Deep Image Comparator: Learning to Visualize Editorial Change

Alexander Black[1] Tu Bui[1] Hailin Jin[2] Vishy Swaminathan[2] John Collomosse[1,2]
[1]CVSSP, University of Surrey [2]Adobe Research
{alex.black,t.v.bui}@surrey.ac.uk {hljin,vishy,collomos}@adobe.com

Abstract

We present a novel architecture for comparing a pair of images to identify image regions that have been subjected to editorial manipulation. We first describe a robust near-duplicate search, for matching a potentially manipulated image circulating online to an image within a trusted database of originals. We then describe a novel architecture for comparing that image pair, to localize regions that have been manipulated to differ from the retrieved original. The localization ignores discrepancies due to benign image transformations that commonly occur during online redistribution. These include artifacts due to noise and recompression degradation, as well as out-of-place transformations due to image padding, warping, and changes in size and shape. Robustness towards out-of-place transformations is achieved via the end-to-end training of a differentiable warping module within the comparator architecture. We demonstrate effective retrieval and comparison of benign transformed and manipulated images, over a dataset of millions of photographs.

1. Introduction

Images tell compelling stories, but are often manipulated to change those stories and spread misinformation. We urgently need tools to support users in making informed trust decisions on images encountered online.

This paper contributes a novel technique for matching an image circulating online to a trusted database of originals, and intuitively visualizing image regions that have been manipulated to differ from the original.

Robustly matching and comparing images is challenging, as image content may be also modified for non-editorial reasons during redistribution. Images are commonly subject to '*benign*' transformations such as changes in size, shape, quality or format by the platforms upon which they are shared. Images may also be '*manipulated*' for editorial reasons, including to alter or falsify their stories. We therefore cannot rely on cryptographic (bit-level) hashing for matching, nor can simple pixel difference operations be used to visualize changes due solely to manipulation. We propose two technical contributions:

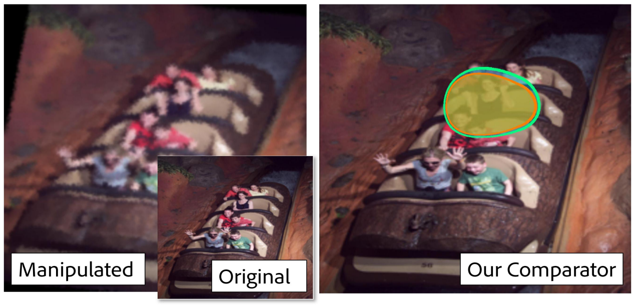


Figure 1. Our deep image comparator is trained to highlight the differences between a pair of images due to editorial manipulation (here, hands up vs. down), whilst ignoring change due to benign transformations of the image during online distribution (here, warping and blurring). Output in green, ground-truth in yellow.

Robust Near-Duplicate Image Search. We learn a visual search embedding that exhibits improved robustness to minor manipulations or benign modifications of images. We apply contrastive training to train a convolutional neural network (CNN) using a dataset of original photographs modified in Adobe Photoshop^{TM}, combined with data augmentations simulating benign image modifications. This yields a search embedding for robustly matching a near-duplicate *query* image circulating 'in the wild' to a trusted database of original images.

Deep Image Comparator. We propose a novel CNN architecture for comparing a pair of images, ignoring any non-editorial (benign) change. The network incorporates both a de-warping and image correlation module, and is trained end-to-end to ignore out-of-place transformation of content e.g. due to padding or warping as well as in-place corruption due to noise. Given a query, and an original image (retrieved from a trusted database via our near-duplicate image search) the network produces a *heatmap* that localizes visual discrepancies due to editorial manipulation (Fig. 1). Further, the network predicts a probability that the query image has undergone benign manipulation, editorial manipulation, or whether the input pair are completely different.

We show the image comparator to be effective both at discriminating between changes due to benign transformation and image manipulation, and at rejecting false positive results returned via the near-duplicate image search for a

