], semi-hard negative mining [33], online hard triplet mining [46].

2.3 Generative adversarial networks

Generative Adversarial Networks (GANs) are first proposed by Goodfellow et al. [8], who formulated the GAN framework as a two-player min-max game between two adversarial networks. Radford et al. proposed Deep Convolutional GANs (DCGANs) [31], which introduced convolutional layers and convolutional-transpose layers to GANs architecture.

Because of the powerful capability for capturing data distribution and generating realistic images, GANs have achieved excellent performance on many image-to-image translation tasks, like image super-resolution [17], image deblurring [15], style transfer [48], image synthesis [13], etc. Despite the tremendous successes, GANs still suffer from the challenges of model collapse and instability in training. Numerous methods have been proposed to address these problems by improving the optimization objectives. For instance, LSGANs [23] uses the least-squares loss function instead of the sigmoid cross-entropy loss function for the discriminator, making the training process more stable.

2.4 Adversarial examples

The adversarial example is first proposed in [40], which proves small and intentional perturbations can mislead machine learning models to make false predictions. After that, much effort has been dedicated to construct adversarial examples to attack machine learning systems. Goodfellow et al. [9] proposed the fast gradient sign method (FSGM), which adds a small error multiplied by the sign of the gradients to the input to generate adversarial examples. Basic Iterative Method (BIM) [16] is the iterative version of FSGM, which produces better adversarial images by applying gradient update and clipping repetitively.

DeepFool [24] assumes that the example space is divided by hyperplane and iteratively move the input along the direction of closet decision boundary in a few iterations and acquire a minimal norm adversarial perturbation. AdvGAN [49] utilizes generative adversarial networks to produce adversarial examples, which visually closer to the natural image.

Unlike methods that generate different perturbations for each image, Universal Adversarial Perturbations (UAP) [25] computes an image-agnostic adversarial perturbation. In order to produce imperceptible adversarial perturbations, most of the methods directly modify pixel values of the digital image. However, in real-world scenarios, such as person ReID, the input of the model is taken directly by the camera. It is impossible to generate adversarial examples by precisely controlling the value of each pixel in such scenarios. Therefore, some researchers turn to the study of adversarial patch attack [3], which produces adversarial example by generating an adversarial patch. The patch can be placed in a certain area, such as the area of a person's shirt and the wall of a building, so that such attacks could be performed from the physical world. DPATCH [21] learns and embeds a small patch in the input image and performs an effective attacking against the state-of-the-art object detector. Perceptual-Sensitive GAN (PS-GAN) [19] focuses on attacking image classification models through generating an adversarial patch with GAN. Akshayvarun et al. [38] proposed Occluding Patch, which is generated to fool both the classifier and the interpretation models of the resulting category, such as Grad-CAM [34]. A recent survey [52] provides a comprehensive review of adversarial example attacks.

The most related literature to AP-GAN includes recently proposed PIRE [22], retrieval-based UAP [18] and TMAA [43]. All three methods are targeting at image retrieval systems. PIRE calculates the perturbations through hundreds of continuous iterations, which is very time-consuming. In addition, PIRE is a white-box attack, which needs to know all the details of the target network when generating adversarial examples. Retrieval-based UAP [18] extends the UAP algorithm [25] from attacking the classification applications to image retrieval. It seeks a universal image-agnostic adversarial perturbation for all query images. TMAA is a targeted mismatch attack for image retrieval and is focused on concealing the query in a privacy preserving scenario. Different from the above works, our AP-GAN focuses on generating visually natural adversarial patches by GANs. The adversarial patch attack is closer to the attack mode of the physical world. To the best of our knowledge, this is the first paper that aims at studying how to generate adversarial patches to attack against image retrieval systems.

3 Methodology

The core goal of this work is to attack image retrieval systems by adding an adversarial patch to the query image. In this section, we will first formulate the problem of adversarial patch attack on deep feature-based image retrieval systems. Then, we present the network architecture and loss function of our AP-GAN in detail. At last, we introduce the training process.

3.1 Problem formulation

Given an input image $\boldsymbol{x}$ of size $H \times W$, the feature maps (or activations) from a convolutional layer l are denote as $\chi \in \mathbb{R}^{c \times h \times w}$. Let $T_\theta()$ be the target image retrieval network with parameters θ , and F be the feature aggregate function (pooling function). We denote f as the final deep representation of x, and $f_{\boldsymbol{x}} = F(T_\theta(\boldsymbol{x}))$. The similarity of two images ($\boldsymbol{x_i}$ and $\boldsymbol{x_j}$) is measured by calculating the distance between the deep features of the two images via a metric function $d(f_{\boldsymbol{x}_i}, f_{\boldsymbol{x}_j})$.

In general, DNNs-based image retrieval systems improves the retrieval performance in four ways: designing more effective feature aggregation functions (e.g., PoC [2], RMAC [42], GeM [30]); fine-tuning pre-trained networks under a more efficient objective function (e.g., classification loss [26], contrastive loss [30], triplet loss [10]); improving network architectures that yield more distinctive features(e.g., attention mechanism [26], multi-branch network [44]); using a re-ranking approach further improving the retrieval performance. In all four cases, the common purpose is to make similar images (or images containing the same instance) have small distances in the deep features space, and vice versa. Therefore, to successfully attack an image retrieval system, we need to invalidate such properties while keeping the adversarial example as real and similar to its original as possible.

We use $\tilde{\boldsymbol{x}}$ represents the adversarial example (with normalized color values), which can be formulated as:

$$\tilde{\boldsymbol{x}} = clip(\boldsymbol{x} + \delta, 0, 1), \tag{1}$$

where δ represents the perturbation, and function $clip(input, min, max)$ is used to limit all elements in $input$ into the range $[min, max]$. Here, we use $clip()$ function to restrict the pixels of the adversarial examples in the range of digital image space.

The adversarial patch is a special kind of adversarial example. The perturbation is only added to a small area on the image, while the other areas remain unchanged. We take a predefined constant binary matrix m as the mask, to determine the location and area of the disturbance. The dimensions of m are the same with the input image, and we set the value to 1 on the location of the adversarial patch added and 0 in everywhere else. In the case of Adversarial patch attack, we can formulate the adversarial example as:

$$\tilde{\boldsymbol{x}} = clip(\boldsymbol{x} \odot (1 - m) + \delta \odot m, 0, 1), \tag{2}$$

where $\odot$ denotes the element-wise multiplication (hadamard product), and the perturbation δ is generated by the generator of AP-GAN.

The generated perturbation is expected to push the query image away from the original image in the deep feature space. This can be described the following objective function:

$$\begin{aligned} maximize \quad & d(f_{\boldsymbol{x}}, f_{\tilde{\boldsymbol{x}}}), \\ s.t. \quad & \|\delta \odot m\|_\infty \le \epsilon, \tilde{\boldsymbol{x}} \in [0, 1], \end{aligned} \tag{3}$$

where $\|\delta \odot m\|_\infty \le \epsilon$ is used to restrict the max volume of adversarial patch through $clip()$ function, and ϵ is a parameter.

However, if we only optimize (3), we will obtain an adversarial example that is visually and significantly different from the original. The distance between the features and the magnitude of the perturbation compete with each other. Excessive distance will result in a big perturbation, and small perturbation may lead to the inability to effectively push away the features. To find a balance between maximizing the distance of the features and minimizing perturbation, we propose a novel loss function based on metric learning, with adaptive margin, in Section 3.3.

3.2 The architecture of AP-GAN

GANs have achieved remarkable results in many image generation tasks recently [15, 17]. AP-GAN is inspired by these works, and employ a GANs-based framework to generate effective yet natural and subtle perturbations. The overall architecture of the AP-GAN is shown in Fig. 2.

