# Targeted Mismatch Adversarial Attack:
## Query with a Flower to Retrieve the Tower

Giorgos Tolias  Filip Radenovic  Ondřej Chum
Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague

## Abstract

*Access to online visual search engines implies sharing of private user content – the query images. We introduce the concept of targeted mismatch attack for deep learning based retrieval systems to generate an adversarial image to conceal the query image. The generated image looks nothing like the user intended query, but leads to identical or very similar retrieval results. Transferring attacks to fully unseen networks is challenging. We show successful attacks to partially unknown systems, by designing various loss functions for the adversarial image construction. These include loss functions, for example, for unknown global pooling operation or unknown input resolution by the retrieval system. We evaluate the attacks on standard retrieval benchmarks and compare the results retrieved with the original and adversarial image.*

## 1. Introduction

Information about users is a valuable article. Websites, service providers, and even operating systems collect and store user data. The collected data have various forms, *e.g*. visited websites, interactions between users in social networks, hardware fingerprints, keyboard typing or mouse movement patterns, *etc*. Internet search engines record what the users search for, as well as the responses, *i.e*. clicks, to the returned results.

Recent development in computer vision allowed efficient and precise large scale image search engines to be launched, such as Google Image Search. Nevertheless, similarly to text search engines, queries – the images – are stored and further analyzed by the provider[1]. In this work, we protect the user image (*target*) by constructing a novel image. The constructed image is visually dissimilar to the target, however, when used as a query, identical results are retrieved as with the target image. Large-scale search methods require short-code image representation, both for storage minimization and for search efficiency, which are usually extracted with Convolutional Neural Networks (CNN). We formulate the problem as an adversarial attack on CNNs.

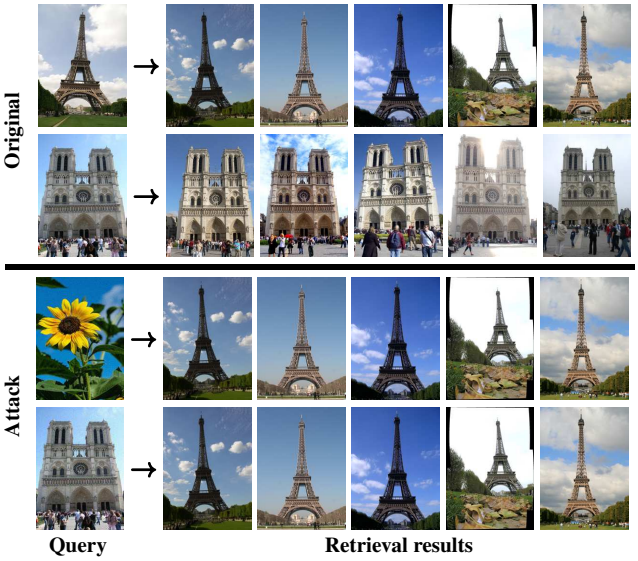


Figure 1. Top two rows show retrieval results to the user query image (target). Bottom two rows show the results of our attack where a carrier image (flower, Notre Dame) is perturbed to have identical descriptor to that of the target in the first row. Identical results are obtained without disclosing the target.

Adversarial attacks, as introduced by Szegedy *et al*. [35], study *imperceptible* non-random image perturbations to mislead a neural network. The first attacks were introduced and tested on image classification. In that context, adversarial attacks are divided into two categories, namely *non-targeted* and *targeted*. The goal of non-targeted attacks is to change the prediction of a test image to an arbitrary class [25, 24], while targeted attacks attempt to make a specific change of the network prediction, *i.e*. to misclassify the test image to a predefined target class [35, 7, 10].

Similarly to image classification, adversarial attacks have been proposed in the domain of image retrieval too. An non-targeted attack attempts to generate an image that for a human observer carries the same visual information, while for the neural network it appears dissimilar to other images of the same object [19, 20, 37]. This way, a user protects personal images and does not allow them to be in-

[1] Google Search Help: "The pictures you upload in your search may be stored by Google for 7 days. They won't be a part of your search history, and we'll only use them during that time to make our products and services better."

dexed for content-based search, even when the images are publicly available. In this paper, we address targeted attacks aiming to retrieve images that are related to a hidden target query without explicitly revealing the image (see Figure 1). Example applications include users checking whether their copyrighted image, or personal photo with sensitive content, *etc*. is indexed, *i.e*. used by anyone else, without providing the query image itself. Such case of privacy protection is an example of "legal" motivation. A concept that bears resemblance to ours exists in the speech recognition, but in a malicious context. Carlini *et al*. [6] generate *hidden voice commands* that are imperceivable to human listeners but are interpreted as commands by devices. We investigate adversarial attacks beyond the white-box scenario, in which all the parameters and design choices of the retrieval system are known. Specifically, we analyze the cases of unknown indexing image resolution and unknown global pooling used in the network.

## 2. Related work

**Adversarial attacks on image classification** were introduced by Szegedy *et al*. [35]. Follow up approaches are categorized to *white-box* attacks [35, 12] if there is complete knowledge of the model or to *black-box* [28, 29] otherwise. Adversarial images are generated by various methods in the literature, such as optimization-based approaches using box-constrained L-BFGS optimizer [35], gradient descent with change of variable [7]. A fast gradient sign method [12] and variants [18, 10] are designed to be *fast* rather than optimal, while DeepFool [25] analytically derives an optimal solution method by assuming that neural networks are totally linear. All these approaches solve an optimization problem given a test image and its associated class in the case of non-targeted attacks or a test image and a target class in the case of targeted attacks. A universal non-targeted approach is proposed by Moosavi *et al*. [24], where an image-agnostic Universal Adversarial Perturbation (UAP) is computed and applied to unseen images to cause network misclassification.

**Adversarial attacks on image retrieval** are studied by recent work [19, 20, 37] in a non-targeted scenario for CNN-based approaches. Liu *et al*. [20] and Zheng *et al*. [37] adopt the optimization-based approach [35], while Li *et al*. [19] adopt the UAP [24]. Similar attacks on classical retrieval systems that are based on SIFT local descriptors [21] have been addressed in an earlier line of work by Do *et al*. [9, 8]. To the best of our knowledge, no existing work focuses on targeted adversarial attacks for image retrieval. Targeted attacks for nearest neighbors in high dimensional spaces are studied by Amsaleg *et al*. [2], where they directly perturb the high dimensional vectors and show that the high local intrinsic dimensionality results in high vulnerability.