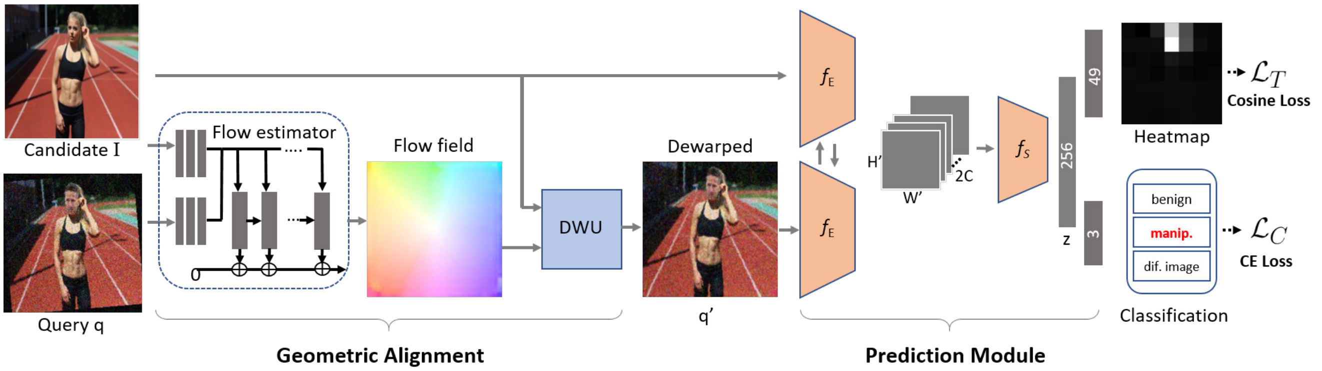


Figure 2. Architecture of the proposed image comparator network. A candidate match to the user queried image is obtained from near-duplicate search (subsec.3.1, not shown). Image alignment is performed via differentiable de-warping unit (DWU) based on a dense optical flow estimate provided by the flow estimator. The resulting image pair are separately encoded via a feature extractor $f_E(.)$ and the concatenated features passed through $f_S(.)$ to obtain the combined feature z, and via further MLP layers to output a heatmap and a 3-way classification score (indicating if the pair are different images, or that change is due to either manipulation or benign transformation).

corpus of millions of images.

2. Related Work

Visual content authenticity has been explored from the perspectives of both detection, and attribution.

Detection of visual tampering or generative ('deep fake') content [10] is typically a 'blind' detection problem. Given a single image, statistics may be learned to localize manipulated regions [34, 36], identify the use of a generative adversarial network (GAN) [35] or even determine (fingerprint) which GAN synthesized an image [38]. Detection of video manipulation similarly exploits temporal anomalies [37] or GAN limitations such as lack of blinking [26].

Image Attribution methods bind an image to data on its provenance, via embedded metadata [3, 11], watermarking [21, 9, 30, 1], or perceptual hashing [29, 27, 4, 22]. Emerging standards securely transport a cryptographically signed edit history within image metadata [3, 11]. Yet image metadata is often stripped by social platforms, and may be replaced to misattribute an image [7]. Watermarking methods similarly embed provenance information, within image content. Both metadata and watermarking methods may instead embed a link to a trusted database (in some cases a blockchain [6]) containing the provenance data.

Perceptual hashing also keys into a trusted database using a robust content-aware hash for visual similarity search [33]. Classical approaches sample the spectral domain using wavelets or DCT coefficients [40, 29]. More recently, deep learning has been applied to learn robust visual hashes. Deep Hashing Networks (DHNs) [41] extended an ImageNet-trained AlexNet [23] feature encoder [14] with a quantization loss, to obtain hashes that retained semantic discrimination. CSQ [39] treats hashing as a retrieval/attribution optimization problem. Both DHN and CSQ but require pairwise labels or semantic annotation unavailable in our use case. Deep Supervised Hashing (DSH) [27] and HashNet [4] train CNNs to learn visual hashes, using a siamese network and ranking loss; such losses are used extensively in visual search [13]. DSDH [25] learns metric ranking and classification directly from the hash code. Our approach is aligned in the sense that we also apply deep metric learning, but differs in that we use contrastive training [5] and data augmentation to learn invariances relevant to benign and editorial image transformation.

Localization of image manipulation focuses on blind detection tasks *e.g.* identifying image splicing [28] or use of photo-retouching tools [34]. Uniquely we approach the problem as a combination of perceptual hashing and pairwise comparison. Our image comparator (the second contribution of this paper) assumes that a trusted 'original' image may be first uncovered by a visual search (the first contribution). Our comparator learns to ignores discrepancies due to benign image transformations, but is sensitized to editorial manipulations. This is achieved through a differential optical flow [31] and dewarping module into our two-stream architecture. Two-stream networks have been employed to predict the kinds of edit operation applied to a pair of images [19]. We differ by producing a heatmap of edit operations, de-sensitized to particular transformation classes. A further feature of our method is a classification score also available at inference to determine whether an image is a benign or manipulated version, or a different image.

3. Methodology

We first describe the process to learn a representation for near-duplicate image search (subsec. 3.1). We assume the existence of a trusted database containing original images and their associated provenance information (*e.g.* curated by a trusted publisher, or via a decentralized immutable data-store such as a blockchain). Images are encoded into a 256-D feature embedding, and binarized into a 128-bit hash for scalable search [20]. The search is used to identify a shortlist of the most similar images to a users' query image.

Next, the top few hundred results are passed to our image comparator network (each in turn, paired with the query image). The classification branch of the comparator is used to cull false positive matches from the shortlist (subsec. 3.2). We then visualize the heatmap for the top ranked remaining result, which informs the user of differences between that image and the query that are due to image manipulation. The classification branch of the comparator is also used to indicate the likelihood of the query image being a manipulated version of that result.

