

Active Image Indexing

Pierre Fernandez^{1,2}, Matthijs Douze¹, Hervé Jégou¹, Teddy Furon²

¹Meta AI, FAIR ²Univ. Rennes, Inria, CNRS, IRISA

Abstract

Image copy detection and retrieval from large databases leverage two components. First, a neural network maps an image to a vector representation, that is relatively robust to various transformations of the image. Second, an efficient but approximate similarity search algorithm trades scalability (size and speed) against quality of the search, thereby introducing a source of error. This paper improves the robustness of image copy detection with *active indexing*, that optimizes the interplay of these two components. We reduce the quantization loss of a given image representation by making imperceptible changes to the image before its release. The loss is back-propagated through the deep neural network back to the image, under perceptual constraints. These modifications make the image more retrievable. Our experiments show that the retrieval and copy detection of activated images is significantly improved. For instance, activation improves by +40% the Recall@1 on various image transformations, and for several popular indexing structures based on product quantization and locality sensitivity hashing.

1 Introduction

The traceability of images on a media sharing platform is a challenge: they are widely used, easily edited and disseminated both inside and outside the platform. In this paper, we tackle the corresponding task of Image Copy Detection (ICD), *i.e.* finding whether an image already exists in the database; and if so, give back its identifier. ICD methods power reverse search engines, photography service providers checking copyrights, or media platforms moderating and tracking down malicious content (*e.g.* Microsoft’s [42] or Apple’s [35]). Image identification systems have to be robust to identify images that are edited (cropping, colorimetric change, JPEG compression ...) after their release [15, 55].

The common approach for content-based image retrieval reduces images to high dimensional vectors, referred to as *representations*. Early representations used for retrieval were hand-crafted features such as color histograms [48], GIST [36], or Fisher Vectors [41]. As of now, a large body of work on self-supervised learning focuses on producing discriminative representations with deep neural networks, which has inspired recent ICD systems. In fact, *all* submissions to the NeurIPS2021 Image Similarity challenge [38] exploit neural networks. They are trained to provide invariance to potential image transformations, akin to data augmentation in self-supervised learning.

Scalability is another key requirement of image similarity search: searching must be fast on large-scale databases, which exhaustive vector comparisons cannot do. In practice, ICD engines leverage approximate neighbor search algorithms, that trade search accuracy against scalability. Approximate similarity search algorithms speed up the search by *not* computing the exact distance between all representations in the dataset [26, 20]. First they lower the number of scored items by partitioning the representation space, and evaluate the distances of only a few subsets. Second, they reduce the computational cost of similarity evaluation with quantization or binarization. These mechanisms make indexing methods subject to the curse of dimensionality. In particular, in high-dimensional spaces, vector representations lie close to boundaries of the partition [5]. Since edited versions of an

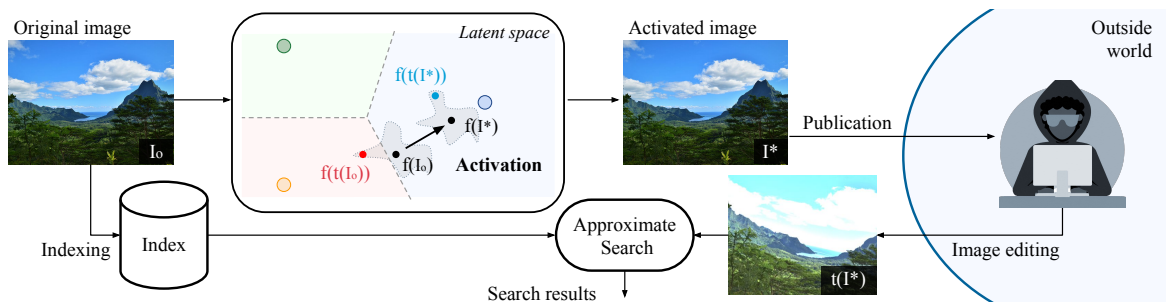


Figure 1: Overview of the method and latent space representation. We start from an original image I_o that can be edited $t(\cdot)$ in various ways: its feature extraction $f(t(I_o))$ spawns the shaded region in the embedding space. The edited versions should be recoverable by nearest neighbor search on quantized representations. In the regular (non-active) case, $f(I_o)$ is quantized by the index as \circ . When the image is edited, $t(I_o)$ switches cells and the closest neighbor returned by the index is the wrong one \circ . In active indexing: I_o is modified in an imperceptible way to generate I^* such that $f(I^*)$ is further away from the boundary. When edited copies $f(t(I^*))$ are queried, retrieval errors are significantly reduced.

original image have noisy vector representations, they sometimes fall into different subsets or are not quantized the same way by the index. All in all, it makes approximate similarity search very sensitive to perturbations of the edited image representations, which causes images to evade detection.

In this paper, we introduce a method that improves similarity search on large databases, provided that the platform or photo provider can modify the images before their release (see Fig. 1). We put the popular saying “attack is the best form of defense” into practice by applying image perturbations and drawing inspiration from adversarial attacks. Indeed, representations produced with neural networks are subject to *adversarial examples* [49]: small perturbations of the input image can lead to very different vector representations, making it possible to create adversarial queries that fool image retrieval systems [32, 51, 13]. In contrast, we modify an image to make it *more* indexing friendly. With minimal changes in the image domain, the image representation is pushed towards the center of the indexing partition, rising the odds that edited versions will remain in the same subset. This property is obtained by minimizing an indexation loss by gradient descent back to the image pixels, like for adversarial examples. For indexing structures based on product quantization [24], this strategy amounts to pushing the representation closer to its quantized codeword, in which case the indexation loss is simply measured by the reconstruction error. Since the image quality is an important constraint here, the perturbation is shaped by perceptual filters to remain invisible to the human eye.

Our contributions are:

- a new approach to improve ICD and retrieval, when images can be changed before release;
- an adversarial image optimization scheme that adds minimal perceptual perturbations to images in order to reduce reconstruction errors, and improve vector representation for indexing;
- experimental evidence that the method significantly improves index performance.

2 Preliminaries: Representation Learning and Indexing

For the sake of simplicity, the exposure focuses on image representations from SSCD networks [43] and the indexing technique IVF-PQ [24], since both are typically used for ICD. Extensions to other methods can be found in Sec. 5.4.

2.1 Deep descriptor learning

Metric embedding learning aims to learn a mapping $f : \mathbb{R}^{c \times h \times w} \rightarrow \mathbb{R}^d$, such that measuring the similarity between images I and I' amounts to computing the distance $\|f(I) - f(I')\|$. In recent works, f is typically a neural network trained with self-supervision on raw data to learn metrically meaningful representations. Methods include contrastive learning [9], self-distillation [19, 7], or masking random patches of images [21, 2]. In particular, SSCD [43] is a training method specialized for ICD. It employs the contrastive self-supervised method SimCLR [9] and entropy regularization [44] to improve the distribution of the representations.

2.2 Indexing

Given a dataset $\mathcal{X} = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$ of d -dimensional vector representations extracted from n images and a query vector x_q , we consider the indexing task that addresses the problem:

$$x^* := \operatorname{argmin}_{x \in \mathcal{X}} \|x - x_q\|. \quad (1)$$

This exact nearest neighbor search is not tractable over large-scale databases. Approximate search algorithms lower the amount of scored items thanks to space partitioning and/or accelerate the computations of distances thanks to quantization and pre-computation.

Space partitioning and cell-probe algorithms. As a first approximation, nearest neighbors are sought only within a fraction of \mathcal{X} : at indexing time, \mathcal{X} is partitioned into $\mathcal{X} = \bigcup_{i=1}^b \mathcal{X}_i$. At search time, an algorithm $Q : \mathbb{R}^d \rightarrow \{1, \dots, b\}^{k'}$ determines a subset of k' buckets in which to search, such that $k' = |Q(x_q)| \ll b$, yielding the approximation:

$$\operatorname{argmin}_{x \in \mathcal{X}} \|x - x_q\| \approx \operatorname{argmin}_{x \in \bigcup_{i \in Q(x_q)} \mathcal{X}_i} \|x - x_q\|. \quad (2)$$

A well known partition is the KD-tree [4] that divides the space along predetermined directions. Subsequently, locality sensitive hashing (LSH) [23, 18] and derivative [12, 40] employ various hash functions for bucket assignment, which implicitly partitions the space.

We focus on the popular clustering and Inverted Files methods [47], herein denoted by IVF. They employ a codebook $\mathcal{C} = \{c_i\}_{i=1}^k \subset \mathbb{R}^d$ of k centroids (also called “visual words” in a local descriptor context), for instance learned with k -means over a training set of representations. Then, Q associates x to its nearest centroid $q_c(x)$ such that the induced partition is the set of the k Voronoï cells. When indexing x , the IVF stores x in the bucket associated with $c_i = q_c(x)$. When querying x_q , IVF searches only the k' buckets associated to centroids c_i nearest to x_q .