## 3. Background

We provide the background for non-targeted and targeted adversarial attacks in the domain of image classification, then detail the basic components of CNN-based image retrieval approaches, and finally discuss non-targeted attacks for image retrieval. All variants presented in this section assume white-box access to the network classifier for classification or the feature extractor network for retrieval.

### 3.1. Image classification attacks

We denote the initial RGB image, called the *carrier image*, by tensor $\mathbf{x}_c \in [0,1]^{W\times H\times 3}$, and its associated label by $y_c \in \{1 \dots K\}$. A CNN trained for $K$-way classification, denoted by function $f : \mathbb{R}^{W\times H\times 3} \rightarrow \mathbb{R}^K$, produces vector $f(\mathbf{x}_c)$ comprising class confidence values. Adversarial attack methods for classification typically study the case of images with correct class prediction, *i.e*. $\arg\max_i f(\mathbf{x}_c)_i$ is equal to $y_c$, where $f(\mathbf{x}_c)_i$ is the $i$-th dimension of vector $f(\mathbf{x}_c)$. An adversary aims at generating *adversarial image* $\mathbf{x}_a$ that is visually similar to the carrier image but is classified incorrectly by $f$. The goal of the attack can vary [1] and corresponds to different loss functions optimizing $\mathbf{x} \in [0,1]^{W\times H\times 3}$.

**Non-targeted misclassification** is achieved by reducing the confidence for class $y_c$, while increasing for all other classes. It is achieved by minimizing loss function

$$L_{\text{nc}}(\mathbf{x}_c, y_c; \mathbf{x}) = -\ell_{\text{ce}}(f(\mathbf{x}), y_c) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \quad (1)$$

Function $\ell_{\text{ce}}(f(\mathbf{x}), y_c)$ is the cross-entropy loss which is maximized to achieve misclassification. In this way, misclassification is performed to any wrong class. Term $||\mathbf{x} - \mathbf{x}_c||^2$ is called *carrier distortion* or simply *distortion* and is the squared $l_2$ norm of the perturbation vector $\mathbf{r} = \mathbf{x} - \mathbf{x}_c$. Other norms, such as $l_\infty$, are also applicable [7].

**Targeted misclassification** has the goal of generating an adversarial image that gets classified into target class $y_t$. It is achieved by minimizing loss function

$$L_{\text{tc}}(\mathbf{x}_c, y_t; \mathbf{x}) = \ell_{\text{ce}}(f(\mathbf{x}), y_t) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \quad (2)$$

In contrast to (1), cross-entropy loss is minimized w.r.t. the target class instead of maximized w.r.t. the carrier class.

**Optimization** of (1) or (2) generates the adversarial images given by

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{nc}}(\mathbf{x}_c, y_c; \mathbf{x}), \quad (3)$$

or

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{tc}}(\mathbf{x}_c, y_t; \mathbf{x}), \quad (4)$$

respectively. In the literature [35, 7], various optimizers such as Adam [16], or L-BFGS [5] are used. The box constraints, *i.e*. $\mathbf{x} \in [0,1]^{W\times H\times 3}$, are ensured by projected

gradient descent, clipped gradient descent, change of variables [7], or optimization algorithms that support box constraints such as L-BFGS. It is a common practice to perform line search for weight $\lambda > 0$ and keep the attack of minimum distortion. The optimization is initialized by the carrier image.

### 3.2. Image retrieval components

This work focuses on attacks on CNN-based image retrieval with global image descriptors. An image is mapped to a high dimensional descriptor by a CNN with a global pooling layer. The descriptor is consequently normalized to have unit $l_2$ norm. Then, retrieval from a large dataset w.r.t. a *query image* reduces to nearest neighbor search via inner product evaluation between the query descriptor and dataset descriptors. The model for descriptor extraction consists of the following components or parameters.

*Image resolution:* The input image $\mathbf{x}$ is re-sampled to image $\mathbf{x}^s$ to have the largest dimension equal to $s$.

*Feature extraction:* Image $\mathbf{x}^s$ is fed as an input to a Fully Convolutional Network (FCN), denoted by function $g : \mathbb{R}^{W\times H\times 3} \to \mathbb{R}^{w\times h\times d}$, which maps $\mathbf{x}^s$ to tensor $g(\mathbf{x}^s)$. When the image is fed at its original resolution we denote it by $g(\mathbf{x})$.

*Pooling:* A global pooling operation $h : \mathbb{R}^{w\times h\times d} \to \mathbb{R}^d$ maps the input tensor $g(\mathbf{x}^s)$ to descriptor $(h \circ g)(\mathbf{x}^s)$. We assume that $l_2$ normalization is included in this process, so that the output descriptor has unit $l_2$ norm. We consider various options for pooling, namely, max pooling (MAC) [32, 36], sum pooling (SPoC) [4], generalized mean pooling (GeM) [31], regional max pooling (R-MAC) [36], and spatially and channel-wise weighted sum pooling (CroW) [15]. The framework can be extended to multiple other variants [27, 23, 3].

*Whitening:* Descriptor post-processing is performed by function $w : \mathbb{R}^d \to \mathbb{R}^d$, which includes centering, whitening and $l_2$ re-normalization [31]. Finally, input image $\mathbf{x}^s$ is mapped to descriptor $(w \circ h \circ g)(\mathbf{x}^s)$.

For brevity we denote $\mathbf{g}_\mathbf{x} = g(\mathbf{x})$, $\mathbf{h}_\mathbf{x} = (h \circ g)(\mathbf{x})$, and $\mathbf{w}_\mathbf{x} = (w \circ h \circ g)(\mathbf{x})$. In the following, we consider an extraction model during the adversarial image optimization and another one during the testing of the retrieval/matching performance. In order to differentiate between the two cases we refer to the components of the former as *attack-model*, *attack-resolution*, *attack-FCN*, *attack-pooling* and *attack-whitening* and the latter as *test-model*, *test-resolution*, *test-FCN*, *test-pooling* and *test-whitening*.

### 3.3. Image retrieval attacks

Adversarial attacks for image retrieval are so far limited to the non-targeted case.

