

Universal Perturbation Attack Against Image Retrieval

Jie Li¹, Rongrong Ji^{1,2*}, Hong Liu¹, Xiaopeng Hong^{3,2,4}, Yue Gao⁵, Qi Tian⁶

¹Department of Artificial Intelligence, School of Informatics, Xiamen University,

²Peng Cheng Lab, Shenzhen, China,

³MOE Key Lab. for Intelligent Networks and Network Security/Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, PRC

⁴University of Oulu, Finland, ⁵Tsinghua University, ⁶Huawei Noah's Ark Lab

lijie32@stu.xmu.edu.cn, rrji@xmu.edu.cn, lynnlou.xmu@gmail.com,

hongxiaopeng@mail.xjtu.edu.cn, kevin.gaoy@gmail.com, tian.qil@huawei.com,

Abstract

Universal adversarial perturbations (UAPs), a.k.a. input-agnostic perturbations, has been proved to exist and be able to fool cutting-edge deep learning models on most of the data samples. Existing UAP methods mainly focus on attacking image classification models. Nevertheless, little attention has been paid to attacking image retrieval systems. In this paper, we make the first attempt in attacking image retrieval systems. Concretely, image retrieval attack is to make the retrieval system return irrelevant images to the query at the top ranking list. It plays an important role to corrupt the neighbourhood relationships among features in image retrieval attack. To this end, we propose a novel method to generate retrieval-against UAP to break the neighbourhood relationships of image features via degrading the corresponding ranking metric. To expand the attack method to scenarios with varying input sizes or untouchable network parameters, a multi-scale random resizing scheme and a ranking distillation strategy are proposed. We evaluate the proposed method on four widely-used image retrieval datasets, and report a significant performance drop in terms of different metrics, such as mAP and mP@10. Finally, we test our attack methods on the real-world visual search engine, i.e., Google Images, which demonstrates the practical potentials of our methods.

1. Introduction

Convolutional neural networks (CNN) have been the state-of-the-art solution for a wide range of computer vision tasks, such as image classification, image segmentation and objective detection. Despite the remarkable success, deep learning models have shown to be vulnerable to

*Corresponding author.

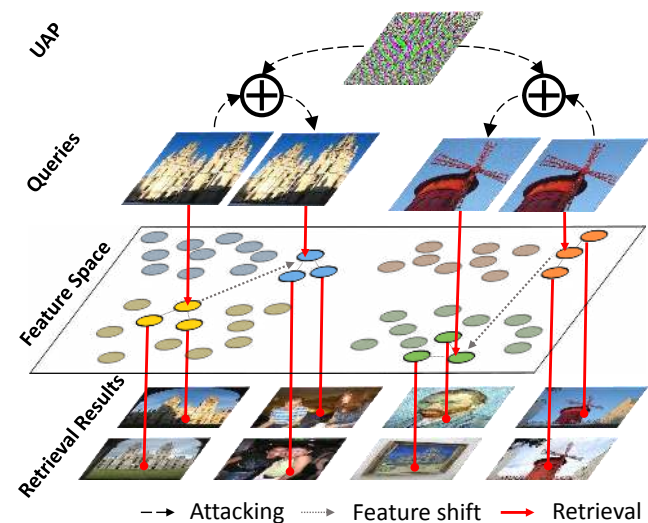


Figure 1. When added to natural images, a single universal perturbation that is invisible to human eyes causes most images to shift significantly in the feature space without preserving original neighbourhood relationships. The top is the perturbation and dots represent the features of images. (Best viewed in color.)

small perturbations to the input image. Various attack techniques have been proposed, like model distillation [30, 42], transfer learning [24, 45], and gradient updating [1]. In contrast to previous methods called image-specific perturbations having to perform computation every time to generate a particular perturbation for any given image, Moosavi-Dezfooli *et al.* [25] proposed an image-agnostic perturbation termed universal adversarial perturbation (UAP), which can fool most images from a data distribution. Being universal, UAPs can be conveniently exploited to perturb unseen datapoints on-the-fly without extra computation. Therefore, UAPs are particularly useful in a wide range of applications.

However, existing methods regardless of whether image-agnostic or not, mainly focus on image classification while

no existing work has touched the topic of attacking image retrieval systems. As a long-standing research topic in computer vision [44], image retrieval aims to find relevant images from a dataset given a query image. Despite the extensive efforts in improving the search accuracy (*e.g.*, new features like NetVLAD [2] and generalized-mean pooling [33]) or efficiency (*e.g.*, indexing schemes like Hamming Embedding [17] or hashing [22, 41]), very little attention has been paid to the vulnerability of the state-of-the-art retrieval systems. It is difficult, or even infeasible to apply existing UAPs methods in image retrieval directly. The reasons come from four aspects.