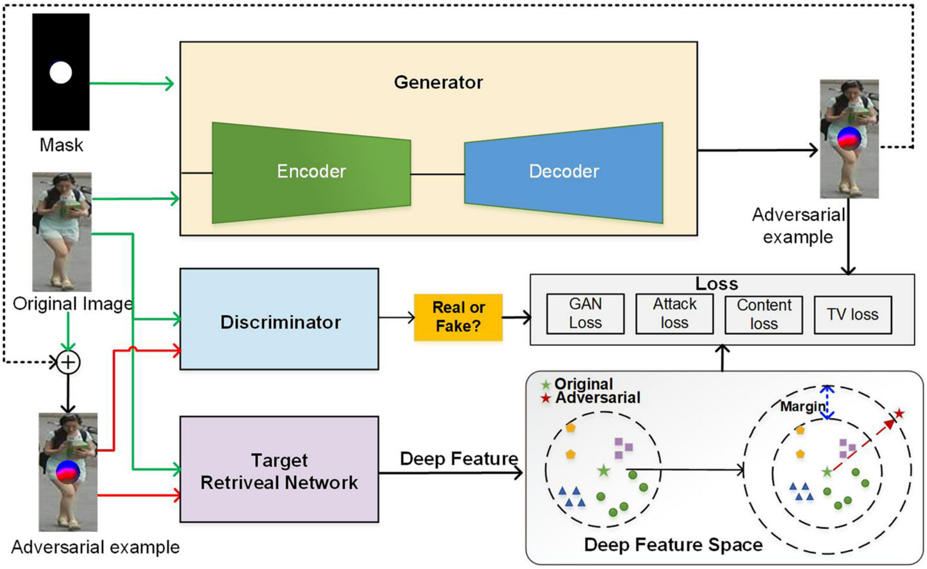


Fig. 2 The overall architecture of the AP-GAN attack framework. We designed four loss functions: the GAN loss enforces that the adversarial example looks like a natural image as much as possible; the attack loss is used to push the adversarial example always from the original query image in the deep feature space; the content loss and TV loss act as regularization to restrict the level of perturbations

AP-GAN contains a generator and a discriminator. The generator that learns to generate a deceptive adversarial patch for each input image. By adding the adversarial patch to the input image, we obtain the adversarial example. The discriminator is used to distinguish generated adversarial examples from the real images in the training phase. In addition, during training, we need to be able to obtain the final features of the input images from the target network. The features are used to calculate the distance between the images in the deep feature space. In this case, the target network is the feature extraction network of the target image retrieval system.

3.2.1 Generator

In AP-GAN, the generator G is used to generate query-specific perturbations by mapping the input query images to the adversarial perturbations manifold. We use the encoder-decoder CNN architecture to build the generator, which allows the network to fuse the hierarchical features to generate perturbation. G begins with three convolutional layers and four residual blocks, with skip-connections [12], down-samples, and encodes the input image into a latent-space representation. The residual block is composed of two convolutional layers followed by a batch normalization layer, and a ReLU activation layer is placed after the first convolutional layer. Each convolutional layer in the residual block contains 32 kernels using kernel size of 3 and stride of 1 and padding. The skip-connection in the residual block is used to accelerate the convergence process by elevating the vanishing gradient phenomenon. As the deconvolution layer often creates checkerboard pattern of artifacts, to obtain more realistic images, we use the resize-convolution approaches (an upsampling layer followed a convolutional layer) instead of deconvolution layer for up-sampling. The decoder maps the latent-space representation to the adversarial perturbation space. The size of perturbation is the same as the input image. Finally, the adversarial patch is obtained by the element-wise multiplication of the perturbation and predefined mask matrix.

3.2.2 Discriminator

For AP-GAN, the role of the discriminator D is to determine whether the input image is a generated adversarial example or a real image. The architecture of D is fairly straightforward: it consists of four convolution blocks, each of which is composed of a convolutional layer, a batch normalization layer, and a Leaky ReLU function. The last layer of D is a global average pooling layer followed with the sigmoid function, which generates a one-dimensional output that represents the probability of input being a real image.

3.2.3 Target network

In the inference phase, the AP-GAN does not need to access the target network. But in the training stage, in order to calculate the similarity between images, AP-GAN needs to obtain the representation of the images in the image retrieval system. No matter how diverse the network structure is, most DNNs-based image retrieval systems represent the image as a fixed-length feature vector. The similarity between images is measured by a distance metric function, such as L2 distance or cosine similarity. For AP-GAN, details about the target network's architecture is insignificant, the only information required is the representation of the images. The architecture of the network could remain a black box to AP-GAN. This shows that AP-GAN provides an attack method with excellent transferability and generalizability.

3.3 Loss function

The optimization objective is used to reduce the retrieval accuracy while ensuring the perturbation adapts the style of the image. In order to achieve the above objective, we use four loss functions to jointly guide the training process of the AP-GAN, include a content loss, a GAN Loss, an attack loss, and a total variation (TV) Loss. The content loss is used to penalizes the generator for introducing differences from the adversarial example to the input image. The GAN loss is used to make the generated adversarial examples look like real images. The attack loss is used to break the similarity relationship between images, which is the key to achieve an effective attack. The TV loss is a regularization item, to make the adversarial example smooth.

3.3.1 Content loss

The content loss uses the square error function to measure the difference between the input image and the adversarial example. We minimize the content loss to ensure that the adversarial example is as visually similar as possible to the original image. The content loss is formulated as below:

$$\mathcal{L}_{content} = \left\| \tilde{\boldsymbol{x}} - \boldsymbol{x} \right\|_2^2, \tag{4}$$

where $\boldsymbol{x}$ is the original image and $\tilde{\boldsymbol{x}}$ is the generated adversarial example. The $\tilde{\boldsymbol{x}}$ is the finally adversarial example, defined in Eq. 2.

3.3.2 GAN loss

In AP-GAN, the generator produces adversarial patches, and then adds it to the input image as adversarial examples. The goal of the generator is to fool the discriminator and make it mistakenly believe the adversarial example is a sample from the real image distribution. Meanwhile, the discriminator tries to distinguish the generated adversarial example from the real image. For a stable training process, we use the least squares loss [23] instead of the cross entropy loss. The GAN loss for the generator and the discriminator can be defined as follows:

$$\mathcal{L}_{GAN\_D} = \mathbb{E}_{\boldsymbol{x}\sim p_x}[(D(\boldsymbol{x}) - 1)^2] + \mathbb{E}_{\tilde{\boldsymbol{x}}\sim p_{(\tilde{\boldsymbol{x}}|\boldsymbol{x})}}[(D(\tilde{\boldsymbol{x}}))^2], \tag{5}$$

$$\mathcal{L}_{GAN\_G} = \mathbb{E}_{\tilde{\boldsymbol{x}}\sim p_{(\tilde{\boldsymbol{x}}|\boldsymbol{x})}}[(D(\tilde{\boldsymbol{x}}) - 1)^2], \tag{6}$$

where p_x is the distribution of real images and $p_{(\tilde{\boldsymbol{x}}|\boldsymbol{x})}$ is the conditional distribution of adversarial examples, given $\boldsymbol{x} \sim p_x$.

3.3.3 Attack loss

The goal of attack loss is to push the adversarial example away from the original image and its neighbors in the deep feature space. However, this simple intuition requires much effort to formulate into feasible optimization objectives. A large distance may lead to the adversarial patch abrupt, on the other hand, a small distance can not produce an effective attack. So, how far is it appropriate to push the adversarial example?

To address the problem, we propose an adaptive strategy based on triplet loss [33] and online hard negative mining [46], It is illustrated at the bottom right of Fig. 2. Let $< \boldsymbol{x}, \tilde{\boldsymbol{x}}, \boldsymbol{x}' >$ denote a triplet, where $\boldsymbol{x}$ is the original input image, $\tilde{\boldsymbol{x}}$ is the generated adversarial example and $\boldsymbol{x}'$ is the hardest example (i.e., the image with the largest distance from $\boldsymbol{x}$ in the batch). AP-GAN determines the appropriate distance by making the distance between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ greater than that of $\boldsymbol{x}$ and $\boldsymbol{x}'$ by a given margin α. The constraint can be written as:

$$d(f_{\boldsymbol{x}}, f_{\boldsymbol{x}'}) + \alpha + \leq d(f_{\boldsymbol{x}}, f_{\tilde{\boldsymbol{x}}}), \tag{7}$$

where α is a given scalar, used to control the margin. The attack loss function can be defined as:

$$\mathcal{L}_{attack} = max(d(f_{\boldsymbol{x}}, f_{\boldsymbol{x}'}) + \alpha - d(f_{\boldsymbol{x}}, f_{\tilde{\boldsymbol{x}}}), 0). \tag{8}$$