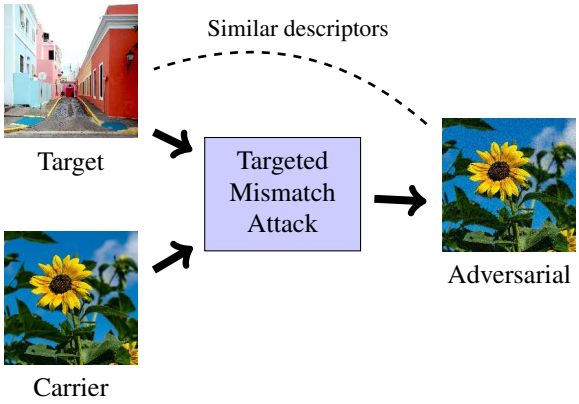


Figure 2. In targeted mismatch attacks an adversarial image is generated given a carrier and a target image. The adversarial image should match the descriptor of the target image but be visually dissimilar to the target; visual dissimilarity to the target is achieved via visual similarity to the carrier. The attack is formed by a retrieval query using the adversarial image, where the goal is to obtain identical results as with the target query while keeping the target image private.

**Non-targeted mismatch** aims at generating an adversarial image with small perturbation compared to the carrier image and descriptor that is dissimilar to that of the carrier. This is formulated by loss function

$$\begin{aligned} L_{\text{nr}}(\mathbf{x}_c;\mathbf{x}) &= \ell_{\text{nr}}(\mathbf{x},\mathbf{x}_c) &&+\lambda\,||\mathbf{x}-\mathbf{x}_c||^2 \\ &= \mathbf{h}_\mathbf{x}^\top \mathbf{h}_{\mathbf{x}_c} &&+\lambda\,||\mathbf{x}-\mathbf{x}_c||^2. \end{aligned} \tag{5}$$

The adversarial image is given by minimizer

$$\mathbf{x}_a = \arg\min_\mathbf{x} L_{\text{nr}}(\mathbf{x}_c;\mathbf{x}). \tag{6}$$

In this way, the adversary modifies images into their non-indexable counterpart. The exact formulation in (5) has not been addressed; the closest is the work of Li *et al.* [19] where they are seeking of a UAP by maximizing $l_1$ descriptor distance instead of minimizing cosine similarity.

## 4. Method

We formulate the problem of targeted mismatch attack and then propose various loss functions to address it and to construct concealed query images.

### 4.1. Problem formulation

The adversary tries to generate an adversarial image with the goal of using it as a (concealed) query for image retrieval instead of a *target image*. The goal is to obtain the same retrieval results without disclosing any information about the target image itself.

We assume a target image $\mathbf{x}_t \in \mathbb{R}^{W\times H\times 3}$ and a carrier image $\mathbf{x}_c$ with the same resolution (see Figure 2). The goal

of the adversary is to generate an adversarial image $\mathbf{x}_a$ that has high *descriptor similarity* but very low *visual similarity* to the target. Visual (human) dissimilarity is not straightforward to model; we model visual similarity w.r.t. another image, *i.e*. the carrier, instead. We refer to this problem as *targeted mismatch attack* and the corresponding loss function is given by

$$L_{\text{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}, \mathbf{x}_t) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \quad (7)$$

In Section 4.2 we propose different instantiations of the *performance loss* $\ell_{\text{tr}}$ according to the known and unknown components of the test-model.

### 4.2. Targeted mismatch attacks

In all the following, we assume a white-box access to the FCN, while the whitening is assumed unknown and is totally ignored during the optimization of the adversarial image; its impact on the attack is evaluated by adding it to the test-model. In general, if all the parameters of the test-model are known, the task is to generate an adversarial image that reproduces the descriptor of the target image. Then, nearest neighbor search will retrieve identical results as if querying with the target image. Choosing a different performance loss introduces invariance or robustness to some parameters of the attacked retrieval system, when these parameters are unknown. We list different performance loss functions used to minimize (7).

**Global descriptor.** Loss function

$$\ell_{\text{desc}}(\mathbf{x}, \mathbf{x}_t) = 1 - \mathbf{h}_{\mathbf{x}}^\top \mathbf{h}_{\mathbf{x}_t}. \quad (8)$$

is suitable when all parameters of the retrieval system are known, including the pooling, and when the image is processed by the neural network at its original resolution. Pooling function $h$ is MAC, SPoC, or GeM in our experiments.

**Activation tensor.** In this scenario, the output of the FCN should be the same for the adversarial and target image, at the original resolution. This is achieved by minimizing the mean squared difference of the two activation tensors

$$\ell_{\text{tens}}(\mathbf{x}, \mathbf{x}_t) = \frac{||\mathbf{g}_{\mathbf{x}} - \mathbf{g}_{\mathbf{x}_t}||^2}{w \cdot h \cdot d}. \quad (9)$$

Identical tensors guarantee identical descriptors computed on top of these tensors, including those where spatial information is taken into account. This covers all global or regional pooling operations, and even deep local features, *e.g*. DELF [26]. However, our experiments show that preserving the activation tensor may result in transferring the target's visual content on the adversarial image (see Figure 7). Further, the visual appearance of the target image can be partially recovered by inverting [22] the activation tensor of the adversarial image.

**Activation histogram.** Preserving channel-wise first order statistics of the activation tensor, at the original resolution, is a weaker constraint than preserving the exact activation tensor. It guarantees identical descriptors for all global pooling operations that ignore spatial information. Activation histogram loss function is defined as

$$\ell_{\text{hist}}(\mathbf{x}, \mathbf{x}_t) = \frac{1}{d} \sum_{i=1}^{d} ||u(\mathbf{g}_{\mathbf{x}}, \mathbf{b})_i - u(\mathbf{g}_{\mathbf{x}_t}, \mathbf{b})_i||, \quad (10)$$

where $u(\mathbf{g}_{\mathbf{x}}, \mathbf{b})_i$ is the histogram of activations from the $i$-th channel of $\mathbf{g}_{\mathbf{x}}$ and $\mathbf{b}$ is the vector of histogram bin centers. Histograms are created with soft assignment by an RBF kernel[2]. Compared with the tensor case, the histogram optimization does not preserve the spatial distribution, is significantly faster, and does not suffer from undesirable disclosure artifacts.

**Different image resolution.** We require an adversarial image at the original resolution of the target ($W \times H$), which when down-sampled to resolution $s$, it retrieves similar results as the target image down-sampled to the same resolution. This is achieved by loss function

$$L^s_{\text{tr}}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}^s, \mathbf{x}^s_t) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2, \quad (11)$$

where $\ell_{\text{tr}}$ can be any of the descriptor, tensor, or histogram based performance loss functions. Note that (11) is different from (7), the performance loss is computed from resampled images, while the distortion loss is still on the original images.