Efficient metric computation and product quantization. Another approximation comes from compressed-domain distance estimation. Vector Quantization (VQ) maps a representation $x \in \mathbb{R}^d$ to a codeword $q_f(x) \in \mathcal{C} = \{C_i\}_{i=1}^K$. The function q_f is often referred to a *quantizer* and C_i as a *reproduction value*. The vector x is then stored as an integer in $\{1, \dots, K\}$ corresponding to $q_f(x)$. The distance between x and query x_q is approximated by $\|q_f(x) - x_q\|$, which is an “asymmetric” distance computation (ADC) because the query is not compressed. This leads to:

$$\operatorname{argmin}_{x \in \mathcal{X}} \|x - x_q\| \approx \operatorname{argmin}_{x \in \mathcal{X}} \|q_f(x) - x_q\|. \quad (3)$$

Binary quantizers (a.k.a. sketches, [8] lead to efficient computations but inaccurate distance estimates [58]. Product Quantization (PQ) [24] or derivatives [17] offer better estimates. In PQ, a vector $x \in \mathbb{R}^d$ is split into m subvectors in $\mathbb{R}^{d/m}$: $x = (x^1, \dots, x^m)$. The product quantizer then quantizes

the subvectors: $q_f : x \mapsto (q^1(x^1), \dots, q^m(x^m))$. If each subquantizer q^j has K_s reproduction values, the resulting quantizer q_f has a high $K = (K_s)^m$. The squared distance estimate is decomposed as:

$$\|q_f(x) - x_q\|^2 = \sum_{j=1}^m \|q^j(x^j) - x_q^j\|^2. \quad (4)$$

This is efficient since x is stored by the index as $q_f(x)$ which has $m \log_2 K_s$ bits, and since summands can be precomputed without requiring decompression at search time.

3 Active Indexing

Active indexing takes as input an image I_o , adds the image representation to the index and outputs an activated image I^* with better traceability properties for the index. It makes the feature representation produced by the neural network more compliant with the indexing structure. The activated image is the one that is disseminated on the platform, therefore the alteration must not degrade the perceived quality of the image.

Images are activated by an optimization on their pixels. The general optimization problem reads:

$$I^* := \operatorname{argmin}_{I \in \mathcal{C}(I_o)} \mathcal{L}(I; I_o), \quad (5)$$

where \mathcal{L} is an indexation loss dependent on the indexing structure, $\mathcal{C}(I_o)$ is the set of images perceptually close to I_o . Algorithm 1 and Figure 1 provide an overview of active indexing.

3.1 Image optimization dedicated to IVF-PQ (“activation”)

The indexing structure IVF-PQ involves a coarse quantizer q_c built with k-means clustering for space partitioning, and a fine product quantizer q_f on the residual vectors, such that a vector $x \in \mathbb{R}^d$ is approximated by $q(x) = q_c(x) + q_f(x - q_c(x))$.

We solve the optimization problem (5) by iterative gradient descent, back-propagating through the neural network back to the image. The method is classically used in adversarial example generation [49, 6] and watermarking [54, 16].

Given an original image I_o , the loss is an aggregation of the following objectives:

$$\mathcal{L}_f(x, q(x_o)) = \|x - q(x_o)\|^2 \quad \text{with } x_o = f(I_o), x = f(I) \quad (6)$$

$$\mathcal{L}_i(I, I_o) = \|I - I_o\|^2. \quad (7)$$

\mathcal{L}_i is a regularization on the image distortion. \mathcal{L}_f is the indexation loss that operates on the representation space. \mathcal{L}_f is the Euclidean distance between x and the target $q(x_o)$ and its goal is to push the image feature towards $q(x_o)$. With IVF-PQ as index, the representation of the activated image gets closer to the quantized version of the original representation, but also closer to the coarse centroid. Finally, the losses are combined as $\mathcal{L}(I; I_o) = \mathcal{L}_f(x, q(x_o)) + \lambda \mathcal{L}_i(I, I_o)$.

3.2 Perceptual attenuation

It is common to optimize a perturbation δ added to the image, rather than the image itself. The adversarial example literature often considers perceptual constraints in the form of an ℓ_p -norm bound

applied on δ ([33] use $\|\delta\|_\infty < \varepsilon = 8/255$). Although a smaller ε makes the perturbation less visible, this constraint is not optimal for the human visual system (HVS), *e.g.* perturbations are more noticeable on flat than on textured areas of the image (see App. A.2).

We employ a handcrafted perceptual attenuation model based on a Just Noticeable Difference (JND) map [59], that adjusts the perturbation intensity according to luminance and contrast masking. Given an image I , the JND map $H_{\text{JND}}(I) \in \mathbb{R}^{c \times h \times w}$ models the minimum difference perceivable by the HVS at each pixel and additionally rescales the perturbation channel-wise since the human eye is more sensible to red and green than blue color shift (see App. A for details).

The relation that links the image I sent to f , δ being optimized and the original I_o , reads:

$$I = I_o + \alpha \cdot H_{\text{JND}}(I_o) \odot \tanh(\delta), \quad (8)$$

with α a global scaling parameter that controls the strength of the perturbation and \odot the point-wise multiplication. Coupled with the regularization \mathcal{L}_i (6), it enforces that the activated image is perceptually similar, *i.e.* $I^* \in \mathcal{C}(I_o)$ as required in (5).

3.3 Impact on the indexing performance

Figure 1 illustrates that the representation of the activated image gets closer to the reproduction value $q(f(I_o))$, and farther away from the Voronoï boundary. This is expected to make image similarity search more robust because (1) it decreases the probability that $x = f(t(I_o))$ “falls” outside the bucket; and (2) it lowers the distance between x and $q(x)$, improving the PQ distance estimate.

Besides, by design, the representation stored by the index is invariant to the activation. Formally stated, consider two images I, J , and one activated version J^* together with their representations x, y, y^* . When querying $x = f(I)$, the distance estimate is $\|q(y^*) - x\| = \|q(y) - x\|$, so the index is oblivious to the change $J \rightarrow J^*$. This means that the structure can index passive and activated images at the same time. Retrieval of activated images is more accurate but the performance on passive images does not change. This compatibility property makes it possible to select only a subset of images to activate, but also to activate already-indexed images at any time.

4 Analyses

We provide insights on the method for IVF-PQ, considering the effects of quantization and space partitioning. For an image I whose representation is $x = f(I) \in \mathbb{R}^d$, \hat{x} denotes the representation of a transformed version: $\hat{x} = f(t(I)) \in \mathbb{R}^d$, and x^* the representation of the activated image I^* . For details on the images and the implementation used in the experimental validations, see Sec. 5.1.

4.1 Product quantization: impact on distance estimate

We start by analyzing the distance estimate considered by the index:

$$\|\hat{x} - q(x)\|^2 = \|x - q(x)\|^2 + \|\hat{x} - x\|^2 + 2(x - q(x))^\top (\hat{x} - x). \quad (9)$$

The activation aims to reduce the first term, *i.e.* the quantization error $\|x - q(x)\|^2$, which in turn reduces $\|\hat{x} - q(x)\|^2$. Figure 3 shows in blue the empirical distributions of $\|x - q(x)\|^2$ (passive) and $\|x^* - q(x)\|^2$ (activated). As expected the latter has a lower mean, but also a stronger variance. The variation of the following factors may explain this: *i)* the strength of the perturbation (due to the HVS modeled by H_{JND} in (8)), *ii)* the sensitivity of the feature extractor $\|\nabla_x f(x)\|$ (some features are easier to push than others), *iii)* the shapes and sizes of the Voronoï cells of PQ.

The second term of (9) models the impact of the image transformation in the feature space. Comparing the orange and blue distributions in Fig. 3, we see that it has a positive mean, but the shift is bigger for activated images. We can assume that the third term has null expectation for two reasons:

i) the noise $\hat{x} - x$ is independent of $q(x)$ and centered around 0, *ii*) in the high definition regime, quantification noise $x - q(x)$ is independent of x and centered on 0. Thus, this term only increases the variance. Since $x^* - q(x)$ has smaller norm, this increase is smaller for activated images.

All in all, $\|\hat{x}^* - q(x)\|^2$ has a lower mean but a stronger variance than its passive counterpart $\|\hat{x} - q(x)\|^2$. Nevertheless, the decrease of the mean is so large that it compensates the larger variance. The orange distribution in active indexing is further away from the green distribution for negative pairs, *i.e.* the distance between an indexed vector $q(x)$ and an independent query y .