3.3.4 Total variation loss

The TV loss is most often used for image denoising and image inpainting, for imposing local spatial continuity. This motivates us to use the TV loss in PS-GAN, making the generated adversarial patch look smooth and coordinated with the surrounding. The TV loss is:

$$\mathcal{L}_{tv} = \sum_{i,j} \sqrt{(\tilde{\boldsymbol{x}}_{\boldsymbol{i},\boldsymbol{j}+\boldsymbol{1}} - \tilde{\boldsymbol{x}}_{\boldsymbol{i},\boldsymbol{j}})^2 + (\tilde{\boldsymbol{x}}_{\boldsymbol{i}+\boldsymbol{1},\boldsymbol{j}} - \tilde{\boldsymbol{x}}_{\boldsymbol{i},\boldsymbol{j}})^2}. \tag{9}$$

3.3.5 Total Loss

Finally, the total loss of the proposed AP-GAN is defined by the combination of the loss functions aforementioned:

$$\mathcal{L}_G = \mathcal{L}_{GAN\_G} + \lambda_c \mathcal{L}_{content} + \lambda_a \mathcal{L}_{attack} + \lambda_{tv} \mathcal{L}_{tv}, \tag{10}$$

$$\mathcal{L}_D = \mathcal{L}_{GAN\_D}, \tag{11}$$

where λ_c, λ_a, and λ_{tv} are the corresponding weights, as hyper-parameters balance the contribution of each part.

3.4 Training process

Training AP-GAN is an iterative process that G and D perform alternating gradient descent over mini-batches. The detailed training procedure is formally presented in Algorithm 1. In the first step, we sample a batch data from the training set as input images, and send them to the fixed G (with a mask), to generate adversarial patches. Then we paste these patches on the original input image as fake images. Both the real images and fake images are then sent to D to calculate the GAN loss for D ($\mathcal{L}_{GAN\_D}$) for the optimization of D's parameters. In the next step, we fix the parameters of D, and send the generated fake images to D to calculate the GAN loss for G ($\mathcal{L}_{GAN\_D}$). G is then optimized by the weighted sum of four losses (i.e., $\mathcal{L}_{GAN\_G}$, $\mathcal{L}_{content}$ and $\mathcal{L}_{metric}$). For each batch, D and G are alternately optimized in such training process. The final output of the training process is the generator model, which used to generate adversarial patches during the inference time. After the training is completed, the G can instantly generate the adversarial patches without D.

Algorithm 1 The training procedure of AP-GAN.

```
Require: training set: $X = \{x^i\}_{i=1}^N$; batch size: $b$; mask: $m$; The number of steps to apply
    to the discriminator: k(we used k = 1)
for the number of training iterations do
    for i=1 to k do
        Sample minibatch of n images $\{x^1, ..., x^n\}$;
        Generate minibatch of n perturbations $\{\delta^1, ..., \delta^n\}$;
        Construct minibatch of n adversarial patches $\{\gamma^i = \delta^i \odot m | \gamma^1, ..., \gamma^n\}$;
        Get minibatch of n adversarial examples $\{\tilde{x}^i = x^i \odot (1 - m) + \gamma^i | \tilde{x}^1, ..., \tilde{x}^n\}$;
        Calculate the loss $\mathcal{L}_D = \mathcal{L}_{GAN\_D}$;
        Update the parameters of discriminator by Adam optimiser.
    end for
    Calculate the loss $\mathcal{L}_G = \mathcal{L}_{GAN\_G} + \lambda_c \mathcal{L}_{content} + \lambda_m \mathcal{L}_{attack} + \lambda_{tv} \mathcal{L}_{tv}$
    Update the parameters of generator by Adam optimiser.
end for
return The trained generator model
```

4 Experiments

In this section, we first describe the experiment setups, including the used datasets, the evaluation metrics and the implementation details. We then report the performance of our approach on three tasks: image retrieval, person ReID and vehicle ReID, respectively. At last, we conduct a series of ablation studies to analyse the impact of each component on the performance and show the robustness of adversarial patches generated by AP-GAN.

4.1 Datasets and evaluation protocols

We evaluate the proposed AP-GAN on three tasks: image retrieval, person ReID and vehicle ReID. Public benchmark datasets are used in our experiments: Oxford5K [27] and Paris6K [28] for image retrieval, Market-1501 [56] and DukeMTMC-ReID [59] for person ReID and VeRi776 [20] for Vehicle ReID.

4.1.1 Evaluation Datasets

- **Oxford5K** is the Oxford Buildings Dataset, which contains 5062 images collected from Flickr. It offers a set of 55 queries for 11 landmark buildings, five for each landmark.
- **Paris6K**, similar to Oxford5k, the Paris6k dataset are composed of 6,412 images collected from Flickr by searching for Paris landmarks.
- **Market1501** contains 32,688 annotated bounding boxes of 1,501 individuals. 751 persons' images are used for training and 750 persons' are used for testing.
- **DukeMTMC-ReID** is constructed from a large-scale multi-target multi-camera tracking dataset DukeMTMC. We use 702 persons for training and the remaining 702 persons for testing from DukeMTMC-ReID.
- **VeRi776** contains 51,035 images of 776 vehicles, which were captured by 20 cameras on a circular road of 1.0 km^2 areas. The training set contains 576 identities and the test set contains the remaining 200 identities.

It should be noted that Oxford5K and Paris6K do not contain training sets and can only be used to evaluate the performance of image retrieval systems. Therefore, we use the images from **retrieval-SfM-30k** [29] as the training set, when training AP-GAN to attack image retrieval models. Retrieval-SfM-30k is composed of 30,012 images, downloaded from Flickr using keywords of landmarks. When training AP-GAN, we only use 1,691 query images in retrieval-SfM-30k.

4.1.2 Evaluation metrics

We follow widely-used evaluation metrics reported in other research papers, using mean Average Precision (mAP) and accuracy at Rank-1, Rank-5 and Rank-10 as the evaluation protocols. We evaluate the attacking performance of AP-GAN by comparing the values of these metrics before and after the adversarial attacks. Note that lower mAP and accuracy at Rank-[1, 5, 10] indicate more successful attacks, and hence better attack performance for the corresponding model.

4.2 Implementation and parameter settings

All of our models and experiments are implemented in PyTorch framework.[1] The experimental server is equipped with 4 NVIDIA TITAN Xp GPUs, 4 Intel Xeon Silver CPUs and 128GB of RAM. Adam is used to optimize the discriminator and generator with the

[1] https://pytorch.org/

following hyper-parameters: $beta1 = 0.9$, $beta2 = 0.999$, $epsilon = 1e - 8$. The AP-GAN model is trained for 10 epochs and the learning rate set to is 0.001 for the generator and 0.004 for the discriminator. ϵ in Eq. 1 is set to 0.5, and the margin α in attack loss is set to 1. We set batch size to 32, $\lambda_a = 8$, $\lambda_{tv} = 10$ and $\lambda_c = 1$ for image retrieval; and batch size 256, $\lambda_a = 8$, $\lambda_{tv} = 10$ and $\lambda_c = 1$ for person ReID and vehicle ReID.

The adversarial patch can be of any shape and size, placed anywhere in the image. For convenience, in the experiments, we place it in the center of the picture. To verify that the adversarial patch can adapt to any shape, we set the adversarial patch to:

- a square shape with 200 pixels side length for image retrieval;
- a circle shape with 60 pixels diameter for person ReID;
- a triangle shape with 80 pixels side length for vehicle ReID.

In section 4.4 we will discuss the impact of each component, including the size and position of the adversarial patch and the weights used in loss functions.

4.3 Adversarial attacks results

4.3.1 Results for image retrieval

For the image retrieval task, we evaluate the attacking performance of AP-GAN on two target retrieval networks (VGG16 and ResNet101) with two feature aggregate functions (MAC, GeM). **need citations here** We performed attacks on four target image retrieval networks: VGG-MAC, VGG-GeM, ResNet-MAC, ResNet-GeM. The target image retrieval models are provided by [30],[2] including the public released code and pre-trained models. For Oxford5K and Paris6K datasets, the image size is usually 1024 × 768 for horizontal images and 768 × 1024 for vertical images. We set the adversarial patches to a square of 200 × 200 pixels, accounting for only 5.09% of the total pixels of the image.