A common down-sampling method used in CNNs is bilinear interpolation. We have observed that different implementations of such a layer result in different descriptors. The difference is caused by the presence of high-frequencies in the high-resolution image. The adversarial perturbation tends to be high-frequency, therefore different down-sampling results may significantly alternate the result of attack. In order to reduce the sensitivity to down-sampling, we introduce high-frequency removal by Gaussian blurring in the optimization. Instead of (11), the following loss is used

$$L^{\hat{s}}_{\text{tr}}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}^{\hat{s}}, \mathbf{x}^{\hat{s}}_t) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2, \quad (12)$$

where $\mathbf{x}^{\hat{s}}$ is image $\mathbf{x}$ blurred with Gaussian kernel with $\sigma_b$ and then down-sampled. Our experiments show, that blurring plays an important role when the attack-resolution $s$ does not exactly match the test-resolution $s'$, *i.e*. $s' = s + \Delta$.

**Ensembles.** We perform the adversarial optimization for a combination of the aforementioned loss functions by minimizing their sum. Some examples follow.

[2] We use $e^{\frac{(x-b)^2}{2\sigma^2}}$, where $\sigma = 0.1$, $x$ is a scalar activation normalized by the maximum activation value of the target, and $b$ is the bin center. We uniformly sample bin centers in [0,1] with step equal to 0.05.

- The test-pooling operation is unknown but there is a set $\mathcal{P}$ of possible pooling operations. Minimization of (7) is performed for performance loss

$$\ell_{\mathcal{P}}(\mathbf{x}, \mathbf{x}_t) = \frac{\sum_{p \in \mathcal{P}} \ell_p(\mathbf{x}, \mathbf{x}_t)}{|\mathcal{P}|}. \quad (13)$$

- The test-resolution is unknown. Joint optimization for a set $\mathcal{S}$ of resolutions is performed with

$$L^{S}_{\text{tr}}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \frac{\sum_{s \in \mathcal{S}} \ell_{\text{tr}}(\mathbf{x}^s, \mathbf{x}^s_t)}{|S|} + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \quad (14)$$

Any performance loss $\ell_{\text{tr}}$ is used, with or without blurring.

### 4.3. Optimization

The optimization is performed with Adam and projected gradient descent is used to apply the box constraints, *i.e.* $\mathbf{x} \in [0,1]^{W \times H \times 3}$. The adversarial image is initialized by the carrier image, while after every update its values are clipped to be in $[0,1]$. The adversarial image is given by

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x}), \quad (15)$$

where $L_{\text{tr}}$ can be $L_{\text{desc}}$ (with "desc" equal to MAC, SPoC, or GeM), $L_{\mathcal{P}}$, $L_{\text{hist}}$, or $L_{\text{tens}}$ according to the variant, while the variants with multiple scales are denoted *e.g.* by $L^{\mathcal{S}}_{\text{hist}}$ without blur or $L^{\hat{\mathcal{S}}}_{\text{hist}}$ with blur.

## 5. Experiments

Given a test architecture, we validate the success of the targeted mismatch attack in two ways. First, by measuring the cosine similarity between descriptors of the adversarial image $\mathbf{x}_a$ and the target $\mathbf{x}_t$ (should be as high as possible), and second, by using $\mathbf{x}_a$ as an image retrieval query and compare its performance with that of the target query (should be as close as possible)[3].

### 5.1. Datasets and evaluation protocol

We perform experiments on four standard image retrieval benchmarks, namely Holidays [14], Copydays [11], $\mathcal{R}$Oxford [30], and $\mathcal{R}$Paris [30]. They all consist of a set of query images and a set of database images, while the ground-truth denotes which are the relevant dataset images per query. We choose to perform attacks only with the first 50 queries for Holidays and Copydays to form adversarial attack benchmarks of reasonable size, while for $\mathcal{R}$Oxford and $\mathcal{R}$Paris we keep all 70 of them and use the *Medium* evaluation setup. All queries are used as targets to form an attack and retrieval performance is measured with mean Average Precision (mAP). Unless otherwise stated we use the "flower" of Figure 1 as the carrier; it is cropped to match the aspect ratio of the target. All images are re-sampled to have the largest dimension equal to 1024, this is the original image resolution. $\mathcal{R}$Oxford and $\mathcal{R}$Paris are treated differently than the other two due to the cropped image queries; the cropped image region that defines the query is used as a target and the relative scale change between queries and database images should be preserved not to affect the ground truth. When the image resolution for descriptor extraction is different than the original one, we down-sample the cropped image with the same scaling factor that the uncropped one should have been down-sampled with.

[3]Public implementation: https://github.com/gtolias/tma

### 5.2. Implementation details and experimental setup

We set the learning rate equal to 0.01 in all our experiments and perform 100 iterations for $L_{\text{desc}}$ and $L_{\text{hist}}$, while 1000 iterations for $L_{\text{tens}}$. If there is no convergence, we decrease the learning rate by a factor of 5 and increase the number of iterations by a factor 2 and re-start. We normalize the distortion term with the dimensionality of $\mathbf{x}$; this is skipped in the loss function of Sections 3 and 4 for brevity. Moreover, in order to handle the different range of activations for different FCNs, we normalize activation tensors with the maximum target activation before computing the mean squared error in (9). Image blurring at resolution $s$ in (12) is performed by a Gaussian kernel with $\sigma_b = 0.3 \max(W, H)/s$. The exponent of GeM pooling is always set to 3.

Setting $\lambda = 0$ provides a trivial solution to (7), *i.e.* $\mathbf{x}_a = \mathbf{x}_t$. However, we observe that initialization by $\mathbf{x}_c$ converges to local minima that are significantly closer to $\mathbf{x}_c$ than $\mathbf{x}_t$ even for the case of $\lambda = 0$. In this way, we satisfy the non-disclosure constraint, *i.e.* the adversarial image is visually dissimilar to the target, and do not sacrifice the performance loss. The image distortion w.r.t. to the carrier image does not sacrifice the goal of concealing the target and preserving user privacy. Therefore, in our experiments we mostly focus on cases with $\lambda = 0$, but also validate cases with $\lambda > 0$ to show the impact of the distortion term or in order to promote the non-disclosure constraint for the case of $L_{\text{tens}}$.