- Different dataset label formats. Most existing UAP methods designed for image classification work on datasets labeled by categories [10], which need UAPs pushing datapoints across decision boundary [25]. However, datasets in retrieval are usually labeled by similarity [32], which require UAPs to capture complex relationships among features instead.
- Different goals. The goal of existing UAP methods is to disturb unary and binary model outputs for single instance, *e.g.*, to change the most likely predict label. However, merely corrupting the top-1 result is still not enough since the retrieval evaluation is usually done on a ranking list. Thus, to attack retrieval systems, one should disturb the ranking list via lowering the positions of positive samples there.
- Different sizes of model input. Generally, models which existing UAPs trained on ask for fixed-size input images, accordingly the size of UAPs is fixed as the input. However, these UAPs are fragile and can be defended by varying the size of input [43]. Note that, the size of images in retrieval usually vary, which restricts the direct usage of the traditional UAPs and thus poses a higher demand for generating UAP for the task of image retrieval.
- Different model output and optimization methods. It is often assumed predict confidence of each category can be fetched [6, 9], and the confidences are a group of continuous and floating numbers responding to the changes of input rapidly. It indicates a way to estimate gradient for optimization. However, the large-scale discrete ranking list returned by retrieval systems offers little guidance on approximating gradient. This fact makes it infeasible to apply existing UAPs to retrieval systems with network parameters unaccessible.

In this paper, we make the first attempt in attacking image retrieval, especially the cutting-edge image retrieval model that are deployed upon deep features. In principle,

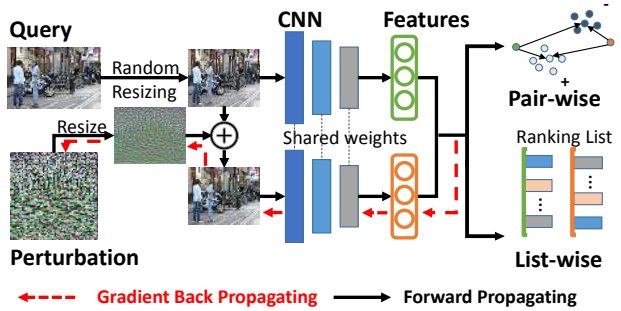


Figure 2. The pipeline of the proposed method. Perturbation is first resized to the same size of the input image which goes through random resizing layer with a random scale. Then both the resized input image and the sum of perturbation and input image are fed into CNN model to corrupt three relationships. Only gradient of the perturbation will be calculated during back propagation to update the perturbation.

we aim to generate a UAP for corrupting neighbourhood relationships in the feature space as depicted in Fig. 1. To address the challenges mentioned above, we propose a novel universal adversarial perturbation attack method for image retrieval. In detail, we build a general model to craft the UAP that breaks the neighborhood relationships among feature points by altering the input slightly. Pair-wise relationship among neighborhood structures is first considered via constructing tuples based on the nearest and farthest groups. We corrupt this relationship by swapping the similarity relationship in the tuples. Although corrupting pair-wise relationship is simple and efficient, the pair-wise information focuses on the local relationship between query and two data samples each time without considering global ranking list which is more significant for retrieval. We argue it can not solve the retrieval attack problem fundamentally. Eventually, we propose the approach to generate UAPs from list-wise aspect that goes further to permute the entire ranking list via destroying the corresponding ranking metrics to lower positions of relative references. In addition, we propose a multi-scale random resizing scheme to apply UAP to input images at different resolutions, which shows better attack performance than fixed-scale methods experimentally. The pipeline of the proposed method is shown in Fig. 2.

Our scheme further enables attack without touching network parameters via a coarse-to-fine strategy to distill victim model by regressing ranking list as depicted in Fig. 3. First, we construct coarse-grained subsets which preserves global ranking information sampled from the entire large-scale ranking list, and prompt distilled model to fit the ordinal relation in the subsets. Then from the fine-grained level, we focus on the top-*k* most related instances for retrieval to refine the distilled model.

The proposed method achieves high attack performance substantially and leads to a large performance drop on stan-

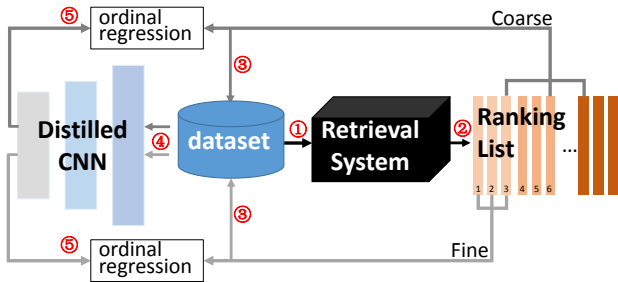


Figure 3. The pipeline of ranking distilling. First, ranking list of the unknowable black-box retrieval systems for the dataset is obtained and divided into groups (①②). Then the datapoints of the coarse-grained subset randomly sampled from each group and the fine-grained top- k references are fetched from dataset (③) to optimize the distilled model via regressing ordinal information among them (④⑤).

standard image retrieval benchmarks, *i.e.*, Oxford Buildings and Paris with their revised versions. The retrieval performance is tested on two CNN-based image representation [33, 34, 40] with three different CNN models [13, 20, 38]. Quantitatively, the universal adversarial perturbation can drop the performance such as mAP and $mP@10$ by at least 50%, which reveals the cutting-edge image retrieval systems is quite vulnerable to adversarial examples. Interestingly, we further evaluate our universal perturbation on the real-world image search engine, *i.e.*, Google Images, and conclude that the perturbation can also corrupt the output ranking list.