Table 1 summarizes the attacking results of AP-GAN on Oxford5K and Paris6K. The name ‘original’ represents the case with no adversarial attack. We observe that, on both Oxford5K and Paris6K datasets, AP-GAN has achieved superior attacking performance against the four image retrieval models. For example, on the Oxford5K dataset, the original mAP of VGG-MAC, VGG-GeM, ResNet-MAC, ResNet-GeM are 0.818, 0.849, 0.769 and 0.862, respectively. Then, the mAP drops drastically to 0.149, 0.289, 0.413 and 0.276 after the attack, indicating that the four image retrieval systems are completely disrupted by the AP-GAN. From the accuracy at Rank-[1, 5, 10] evaluation metrics, we can see similar results where AP-GAN leads to an drastic drop in retrieval accuracy and renders the retrieval completely a failure.

We compare our proposed AP-GAN with recent released retrieval-based UAP [18], and the results are shown in Table 1. It is evident that, apart from the ResNet-MAC model, AP-GAN consistently outperforms retrieval-based UAP on the other target image retrieval models and all the datasets. For retrieval-based UAP in the ResNet-MAC model, AP-GAN also obtains close results to the winning model. It is particularly important to note that AP-GAN only modifies about 5% of the pixels in the image, but the retrieval-based UAP affects all pixels in the image, which makes UAP unviable as an attack approach from the physical world for Smart City applications. Figure 3 illustrates a set of examples of adversarial patch attack results on VGG16-MAC models on Oxford5K and Paris6K datasets. As the red

[2] http://cmp.felk.cvut.cz/cnnimageretrieval/

Table 1 Detailed experimental results of adversarial patch attack on image retrieval.

		Oxford5K				Paris6K			
Target Network	Attack Method	mAP	rank1	rank5	rank10	mAP	rank1	rank5	rank10
VGG-MAC	original	0.818	1	1	1	0.788	1	1	1
	UAP	0.341	-	-	-	0.239	-	-	-
	AP-GAN	0.149	0.163	0.200	0.200	0.126	0.009	0.200	0.218
VGG-GeM	original	0.849	0.981	1	1	0.860	1	1	1
	UAP	0.409	-	-	-	0.361	-	-	-
	AP-GAN	0.289	0.418	0.509	0.581	0.267	0.309	0.436	0.527
ResNet-MAC	original	0.769	0.963	0.981	1	0.852	1	1	1
	UAP	0.317	-	-	-	0.337	-	-	-
	AP-GAN	0.413	0.581	0.709	0.745	0.374	0.436	0.690	0.745
ResNet-GeM	original	0.862	1	1	1	0.907	1	1	1
	UAP	0.329	-	-	-	0.265	-	-	-
	AP-GAN	0.276	0.290	0.545	0.672	0.295	0.254	0.454	0.527

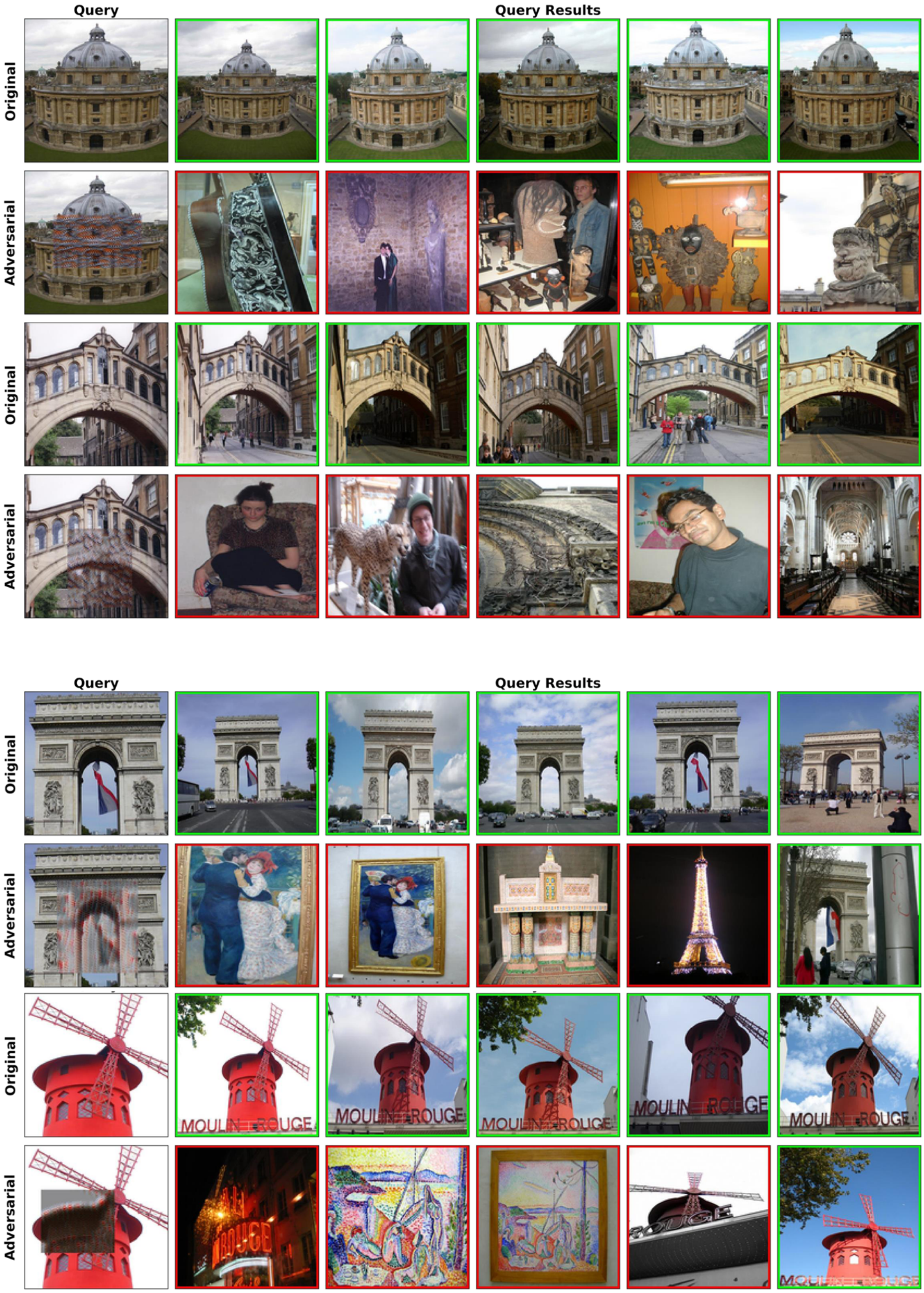


Fig. 3 Examples of adversarial attack results on Oxford5K(top 4 rows) and Paris6K(bottom 4 rows). Each group contains two rows of images, the first row is the original retrieval results, and the second row is the image retrieval results after being attacked by AP-GAN

borders indicate, AP-GAN's adversarial patches successfully cripple the image retrieval system and have it return completely irrelevant images. Observing the adversarial samples, we can find that the adversarial patch is located in the middle of the pictures as designated and has a good fusion with the surrounding area.

4.3.2 Results for person ReID

For person ReID, we study two typical networks: ResNet50 [57][3] and MGN [44].[4] Resnet50 is a classic method based on global features, and MGN is based on multi-scale features, using both global and local features. We carry out detailed experiments on Market1501 and DukeMTMC-ReID datasets.

Table 2 reports the quantitative results recorded at ranks 1, 5, and 10 and mAP on Market1501 and DukeMTMC-ReID datasets. We can see that AP-GAN greatly reduces the effectiveness of the MGN model and the ResNet50 model on both datasets. From empirical results, we confirm that the person ReID systems are also very vulnerable to adversarial patches generated by AP-GAN. Comparing the attack results of ResNet50 and MGN, it can be seen that the attack results on MGN are slightly worse. In other words, the ReID system based on the MGN model is more robust to attacks. Analysis show that it could be that the patch only exists in some areas of the original image (usually only 5%), while the other areas are still unmodified images. Therefore, the methods, such as MGN, that use local features will be less affected than the methods based on global features.