4.2 Space partitioning: impact on the IVF probability of failure

We denote by $p_f := \mathbb{P}(q_c(x) \neq q_c(\hat{x}))$ the probability that \hat{x} is assigned to a wrong bucket by IVF assignment q_c . In the single-probe search ($k' = 1$), the recall (probability that a pair is detected when it is a true match, for a given threshold τ on the distance) is upper-bounded by $1 - p_f$:

$$R_\tau = \mathbb{P}(\{q_c(\hat{x}) = q_c(x)\} \cap \{\|\hat{x} - q(x)\| < \tau\}) \leq \mathbb{P}(\{q_c(\hat{x}) = q_c(x)\}) = 1 - p_f. \quad (10)$$

In other terms, even with a high threshold $\tau \rightarrow \infty$ (and low precision), the detection misses representations that ought to be matched, with probability p_f . It explains the sharp drop at recall $R = 0.13$ in Fig. 2. This is why it is crucial to decrease p_f . The effect of active indexing is to reduce $\|\hat{x} - q_c(x)\|$ therefore reducing p_f and increasing the upper-bound for R : the drop shifts towards $R = 0.32$.

This explanation suggests that pushing x towards $q_c(x)$ decreases even more efficiently p_f . This makes the IVF more robust to transformation but this may jeopardize the PQ search because features of activated images are packed altogether. In a way, our strategy, which pushes x towards $q(x)$, dispatches the improvement over the IVF and the PQ search.

5 Experimental Results

5.1 Experimental setup

Dataset. We use DISC21 [15] a dataset dedicated to ICD. It includes 1M reference images and 50k query images, 10k of which are true copies from reference images. A disjoint 1M-image set with same distribution as the reference images is given for training. Images resolutions range from 200×200 to 1024×1024 pixels (most of the images are around 1024×768 pixels).

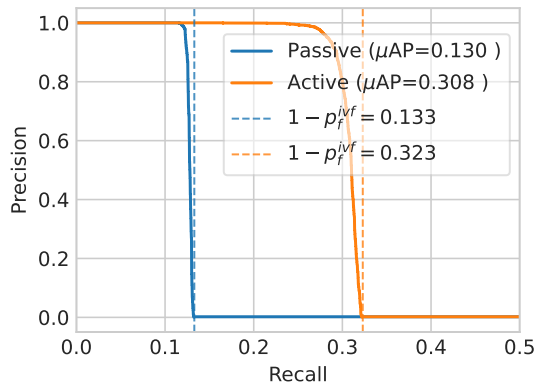


Figure 2: Precision-Recall curve for ICD with 50k queries and 1M reference images (more details for the experimental setup in Sec. 5.1). p_f^{IVF} is the probability of failure of the IVF (Sec. 4.2).

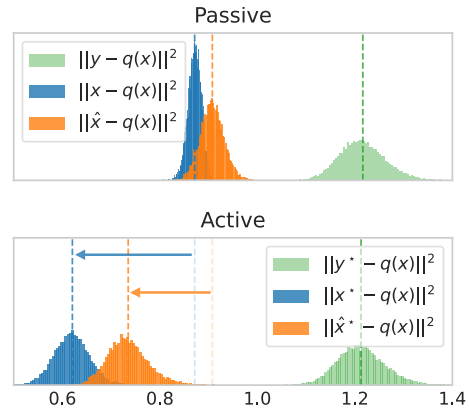


Figure 3: Distance estimates histograms (sec. 4.1). With active indexing, $\|x - q(x)\|^2$ is reduced (\leftarrow), inducing a shift (\leftarrow) in the distribution of $\|\hat{x} - q(x)\|^2$, where $t(I)$ a hue-shifted version of I . y is a random query.

The queries used in our experiments are *not* the queries in DISC21, since we need to control the image transformations in our experiments, and most transformations of DISC21 were done manually so they are not reproducible. Our queries are transformations of images *after active indexing*. These transformations range from simple attacks like rotation to more realistic social network transformations which created the original DISC21 queries (see App. B.1).

Metrics. For retrieval, our main metric is Recall 1@1 ($R@1$ for simplicity), which corresponds to the proportion of positive queries where the top-1 retrieved results is the reference image.

For copy detection, we use the same metric as the NeurIPS Image Similarity Challenge [15]. We retrieve the $k = 10$ most similar database features for every query; and we declare a pair is a match if the distance is lower than a threshold τ . To evaluate detection efficiency, we use the 10k matching queries above-mentioned together with 40k negative queries (*i.e.* not included in the database). We use precision and recall, as well as the area under the precision-recall curve, which is equivalent to the micro average precision (μAP). While $R@1$ only measures ranking quality of the index, μAP takes into account the confidence of a match.

As for image quality metric, we use the Peak Signal-to-Noise Ratio (PSNR) which is defined as $10 \log_{10} (255^2 / \text{MSE}(I, I')^2)$, as well as SSIM [56] and the norm $\|I - I'\|_{\infty}$.

Implementation details. The evaluation procedure is: (1) we train an index on the 1M training images, (2) index the 1M reference images, (3) activate (or not) 10k images from this reference set. (4) At search time, we use the index to get closest neighbors (and their distances) of transformed versions from a query set made of the 10k images.

Unless stated otherwise, we use a IVF4096,PQ8x8 index (IVF quantizer with 4096 centroids, and PQ with 8 subquantizers of 2^8 centroids), and use only one probe on IVF search for shortlist selection ($k' = 1$). Compared to a realistic setting, we voluntarily use an indexing method that severely degrades learned representations to showcase and analyze the effect of the active indexing. For feature extraction, we use an SSCD model with a ResNet50 trunk [22]. It takes image resized to 288×288 and generates normalized representations in \mathbb{R}^{512} . Optimization (5) is done with the Adam optimizer [27], the learning rate is set to 1, the number of iterations to $N = 10$ and the regularization to $\lambda = 1$. In (8), the distortion scaling is set to $\alpha = 3$ (leading to an average PNSR around 43 dB). In this setup, activating 128 images takes around 6s ($\approx 40\text{ms}/\text{image}$) with a 32GB GPU. It can be sped-up at the cost of some accuracy (see App. C.2).

5.2 Active vs. Passive

This section compares retrieval performance of active and passive indexing. We evaluate $R@1$ when different transformations are applied to the 10k reference images before search. The “Passive” lines of Tab. 1 show how the IVF-PQ degrades the recall. This is expected, but the IVF-PQ also accelerates search $500\times$ and the index is $256\times$ more compact, which is necessary for large-scale applications. Edited images are retrieved more often when they were activated for the index: increase of up to $+60 R@1$ for strong brightness and contrast changes, close to results of the brute-force search. We also notice that the performance of the active IVF-PQ $^{k'=1}$ is approximately the same as the one of the passive IVF-PQ $^{k'=16}$, meaning that the search can be made more efficient at equal performance. For the IVF-PQ † that does less approximation in the search (but is slower and takes more memory), retrieval on activated images is also improved, though to a lesser extent.

As for copy detection, Figure 2 gives the precision-recall curves obtained for a sliding value of τ , and corresponding μAP . Again, we observe a significant increase ($\times 2$) in μAP with active indexing. Note that the detection performance is much weaker than the brute-force search even in the active case because of the strong approximation made by space partitioning (more details in Sec. 4.2).

Table 1: Comparison of the index performance between activated and passive images. The search is done on a 1M image set and $R@1$ is averaged over 10k query images submitted to different transformations before search. **Random**: randomly apply 1 to 4 transformations. **Avg.**: average on the transformations presented in the table (details in App. B.2). **No index**: exhaustive brute-force nearest neighbor search. **IVF-PQ**: IVF4096,PQ8x8 index with $k'=1$ (16 for IVF-PQ¹⁶). **IVF-PQ[†]**: IVF512,PQ32x8 with $k'=32$.

	Search (ms)	Bytes/vector	Activated	Identity	Contr. 0.5	Contr. 2.0	Bright. 0.5	Bright. 2.0	Hue 0.2	Blur 2.0	JPEG 50	Rot. 25	Rot. 90	Crop 0.5	Resi. 0.5	Meme	Random	Avg.
No index	252	2048	✗	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.90	0.99
IVF-PQ	0.38	8	✗	1.00	0.73	0.39	0.73	0.28	0.62	0.48	0.72	0.07	0.14	0.14	0.72	0.14	0.13	0.45
			✓	1.00	1.00	0.96	1.00	0.92	1.00	0.96	0.99	0.10	0.50	0.29	1.00	0.43	0.32	0.75
IVF-PQ ¹⁶	0.42	8	✗	1.00	1.00	0.90	1.00	0.78	0.99	0.95	0.99	0.35	0.57	0.57	1.00	0.56	0.39	0.79
			✓	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	0.43	0.88	0.75	1.00	0.84	0.50	0.88
IVF-PQ [†]	1.9	32	✗	1.00	1.00	0.99	1.00	0.95	1.00	0.99	1.00	0.72	0.87	0.88	1.00	0.87	0.61	0.92
			✓	1.00	1.00	0.99	1.00	0.98	1.00	1.00	1.00	0.75	0.92	0.91	1.00	0.92	0.63	0.94

Example of activated images are given in Fig. 5 (more in App. E), while the qualitative image metrics are as follows: PSNR= 43.8 ± 2.2 dB, SSIM= 0.98 ± 0.01 , and $\|I - I'\|_\infty = 14.5 \pm 1.2$. These results are computed on 10k images, the \pm indicates the standard deviation.