2. Related Work

Adversarial Examples. Szegedy *et al.* [39] have demonstrated that neural networks can be fooled by adversarial example, which is a clean image being intentionally perturbed, *e.g.* by adding adversarial perturbation that is quasi-imperceptible to human eyes. Subsequently, various methods have been proposed to generate such perturbations [11, 12, 26]. An iterative scheme is proposed in [21] to achieve better attack performance via applying gradient ascent multiple times. Besides, complex approaches like [26] find perturbation from the perspective of classification boundary.

However, these methods compute perturbations for each data point specifically and independently. More recently, Moosavi-Dezfooli *et al.* [25] have shown that there exists a single image-agnostic perturbation termed universal adversarial perturbation (UAP) being able to corrupt most natural images. UAP is a single adversarial noise that is trained offline and can perturb the corresponding outputs of a given model online. Contrast to white-box attack where victim model can be accessed, black-box attack refers to the case that attackers have little knowledge about victim. It is observed that perturbations crafted for specific models or training sets can fool other models and

datasets [12, 39], referred as transfer attack, which is widely adopted in black-box. Another popular method is knowledge distillation [14], which obtains substitute model via regressing output of victim and then applies white-box attack methods [42].

Visual Features for Retrieval. Image retrieval is a long-standing research topic in computer vision [44]. Given a query image, the search engine retrieves related ones from a large set of reference images. A typical setting refers to extracting and comparing features between a query and references, such as global descriptors [28] and local descriptor aggregations [18, 36]. Nowadays, the most prominent retrieval methods are mostly based on CNNs [4, 5, 15, 19, 33, 34, 40]. They mainly use the pre-trained CNNs as a backbone to extract global representation for images. To that effect, CNN models pre-trained with ImageNet [10] (*e.g.* AlexNet [20], VGGNet [38] and ResNet [13]) already provide superior performance over hand-crafted features [5]. Babenko *et al.* [5] further showed that fine-tuning the CNN models can further boost the retrieval performance. In this trend, many recent methods are proposed to construct trainable pooling layers for better feature representation. Representative methods include, but not limited to, maximum activations of convolutions (MAC) [34, 40], weighted sum pooling (CroW) [19], and generalized-mean pooling (GeM) [33]. In this paper, we mainly consider two state-of-the-art pooling methods, *i.e.*, MAC [34, 40] and GeM [33], with three different CNN models, *i.e.*, AlexNet, VGGNet, and ResNet, to evaluate the performance of UAP attack.

3. The Proposed Method

Our method is aimed to seek a universal perturbation δ with constraint of $\|\delta\|_\infty \leq \epsilon$, to corrupt as much similarity relationships as possible in the data distribution \mathcal{X} . By doing so, the originally similar features should be dissimilar after adding a small perturbation.

For convenience, we denote the universal perturbation by δ , denote the feature vectors of the i -th original image x_i and the adversarial one by:

$$f_i = F(R_I(x_i)),$$

$$f'_i = F\left(\max\left(0, \min\left(255, R_P(\delta, R_I(x_i)) + R_I(x_i)\right)\right)\right),$$

where $F(\cdot)$ is the function that outputs the feature vector through a CNN model, $R_I(\cdot)$ and $R_P(\cdot, \cdot)$ are resizing operators for input image and universal perturbation, respectively. Resizing operators will be elaborated in Sec. 3.3. The Euclidean distance between two feature vectors f_i and f_j are characterized as a function $d(f_i, f_j)$. To avoid computational overhead caused by the large-scale dataset, a landmark-based ordinal relation [3] that compares any

Algorithm 1 Universal Perturbation Generating for Attacking Image Retrieval.

Input: Data set $X = \{x_1, x_2, \dots, x_n\}$, parameters λ .

Output: Universal perturbation vector δ .

- 1: Initialize $\delta \leftarrow 0$
 - 2: **repeat**
 - 3: **for** each datapoint $x_i \in X$ **do**
 - 4: Randomly resize x_i then resize perturbation δ accordingly
 - 5: Compute and update the gradients ∇L
 - 6: Update the perturbation by optimizing Eq. 8
 - 7: **if** δ gets saturated **then**
 - 8: $\delta = \delta/2$
 - 9: **end if**
 - 10: **end for**
 - 11: **until** convergence.
-

query point to the landmarks¹ is calculated in advance.

3.1. Baseline

We first attempt to disturb retrieval systems using label-wise information to validate whether UAPs against classification is suitable for retrieval. We define a classifier, equipped by the cross-entropy loss function with FC layer and softmax layer. We recognize the cluster index as pseudo-label and use them for all experiments in order to reduce computational cost and to ensure every experiment is conducted under the same setting. Pseudo-labels are crafted with features from victim model, which include more victim attributes than exact labels and could benefit attack. Furthermore, pseudo-labels can be easily extended to scenarios where exact labels are unavailable. The classifier is trained via pseudo-labels and then fooled by minimizing the widely-used classification attack loss [8] as follows:

$$L(\delta) = [Z(x')_t - \max(Z(x')_i : i \neq t)]_+, \quad (1)$$

where $[x]_+$ is the $\max(x, 0)$ function, $Z(\cdot)$ is the output before the softmax layer, t is the label of clean input and x' is the input perturbed by δ .