We quantitatively compare the performance of the proposed method with the previous state-of-the-art adversarial attack methods on the classification task, including FGSM [9], BIM [16], DeepFool [24]. Table 2 shows substantial performance advantage of AP-GAN over all state-of-the-arts with significant leads in both Rank-1 and mAP. The experimental results reflect that the adversarial sample model proposed for image classification is not suitable for image retrieval tasks. And AP-GAN has achieved very competitive empirical results, by tampering with the similarity relationship of the images, through the proposed loss functions.

Adversarial attack results on Market1501 and DukeMTMC-ReID datasets are presented in Fig. 4. Again, red borders indicate incorrect images retrieved and green indicates correct ones. The attack's effect is very obvious, with almost all the nearest neighbor images returned being false matches. Looking closely at the images of the adversarial examples, e.g. the first image in the bottom row, we can observe the unobtrusive circular adversarial patch in the middle of the person. It can be a sticker physically on the person if we aim to perform attacks on suivallance systems through live camera feeds.

4.3.3 Results for vehicle ReID

For Vehicle ReID, we tested the attacking performance of AP-GAN on a competitive existing model: open-VehicleReID,[5] which achieved the state-of-the-art performance on VeRi776 [20] dataset. We report the experimental results of open-VehicleReID model before and after the adversarial patch attack.

[3] https://github.com/layumi/Person_reID_baseline_pytorch/

[4] https://github.com/seathiefwang/MGN-pytorch/

[5] https://github.com/BravoLu/open-VehicleReID

Table 2 Detailed experimental results of adversarial attack on person re-identification

Target Network	Attack Method	Market1501				DukeMTMC-ReID			
		mAP	rank1	rank5	rank10	mAP	rank1	rank5	rank10
ResNet50	original	0.722	0.895	0.955	0.971	0.650	0.809	0.907	0.934
	FGSM	0.598	0.818	0.918	0.946	0.335	0.534	0.685	0.744
	BIM	0.597	0.817	0.918	0.945	0.232	0.386	0.560	0.630
	DeepFool	0.597	0.818	0.919	0.945	0.231	0.382	0.557	0.630
	AP-GAN	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501
MGN	original	0.870	0.945	0.983	0.989	0.874	0.897	0.942	0.953
	FGSM	0.734	0.851	0.936	0.972	0.618	0.669	0.787	0.832
	BIM	0.725	0.850	0.922	0.969	0.665	0.587	0.720	0.805
	DeepFool	0.708	0.842	0.931	0.973	0.706	0.734	0.798	0.825
	AP-GAN	0.298	0.340	0.459	0.528	0.313	0.327	0.471	0.559

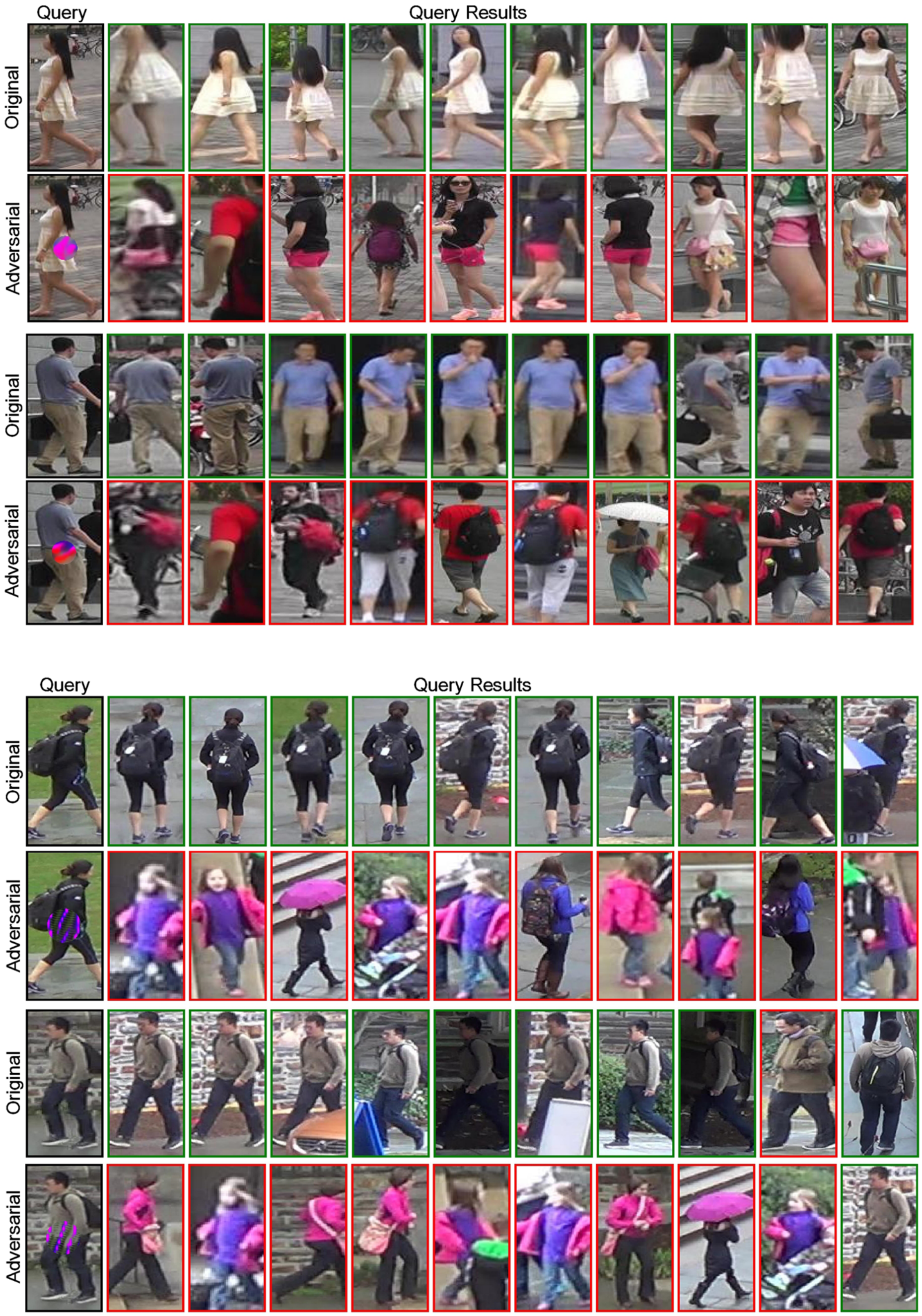


Fig. 4 Examples of attack results on Market1501 (top) and DukeMTMC-ReID (bottom) datasets. The green and red borders denote correct and incorrect results, respectively

The attack method named ‘original’ represents the situation with no adversarial attack. As can be observed from Table. 3, the mAP, accuracy at rank-1, 5, and 10 have dropped down to a very low level. Even the Rank-10 accuracy is below 0.4, which means that we can hardly find the correct vehicle image from the top 10 results. The results demonstrate that the vehicle search model has almost no defense ability against the adversarial attacks from AP-GAN. Some of the adversarial attack results are presented in Fig. 5.

4.4 Ablation study

To verify the effectiveness of each individual component in AP-GAN, we conduct ablation experiments on the Market-1501 dataset for the person ReID task. First, to investigate the contribution of each part in the overall loss function, we compare the performance of AP-GAN by deliberately removing certain parts of the loss. Then, we compare performance of AP-GAN by setting the hyperparameters at different values. At last, we report the performance of our approach when setting the different sizes to adversarial patches.

4.4.1 Different loss functions

To optimize the AP-GAN, we use 4 kinds of loss in the final loss function, they are: content loss, GAN Loss, attack loss, and TV Loss. We use ‘no-content’, ‘no-attack’ and ‘no-tv’ to represent the AP-GAN without content loss, attack loss, and TV Loss, respectively. The named ‘original’ represents the results with no adversarial attack. The results are reported in Table 4.

We found empirically that the ‘no-attack’ model has little impact on reducing the retrieval performance of resnet50 on the two datasets. The ‘no-content’ and ‘no-tv’ get better attack results than complete AP-GAN. Such results confirm our expectations when designing the loss function. That is, the attack effect is controlled by attack loss, and the quality of the generated adversarial examples are controlled by content loss and TV loss.