We experiment with different loss functions for targeted mismatch attacks. We define $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ sets of attack-resolutions[4]. We denote AlexNet [17], ResNet18 [13], and VGG16 [34] by $\mathcal{A}$, $\mathcal{R}$, and $\mathcal{V}$, respectively. We use networks that are pre-trained on ImageNet [33] and keep only their fully convolutional part. The AlexNet and ResNet18 ensemble is denoted by $\mathcal{E}$; mean loss over two networks is minimized. We report the triplet attack-model, loss function and value of $\lambda$ to denote the kind of adversarial optimization, for example $(\mathcal{A}, L^{\mathcal{S}_1}_{\text{hist}}, 0)$. For testing, we report the triplet test-model, test-pooling and test-resolution, for example $[\mathcal{A}, \text{GeM}, \mathcal{S}_0]$.

[4]$\mathcal{S}_0 = \{1024\}$, $\mathcal{S}_1 = \mathcal{S}_0 \cup \{300, 400, 500, 600, 700, 800, 900\}$, $\mathcal{S}_2 = \mathcal{S}_1 \cup \{350, 450, 550, 650, 750, 850, 950\}$, $\mathcal{S}_3 = \mathcal{S}_0 \cup \{262, 289, 319, 351, 387, 427, 470, 518, 571, 630, 694, 765, 843, 929\}$

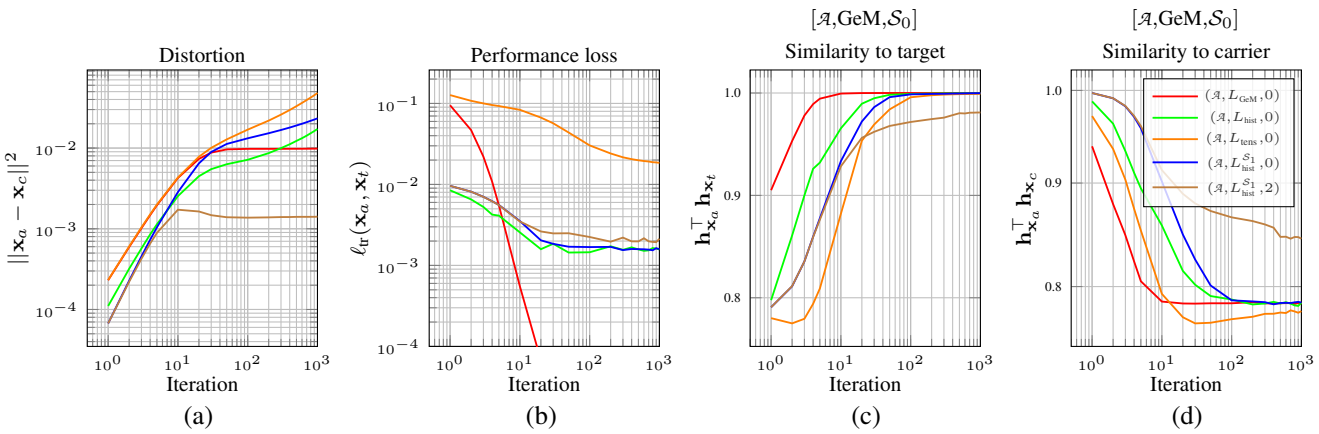


Figure 3. Adversarial images are generated with different loss functions and various measurements are reported as they evolve with the number of iterations. The presented measurements are: (a) the distortion w.r.t. the carrier image, (b) the performance loss from (7), (c) descriptor similarity of the adversarial image to the target for test case [$\mathcal{A}$,GeM,$\mathcal{S}_0$] and (d) descriptor similarity of the adversarial image to the carrier for test case [$\mathcal{A}$,GeM,$\mathcal{S}_0$]. The target and carrier images are the ones shown in Figure 7.

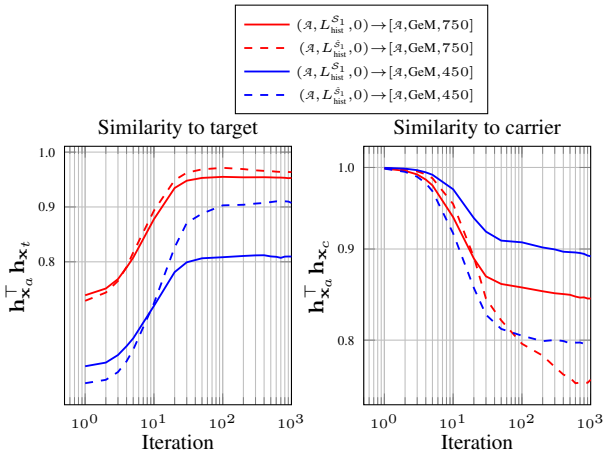


Figure 4. Descriptor similarity between the adversarial image and the target or the carrier as it evolves with the number of iterations. Comparison for test-resolutions that are not in the attack-resolutions for cases without (solid) and with (dashed) blurring. The adversarial optimization (left) and the test model (right) are denoted with $\rightarrow$. The target and carrier images are from Figure 7.

| $L_{\text{tr}}$ / $h$ | Original | $L_{\text{GeM}}$ | $L_{\mathcal{P}}$ | $L_{\text{hist}}$ | $L_{\text{tens}}$ |
|---|---|---|---|---|---|
| | mAP | mAP difference to original | | | |
| GeM | 41.3 | −0.0 | −0.0 | −0.2 | −0.1 |
| MAC | 37.0 | −0.5 | −0.0 | −0.8 | −0.0 |
| SPoC | 32.9 | −4.4 | −0.1 | −0.1 | −0.7 |
| R-MAC | 44.1 | −1.2 | −0.5 | −0.7 | −0.0 |
| CroW | 38.2 | −1.3 | −0.4 | −0.2 | −0.0 |
| | $\mathbf{x}_t^\top \mathbf{x}_a$ | | | | |
| GeM | 1.000 | 1.000 | 1.000 | 0.997 | 0.998 |
| MAC | 1.000 | 0.972 | 1.000 | 0.985 | 0.996 |
| SPoC | 1.000 | 0.909 | 1.000 | 0.999 | 0.996 |
| R-MAC | 1.000 | 0.972 | 0.978 | 0.979 | 0.997 |
| CroW | 1.000 | 0.968 | 0.994 | 0.995 | 0.998 |

Table 1. Performance evaluation for attacks based on AlexNet and various loss functions optimized at the original image resolution $\mathcal{S}_0$. Testing is performed on [$\mathcal{A}$,desc,$\mathcal{S}_0$] for multiple types of descriptor/pooling. Mean average Precision on $\mathcal{R}$Paris and mean descriptor similarity between the adversarial image and the target across all queries is reported. *Original* corresponds to queries without attack.