5.3 Image quality trade-off

For a fixed index and neural extractor, the performance of active indexing mainly depends on the scaling α that controls the activated image quality. In Fig. 4, we repeat the previous experiment for different values of α and plot the μ AP against the average PSNR. As expected, lower PSNR implies better μ AP. For instance, at PSNR 30 dB, the μ AP is augmented threefold compared to the passive case. Indeed, for strong perturbations the objective function of (6) can be further lowered, reducing even more the gap between representations and their quantized counterparts.

5.4 Generalization

Generalization to other neural feature extractors. We first reproduce the experiment of Sec. 5.1 with different extractors, that cover distinct training methods and architectures. Among them, we evaluate a ResNext101 [60] trained with SSCD [43], a larger network than the ResNet50 used in our main experiments ; the winner of the descriptor track of the NeurIPS ISC, LYAKAAP-dt1 [62],

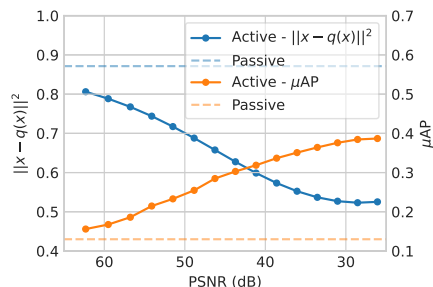


Figure 4: PSNR trade-off. As the PSNR decreases, the μ AP (orange) gets better, because the distance (blue) between activated representations x and $q(x)$ decreases.



Figure 5: Activated images. *Left*: reference from DISC (R000643.jpg and R000761.jpg), *middle*: activated image, *right*: pixel-wise difference.

Table 2: $R@1$ for different transformations before search. We use our method to activate images for indexing with IVF-PQ, with different neural networks used as feature extractors.

Name	Arch.	Activated	Identity	Contr. 0.5	Contr. 2.0	Bright. 0.5	Bright. 2.0	Hue 0.2	Blur 2.0	JPEG 50	Rot. 25	Rot. 90	Crop 0.5	Resi. 0.5	Meme	Random	Avg.
SSCD	ResNet50	✗	1.00	0.73	0.39	0.73	0.28	0.62	0.48	0.72	0.07	0.14	0.14	0.72	0.14	0.13	0.45
		✓	1.00	1.00	0.96	1.00	0.92	1.00	0.96	0.99	0.10	0.50	0.29	1.00	0.43	0.32	0.75
	ResNext101	✗	1.00	0.88	0.68	0.88	0.57	0.84	0.46	0.79	0.46	0.63	0.53	0.80	0.48	0.28	0.66
		✓	1.00	1.00	0.96	1.00	0.90	0.99	0.77	0.97	0.53	0.85	0.64	1.00	0.74	0.37	0.84
DINO	ResNet50	✗	1.00	0.66	0.65	0.65	0.52	0.71	0.52	0.82	0.07	0.20	0.51	0.84	0.62	0.18	0.57
		✓	1.00	0.99	0.88	0.99	0.75	0.93	0.72	0.94	0.08	0.25	0.57	0.99	0.82	0.23	0.72
	ViT-s	✗	1.00	0.89	0.71	0.86	0.64	0.75	0.74	0.90	0.14	0.18	0.57	0.88	0.61	0.25	0.65
		✓	1.00	0.99	0.94	0.99	0.92	0.98	0.89	0.99	0.15	0.28	0.63	0.99	0.77	0.32	0.77
ISC-dt1	EffNetv2	✗	1.00	0.25	0.08	0.16	0.01	0.51	0.54	0.84	0.18	0.16	0.23	0.79	0.16	0.18	0.36
		✓	1.00	0.57	0.16	0.33	0.01	0.88	0.79	0.97	0.20	0.24	0.29	0.97	0.26	0.26	0.49

that uses an EfficientNetv2 architecture [50]; networks from DINO [7], either based on ResNet50 or ViT [14], like the ViT-S model [52].

Table 2 presents the $R@1$ obtained on 10k activated images when applying different transformations before search. The $R@1$ is better for activated images for all transformations and all neural networks. The average improvement on all transformations ranges from +12% for DINO ViT-s to +30% for SSCD ResNet50.

Generalization to other indexes. The method easily generalizes to other types of indexing structures, the only difference being in the indexation loss \mathcal{L}_f (6). We present some of them below:

- **PQ and OPQ.** In PQ [24], a vector $x \in \mathbb{R}^d$ is approximated by $q_f(x)$. \mathcal{L}_f reads $\|x - q_f(x_o)\|$. In OPQ [17], vectors are rotated by matrix R before codeword assignment, such that $RR^\top = I$. \mathcal{L}_f becomes $\|x - R^\top q_f(Rx_o)\|$.
- **IVF.** Here, we only do space partitioning. Employing $\mathcal{L}_f = \|x - q_c(x_o)\|$ (“pushing towards the cluster centroid”) decreases the odds of x falling in the wrong cell (see Sec. 4.2). In this case, an issue can be that similar representations are all pushed together to a same centroid, which makes them less discriminate. Empirically, we found that this does not happen because perceptual constraint in the image domain prevents features from getting too close.
- **LSH.** Locality Sensitive Hashing maps $x \in \mathbb{R}^d$ to a binary hash $b(x) \in \mathbb{R}^L$. It is commonly done with projections against a set of vectors, which give for $j \in [1, \dots, L]$, $b_j(x) = \text{sign}(w_j^\top x)$. The objective $\mathcal{L}_f = -1/L \sum_j \text{sign}(b(x_o)) \cdot w_j^\top x$, allows to push x along the LSH directions and to improve the robustness of the hash.

Table 3 presents the $R@1$ and μAP obtained on the 50k query set. Again, results are always better in the active scenario. We remark that active indexing has more impact on space partitioning techniques: the improvement for IVF is higher than with PQ and the LSH binary sketches. As to be expected, the impact is smaller when the indexing method is more accurate.

Index	Search time	$R@1$ avg.		μAP	
		Passive	Activated	Passive	Activated
IVF 1024	0.32 ms	0.47	0.83	0.16	0.43
OPQ 8x8	5.71 ms	0.92	0.94	0.48	0.55
PCA64, LSH	0.99 ms	0.72	0.83	0.25	0.39

Table 3: $R@1$ averaged on transformations presented in Tab. 1 and μAP for different indexing structures

6 Related Work

Image watermarking hides a message in a host image, such that it can be reliably decoded even if the host image is edited. Early methods directly embed the watermark signal in the spatial or transform domain like DCT or DWT [11]. Recently, deep-learning based methods jointly train an encoder and a decoder to learn how to watermark images [66, 1, 63].

Watermarking is an alternative technology for ICD. Our method bridges indexing and watermarking, where the image is modified before publication. Regarding retrieval performance, active indexing is more robust than watermarking. Indeed, the embedded signal reinforces the structure naturally present in the original image, whereas watermarking has to hide a large secret keyed signal independent of the original feature. App. D provides a more thorough discussion and experimental results comparing indexing and watermarking.

Active fingerprint is more related to our work. As far as we know, this concept was invented by Voloshynovskiy *et al.* [53]. They consider that the image $I \in \mathbb{R}^N$ is mapped to $x \in \mathbb{R}^N$ by an invertible transform W such that WW^T . The binary fingerprint is obtained by taking the sign of the projections of x against a set of vectors $b_1, \dots, b_L \in \mathbb{R}^N$ (à la LSH). Then, they change x to strengthen the amplitude of these projections so that their signs become more robust to noise. They recover I^* with W^T . This scheme is applied to image patches in [28] where the performance is measured as a bit error rate after JPEG compression. Our paper adapts this idea from fingerprinting to indexing, with modern deep learning representations and state-of-the-art indexing techniques. The range of transformations is also much broader and includes geometric transforms.