4.4.2 Different weights of loss function

In order to further clarify the impact of the weights of different loss functions on the model, we compared the performance of AP-GAN when different hyperparameters were adopted. In Table 5, we present the mAP and accuracy at rank-[1, 5, 10] on Market1501 and DukeMTMC-ReID varying different weights. We also visualize the comparison results in Fig. 6, to show off how weight parameters influence the attacking performance of AP-GAN.

We can draw the following conclusions from Table 5 and Fig. 6: The weight parameters λ_c and are used to control the difference between origin images and adversarial examples, and λ_{tv} is used to make the generated adversarial patch looks smooth and coordinated with the surrounding. Therefore, increasing these two parameters will improve the visual quality

Table 3 Detailed experimental results of adversarial attack on vehicle re-identification

		VeRi776			
Target network	Attack method	mAP	rank1	rank5	rank10
open-VehicleReID	original	0.747	0.948	0.983	0.992
	AP-GAN	0.117	0.156	0.278	0.351

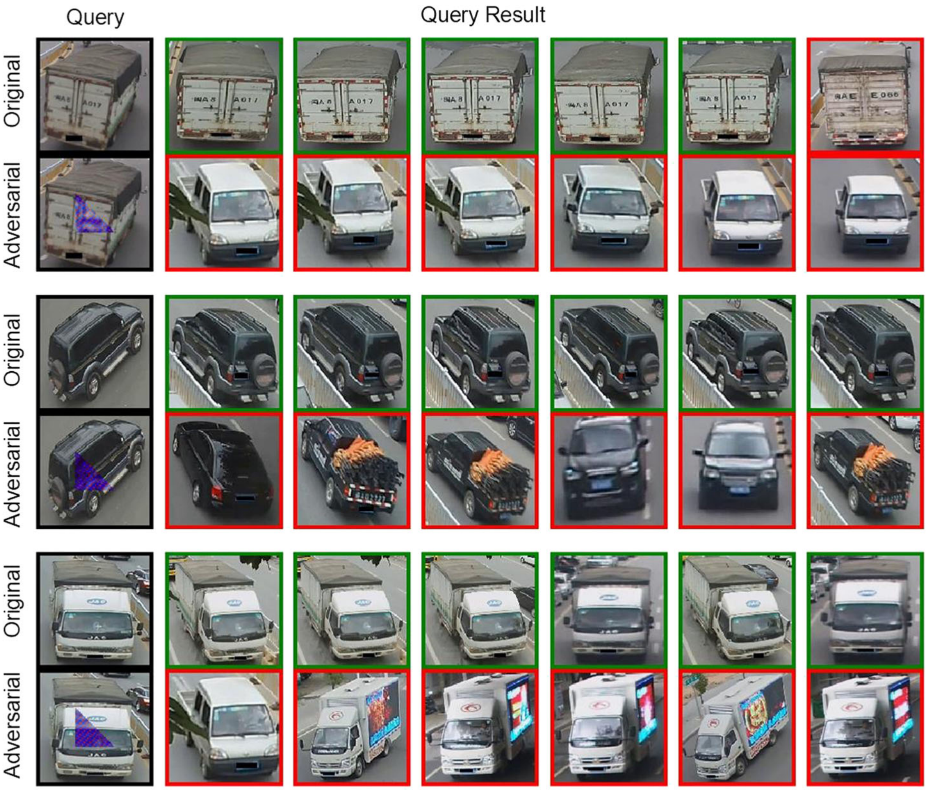


Fig. 5 Examples of attack results on open-VehicleReID on VeRi776 dataset. Green box for correct samples and red for incorrect samples

of the adversarial samples, but will result in a decrease in the performance of the adversarial patch attack at the same time. The parameter λ_a is used to control the attacking performance of AP-GAN, and the experimental results show that the larger value of λ_a, the better the attacking performance.

4.4.3 Different patch sizes

Another factor that significantly affects the effectiveness of adversarial examples is the size of the adversarial patch. The patch size determines how many pixels AP-GAN can modify in an image. In general, the more pixels is modified, the more significant the attack result is. But large patch sizes will cause a decrease in image quality, and lead to greater difficulties to perform attack in the physical world. The obtained detailed results are shown in Table 6, as we set the shape of adversarial patches to be circle shape for person ReID, the value of patch size represents the diameter of the adversarial patches.

4.5 Robustness of attack

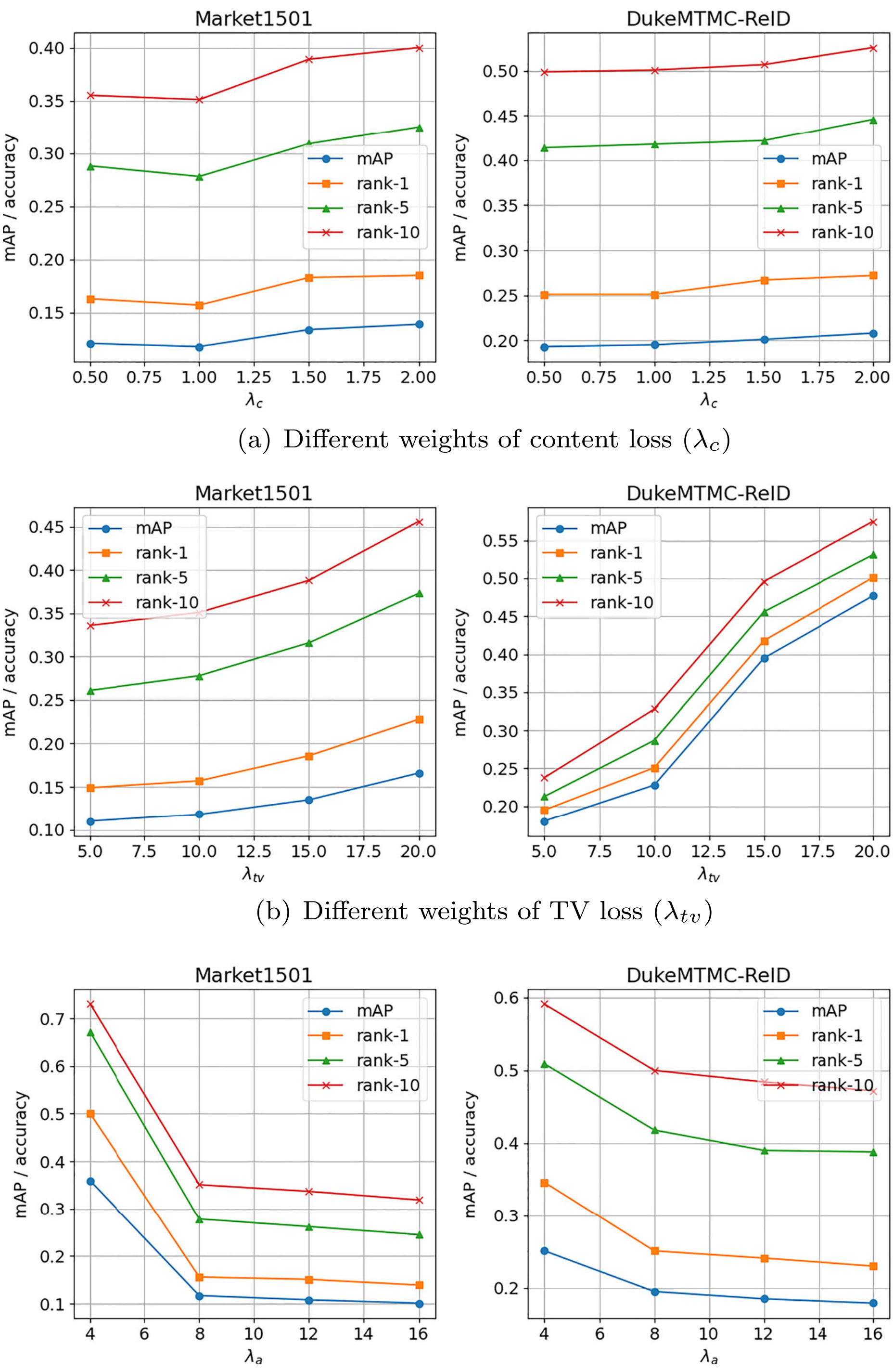
The goal of AP-GAN is to be able to attack various image retrieval systems in the real physical world. When considering the adversarial attack in the physical world, it is very

Table 4 Ablation study on person ReID with different loss functions.