3.1. Near-Duplicate Visual Search

Representation learning. We train a single CNN model to encode a whole image into a compact embedding space. The model architecture is ResNet50 with the N-way classifier layer replaced by a 256-D fully connected (fc) layer as the embedding; an image I is encoded to descriptor $z = f(I) \in \mathbb{R}^{256}$. The model is initialized with the DeepAugMix pretrained weight [17] and trained with loss:

$$\mathcal{L}(z) = -\log \frac{\sum_i e^{d(z,z_i^+)/\tau} + \sum_t e^{d(z,z_t^+)/\tau}}{\sum_{z^-} e^{d(z,z^-)/\tau}} \quad (1)$$

$$\text{where } d(u,v) = \frac{E_b(u) \cdot E_b(v)}{|E_b(u)|\,|E_b(v)|} \quad (2)$$

where $E_b(.)$ is a fc layer which serves as a buffer between the embedding and loss; $d(u,v)$ measures the cosine similarity between the intermediate embeddings $E_b(u)$ and $E_b(v)$; τ is the contrastive temperature (per [5]). z_i^+ and z_t^+ refer to the embeddings of the benign-transformed and manipulated (also subjected to benign transformations) versions of image I respectively; while z^- is the embeddings of other images and its transformed versions in the mini-batch. Our loss resembles SimCLR [5] with 2 key differences: (i) our loss leverages multiple positive images for a given input image I instead of just one pair in [5]; and (ii) we adapt SimCLR in a near-duplicate retrieval problem treating manipulated images as positives. During training, we ensure there is at least one benign-transformed and one manipulated versions for any given image I in a mini batch.

Hashing. Although our 256-D embedding z is already compact, it is difficult to scale search to millions of images while retaining interactive speed. Inspired from [20], we further binarize the embedding features via a 2-step quantization:

$$b = q_1(z) + q_2(z - q_1(z)) \in \{0,1\}^D \quad (3)$$

where $q_1(.)$ is a coarse quantizer to allocate the feature z into one of several clusters, and $q_2(.)$ is a fine quantizer encoding the residual of z and its corresponding centroid. $q_1(.)$ behaves like an inverted list enabling search within a fraction of the database, while $q_2(.)$ delivers a compact binary code efficient for search in the Hamming space. We use KMeans with 1024 clusters for $q_1(.)$ also extending the search to nearby 10 clusters, and Product Quantization for $q_2(.)$ resulting in total a 128-bit descriptor.

3.2. Detecting and Localizing Editorial Change

We propose an Image Comparator Network (ICN) that accepts a pair of images as input: the query image, and an original image retrieved by the near-duplication search model (subsec. 3.1). The ICN outputs both: (i) a heatmap highlighting areas of the pair that mismatch due to manipulation; and (ii) a 3-way classification, indicating the probability that the query image has been subject to benign transformation (non-editorial change), manipulation (editorial change), or that they are completely different (distinct).

The ICN architecture consists of 2 modules: a geometrical alignment module, $\mathcal{F}_A$, followed by a prediction module, $\mathcal{F}_P$ (Fig. 2). Below we describe our designs for $\mathcal{F}_A$ and $\mathcal{F}_P$ as well as the learning objectives.

Geometric alignment module $\mathcal{F}_A$. In practice, the query (hereafter, q) may undergo through arbitrary transformations some of which alter the pixel placement *e.g.* affine transformations or padding. It is important to correct its alignment prior to detection. $\mathcal{F}_A$ comprises an optical flow estimator and a de-warping unit (DWU). We build our flow estimator based on RAFT [31] which was originally designed to the estimate optical flow between video frames; here we determine the alignment between two images instead. Supposed the query image q and the retrieved original image I are both resized to a fixed height (H) and width (W), RAFT identifies a dense pixel displacement field $\{\rho^x, \rho^y\} \in \mathbb{R}^{H\times W}$ from q to I by computing correlation between the per-pixel features from all pairs of pixels. See [31] for detail.

Our DWU then applies the predicted optical flow to the query for the best alignment to the candidate image:

$$M \ : \ (x,y) \mapsto (x + \rho^x(x), y + \rho^y(y)) \quad (4)$$

$$\text{DWU}\,(q|\rho^x,\rho^y) = \mathcal{S}(M) \in \mathbb{R}^{H\times W} \quad (5)$$

where (x,y) refers to the pixel coordinates in the query q which are mapped into its estimated correspondence M according to the optical flow $\{\rho^x,\rho^y\}$. $\mathcal{S}(.)$ is a bilinear sampler that effectively fits a local grid around M: $\mathcal{S}(M) = \{M + \Delta M | \Delta M \in \mathbb{R}^2, |\Delta M| <= 1\}$ where output coordinates are computed by linear interpolation.