### 5.3. Results

For each adversarial image we perform the following measurements. We compute its similarity to the target and to the carrier by cosine similarity of the corresponding descriptors, we measure the carrier distortion and, lastly, we perform an attack by using it as a query and measure the average precision which is compared to that of the target.

**Optimization iterations.** We perform the optimization for different loss functions and report the measurements over iteration in Figure 3. Optimizing global descriptor or histogram converges much faster than the tensor case and results in significantly lower distortion. This justifies our choice of using a lower number of iterations for the two approaches. Increasing the value of $\lambda$ keeps the distortion lower but sacrifices the performance loss, as expected.

In Figure 4 we show how the similarity to the target and the carrier evolves for test-resolution that is not included in the set of attack-resolutions. Processing the images with image blurring offers significant improvements, especially for the smaller resolutions.

**Robustness to unknown test-pooling.** In Table 1 we present the evaluation comparison for different loss functions and test-pooling. The case of same attack-resolution and test-resolution is examined first. If the test-pooling is directly optimized ($L_{\text{GeM}}$ or $L_{\mathcal{P}}$ case), then perfect performance is achieved. The histogram and tensor based approaches both perform well for a variety of test-descriptors.

**Robustness to unknown test-resolution.** Cases with different attack-resolution and test-resolution are evaluated and results are presented in Figure 5. Resolutions that

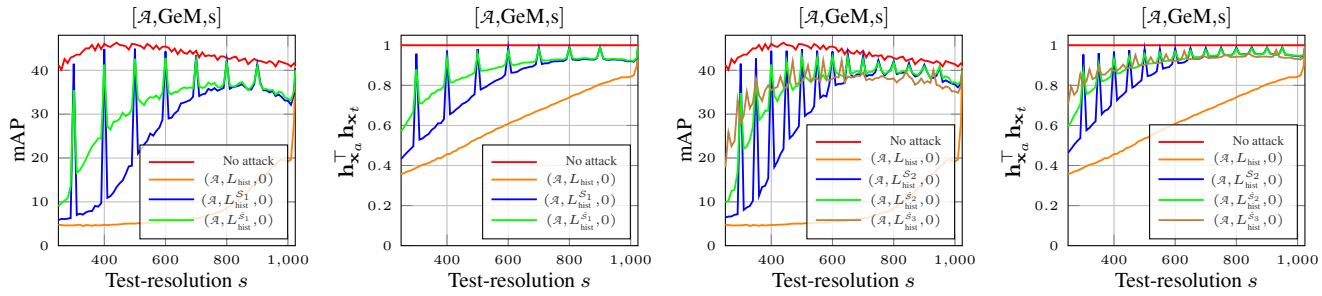


Figure 5. Performance evaluation for attack based on AlexNet and a set of attack-resolutions. Mean average Precision on $\mathcal{R}$Paris and mean descriptor similarity between the adversarial image and the target across all queries is shown for increasing test-resolution. Comparison using different sets of attack-resolutions and comparison for optimization without ($\mathcal{S}$) and with ($\hat{\mathcal{S}}$) image blurring.

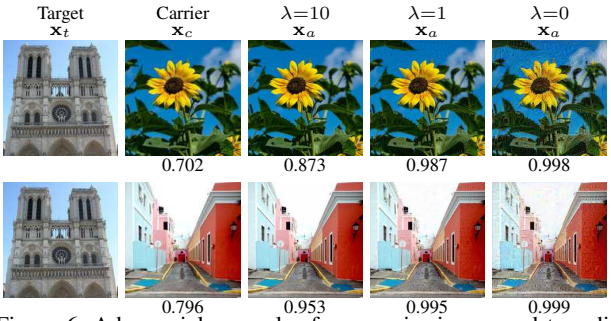


Figure 6. Adversarial examples for a carrier image and two different targets while optimizing $(\mathcal{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, \lambda)$ for various values of $\lambda$. Descriptor similarity is reported for $[\mathcal{A}, \text{GeM}, \mathcal{S}_0]$.

are not part of the attack-resolutions suffer from significant drop in performance when blurring is not performed, while blurring improves it. We observe how the retrieval performance and descriptor similarity between adversarial image and target are correlated. Moreover, the optimization for multiple resolutions is clearly better than that for single resolution, while logarithmic sampling of attack-resolutions ($\mathcal{S}_3$) significantly improves the performance for very small test-resolution but harms it for larger ones.

**Impact of the distortion term.** We evaluate $[\mathcal{A}, \text{GeM}, \mathcal{S}_0]$ on queries of $\mathcal{R}$Paris for $(\mathcal{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, \lambda)$ and $\lambda$ equal to 0, 0.1, 1, 10. The average similarity between the adversarial image and the target is 0.990, 0.987, 0.956, and 0.767, respectively, while the average distortion is 0.0177, 0.0083, 0.0026, and 0.0008, respectively. Examples of adversarial images are shown in Figure 6.

**Impact of the whitening in the test-model.** We now consider the case that the test-model includes descriptor whitening. The whitening is unknown during the time of the adversarial optimization. We evaluate the performance on $\mathcal{R}$Paris while learning whitening with PCA on $\mathcal{R}$Oxford. Testing without whitening and $[\mathcal{A}, \text{GeM}, \mathcal{S}_0]$ or $[\mathcal{A}, \text{GeM}, 768]$ achieves 41.3, and 40.2 mAP, respectively. After applying whitening the respective performances increase to 47.5 and 48.0 mAP. Attacks with $(\mathcal{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, 0)$ achieve 40.2, and 39.4 mAP when tested in the aforementioned cases without whitening. Attacks with $(\mathcal{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, 0)$ achieve 47.3, and 42.9 mAP when tested in the aforementioned cases with whitening. Whitening introduces additional challenges, but the attacks seem effective with slightly reduced performance.