3.2. The Proposed UAP

Image retrieval can be viewed as a ranking problem, from which perspective the relationship between query and references plays an important role [23]. Therefore, such relationship should be fully utilized, that can further improve the attack performance. To this end, we consider two relationships to be corrupted *i.e.*, pair-wise and list-wise.

Corrupting Pair-wise Relationship. Here we use ordinal relationship between the nearest and the farthest references to approximate the pair-wise information, which can

¹Landmarks are generated via K-means clustering.

be constructed directly via the classical triplet loss. Formally, an ordered relation set C can be written as follows:

$$\begin{aligned} \eta_{ij} < \eta_{ik} &\Rightarrow d(f_j, f_i) > d(f_k, f_i) \\ &\Rightarrow d(f_j, f'_i) < d(f_k, f'_i), \forall (i, j, k) \in C. \end{aligned} \quad (2)$$

We define $\eta_{ik} = 1$ as similar pairs of x_i and x_k that share the same cluster. $\eta_{ij} = 0$ means the distance between clusters corresponding to samples x_i and x_j is the farthest. Therefore, a set of tuples belonging to the subset of C can be re-computed. To attack the retrieval system, we minimize the traditional triplet loss as follows:

$$L(\delta) = \sum_{\eta_{ik}=1, \eta_{ij}=0} [\alpha + d_1(f_j, f'_i) - d_1(f_k, f'_i)]_+, \quad (3)$$

where α is the parameter representing the margin between the matched and unmatched samples.

Corrupting List-wise Relationship. Unlike corrupting pair-wise one that focuses merely on the local relationship, we further permute the entire ranking list for list-wise relationship to destroy the corresponding ranking metric.

Since the list is typically too large to be directly processed, we re-use the landmark employed above and construct a subset of the ranking list with suitable size by sampling references from each landmark each time. We treat the reversed ranking list of cluster centers as the ideal ranking sequence, and destroy the normalized Discounted Cumulative Gain (NDCG) metric [16] as it is the most classical measurement well suited to information retrieval [35]. NDCG is multilevel measures, which is aimed to measure the instance's gain based on its position in result list. The gain is accumulated from the top of the list to the bottom, and gain of reference at lower rank will be discounted. Given any permutation g of the set S and its ratings sets $\{y_i\}_{i=1}^{|g|}$, DCG is defined as follows:

$$DCG(R) = \sum_{i=1}^{|g|} \frac{2^{y_i} - 1}{\log_2(i + 1)}. \quad (4)$$

NDCG divides DCG by value of ideal ranking sequence to ensure a range of $[0, 1]$.

However, the function in Eq. (4) is non-convex and non-smooth, which makes the optimization problematic. To this end, we approximate the gradient by accumulating the influence via swapping references. After sorting the images by score for a given query image feature f_i , if y_j and y_k are the ideal rank indices of current the i -th and j -th images feature f_j and f_k respectively, we have the tangent of the distance function that has the property as follows:

$$\begin{aligned} \frac{\partial d(f_i, f_j)}{\partial \delta} - \frac{\partial d(f_i, f_k)}{\partial \delta} &\gg 0, \\ \text{whenever } j &\gg k \text{ and } y_j \ll y_k. \end{aligned} \quad (5)$$

Therefore, given a ranking list, we can directly calculate the sum of the gradient residuals in Eq. (5), which roughly approximate the gradient of the NDCG loss in Eq. (4). Due to the discounted factor in DCG, following the similar strategy in [7], we also introduce the λ parameter to weight the gradient residual, whose gradient can be defined as follows:

$$\nabla\delta = \frac{\partial NDCG(R)}{\partial\delta} \approx \sum_{j \neq k} \lambda_{jk} \left(\frac{\partial d(f, f_j)}{\partial\delta} - \frac{\partial d(f, f_k)}{\partial\delta} \right),$$

$$\lambda_{jk} = \frac{-1}{1 + e^{(d(f, f_j) - d(f, f_k))}} |\Delta_{NDCG_{jk}}|, \quad (6)$$

where $|\Delta_{NDCG_{ij}}|$ is the change of NDCG metric if swap positions of the i -th and the j -th references.

3.3. Random Resizing

Unlike classification models, where input images are cropped and padded to a fixed size, retrieval model can accept inputs at different scales. Therefore, resizing is a mean for defense attack [43], which not only affects the retrieval performance, but also influences the attack quality.

To make the proposed universal perturbation be suitable for different scales, a random resizing process $R_I(\cdot)$ is employed, which resizes the original input image x with size $W \times H \times 3$ to a new image $R_I(x)$ with random size $W' \times H' \times 3$. Note that, W' along with H' is within a specific range, and $|\frac{W'}{W} - \frac{H'}{H}|$ should be within a reasonably small range to prevent image distortion. Then, the UAP δ is resized to a new perturbation $R_P(\delta, R_I(x))$ with the same size as $R_I(x)$ to be added to the input image.

3.4. Rank Distillation

Above methods require accessing model parameters which is not realistic in general. To overcome it, we propose a coarse-to-fine rank distillation method to build a substitute model. Note that the gap between different architectures exists, and distillation can be viewed as an effective defense [29, 31] as well. Therefore, distilling with diverse architecture may not work. Similar to [42], we assume the architecture of model is known.