Prediction module $\mathcal{F}_P$. Given the candidate I and the aligned query, $q' = \mathcal{F}_A(q|I)$, we first extract local features of each image using a shared CNN module:

$$z_q = f_E(q');\ z_I = f_E(I) \in \mathbb{R}^{H'\times W'\times C} \quad (6)$$

where H', W' and C are the new height, width and feature dimension respectively. Our feature extractor $f_E(.)$ is 3 convolution layers separated by ReLU, batch norm and

Method	Benign			Manip			Manip+Benign			Average		
	IR@1	IR@10	IR@100	IR@1	IR@10	IR@100	IR@1	IR@10	IR@100	IR@1	IR@10	IR@100
Ours (stage 1 + 2)	**0.9003**	**0.9331**	**0.9408**	**0.9206**	0.9300	0.9300	**0.8028**	**0.8312**	**0.8412**	**0.8745**	**0.8981**	**0.9039**
Ours (stage 1)	0.7585	0.8788	0.9408	0.8108	0.8817	0.9300	0.5723	0.7323	0.8412	0.7138	0.8309	0.9039
MSResNet fine.	0.7203	0.8499	0.9202	0.8162	0.8905	0.9314	0.5582	0.7147	0.8289	0.6982	0.8184	0.8935
ImageNet fine.	0.6691	0.8226	0.9084	0.7858	0.8676	0.9128	0.4801	0.6614	0.7902	0.6450	0.7839	0.8705
DeepAugMix [17]	0.4899	0.5907	0.6743	0.8885	**0.9402**	0.9611	0.3828	0.4962	0.5965	0.5870	0.6757	0.7439
MSResNet [24]	0.1548	0.2567	0.3885	0.8807	0.9358	**0.9662**	0.1020	0.1898	0.3108	0.3792	0.4608	0.5552
ImageNet [15]	0.220	0.3113	0.4056	0.8797	0.9260	0.9527	0.1640	0.2498	0.3441	0.4212	0.4957	0.5675
CSQ [39]	0.0353	0.0873	0.2286	0.3274	0.4257	0.6179	0.0222	0.0611	0.1818	0.1283	0.1914	0.3428
HashNet [4]	0.0635	0.1274	0.2310	0.4611	0.5618	0.6726	0.0349	0.0797	0.1606	0.1865	0.2563	0.3547
pHash [40]	0.3218	0.3282	0.3321	0.3662	0.3743	0.3760	0.1529	0.1612	0.1658	0.2803	0.2879	0.2913

Table 1. Retrieval performance (on 2M images, PSBat-Ret) reported as IR score at ranks [1,10,100], for query images subjected to benign transforms, manipulation, or both. Stage 1 refers to nearest-neighbor search only. Stage 1+2 is the search reranked via the ICN classifier.

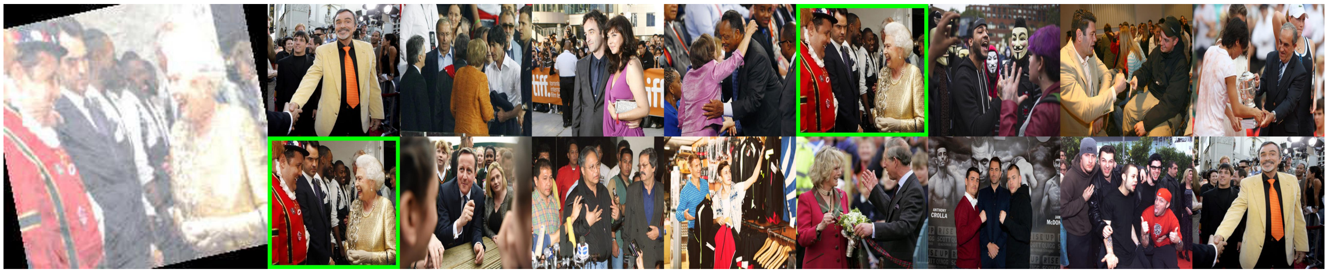


Figure 3. Re-ranking using ICN Classifier. The query image is shown left. Top: Stage 1 near-duplication retrieval (top 8 ranked results). Bot.: Stage 1+2 ranked results, due to rerank on 'distinct' score from ICN classifier. The correct result is promoted to rank 1.

max pooling. It outputs features at $\frac{1}{4}$ resolution ($H' = H/4, W' = W/4$ and we set $C = 128$). The combined features feed another CNN to learn a combined feature z:

$$z = f_S([z_q, z_I]) \in \mathbb{R}^{256} \tag{7}$$

where $[,]$ is a concatenation, and $f_S(.)$ is formed from 4 ResNet residual blocks [15] followed by average pooling and a FC layer outputting 256-D features.