7 Conclusion & Discussion

We introduce a way to improve ICD in large-scale settings, when images can be changed before release. It leverages an optimization scheme, similar to adversarial examples, that modifies images so that (1) their representations are better suited for indexing, (2) the perturbation is invisible to the human eye. We provide grounded analyses on the method and show that it significantly improves retrieval performance of activated images, on a number of neural extractors and indexing structures.

Activating images takes time (in the order of 10 ms/image) but one advantage is that the database may contain both active and passive images: active indexing does not spoil the performance of passive indexing and vice-versa. This is good for legacy compliance and also opens the door to flexible digital asset management strategies (actively indexing images of particular importance).

The method has several limitations. First, it is not agnostic to the indexing structure and extractor that are used by the similarity search. Second, an adversary could break the indexing system in several ways. In a black-box setting (no knowledge of the indexing structure and neural network extractor), adversarial purification [45] could get rid of the perturbation that activated the image. In a semi-white-box setting (knowledge of the feature extractor), targeted mismatch attacks against passive indexing like [51] may also work. Adversarial training [33] could be a defense. For instance, it is interesting to know if adversarial training prevents active indexing, or if the perceptual perturbation that is used in our method is still able to push features in the latent space of a robust and defended neural network.

Ethics Statement

Societal impact statement. Content tracing is a double-edged sword. On the one hand, it allows media platforms to more accurately track malicious content (pornographic, terrorist, violent images, e.g. Apple’s NeuralHash and Microsoft’s PhotoDNA) and to protect copyright (e.g. Youtube’s Content ID). On the other hand it can be used as a means of societal and political censorship, to restrict free speech of specific communities. However, we still believe that research needs to be advanced to improve global moderation in the internet. We also believe that advantages that a better copy detection could bring are more numerous than its drawbacks.

Environmental impact statement. We roughly estimated that the total GPU-days used for running all our experiments to 200, or ≈ 5000 GPU-hours. Experiments were conducted using a private infrastructure and we estimate total emissions to be in the order of a ton CO₂eq. Estimations were conducted using the MachineLearning Impact calculator presented in [30]. We do not consider in this approximation: memory storage, CPU-hours, production cost of GPUs/ CPUs, etc. as well as the environmental cost of training the neural networks used as feature extractors. Although the cost of the experiments and the method is high, it could possibly allow a reduction of the computations needed in large data-centers thanks to improved performance of indexing structures.

Reproducibility Statement

The implementation will be made available. Models used for feature extraction (SSCD, DINO, ISC-dt1) can be downloaded in their respective repositories. It builds upon the open-source Pytorch [39] and FAISS [26] libraries.

The main dataset used in the experiments (DISC21) can be freely downloaded on its webpage <https://ai.facebook.com/datasets/disc21-dataset/>. Dataset processing is described in App. B.1.

References

- [1] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 2020.
- [2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. *arXiv preprint arXiv:2204.07141*, 2022.
- [3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*. PMLR, 2018.
- [4] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
- [5] Christian Böhm, Stefan Berchtold, and Daniel A Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.
- [6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.
- [7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*. IEEE, 2021.

- [8] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.
- [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 2020.
- [10] Chun-Hsien Chou and Yun-Chin Li. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transactions on circuits and systems for video technology*, 5(6):467–476, 1995.
- [11] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007.
- [12] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.
- [13] Brian Dolhansky and Cristian Canton Ferrer. Adversarial collision attacks on image hashing functions. *arXiv preprint arXiv:2011.09473*, 2020.
- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [15] Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 image similarity dataset and challenge. *arXiv preprint arXiv:2106.09672*, 2021.
- [16] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.
- [17] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *CVPR*. IEEE, 2013.
- [18] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.
- [19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *NeurIPS*, 2020.
- [20] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *ICML*. PMLR, 2020.
- [21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*. IEEE, 2016.
- [23] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.
- [24] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

- [25] Qiuping Jiang, Zhentao Liu, Shiqi Wang, Feng Shao, and Weisi Lin. Towards top-down just noticeable difference estimation of natural images. *IEEE Transactions on Image Processing*, 2022.
- [26] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.
- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [28] Dimche Kostadinov, Slava Voloshynovskiy, Maurits Diephuis, and Taras Holotyak. Local active content fingerprinting: Optimal solution under linear modulation. In *ICIP*, 2016.
- [29] Lester E Krueger. Reconciling fechner and stevens: Toward a unified psychophysical law. *Behavioral and Brain Sciences*, 12(2):251–267, 1989.
- [30] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.
- [31] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.
- [32] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Who’s afraid of adversarial queries? the impact of image modifications on content-based image retrieval. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 306–314, 2019.
- [33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- [34] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *International Conference on Multimedia*. ACM, 2010.
- [35] NeuralHash. Apple. <https://www.apple.com/child-safety/pdf>, 2021.
- [36] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.
- [37] Zoe Papanikou and Joanna Bitton. Augly: Data augmentations for robustness. *arXiv preprint arXiv:2201.06494*, 2022.
- [38] Zoë Papanikou, Giorgos Toliás, Tomas Jeníček, Ed Pizzi, Shuhei Yokoo, Wenhao Wang, Yifan Sun, Weipu Zhang, Yi Yang, Sanjay Addicam, et al. Results and findings of the 2021 image similarity challenge. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 1–12. PMLR, 2022.
- [39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*. Curran Associates, Inc., 2019.
- [40] Loïc Paulevé, Hervé Jégou, and Laurent Amsaleg. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern recognition letters*, 31(11), 2010.
- [41] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, pages 3384–3391. IEEE, 2010.
- [42] PhotoDNA. Microsoft. <https://www.microsoft.com/en-us/photodna>, 2009.

- [43] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *CVPR*. IEEE, 2022.
- [44] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *ICML*, 2019.
- [45] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervised learning. In *ICLR*, 2021.
- [46] Richard Shin and Dawn Song. Jpeg-resistant adversarial images. In *NeurIPS Workshop on Machine Learning and Computer Security*, 2017.
- [47] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society, 2003.
- [48] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.
- [49] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2013.
- [50] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *ICML*. PMLR, 2021.
- [51] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *ICCV*. IEEE, 2019.
- [52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.
- [53] Sviatoslav Voloshynovskiy, Farzad Farhadzadeh, Oleksiy Koval, and Taras Holotyak. Active content fingerprinting: a marriage of digital watermarking and content fingerprinting. In *International Workshop on Information Forensics and Security (WIFS)*, pages 175–180. IEEE, 2012.
- [54] Vedran Vukotić, Vivien Chappelier, and Teddy Furon. Are classification deep neural networks good for blind image watermarking? *Entropy*, 2020.
- [55] Wenhao Wang, Yifan Sun, and Yi Yang. A benchmark and asymmetrical-similarity learning for practical image copy detection. *arXiv preprint arXiv:2205.12358*, 2022.
- [56] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on image processing*, 2004.
- [57] Andrew B Watson. Dct quantization matrices visually optimized for individual images. In *Human vision, visual processing, and digital display IV*, volume 1913, pages 202–216. SPIE, 1993.
- [58] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *NeurIPS*, 21, 2008.
- [59] Jinjian Wu, Leida Li, Weisheng Dong, Guangming Shi, Weisi Lin, and C-C Jay Kuo. Enhanced just noticeable difference model for images with pattern complexity. *IEEE Transactions on Image Processing*, 2017.
- [60] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*. IEEE, 2017.

- [61] XK Yang, WS Ling, ZK Lu, Ee Ping Ong, and SS Yao. Just noticeable distortion model and its applications in video coding. *Signal processing: Image communication*, 20(7):662–680, 2005.
- [62] Shuhei Yokoo. Contrastive learning with large memory bank and negative embedding subtraction for accurate copy detection. *arXiv preprint arXiv:2112.04323*, 2021.
- [63] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *NeurIPS*, 2020.
- [64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. IEEE, 2018.
- [65] Xiaohui Zhang, Weisi Lin, and Ping Xue. Just-noticeable difference estimation with pixels in images. *Journal of Visual Communication and Image Representation*, 19(1):30–41, 2008.
- [66] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, 2018.

Appendix - Active Image Indexing

A Details on the Perceptual Attenuation Model

A.1 Just Noticeable Difference map

The maximum change that the human visual system (HVS) cannot perceive is sometimes referred to as the just noticeable difference (JND) [29]. It is used in many applications, such as image/video watermarking, compression, quality assessment (JND is also used in audio).

JND models in pixel domain directly calculate the JND at each pixel location (*i.e.* how much pixel difference is perceivable by the HVS). The JND map that we use is based on the work of [10]. We use this model for its simplicity, its efficiency and its good qualitative results. More complex HVS models could also be used if even higher imperceptibility is needed ([57, 61, 65, 25] to cite a few). The JND map takes into account two characteristics of the HVS, namely the luminance adaptation (LA) and the contrast masking (CM) phenomena. We follow the same notations as [59].