Since regressing large-scale ranking indices is very computational and memory intensive, we turn to adopt a hierarchical strategy that first considers coarse-grained subset and then focuses on fine-grained top- k references.

For coarse-grained part, a subset of the entire ranking list which preserves the global ranking information for distilled model to regress is considered. Concretely, a large ranking list is divided into many bins according to the indices, and a subset is constructed by sampling one reference from each bin. We optimize the distillation model on the subset to fit the ordinal relation between the corresponding bins. For-

mally, the ordinal regression objective is defined as follows:

$$\min \sum_i \sum_{m>n} \lambda_m [d(q_i, r_{im}) - d(q_i, r_{in}) + \beta]_+, \quad (7)$$

where q_i is the feature from the distilled model of the i -th query, r_{im} is the feature of the m -th similar reference in subset for the i -th query, λ_m is the discount factor ensuring top references have more importance, and β is the margin to avoid all features falling into a single point.

Subsequently, for fine-grained part, a refined procedure focusing on the top- k references are conducted. We adopt the similar strategy as coarse part with decreasing arguments (*e.g.* learning rate and margin), while r_{im} in Eq. (7) refers to the m -th similar feature in top- k list instead.

Then, the same attack strategy as described in Sec. 3.2 is carried out on the distilled model, and the learned perturbation is transferred to attack the true target victim.

3.5. The Optimization

Since the gradient of δ can be got easily, we adopt the stochastic gradient descent with momentum [11] to update the perturbation vector at the i -th iteration:

$$g_i = \mu \cdot g_{i-1} + \frac{\nabla\delta}{\|\nabla\delta\|_1},$$

$$\delta_i = \delta_{i-1} + \lambda \cdot \text{sign}(g_i), \quad (8)$$

$$\delta_i = \min(\max(-\epsilon, \delta_i), \epsilon),$$

where g_i is the momentum of the i -th iteration and λ is the learning rate. The clipping operation that ensures constraint $\|\delta\| \leq \epsilon$ may invalidate the update after δ reaches a constraint. We tackle this issue by following [27], which rescales δ to half when the perturbation gets saturated. The detailed algorithm is provided in Alg. 1.

4. Experiments

In this section, we present quantitative results and analysis to evaluate the proposed attack schemes. We train our universal perturbations on the 30k Structure-of-Motion Reconstruction dataset. Two recent CNN-based image descriptors (*i.e.*, MAC [34, 40] and GeM [33]) with three different CNN models (*e.g.* AlexNet [20], VGGNet [38] and ResNet [13]) are used, forming six CNN models that are trained on the 120k Structure-of-Motion Reconstruction dataset. We use *Oxford5k* and *Paris6k* with their revised versions [32] to evaluate the attack performance.

Training datasets. The *SfM* dataset [37] consists of 7.4 million images downloaded from Flickr. It contains two large-scale training sets named *SfM-30k* and *SfM-120k*, respectively. We utilize K-Means clustering on 6,403 validation images from *SfM-30k* to obtain the list-wise relationship, and use the clustering index as pseudo-label to train