Learning objectives. To predict the query-candidate relationship and visualize the possible manipulated regions, we apply two losses on top of the fusion feature z. The first loss is a 3-way cross entropy predicting whether the pair is benign (*i.e.* the query q is either identical or a benign transformed version of the candidate I), manipulated (*i.e.* z is a manipulated version of I) or of distinct images (*i.e.* z and q are two different instances):

$$c = E_c(z) \in \mathbb{R}^3 \tag{8}$$

$$\mathcal{L}_C = -\log \frac{e^{c_y}}{\sum_{i=1}^{3} e^{c_i}} \tag{9}$$

where $E_c(.)$ is a FC layer projecting z to a 3-D feature c, and y is the classification target of the pair (q, I).

The second loss minimizes the cosine distance between the manipulation heatmap derived from z and the ground truth heatmap. We produce the heatmap at resolution $t \times t$ from z via a FC layer, $E_t(z) \in \mathbb{R}^{t^2}$, and compute loss:

$$\mathcal{L}_T = 1 - \frac{E_t(z) \cdot T}{|E_t(z)|\,|T|} \tag{10}$$

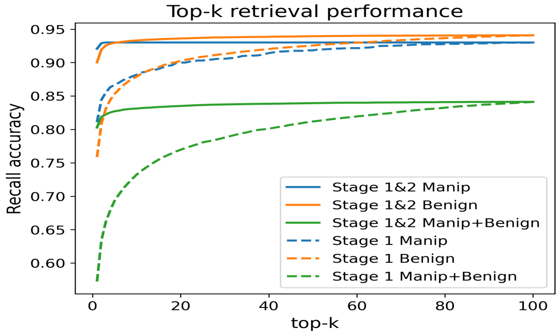


Figure 4. Recall accuracy versus top-k in retrieving: (i) benign; (ii) manipulated and (iii) both manipulated and transformed images.

where T is the ground truth manipulation heatmap. T is a matrix of zeros if the pair (q, I) is benign, ones if different (distinct), and if a manipulated pair $T \in [0, 1]$ derived from human ground truth annotations (subsec. 4.1). We define the output heatmap resolution $t = 7$ during training. At test time, the 7×7 heatmap is interpolated to the original resolution $H \times W$ and super-imposed on the query image. The heat map is continuous but can be thresholded for more intuitive visualization.

The total ICN loss is $\mathcal{L}(.) = w_c\mathcal{L}_C(.) + w_t\mathcal{L}_T(.)$ where loss weight $w_c = w_t = 0.5$ is set empirically.

3.3. Robust Search and Re-Ranking using ICN

Manipulation may introduce substantial change in an image. For a corpus of millions of images, our near-duplicate search model may not always retrieve the correct original image as the top ranked (top-1) result. Therefore we apply

Method	Accuracy (IoU)		Interp. (%)	
	T	No T	No T	T
ICN (Ours)	**0.551**	**0.563**	**77.8**	**85.2**
ResNetConv+Geo. Align. $\mathcal{F}_A$	0.243	0.243	7.84	3.87
ResNetConv [15]	0.238	0.239	5.88	2.58
SSD+Geo. Align. $\mathcal{F}_A$	0.231	0.231	1.31	3.23
SSD	0.149	0.154	3.27	0.65
ErrAnalysis+Geo. Align. $\mathcal{F}_A$	0.143	0.011	1.31	0.00
ErrAnalysis [36]	0.109	0.025	1.31	0.00
MantraNet+Geo. Align. $\mathcal{F}_A$	0.061	0.091	0.65	1.29
MantraNet [37]	0.027	0.036	0.65	3.23

Table 2. Evaluating heatmap accuracy and intrepretability for thresholded (T) and non-thresholded (No T) methods. Our proposed ICN is compared against baselines both objectively for accuracy (IoU) and subjectively via users to determine which exhibits best intrepretability (% method preference). $+\mathcal{F}_A$ indicates geometric alignment module applied.

a re-ranking to the top-k candidate images obtained from the initial (*stage 1*) near-duplicate retrieval search.

Typical visual search pipelines apply second stage (*stage 2*) processing via geometric verification (GV) to discard false positives within the top-k results (*stage 2*). This process is slow (typically up to one second per image doing GV via MLESAC [32]). For interactive search speeds, choice of a low k is therefore forced. Instead, we propose to use our ICN classifier for second stage processing. We re-rank our top-k results based on the probability of the image pair being distinct. Inference takes around 4 ms per pair, enabling larger k. As we later show (subsec. 4.5.2), the accuracy at detecting 'distinct' images is 98.95% and we pick $k = 100$.

4. Experiments and Discussion

We evaluate both the near-duplicate search and the performance of the ICN heatmaps and classification.