The CM map \mathcal{M}_C is a function of the image gradient magnitude \mathcal{C}_l (the Sobel filter of the image):

$$\mathcal{M}_C(x) = 0.115 \times \frac{\alpha \cdot \mathcal{C}_l(x)^{2.4}}{\mathcal{C}_l(x)^2 + \beta^2}, \quad \text{with } \mathcal{C}_l = \sqrt{\nabla_x I(x)^2 + \nabla_y I(x)^2}, \quad (11)$$

where x is the pixel location, $I(x)$ the image intensity, $\alpha = 16$, and $\beta = 26$. It is an increasing function of \mathcal{C}_l , meaning that the stronger the gradient is at x , the more the image is masking a local perturbation, and the higher the noticeable pixel difference is.

LA takes into account the fact that the HVS presents different sensitivity to background luminance (*e.g.* it is less sensible in dark backgrounds). It is modeled as:

$$\mathcal{L}_A(x) = \begin{cases} 17 \times \left(1 - \sqrt{\frac{B(x)}{127}}\right) & \text{if } B(x) < 127 \\ \frac{3 \times (B(x) - 127)}{128} + 3 & \text{if } B(x) \geq 127, \end{cases} \quad (12)$$

where $B(x)$ is the background luminance, which is calculated as the mean luminance value of a local patch centered on x .

Finally, both effects are combined with a nonlinear additivity model:

$$H_{\text{JND}} = \mathcal{L}_A + \mathcal{M}_C - C \cdot \min\{\mathcal{L}_A, \mathcal{M}_C\}, \quad (13)$$

where C is set to 0.3 and determines the overlapping effect. For color images, the final RGB heatmap is $H_{\text{JND}} = [\alpha_R H, \alpha_G H, \alpha_B H]$, where $(\alpha_R, \alpha_G, \alpha_B)$ are inversely proportional to the mixing coefficients for the luminance: $(\alpha_R, \alpha_G, \alpha_B) = 0.072/(0.299, 0.587, 0.114)$.



Figure 6: A reference image I from DISC21 (R002815.jpg), and the associated perceptual heatmap $H_{\text{JND}}(I)$.

A.2 Comparison with ℓ_∞ Constraint Embedding

Figure 7 shows the same image activated using either the ℓ_∞ constraint (commonly used in the adversarial attack literature) or our perceptual constraint based on the JND model explained above. Even with very small ε (4 over 255 in the example bellow), the perturbation is visible especially in the flat regions of the images, such as the sea or sky.

[31] also show that the ℓ_∞ is not a good perceptual constraint. They use the LPIPS loss [64] as a surrogate for the HVS to develop more imperceptible adversarial attacks. Although a similar approach could be used here, we found that at this small level of image distortion the LPIPS did not capture CM and LA as well as the handcrafted perceptual models present in the compression and watermarking literature.

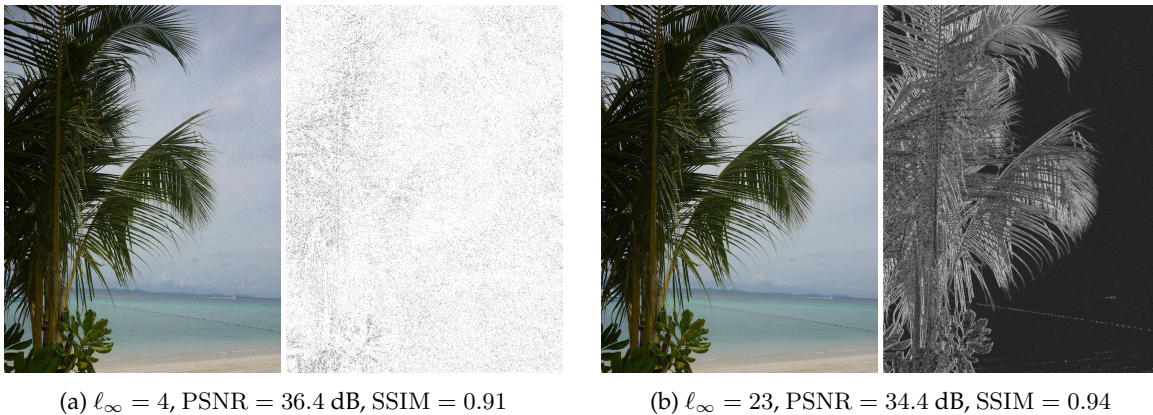


Figure 7: Activated images, either with (a) the $\ell_\infty \leq 4$ constraint or with (b) our perceptual model (best viewed on screen). We give the corresponding measures between the original and the protected image, as well as the pixel-wise difference. The perturbation on the right is much less perceptible thanks to the perceptual model, even though its ℓ_∞ distance with the original image is much higher.

B More Experiments Details

B.1 Dataset

The dataset DISC 2021 was designed for the Image Similarity Challenge [15] and can be downloaded in the dataset webpage: <https://ai.facebook.com/datasets/disc21-dataset/>.

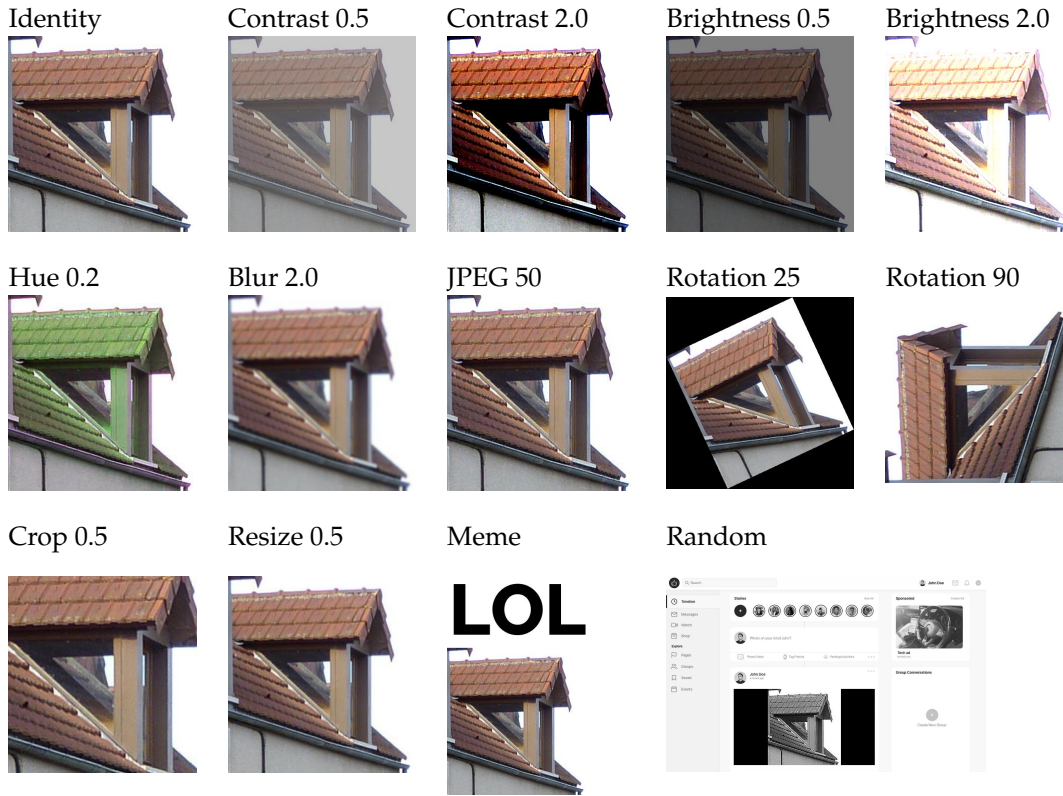
We want to test performance on edited versions of activated images but in DISC query set transformations are already applied to images. Therefore the query set cannot be used as it is.

We create a first test set “Ref10k” by selecting the 10k images from the reference set that were originally used to generate the queries (the “dev queries” from the downloadable version). We also re-create a query set “Query50k”. To be as close as possible, we use the same images that were used for generating queries in DISC. Edited images are generated using the AugLy library [37], following the guidelines given in the “Automatic Transformations” section of the DISC paper. Therefore, the main difference between the query set used in our experiments and the original one is that ours do not have manual augmentations.

B.2 Transformations seen at test time

They cover both spatial transformations (crops, rotation, etc.), pixel-value transformations (contrast, hue, jpeg, etc.) and “everyday life” transformations with the AugLy augmentations. All transforma-

Table 4: Illustration of all transformations evaluated in Tab. 1.



tions are illustrated in Fig. 4. The parameters for all transformations are the ones of the torchvision library [34], except for the crop and resize that represent area ratios. For the Gaussian blur transformation we use alternatively σ , the scaling factor in the exponential, or the kernel size k_b (in torchvision $k_b = (\sigma - 0.35)/0.15$). The “Random” transformation is the one used to develop the 50k query set. A series of simple 1-4 AugLy transformations are picked at random, with skewed probability for a higher number. Among the possible transformations, there are pixel-level, geometric ones, as well as embedding the image as a screenshot of a social network GUI.