		Market1501				DukeMTMC-ReID			
Target network	Attack method	mAP	rank1	rank5	rank10	mAP	rank1	rank5	rank10
ResNet50	original	0.722	0.895	0.955	0.971	0.650	0.809	0.907	0.934
	no-content	0.110	0.157	0.266	0.341	0.191	0.251	0.409	0.495
	no-tv	0.102	0.133	0.243	0.316	0.161	0.192	0.359	0.445
	no-attack	0.628	0.811	0.916	0.946	0.621	0.789	0.900	0.931
	AP-GAN	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501

Table 5 Ablation study on person ReID with different weights of the loss function

		Market1501				DukeMTMC-ReID			
Parameters	Values	mAP	rank1	rank5	rank10	mAP	rank1	rank5	rank10
λ_c	0.5	0.121	0.163	0.288	0.355	0.193	0.251	0.414	0.499
	1	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501
	1.5	0.133	0.182	0.309	0.388	0.201	0.266	0.422	0.507
	2	0.139	0.185	0.325	0.400	0.208	0.272	0.446	0.526
λ_{tv}	5	0.109	0.149	0.260	0.336	0.180	0.228	0.394	0.476
	10	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501
	15	0.134	0.185	0.315	0.394	0.213	0.286	0.455	0.530
	20	0.166	0.228	0.372	0.455	0.238	0.327	0.495	0.574
λ_a	4	0.359	0.500	0.671	0.731	0.251	0.346	0.509	0.591
	8	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501
	12	0.109	0.151	0.263	0.338	0.186	0.242	0.391	0.484
	16	0.101	0.139	0.245	0.319	0.179	0.230	0.388	0.472



(a) Different weights of content loss (λ_c)

(b) Different weights of TV loss (λ_{tv})

(c) Different weights of attack loss (λ_a)

Fig. 6 Results of attacking the ResNet50 model, when setting the weights of the loss function to different values

different from the digital world. By introducing the adversarial patch methods, we avoid the need to consider and modify each pixel. Adversarial patch enables a potential risk in the physical world, that is, successful attacks can be achieved by adding a specific sticker on the pedestrian's clothes, the vehicle's surface, and the building walls, to work against the retrieval-based systems.

There are two other important issues that need to be resolved if such attacks could be viable in the physical world:

- First, in the physical world, objects are always moving, which will cause changes in the position and shape of the adversarial patches.
- Second, usually in the physical world, the attacked target models are not accessible.

For the first problem, we simulate the situation of morphological changes in the real world by applying affine transformations to the adversarial patches. For the second problem, we will verify the transferability of AP-GAN across different models, to verify the attacking performance of an AP-GAN trained by a retrieval model against the other models.

4.5.1 Affine transformation

In the real world, the angle and distance between the object and the camera may change. For example, if the target is further away from the camera, the adversarial patches on the target would be smaller. In view of different possibilities in the physical world, we deal with the adversarial patches by selecting four types of affine transformation, namely translation, scale, flip, and rotation, to verify the attack ability of the adversarial patches when encountering a morphological change. In Table 7, we show the results under different types of an affine transformation. For example, we randomly select a rotate angle between −180° and 180° for each patch. Then we take the transformed adversarial patches as new adversarial patches to test the adversarial effect.

From the Experimental results, we can find that despite making various affine transformations on adversarial patches, AP-GAN still maintains effective attack results. In addition, it can be seen that the degree of influence of different types of transformations is different. After the flip and the rotation transformations, the adversarial patches still maintain good attack performance. However, the translation and the scale transformation greatly weakened the attack ability of the adversarial patches.

4.5.2 Transferability

The transferability of adversarial attack methods means that the adversarial patch generated for one target model will also mislead the other models. We evaluate the performance of transfer attacks by cross-evaluating the performance of AP-GAN trained for different target image retrieval networks.

Quantitative results are summarized in Table 8. The first column represents the target model used to train the AP-GAN and the first row indicates the retrieval performance of the original model without attack. Each subsequent row represents the retrieval performance of these target retrieval models after being attacked by AP-GAN. Experiments demonstrate that transferability exists commonly between different image retrieval models. In addition, it can be seen that the transferability is more obvious between retrieval models based on the same backbone networks, and weakened between models with different backbone network structures. For example, the attacking model training for VGG-MAC has the similar attacking performance to VGG-GeM, but has the lower attacking performance to ResNet-MAC

Table 6 Ablation study on person ReID with different patch sizes

		Market1501				DukeMTMC-ReID			
Parameters	Values	mAP	rank1	rank5	rank10	mAP	rank1	rank5	rank10
Patch Size	20	0.607	0.784	0.899	0.931	0.554	0.720	0.851	0.888
	40	0.313	0.438	0.614	0.687	0.439	0.606	0.759	0.813
	60	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501
	80	0.071	0.087	0.169	0.217	0.056	0.040	0.098	0.153
	100	0.017	0.015	0.027	0.035	0.022	0.009	0.031	0.048

Table 7 The results of AP-GAN on person ReID with different types of affine transformation

	Market1501				DukeMTMC-ReID			
Transformation	mAP	rank1	rank5	rank10	mAP	rank1	rank5	rank10
original	0.722	0.895	0.955	0.971	0.650	0.809	0.907	0.934
AP-GAN	0.117	0.156	0.278	0.351	0.195	0.251	0.418	0.501
translation	0.505	0.674	0.810	0.853	0.339	0.463	0.584	0.646
scale	0.279	0.363	0.448	0.532	0.615	0.796	0.925	0.925
flip	0.299	0.411	0.573	0.641	0.202	0.269	0.438	0.514
rotation	0.287	0.387	0.550	0.628	0.259	0.366	0.524	0.597

Table 8 Transfer attacking results for image retrieval models on Paris6K

Attack target / Trained with	VGG-MAC	VGG-GeM	Res-MAC	Res-GeM
no attack	0.788	0.860	0.852	0.907
VGG-MAC	0.126	0.295	0.438	0.424
VGG-GeM	0.180	0.267	0.399	0.377
Res-MAC	0.562	0.439	0.374	0.378
Res-GeM	0.473	0.436	0.382	0.295

and ResNet-GeM. The transferability of AP-GAN shows that even if the target retrieval network is completely unknown, by training models on other networks, certain attacking performance can be achieved on unknown models. It indicates potential of black-box attacks in the physical world.

5 Conclusion

In this article, we propose a novel AP-GAN model, which can effectively generate adversarial patch to attack the image retrieval systems, and making them return irrelevant results. Our method demonstrates a viable method to perform adversarial attacks on image retrieval-based Smart City applications in the physical world, where it is impossible to arbitrarily modify the values of pixels in the image. AP-GAN is trained in a self-supervised way, using only a few unlabeled images. Once trained, it is able to produce effectively image-specific adversarial patches for any input image. Furthermore, AP-GAN is a semi-white box attack method because it does not need to access the target network during inference stage. And it has demonstrated a significant level of transferability across backbone models for the retrieval systems. We conduct extensive experiments on several widely used benchmark datasets, demonstrating that AP-GAN is able to effectively cripple the performance of various retrieval systems. The experiment results also show that the adversarial patch generated by AP-GAN has considerable transferable attack capabilities across different target networks, and is robustness to affine transformations.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 61602487, 61832017), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (Grant No. 2015030275, 2018030202).