4.1. Datasets and Augmentation

We train and evaluate on PSBattles [16]; a dataset of images manipulated in Adobe Photoshop$^{\text{TM}}$, collected from the 'Photoshopbattles' forum on Reddit. The dataset contains more than 10k original images and, for each, of these, several manipulated variants; in total 102,028 variants contributed by 31K artists. To make our task challenging, we remove the original-manipulated pairs that are obviously different, retaining only similar pairs $\mathcal{D} = \{(O_i, P_i) | \, ||f(O_i) - f(P_i)||_2 < \beta\}$ where $f(.)$ is a pretrained ImageNet ResNet50 feature extractor and $\beta = 150$ is the distance threshold. That leaves 7,171 originals and 24,157 manipulated images. The data is split into training (**PSBat-Train**) and test (**PSBat-Test**) sets, the former has 6,364/21,197 and the latter has 807/2,960 original/manipulated images. The PSBat-Train is used to train both our image retrieval and ICN models (sec. 3.1-3.2) while PSBat-Test is used for the two benchmarks below.

PSBat-Ret. We construct a database of 807 original images from PSBat-Test plus 2 million diverse distractor images scraped from the Adobe Stock website. Next we created 3 query sets: (i) *Manip* contains 2,960 manipulated images in PSBat-Test; (ii) *Benign* contains 29.6k images created by transforming the PSBat-Test original images; and (iii) *Manip+Benign* also contains 29.6k images but via transforming the manipulated set instead. To obtain Benign and Manip+Benign, we applied a suite of benign transformations common in online image re-distribution. These include JPEG compression (40%-90%), random crop (90% area), padding (max 10% each side), rotation (max 15 degree), flipping and ImageNet-C [18] transformations containing various additive noise (*e.g.* Gaussian, shot, impulse noise) and blur (*e.g.* Gaussian, motion, defocus blur) and enhancement (*e.g.* brightness, contrast, snow) for all 5 severity levels in ImageNet-C [18]. We divide benign transformations to 3 groups: the *primary* group contains resize and JPEG re-compression; the *in-place* group contains in-place transformations from ImageNet-C transformations; and the *out-place* group contains those transformations that change pixel coordinates such as padding and affine warps. When transforming an image we apply all those in the primary group, followed by a random transformation in either the in-place group, or out-place group, or both.

PSBat-Pair. To evaluate ICN capability in detecting manipulation as well as generating heatmap, we create 4 evaluation sets out of PSBat-Test, each has 2,960 query-candidate image pairs. In the first set, *Benign*, each original (candidate) image is paired with a benign transformation of itself (the same transformation settings in PSBat-Ret are applied). The second set, *Manip*, has the queries made from the corresponding manipulated images. We also transform these manipulated images to create queries for the third set, *Manip+Benign*. In the fourth set, *Distinct*, each original image is paired with a random different image which is also subjected to benign transformations.

Training and evaluating ICN requires labelled manipulated regions. We identify these regions via crowd sourced annotation. For each original-manipulated pair in PSBat-Train/Test, 3 workers draw bounding boxes around the manipulated areas, obtaining a binary heatmap each $G^k = \{0,1\}^{H \times W}, k = 1,2,3$ where $G^k(x,y) = 1$ if pixel (x,y) is contained in a bounding box drawn by worker k. We normalize G^k w.r.t 7x7 image size and combine to a ground truth heatmap $T = \{\sum_k G^k(x,y)/3 \in \mathbb{R}^{7 \times 7}\}$ (Fig. 6).

4.2. Training Details

Near-Duplicate Search is trained on the PSBat-Train set (subsec. 3.1) for 20 epochs with SGD optimizer and learning rate starting from 1e^{-3} and step decays after 60% and 90% number of epochs. The training is terminated early if the loss stops increasing. To create a batch, we randomly sample 16 unique original images and 16 corresponding manipulated images, then create 2 transformation versions of each image via data augmentation. This leads to a batch

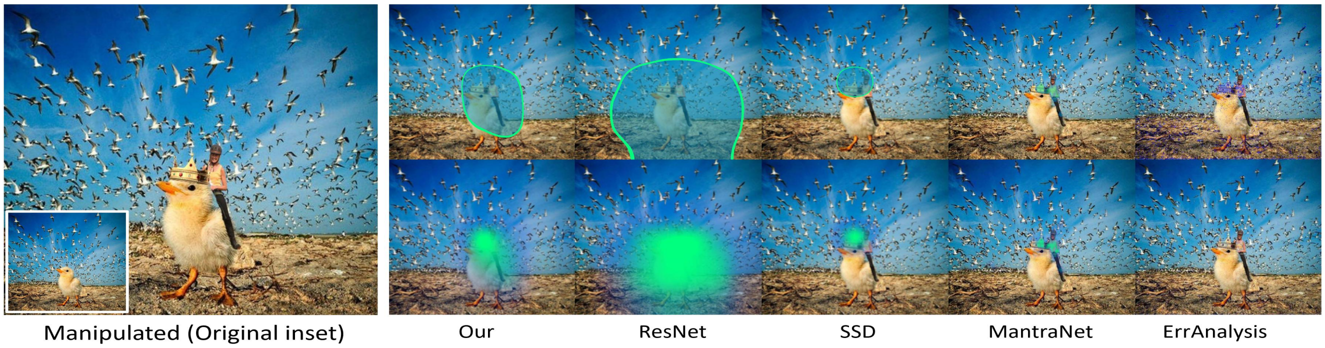


Figure 5. Comparison of heatmap visualizations (in green) from our ICN method, and baseline methods for thresholded (top) and non-thresholded (bot.) heatmaps. The heatmap visualizes manipulation of an image (crown/rider added on bird).