**Concealing/revealing the target.** We generate adversarial images for different loss functions and show examples in Figure 7. The corresponding tensors show that spatial information is only preserved in the tensor-based loss function. The tensor-based approach requires the distortion term to avoid revealing visual structures of the target (adversarial images in 6-th and 7-th column). We now pose the question "can the FCN activations of the adversarial image reveal the content of the target?". To answer, we invert tensor $\mathbf{g}_{\mathbf{x}_a}$ at multiple resolutions using the method of Mahendran and Vedaldi [22]. The tensor-based approach indeed reveals the target's content in the reconstruction, while no other approach does. This highlights the benefits of the proposed histogram-based optimization. Note that the reconstructed image resembles the target less if the resolutions used in the reconstruction are not the same as the attack-resolutions (rightmost column).

**Timings.** We report the average optimization time per target image on Holidays dataset and on a single GPU (Tesla P100) for some indicative cases. Optimizing $(\mathcal{A}, L_{\text{GeM}}, 0)$, $(\mathcal{A}, L_{\text{GeM}}^{\hat{\mathcal{S}}_1}, 0)$, $(\mathcal{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_1}, 0)$, $(\mathcal{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, 0)$, and $(\mathcal{A}, L_{\text{tens}}^{\hat{\mathcal{S}}_1}, 0)$ takes 1.9, 7.5, 12.3, 22.9, and 68.4 seconds, respectively. Using ResNet18 $(\mathcal{R}, L_{\text{GeM}}, 0)$ and $(\mathcal{R}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, 0)$ take 3.9 and 40.6 seconds, respectively.

**Multiple attacks.** We show results of multiple attacks in Table 2. We present the original retrieval performance together with the difference in the performance caused by the attack. It summarizes the robustness of the histogram and tensor based optimization to unknown pooling operations. It emphasizes the challenges of unknown test-resolution and the impact of the blurring; this outcome can be useful in various different attack models. The very last row suggests that transferring attacks to different FCNs (optimizing on $\mathcal{E}$, which includes $\mathcal{A}$ and $\mathcal{R}$, and testing on $\mathcal{V}$) is hard to achieve; it is harder than for classification [35].

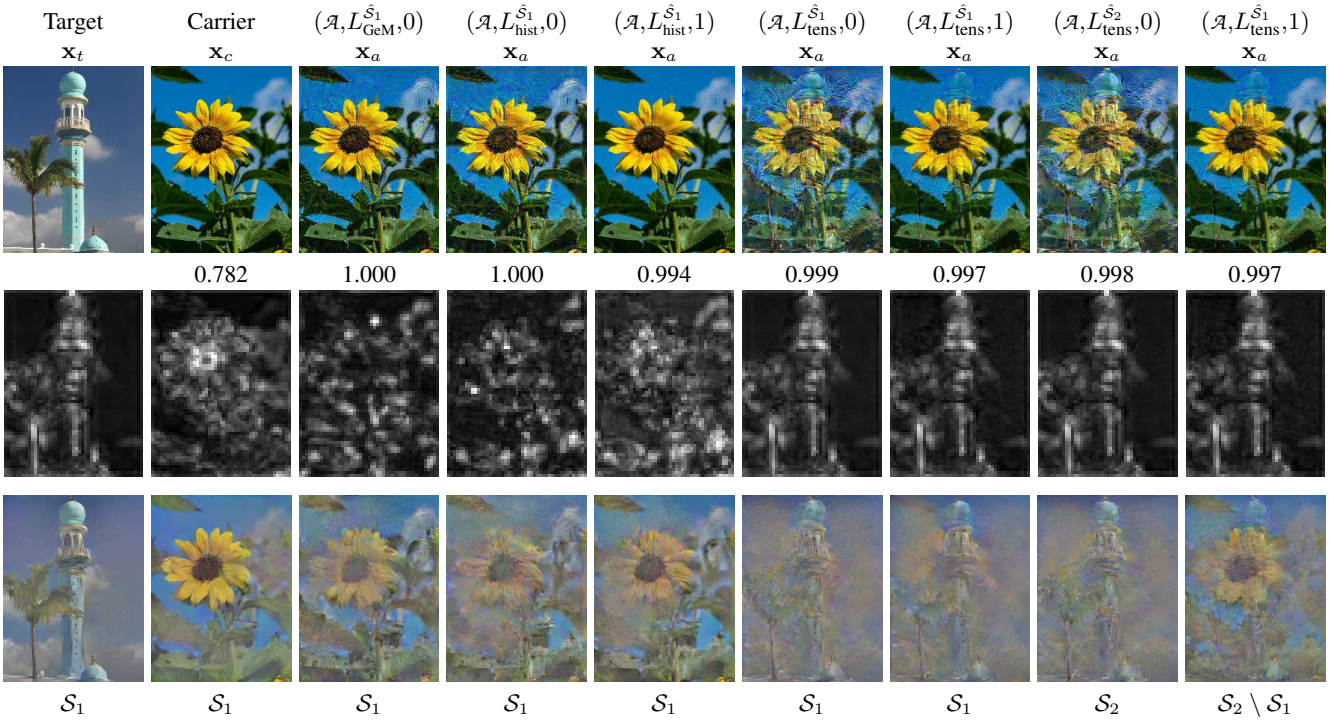


Figure 7. Target, carrier and adversarial images for different variants (top image row), a summary of tensor $\mathbf{g}_{\mathbf{x}}$ by depth-wise maximum (middle image row) and the inversion of $\mathbf{g}_{\mathbf{x}_t}$, $\mathbf{g}_{\mathbf{x}_c}$, or $\mathbf{g}_{\mathbf{x}_a}$, respectively, over multiple resolutions (bottom image row). The resolutions for inversion are reported below the bottom row. The tensor inversion shows whether the target, or any information about it, can be reconstructed from the adversarial image. The first two inversions are presented as a reference. Descriptor similarity to the target is reported below the first image row for $[\mathcal{A},\text{GeM},1024]$.