References

1. Akhtar N, Liu J, Mian A (2018) Defense against universal adversarial perturbations. In: CVPR, pp 3389–3398
2. Babenko A, Lempitsky V (2015) Aggregating local deep features for image retrieval. In: ICCV, pp 1269–1277
3. Brown TB, Mané D, Roy A, Abadi M, Gilmer J (2017) Adversarial patch. CoRR http://arxiv.org/abs/1712.09665
4. Chen C, Seff A, Kornhauser A, Xiao J (2015) Deepdriving: Learning affordance for direct perception in autonomous driving. In: ICCV, pp 2722–2730

5. Chen L, Shang S (2019) Region-based message exploration over spatio-temporal data streams. AAAI, vol 33, pp 873–880
6. Cheng D, Gong Y, Zhou S, Wang J, Zheng N (2016) Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In: CVPR, pp 1335–1344
7. Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, Prakash A, Kohno T, Song D (2018) Robust physical-world attacks on deep learning visual classification. In: CVPR, pp 1625–1634
8. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS, pp 2672–2680
9. Goodfellow I, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: ICLR
10. Gordo A, Almazan J, Revaud J, Larlus D (2017) End-to-end learning of deep visual representations for image retrieval. IJCV 124(2):237–254
11. Han P, Yang P, Zhao P, Shang S, Liu Y, Zhou J, Gao X, Kalnis P (2019) Gcn-mf: Disease-gene association identification by graph convolutional networks and matrix factorization. In: SIGKDD, pp 705–713
12. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: ECCV, pp 630–645
13. Huang R, Zhang S, Li T, He R (2017) Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In: ICCV, pp 2439–2448
14. Kalantidis Y, Mellina C, Osindero S (2016) Cross-dimensional weighting for aggregated deep convolutional features. In: ECCV, pp 685–701
15. Kupyn O, Budzan V, Mykhailych M, Mishkin D, Matas J (2018) Deblurgan: Blind motion deblurring using conditional adversarial networks. In: CVPR
16. Kurakin A, Goodfellow I, Bengio S (2017) Adversarial machine learning at scale. In: ICLR
17. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR, pp 4681–4690
18. Li J, Ji R, Liu H, Hong X, Gao Y, Tian Q (2019) Universal perturbation attack against image retrieval. In: ICCV, pp 4899–4908
19. Liu A, Liu X, Fan J, Ma Y, Zhang A, Xie H, Tao D (2019a) Perceptual-sensitive gan for generating adversarial patches. In: AAAI, vol 33, pp 1028–1035
20. Liu X, Liu W, Ma H, Fu H (2016) Large-scale vehicle re-identification in urban surveillance videos. In: ICME. IEEE, pp 1–6
21. Liu X, Yang H, Liu Z, Song L, Chen Y, Li H (2019b) DPATCH: an adversarial patch attack on object detectors. In: Workshop AAAI, vol 2301
22. Liu Z, Zhao Z, Larson M (2019c) Who's afraid of adversarial queries?: The impact of image modifications on content-based image retrieval. In: ICMR. ACM, pp 306–314
23. Mao X, Li Q, Xie H, Lau RY, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. In: ICCV, pp 2794–2802
24. Moosavi-Dezfooli SM, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR, pp 2574–2582
25. Moosavi-Dezfooli SM, Fawzi A, Fawzi O, Frossard P (2017) Universal adversarial perturbations. In: CVPR, pp 1765–1773
26. Noh H, Araujo A, Sim J, Weyand T, Han B (2017) Large-scale image retrieval with attentive deep local features. In: ICCV, pp 3456–3465
27. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: CVPR, pp 1–8
28. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2008) Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR
29. Radenović F, Tolias G, Chum O (2016) Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In: ECCV, pp 3–20
30. Radenović F, Tolias G, Chum O (2018) Fine-tuning cnn image retrieval with no human annotation. TPAMI 41(7):1655–1668
31. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR
32. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS, pp 91–99
33. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: CVPR, pp 815–823
34. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV, pp 618–626

35. Shang S, Zhu S, Guo D, Lu M (2017) Discovery of probabilistic nearest neighbors in traffic-aware spatial networks. WWW 20(5):1135–1151
36. Simo-Serra E, Trulls E, Ferraz L, Kokkinos I, Fua P, Moreno-Noguer F (2015) Discriminative learning of deep convolutional feature point descriptors. In: ICCV, pp 118–126
37. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. ICLR
38. Subramanya A, Pillai V, Pirsiavash H (2019) Fooling network interpretation in image classification. In: ICCV, pp 2020–2029
39. Sun Y, Zheng L, Yang Y, Tian Q, Wang S (2018) Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In: ECCV, pp 480–496
40. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: ICLR
41. Thys S, Van Ranst W, Goedemé T (2019) Fooling automated surveillance cameras: adversarial patches to attack person detection. In: CVPR Workshops, pp 0–0
42. Tolias G, Sicre R, Jégou H (2016) Particular object retrieval with integral max-pooling of cnn activations. ICLR
43. Tolias G, Radenovic F, Chum O (2019) Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In: Proceedings of the IEEE International Conference on Computer Vision, pp 5037–5046
44. Wang G, Yuan Y, Chen X, Li J, Zhou X (2018a) Learning discriminative features with multiple granularities for person re-identification. In: MM. ACM, pp 274–282
45. Wang H, Yang YY, Pan Y, Han P, Li ZX, Huang HG, Zhu SZ (2020) Detecting thoracic diseases via representation learning with adaptive sampling. Neurocomputing
46. Wang J, Song Y, Leung T, Rosenberg C, Wang J, Philbin J, Chen B, Wu Y (2014) Learning fine-grained image similarity with deep ranking. In: CVPR
47. Wang Y, Chen Z, Wu F, Wang G (2018b) Person re-identification with cascaded pairwise convolutions. In: CVPR, pp 1470–1478
48. Wei L, Zhang S, Gao W, Tian Q (2018) Person transfer gan to bridge domain gap for person re-identification. In: CVPR, pp 79–88
49. Xiao C, Li B, yan Zhu J, He W, Liu M, Song D (2018) Generating adversarial examples with adversarial networks. In: IJCAI-18, pp 3905–3911
50. Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A (2017) Adversarial examples for semantic segmentation and object detection. In: ICCV, pp 1369–1378
51. Xu Y, Wu B, Shen F, Fan Y, Zhang Y, Shen HT, Liu W (2019) Exact adversarial attack to image captioning via structured output learning with latent variables. In: CVPR, pp 4135–4144
52. Yuan X, He P, Zhu Q, Li X (2019) Adversarial examples: Attacks and defenses for deep learning. IEEE Trans Neural Netw Learn Syst 30(9):2805–2824
53. Zhang K, Ni J, Yang K, Liang X, Ren J, Shen XS (2017) Security and privacy in smart city applications: Challenges and solutions. IEEE Commun Mag 55(1):122–129
54. Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In: CVPR, pp 589–597
55. Zhang Y, Foroosh H, David P, Gong B (2019) CAMOU: Learning Physical vehicle camouflages to adversarially attack detectors in the wild. In: ICLR
56. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: a benchmark. In: ICCV
57. Zheng L, Zhang H, Sun S, Chandraker M, Yang Y, Tian Q (2017a) Person re-identification in the wild. In: CVPR, pp 1367–1376
58. Zheng Y, Capra L, Wolfson O, Yang H (2014) Urban computing: concepts, methodologies, and applications. ACM Trans Intell Syst Technol (TIST) 5(3):1–55
59. Zheng Z, Zheng L, Yang Y (2017b) Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In: ICCV. IEEE

Guoping Zhao is currently working toward the PhD degree in the School of Information, Renmin University of China. He received the M.S. in 2012 from the Department of Computer Science at the Beihang University, and obtained his B.S. form Northwestern Polytechnical University. He was a research assistant at The Second Research Institute of China Aerospace Science and Industry Corporation, from 2012 to 2015. His research interests include computer vision, deep learning, multimedia retrieval.

Mingyu Zhang received his B.S. from the Beijing Jiaotong University. He is working toward the graduate degree from the School of Information, Renmin University of China. His research interests include deep learning and multimedia retrieval.

Jiajun Liu is currently an Associate Professor with Renmin University of China. He received his Ph.D. from The University of Queensland, Australia in 2012 and his B.Eng. from Nanjing University, China in 2006. He worked as a Researcher at IBM China Research Labs from 2006 to 2008. His research interests include multimedia retrieval, and management and mining for spatial temporal data. He has published extensively in various venues and has served as invited reviewer for multiple top conferences and journals.

Yaxian Li received the B.S. degree in computer science from Renmin University of China, in 2019. She is currently working toward the M.S. degree in computer science at Renmin University of China. Her research interests include computer vision and machine learning.

Ji-Rong Wen is a full professor at School of Information, Renmin University of China. He worked at Microsoft Research Asia for fourteen years and many of his research results have been integrated into important Microsoft products (e.g. Bing). He serves as an associate editor of ACM Transactions on Information Systems (TOIS). His main research interests include web data management, information retrieval, data mining and machine learning.