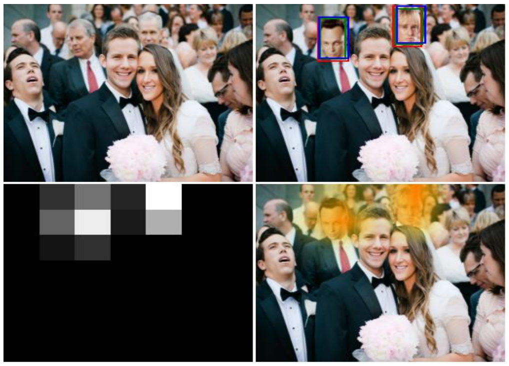


Figure 6. Examples of the crowd-annotation we collected on PS-Battles to identify ground-truth (g-t) manipulated regions: Original image (top-left); manipulated image with MTurk annotations via bounding boxes (top-right); 7x7 ground truth (bottom-left), manipulated superimposed with the g-t heatmap (bottom-right).

of total size of 64 whose every image has 3 other positives and the rest is negatives to be fed to our contrastive learning eq. 1. Augmentation is via random ImageNet-C (in-place) and padding/affine warps (out-of-place), see subsec. 4.1.

Image Comparator Network (ICN) is trained on PSBat-Train. The flow estimation (RAFT) sub-module is initialised with weights pre-trained on the KITTI[12] dataset. We construct a training batch by randomly sample 8 original images from PSBat-Train, then for each image I we pair it with an image randomly selected from three sources with equal probabilities: (i) a manipulated version of I, (ii) an original image different/distinct from I and (iii) the image I itself. We then apply random augmentation on the resulting 8 pairs (per our retrieval model). We train end-to-end using ADAM and learning rate 10^{-4}.

4.3. Metrics

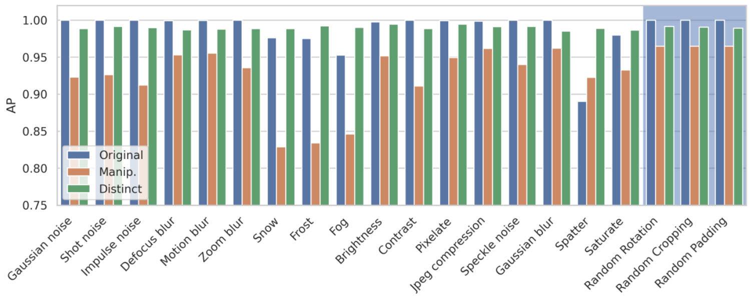
To evaluate near duplicate search, we use Instance Retrieval IR@k metric which measures the ratio of queries that returns the relevant images within top-k retrieval. Formally, $IR@k = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} r(q_i, j)$ where N is number of queries, relevance function $r(q_i, j) = 1$ if the returned image at rank j is relevant to the query q_i (there is only one such image in PSBat-Ret), otherwise 0.

To evaluate our ICN, we use Average Precision (AP) to measure the accuracy of our ICN classifier (eq. 8-9). For the generated heatmap, we up-sample the 7x7 heatmap to the image resolution $H \times W$, convert to binary with a threshold and compute Intersection over Union (IoU) with the ground truth, IoU $= \frac{1}{N}\sum_{i=1}^{N}\frac{S(U_i)\cap T_i}{S(U_i)\cup T_i}$ where T_i is the $H \times W$ binary ground truth heatmap, U_i is the predicted heatmap after interpolation and thresholding. We leverage the image pair classification result to improve the heatmap with $S(U_i) = U_i$ if the query is classified as manipulated, $\{0\}^{H\times W}$ if benign and $\{1\}^{H\times W}$ if distinct.