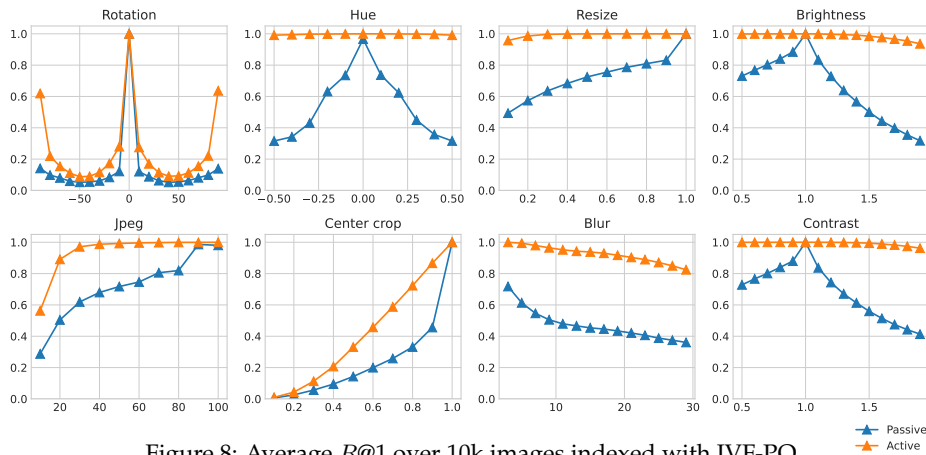


Figure 8: Average $R@1$ over 10k images indexed with IVF-PQ.

C More Experimental Results

C.1 Detailed metrics on different image transformations

On Fig. 8, we evaluate the average $R@1$ over the 10k images from the reference dataset. The experimental setup is the same as for Tab. 1 but a higher number of transformation parameters are evaluated. As expected, the higher the strength of the transformation, the lower the retrieval performance is. The decrease in performance is significantly reduced with activated images.

C.2 Additional ablations

Speeding-up the optimization. In our experiments, the optimization is done using 10 iterations of gradient descent, which takes approximately 40ms/image. If the indexation time is important (often, this is not the case and only the search time is), it can be reduced at the cost of some accuracy.

We activated 10k reference images, with the same IVF-PQ indexed presented in Sec. 5.2 with only one step of gradient descent with a higher learning rate. Activation times are computed on average. The $R@1$ results in Tab. 5 indicate that the speed-up in the image optimization has a small cost in retrieval accuracy. Specifically, it reduces the $R@1$ for unedited images. The reason is that the learning rate is too high: it can cause the representation to be pushed too far and to leave the indexing cell. This is why a higher number number of steps and a lower learning rate are used in practice. If activation time is a bottleneck, it can however be useful to use less optimization steps.

Table 5: $R@1$ for different transformations applied before search, with either 1 step at learning rate 10, or 10 steps at learning rate 1.

	Activation	Identity	Contr. 0.5	Contr. 2.0	Bright. 0.5	Bright. 2.0	Hue 0.2	Blur 2.0	JPEG 50	Rot. 25	Rot. 90	Crop 0.5	Resi. 0.5	Meme	Random	Avg:
Passive	-	1.00	0.73	0.39	0.73	0.28	0.62	0.48	0.72	0.07	0.14	0.14	0.72	0.14	0.13	0.45
lr=1 - 10 steps	39.8 ms/img	1.00	1.00	0.96	1.00	0.92	1.00	0.96	0.99	0.10	0.50	0.29	1.00	0.43	0.32	0.75
lr=10 - 1 step	4.3 ms/img	0.99	0.99	0.92	0.99	0.84	0.99	0.95	0.99	0.10	0.39	0.25	0.99	0.36	0.27	0.72

Data augmentation at indexing time and EoT. Expectation over Transformations [3] was originally designed to create adversarial attacks robust to a set of image transformations. We follow a similar approach to improve robustness of the marked image against a set of augmentations \mathcal{T} . At each optimization step, we randomly sample A augmentations $\{t_i\}_{i=1}^A$ in \mathcal{T} and consider the average loss: $\mathcal{L}_f = \sum_{i=1}^A \mathcal{L}(I, t_i; I_o)/A$. In our experiments, \mathcal{T} encompasses rotations, Gaussian blurs, color jitters and a differentiable approximation of the JPEG compression [46]. A is set to 8 and we always take the un-augmented image in the chosen set of augmentations.

We activated 10k reference images, with the same IVF-PQ as Sec. 5.2 with or without using EoT. Table 6 shows the average $R@1$ performance over the images submitted to different transformations before search. EoT brings a small improvement, specifically on transformations where base performance is low (e.g. rotation or crops here). However, it comes at a higher computational cost since each gradient descent iteration needs A passes through the network, and since fewer images can be jointly activated due to GPU memory limitations (we need to store and back-propagate through A transformations for every image). If the time needed to index or activate an image is not a bottleneck, using EoT can therefore be useful. Otherwise, it is not worth the computational cost.

Table 6: $R@1$ for different transformations applied before search, with or without EoT when activating the images.

	Activation	Identity	Contr. 0.5	Contr. 2.0	Bright. 0.5	Bright. 2.0	Hue 0.2	Blur 2.0	JPEG 50	Rot. 25	Rot. 90	Crop 0.5	Resi. 0.5	Meme	Random	Avg:
Without EOT	40 ms	1.00	1.00	0.96	1.00	0.92	1.00	0.96	0.99	0.10	0.50	0.29	1.00	0.43	0.32	0.75
With EOT	870 ms	1.00	1.00	0.95	1.00	0.92	1.00	0.95	0.99	0.14	0.64	0.33	1.00	0.45	0.33	0.76

D Active Indexing vs. Watermarking

Discussion. Watermarking and active indexing both modify images for tracing and authentication, however there are significant differences between them. Watermarking embeds arbitrary information into the image. The information can be a message, a copyright, a user ID, etc. In contrast, active indexing modifies it to improve the efficiency of the search engine. Watermarking also focuses on the control over the False Positive Rate of copyright detection, *i.e.* a bound on the probability that a random image has the same message as the watermarked one (up to a certain distance).

Although watermarking considers different settings than indexing methods, it could also be leveraged to facilitate the re-identification of near-duplicate images. In this supplemental section, we consider it to address a use-case similar to the one we address in this paper with our active indexing approach. In this scenario, the watermark encoder embeds binary identifiers into database images. The decoded identifier is then directly mapped to the image (as the index of a list of images).

Experimental setup. In the rest of the section, we compare active indexing against recent watermarking techniques based on deep learning.

- For indexing, we use the same setting as in Sec. 5.1 (IVF-PQ index with 1M reference images). When searching for an image, we look up the closest neighbor with the help of the index.
- For watermarking, we encode 20-bit messages into images, which allows to represent $2^{10} \approx 10^6$ images (the number of reference images). When searching for an image, we use the watermark decoder to get back an identifier and the corresponding image in the database.

Like before, we use $R@1$ as evaluation metric. For indexing, it corresponds to the accuracy of the top-1 search result. For watermarking, the $R@1$ also corresponds to the word accuracy of the decoding, that is the proportion of images where the message is perfectly decoded. Indeed, with 20-bit encoding almost all messages have an associated image in the reference set, so an error on a single bit causes a mis-identification (there is no error correction¹).

We use two state-of-the-art watermarking methods based on deep learning: SSL Watermarking [16], which also uses an adversarial-like optimization to embed messages, and HiDDeN [66], which encodes and decodes messages thanks to Conv-BN-ReLU networks. The only difference with the original methods is that their perturbation δ is modulated by the handcrafted perceptual attenuation model presented in App. A. This approximately gives the same image quality, thereby allowing for a direct comparison between active indexing and watermarking.

Results. Tab. 7 compares the $R@1$ when different transformations are applied before search or decoding. Our active indexing method is overall the best by a large margin. For some transformations, watermarking methods are not as effective as passive indexing, yet for some others, like crops for HiDDeN, the watermarks are more robust.

Table 7: $R@1$ for different transformations applied before search, when using either watermarking or active indexing. Results are averaged on 1k images. Best result is in **bold** and second best in *italic*.