| Attack | Test | $\mathcal{R}$Oxford | $\mathcal{R}$Paris | Holidays | Copydays |
|---|---|---|---|---|---|
| $(\mathcal{A},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$ | $[\mathcal{A},\text{GeM},\mathcal{S}_0]$ | 26.9 / +0.2 | 41.3 / -1.2 | 81.5 / +0.2 | 80.4 / -0.4 |
| $(\mathcal{R},L^{\hat{\mathcal{S}}_2}_{\text{GeM}},0)$ | $[\mathcal{R},\text{GeM},\mathcal{S}_0]$ | 21.5 / -0.7 | 46.9 / -0.4 | 82.9 / -0.3 | 69.3 / -0.7 |
| | $[\mathcal{R},\text{GeM},768]$ | 24.0 / -2.5 | 48.0 / -3.9 | 81.7 / -4.4 | 75.6 / -2.8 |
| | $[\mathcal{R},\text{GeM},512]$ | 22.4 / -6.7 | 49.7 /-11.1 | 82.8 / -0.6 | 82.1 /-10.7 |
| $(\mathcal{R},L^{\mathcal{S}_2}_{\text{hist}},0)$ | $[\mathcal{R},\text{GeM},\mathcal{S}_0]$ | 21.5 / -1.2 | 46.9 / -1.9 | 82.9 / -0.6 | 69.3 / -1.3 |
| | $[\mathcal{R},\text{GeM},768]$ | 24.0 / -3.7 | 48.0 / -7.2 | 81.7 / -2.3 | 75.6 / -7.1 |
| | $[\mathcal{R},\text{GeM},512]$ | 22.4 /-11.2 | 49.7 /-20.7 | 82.8 /-17.1 | 82.1 /-20.6 |
| $(\mathcal{R},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$ | $[\mathcal{R},\text{GeM},\mathcal{S}_0]$ | 21.5 / -1.4 | 46.9 / -1.8 | 82.9 / -2.4 | 69.3 / -1.3 |
| | $[\mathcal{R},\text{GeM},768]$ | 24.0 / -5.3 | 48.0 / -6.0 | 81.7 / -1.7 | 75.6 / -4.2 |
| | $[\mathcal{R},\text{GeM},512]$ | 22.4 / -7.4 | 49.7 /-11.9 | 82.8 / -4.9 | 82.1 /-11.3 |
| $(\mathcal{R},L^{\hat{\mathcal{S}}_2}_{\mathcal{P}},0)$ | | 22.0 / -1.1 | 45.0 / -0.5 | 81.0 / +0.9 | 67.0 / -1.6 |
| $(\mathcal{R},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$ | $[\mathcal{R},\text{CroW},\mathcal{S}_0]$ | 22.0 / -0.3 | 45.0 / -0.8 | 81.0 / +1.3 | 67.0 / -1.0 |
| $(\mathcal{R},L^{\hat{\mathcal{S}}_2}_{\text{tens}},0)$ | | 22.0 / -0.7 | 45.0 / -0.0 | 81.0 / -0.6 | 67.0 / -3.0 |
| $(\mathcal{E},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$ | $[\mathcal{A},\text{GeM},\mathcal{S}_0]$ | 26.9 / -2.3 | 41.3 / -5.5 | 81.5 / -3.1 | 80.4 / -4.9 |
| | $[\mathcal{R},\text{CroW},\mathcal{S}_0]$ | 22.0 / -1.1 | 45.0 / -0.8 | 81.0 / +1.0 | 67.0 / -0.8 |
| | $[\mathcal{V},\text{GeM},\mathcal{S}_0]$ | 38.1 /-34.9 | 54.0 /-47.4 | 85.7 /-72.6 | 80.0 /-72.9 |

Table 2. Performance evaluation for multiple attacks, test-models, and datasets. Mean average Precision over the original queries, together with the mAP difference to the original caused by the attack, is reported. The parameters of the adversarial optimization during the attack are shown in the leftmost column, while the type of test-model used is shown in the second column.

## 6. Conclusions

We have introduced the problem of targeted mismatch attack for image retrieval and address it in order to construct concealed query images instead of the initial intended query. We show that optimizing the first order statistics is a good way to generate images that result in the desired descriptors without disclosing the content of the intended query. We analyze the impact of image re-sampling, which is a natural component of image retrieval systems and reveal the benefits of simple image blurring in the adversarial image optimization. Finally, we show that transferring attacks to new FCNs are much more challenging than their image classification counterparts.

We focused on concealing the query in a privacy preserving scenario. In a malicious scenario the adversary might try to corrupt the search results by targeted mismatch attacks on indexed images. This is an interesting direction and an open research problem.

**Acknowledgments** Work supported by GAČR grant 19-23165S and OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

## References


[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 2018. 2

[2] Laurent Amsaleg, James Bailey, Dominique Barbe, Sarah Erfani, Michael E Houle, Vinh Nguyen, and Miloš Radovanović. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. In *IEEE Workshop on Information Forensics and Security (WIFS)*, 2017. 2

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 3

[4] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In *ICCV*, 2015. 3

[5] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SISC*, 1995. 2

[6] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security*, 2016. 2

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *SSP*, 2017. 1, 2, 3

[8] Thanh-Toan Do, Ewa Kijak, Laurent Amsaleg, and Teddy Furon. Security-oriented picture-in-picture visual modifications. In *ICMR*, 2012. 2

[9] Thanh-Toan Do, Ewa Kijak, Teddy Furon, and Laurent Amsaleg. Challenging the security of content-based image retrieval systems. In *MMSP*, 2010. 2

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 1, 2

[11] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of GIST descriptors for web-scale image search. In *CIVR*, 2009. 5

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[14] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 5

[15] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCVW*, 2016. 3

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 5

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLRW*, 2017. 2

[19] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In *arXiv*, 2018. 1, 2, 3

[20] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Who's afraid of adversarial queries? the impact of image modifications on content-based image retrieval. In *arXiv*, 2019. 1, 2

[21] David Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[22] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 4, 7

[23] Eva Mohedano, Kevin McGuinness, Noel E O'Connor, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *ICMR*, 2016. 3

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 1, 2

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 1, 2

[26] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017. 4

[27] Eng-Jon Ong, Sameed Husain, and Miroslaw Bober. Siamese network of deep fisher-vector descriptors for image retrieval. In *arXiv*, 2017. 3

[28] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. In *arXiv*, 2016. 2

[29] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ASIACCS*, 2017. 2

[30] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 5

[31] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *PAMI*, 2018. 3

[32] Ali Sharif Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Trans. MTA*, 2016. 3

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv*, 2014. 5

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1, 2, 7

[36] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*, 2016. 3

[37] Zhedong Zheng, Liang Zheng, Zhilan Hu, and Yi Yang. Open set adversarial examples. In *arXiv*, 2018. 1, 2