Eval		Oxford5k		ROxford5k						Paris6k		RParis6k						mDR
		E	M	H	E	M	H	E	M	H	E	M	H					
		mAP			mP@10			mAP			mP@10							
A-MAC	O	57.11	45.23	32.96	10.43	57.25	55.43	15.36	65.64	63.99	46.93	20.06	88.00	91.29	58.29			
	C	46.99	36.13	27.89	7.86	49.58	48.36	12.71	57.91	52.96	40.33	16.27	80.86	83.00	48.86	15.47%		
	P	29.61	24.52	17.99	4.92	32.06	30.86	6.67	42.89	38.71	30.43	11.13	52.86	54.71	29.14	44.35%		
	L	27.88	21.59	16.31	4.06	28.33	28.57	7.50	41.15	37.40	29.28	10.00	49.29	51.43	25.00	48.33%		
A-GeM	O	59.86	50.21	36.72	14.29	58.10	53.60	23.32	73.66	70.65	51.89	22.80	87.71	88.86	57.86			
	C	35.49	30.07	22.00	7.03	33.62	31.71	10.16	48.27	42.60	33.80	12.55	46.57	50.00	27.00	43.51%		
	P	29.31	22.85	17.57	5.56	25.65	24.79	8.36	40.71	35.17	29.44	10.71	38.86	41.71	20.14	54.12%		
	L	26.48	22.45	17.12	5.29	25.78	24.25	8.03	37.17	32.28	27.42	10.23	34.86	37.14	18.29	56.88%		
V-MAC	O	81.45	75.07	57.15	29.96	78.60	78.33	45.57	88.31	86.39	69.60	44.97	93.57	96.86	84.71			
	C	42.70	37.15	30.14	14.87	35.59	36.14	20.43	34.15	29.88	27.37	12.48	18.57	18.86	12.43	61.80%		
	P	37.60	32.33	26.99	14.49	35.15	35.29	20.57	23.76	21.02	20.12	9.21	13.86	15.57	9.86	66.94%		
	L	35.57	29.83	24.97	13.13	32.79	32.29	19.71	25.38	22.13	20.99	9.23	15.29	17.14	10.43	67.96%		
V-GeM	O	85.24	76.43	59.17	32.26	80.52	81.29	49.71	86.28	84.66	67.06	42.40	95.14	97.57	83.00			
	C	46.08	38.98	31.59	14.20	36.45	36.29	19.57	44.51	38.05	34.44	15.39	27.14	27.29	17.57	57.60%		
	P	43.71	37.84	30.92	15.36	36.76	37.00	21.86	30.92	28.12	25.78	11.91	17.43	17.43	12.86	62.64%		
	L	41.94	37.13	30.00	15.39	34.40	34.00	21.43	32.29	27.39	25.95	11.69	16.86	16.86	10.86	63.72%		
R-MAC	O	81.69	73.85	56.14	29.80	78.33	79.86	46.57	83.55	81.56	63.91	39.06	93.52	96.71	79.57			
	C	58.52	50.65	37.50	15.59	56.47	54.29	24.71	67.57	61.51	49.43	25.01	70.00	72.43	49.57	31.27%		
	P	35.31	30.34	24.73	13.37	36.62	36.43	20.71	35.66	32.61	27.23	12.12	32.57	34.86	21.29	59.71%		
	L	34.08	28.68	23.30	12.09	34.26	32.95	19.86	34.63	30.71	26.16	11.50	28.00	29.71	18.43	62.60%		
R-GeM	O	86.24	80.63	63.13	38.51	82.72	83.14	54.57	90.66	90.33	74.06	51.69	94.96	98.29	88.29			
	C	68.45	59.30	45.57	21.38	66.25	62.52	34.86	79.00	73.48	59.05	33.36	84.00	87.00	68.71	23.76%		
	P	34.81	30.50	24.33	13.79	28.97	28.43	19.71	33.76	31.67	26.54	11.28	27.86	29.43	17.00	66.69%		
	L	31.73	29.21	23.17	13.01	27.21	27.29	18.00	32.07	29.60	25.18	10.35	27.86	28.86	16.14	68.47%		

Table 1. The attack results with different relationships: Original Results (O), Label-wise (C), Pair-wise (P), and List-wise (L). We evaluate the performance with six retrieval models on four evaluated datasets. The ROxford5k along with RParis6k is annotated with three protocol setups: Easy (E), Medium (M), Hard (H). Lower mAP or mP@10 and higher mDR (mean dropping rate) mean better performance in attack.

	A-MAC	A-GeM	V-MAC	V-GeM	R-MAC	R-GeM
A-MAC	48.33	34.94	13.60	10.78	8.57	11.27
A-GeM	38.18	56.88	14.31	12.00	7.64	12.22
V-MAC	14.68	15.26	67.96	60.16	18.46	19.32
V-GeM	15.66	16.30	66.16	63.72	18.24	19.87
R-MAC	16.38	15.53	23.59	19.62	62.60	58.25
R-GeM	14.27	14.29	23.94	22.35	67.91	68.47

Table 2. Results of transfer attack. The mean dropping rates are reported, where a larger number means better attack performance.

a classification model to obtain the label-wise relationship. Our universal perturbations are trained on 1,691 query images from the *SfM-30k*.

Test Datasets. The *Oxford5k* dataset [32] consists of 5,062 images and the collection has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each of which is represented by 5 possible queries. Similar to *Oxford5k*, the *Paris6k* dataset [32] consists of 6,412 images with 55 queries. Recently, Radenović *et al.* [32] have revisited these two datasets to revise the annotation error, the size of the dataset, and the level of challenge. The *Revisited Oxford5k* and *Revisited Paris6k* datasets are referred as ROxford5k and RParis6k, respectively. We report our results on both the original and revisited datasets.

Visual Features. For CNN-based image representation, we use AlexNet (A) [20], VGG-16 (V) [38] and ResNet101 (R) [13] pre-trained on ImageNet [10] as our base models to fine-tune the CNN models on the *SfM-120k* dataset. For the fine-tuned features, we consider two cutting-edge features,

i.e., the generalized mean-pooling (GeM) [33] and the max-pooling (MAC) [34, 40]. As a result, we obtain a total of 6 features to evaluate the attack performance, termed as *A-GeM*, *V-GeM*, *R-GeM*, *A-MAC*, *V-MAC* and *R-MAC*.

Evaluation Metrics. To measure the performance of universal perturbation for retrieval, we mainly consider three evaluation metrics, *i.e.*, mAP, mP@10, and the fooling rate. Unlike classification, the fooling rate of top-1 label prediction can not be computed directly for image retrieval. Therefore, we define a new metric to evaluate the fooling rate for retrieval, termed dropping rate (DR) as follows:

$$DR(M, x, \hat{x}) = \frac{M(x) - M(\hat{x})}{M(x)} \times 100\%, \quad (9)$$

where \hat{x} is an adversarial example of the original feature x , and M is the metric used in retrieval such as mAP. Dropping rate characterizes the attack performance by measuring the performance degeneration of retrieval systems. The higher the dropping rate is, the more successful the attack is.