	Identity	Contr. 0.5	Contr. 2.0	Bright. 0.5	Bright. 2.0	Hue 0.2	Blur 2.0	JPEG 50	Rot. 25	Rot. 90	Crop 0.5	Resi. 0.5	Meme	Random	Avg.
Passive indexing	1.00	0.73	0.39	0.73	0.28	0.62	0.48	0.72	0.07	0.14	0.14	0.72	0.14	0.13	0.45
Active indexing (ours)	1.00	1.00	0.96	1.00	0.92	1.00	0.96	0.99	0.10	0.50	<i>0.29</i>	1.00	0.43	0.32	0.75
SSL Watermarking [16]	1.00	<i>0.98</i>	<i>0.53</i>	<i>0.98</i>	<i>0.63</i>	<i>0.85</i>	0.13	0.00	0.00	<i>0.15</i>	0.11	0.00	<i>0.46</i>	0.07	0.42
HiDDeN ² [66]	<i>0.94</i>	0.87	0.36	0.85	0.55	0.00	<i>0.81</i>	0.00	0.00	0.00	0.92	0.44	0.77	<i>0.16</i>	<i>0.48</i>

¹In order to provide error correction capabilities, one needs longer messages. This makes it more difficult to insert bits: in our experiments, with 64 bits we observe a drastic increase of the watermarking bit error rate.

E More Qualitative Results

Figure 10 gives more examples of activated images from the DISC dataset, using the same parameters as in Sec. 5.2. The perturbation is very hard to notice (if not invisible), even in flat areas of the images because the perceptual model focuses on textures. We also see that the perturbation forms a regular pattern. This is due to the image (bilinear) resize that happens before feature extraction.

Figure 9 gives example of an image activated at several values of perturbation strength α of Eq. (8) (for instance, for $\alpha = 20$ the image has PSNR 27dB and for $\alpha = 1$ the image has PSNR 49dB). The higher the α , the more visible the perturbation induced by the activation is. Nevertheless, even with low PSNR values (< 35 dB), it is hard to notice if an image is activated or not.

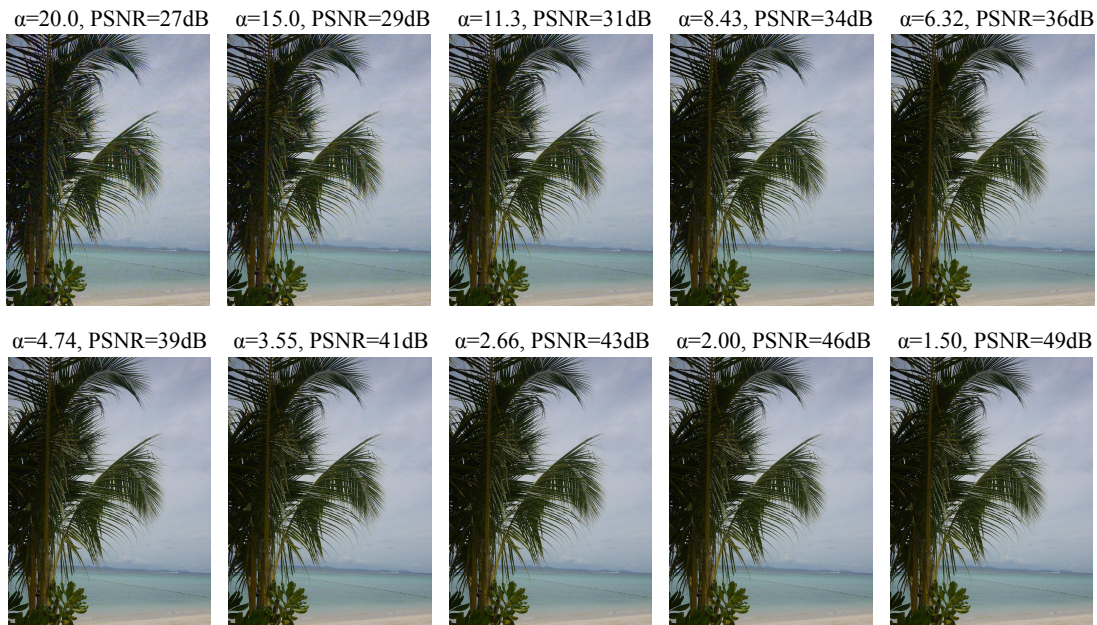


Figure 9: Example of one activated image at different levels of α .

²Our implementation. As reported in other papers from the literature, results of the original paper are hard to reproduce. Therefore to make it work better, our model is trained on higher resolution images (224×224), with a payload of 20-bits, instead of 30 bits embedded into 128×128 . Afterwards, the same network is used on images of arbitrary resolutions, to predict the image distortion which is later rescaled as in Eq. (8). In this setting the watermark can not always be inserted (6% failure).

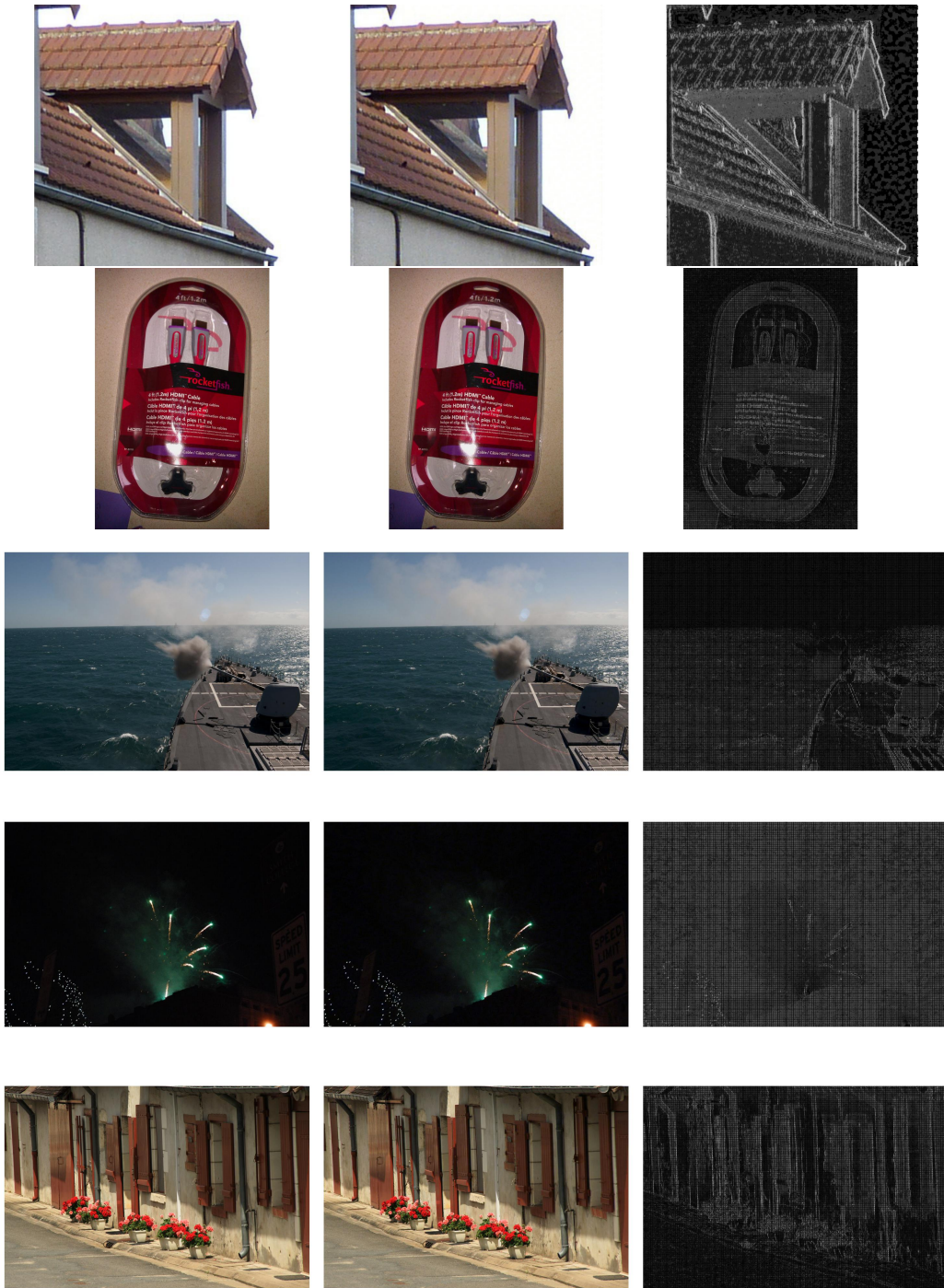


Figure 10: Example of activated images for $\alpha = 3.0$. (Left) original images, (Middle) activated images, (Right) pixel-wise difference. Images are R000005.jpg, R000045.jpg, R000076.jpg, R000172.jpg and R000396.jpg.