4.1. Results of UAP Attack

We evaluate the performance of six state-of-the-art deep visual representations against universal adversarial perturbation, the quantitative results of mean DR, mAP and mP@10 are shown in Tab. 1. Poor dropping rates (except ones for VGG16) prove limited ability of UAPs against classification on retrieval. Although they achieve considerable results for VGG16, they are still worse than our pro-

	Random	Pre-trained	Distillation
A-GeM	5.53%	32.98%	39.72%
V-GeM	1.66%	28.85%	44.68%

Table 3. Results about distillation attack. Random refers to Perturbations on randomly initialized model, pre-trained means the one from model trained on the ImageNet dataset, and distillation is obtained via distilled model.

posed methods. Clearly, for all deep visual features, all kinds of our universal perturbations achieve very high dropping rates on the validation set. Most of them achieve a dropping rate of more than 50%, which means that most relevant images will not be returned on top of the ranking list. Specifically, the universal perturbations computed for V-MAC and R-GeM achieve nearly 68% dropping rate. Notably, list-wise relationship plays an important role in generating universal perturbations. We owe it to more ranking information employed during optimization. We conclude that both the pair-wise and list-wise relationships are suitable for universal perturbation generation, and list-wise relationship achieves better performance.

4.2. Results of Transfer Attack

As mentioned in Sec. 2, transfer attack is to fool models or dataset with a perturbation generated on another model or dataset. Tab. 2 shows the results about the transfer attack across different visual features, in which we report the mDR calculated on all four evaluation datasets. Each row in Tab. 2 shows the mDR s for perturbation crafted by a given model, and each column shows the transfer dropping rates on the target model. The universal perturbation is trained on one architecture (*e.g.*, V-GeM), whose attack ability is evaluated to fool the retrieval system based on the other deep features (*e.g.*, R-MAC or V-MAC²). It is interesting to find that universal perturbations generated from the same network architecture can be transferred well to related models with different pooling methods.

We also measure the power of distillation in Tab. 3 for the case that the architecture is known beforehand. It’s clear that perturbations from randomly initialized models make no sense in spite of using the same architecture. As all the retrieval models are fine-tuned from the ImageNet pre-trained models, perturbations generated from pre-trained models achieve considerable results compared with transfer attack from other architectures in Tab. 2. However, perturbations from distilled model overmatch ones from pre-trained models by at least 6%, showing the power of ranking distillation. We conclude that our proposed ranking distillation attack is practical, when the model parameters can not be touched.

²We consider that different CNN architecture with the same pooling method as different features.

Range	[362, 362]	[1024, 1024]	[128, 1024]	[256, 1024]	[512, 1024]	[768, 1024]
A-GeM	16.89%	24.69%	53.21%	56.88%	51.41%	39.21%
V-GeM	25.87%	30.42%	61.93%	63.72%	55.02%	42.08%

Table 4. The effect of resizing in attack.

4.3. On the Effect of Resizing

As mentioned before, the retrieval system can accept various size of input image, which inspire us to investigate the effect of resizing when attacking the systems. Quantitative results are shown in Tab. 4. We first set the resizing scale to a fixed 362×362 and 1024×1024 , considering that 362×362 is the scale used to training the retrieval model. The dropping rates for A-GeM and V-GeM are lower than half of our multi-scale random resizing method. Finally, we evaluate the influence of the range for our multi-scale random resizing and observe that too broad or narrow range damages attack performance.

4.4. Visualization

Fig. 4 shows the retrieval results for R-GeM features from the *Oxbuild5k* and *Paris6K* evaluation set. In details, to attack the label-wise relationship, the model aims to learn the perturbation to push the original image to other categories. In the second row, we observe that the top 5 retrieved images are relevant to the category of dogs, instead of the true category of building. This phenomenon exists for pair-wise relationship and list-wise relationship that both pursues the farthest landmark to some degree, *e.g.*, most retrieved images are relevant about sculptures or oil paintings. Note that, retrieved images for pair-wise relationship and list-wise are similar since list-wise relationship includes pair-wise information.

We then visualize the perturbations that are trained from different models in Fig. 5. Perturbations in the first row are generated from MAC pooling and the ones in second row are from GeM pooling. The first three perturbation each row generated from different networks show large difference, while perturbations from same column share similar appearances. This is consistent with the result of transfer attack. Besides, perturbations crafted from pair-wise relationship and list-wise relationship are more similar than the one from label-wise, which may also indicate the gap between attack of classification and retrieval.

4.5. The Real-world System Attack

Fig. 6 shows the attack results on a real-world image retrieval system, *i.e.*, Google Image. The even rows show the perturbed images along with the retrieved images and the predicted keywords provided by Google Image, which are completely different from the original ones at the odd rows. For example, the original input is categorized to monochrome, while the adversarial example changes to be tree. Note that it is unable to quantize the mAP drop due to



Figure 4. The visualization results on *Oxford5k* and *Paris6K* for ResNet101-GeM. All the images in red box are the queries, and the retrieved pictures are sorted from left to right. The 4 rows show the retrieval results by using the original images and perturbed images via the label-wise relationship, pair-wise relationship and list-wise relationship, respectively. (Best viewed in color.)

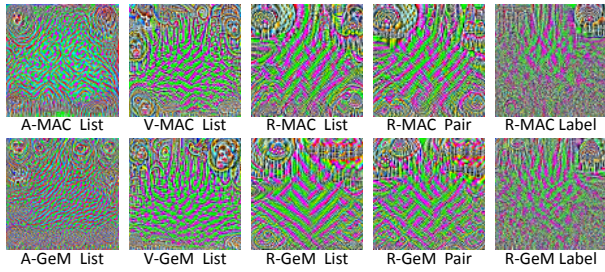


Figure 5. Universal adversarial perturbations crafted by the proposed method for multiple architectures trained on SfM. Corresponding features and deep architectures are mentioned below each image. (Best viewed in color and zoom in.)

the lack of ground truth ranking list. Therefore, we quantify how often the retrieved images from clean query are absent in the retrieved list of the corrupted one for 100 images randomly sampling from *Oxbuild5k* and *Paris6K* datasets. For this metric, our model has a 62.85% absent rate achieved. The attack results have demonstrated that the proposed method can generate universal perturbations to fool the real-world search engine.

5. Conclusion

In this paper, we are the first to propose a set of universal attack methods against image retrieval. We mainly focus on attacking the point-wise, pair-wise, and list-wise neighborhood relationships. We further analyze the impact of resizing operations in generating universal perturbation in details, and employ a multi-scale random resizing method to improve the success rate of the above attack schemes. A coarse-to-fine distillation strategy is also been proposed for black-box attack. We evaluate our proposed method on widely-used image retrieval datasets, *i.e.*, *Oxford5k*, and

Query	Similar Images List via Goolge Image	Keywords
Original 		monochrome
Adversarial 		tree
Original 		ashmolean museum
Adversarial 		palace

Figure 6. Example retrieval results on Google Images. The odd rows and even rows show the images retrieved by original query and the corrupted ones by our universal perturbation, respectively. The predicted keywords via Google Image are also given.

Paris6K, in which our method shows high attack performance that leads to a large retrieval metrics drop in a serial of models. Finally, we also attack the real-world system, *i.e.*, Google Images, which further demonstrates the efficacy of our methods. Last but not least, our work can therefore serve as an inspiration in designing more robust and secure retrieval models against the proposed attack schemes.

Acknowledgements. This work is supported by the National Key R&D Program (No.2017YFC0113000 and No.2016YFB1001503), Nature Science Foundation of China (No.U1705262, No.61772443, and No.61572410), Scientific Research Project of National Language Committee of China (No.YB135-49), and Nature Science Foundation of Fujian Province, China (No.2017J01125 and No.2018J01106).

References

- [1] Naveed Akhtar and Ajmal S Mian. Threat of adversarial attacks on deep learning in computer vision - A Survey. *IEEE Access*, 2018.
- [2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Computer Vision and Pattern Recognition*, 2016.
- [3] Ery Arias-Castro. Some theory for ordinal embedding. *Bernoulli*, 2017.
- [4] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *International Conference on Computer Vision*, 2015.
- [5] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European Conference on Computer Vision*, 2014.
- [6] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Black-box attacks on deep neural networks via gradient estimation. In *Workshop on International Conference on Learning Representations*, 2018.
- [7] Christopher J Burges, Robert Ragno, and Quoc V Le. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems*, 2007.
- [8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.
- [9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, 2017.
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009.
- [11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Computer Vision and Pattern Recognition*, 2018.
- [12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 2016.
- [14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*, 2015.
- [15] Noh Hyeonwoo, Araujo Andre, Sim Jack, Weyand Tobias, and Han Bohyung. Large-scale image retrieval with attentive deep local features. In *International Conference on Computer Vision*, 2017.
- [16] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.
- [17] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, 2008.
- [18] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition*, 2010.
- [19] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision*, 2016.
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 2012.
- [21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
- [22] Hong Liu, Rongrong Ji, Jingdong Wang, and Chunhua Shen. Ordinal constraint binary coding for approximate nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [23] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 2009.
- [24] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.
- [25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal Adversarial Perturbations. In *Computer Vision and Pattern Recognition*, 2017.
- [26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Computer Vision and Pattern Recognition*, 2016.
- [27] Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [28] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 2001.
- [29] Nicolas Papernot and Patrick McDaniel. Extending defensive distillation. *arXiv preprint arXiv:1705.05264*, 2017.
- [30] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM Asia Conference on Computer and Communications Security*, 2017.
- [31] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, 2016.
- [32] Filip Radenovic, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Computer Vision and Pattern Recognition*, 2018.

- [33] Filip Radenović, Giorgos Tolias, and Ondrej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [34] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 2016.
- [35] Stephen Robertson and Hugo Zaragoza. On rank-based effectiveness measures and optimization. *Information Retrieval*, 2007.
- [36] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 2013.
- [37] Johannes L Schonberger, Filip Radenovic, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3d reconstruction. In *Computer Vision and Pattern Recognition*, 2015.
- [38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.
- [40] Giorgos Tolias, Ronan Sifre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *International Conference on Learning Representations*, 2016.
- [41] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [42] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *International Joint Conferences on Artificial Intelligence*, 2018.
- [43] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.
- [44] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [45] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *The European Conference on Computer Vision*, 2018.