4.4. Evaluating Near-Duplicate Search

We compare our retrieval method (both stages) against 8 baselines. **ImageNet** [15], **MSResNet** [24] and **DeepAugMix** [17] are 3 public pretrained CNN models, all use ResNet50 architecture. The classic ImageNet model is trained on ILSVRC2012 [8], MSResNet is built by Microsoft to power its Bing image search engine while DeepAugMix reports state of art performance on the ImageNet-C benchmark. **ImageNet fine.** and **MSResNet fine.** are the finetuned models on PSBat-Train using our training strategy stated in subsec. 3.1, as compared with finetuning DeepAugMix for **Ours (stage 1)**. **CSQ** [39] and **HashNet** [4] are two supervised class-level online hashing methods. For fair comparison, we train these models using the same CNN backbone (ResNet50) with the same data augmentation strategy (sec. 4.1). **pHash** [2] is a classical image hashing method using relative DCT coefficients. All methods produce 128-bit hash code except pHash (64-bit).

Tab. 1 compares retrieval performance. The two online hashing methods, CSQ [39] and HashNet [4], are among the worst performers. CSQ and HashNet struggle to cope with strong ImageNet-C transformations present during training and test, resulting in lower performance than the classical pHash. ImageNet [15], MSResNet [24] and DeepAugMix [17] perform strongly on the Manip set but poorly when they undergo benign transformations. When trained via our contrastive loss (eq. 1), all models gain with the



Test	Average Precision (AP)		
	Not Manip.	Manip.	Diff.
Original	1.000	0.0000	0.0000
Benign	0.9635	0.0361	0.0003
Manip.	0.0250	0.9726	0.0024
Benign+Manip	0.1155	0.8807	0.0037
Distinct	0.0024	0.0081	0.9895

Figure 7. Characterizing the ICN classifier performance on PSBat-Pair. Left: Breakdown of classifier performance in the presence of different benign transformation classes, both in-place (from Imagenet-C) and out of place transformations (*e.g.* warps, padding – shaded in blue). Right: Overall classifier performance when comparing an original image with: itself, benign transformed, manipulated, benign+manipulated, and entirely different (distinct) images.

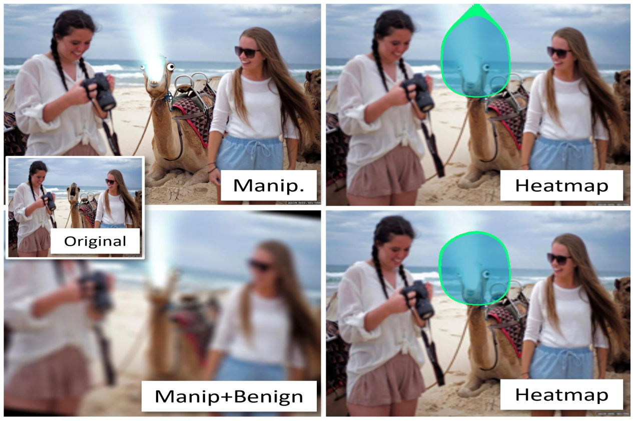


Figure 8. ICN heatmaps showing manipulation of an original image (inset) at threshold 0.35. The region of manipulation is correctly identified both without (top) and with (bottom) benign transformation of the manipulated version.

finetuned DeepAugMix (Ours, stage 1) achieving 1.8× improvement on Benign IR@1 and 1.5× on Manip+Benign versus the pretrained model. Our trained DeepAugMix also outperforms the finetuned ImageNet/MSResNet by 1-7% on all top-k scores and query sets. Re-ranking with ICN (Ours stage 1+2) further helps to improve top-1 performance (16% increase at IR@1 overall vs. stage 1). The advantage of ICN is shown in Fig. 4 where our two-stage retrieval reaches its upper bound performance within top-10 returned images across the 3 query sets. The curve justifies a *top-k* shortlist of $k = 100$; at this level our end-to-end system retains interactive speeds (stage 1: 40ms, stage 2: 400ms) on a GTX 1080 Ti GPU. Examples are in Fig.3.

4.5. Evaluating Image Comparator Network

We compare the localization performance of the proposed method against four baselines. **Sum of Squared Distances (SSD)** - we simply compute SSD between two images at pixel level, resize it to 7×7 then resize it back before thresholding to create continuity in the detected heatmap. **ResNetConv** - we extract $7 \times 7 \times 2048$ features from pretrained ImageNet ResNet50 model for both query and original images. These are averaged across channels to produce a 7×7 heatmap. **ErrAnalysis** - inspired from the blind detection technique in [36], we perform JPEG compression on the query image and compare with itself. **MantraNet** - is a supervised blind detection method [37] that detects anomalous regions. Additionally we evaluate baselines with images passed through our alignment module.

4.5.1 Heatmap Localization and Interpretability

Heatmap Localization Accuracy. We compare the heatmaps generated by our ICN with baseline methods. Heatmaps are produced by upsampling the 7×7 heatmap output of the ICN to the size of the image using bicubic interpolation. Heatmaps may be presented on false-colour scale (*e.g.* jet) in this form, or thresholded to produce an outline of the predicted manipulated region (Fig. 5 shows examples of thresholded and non-thresholded heatmaps). In our experiments, we threshold the normalized heatmaps at 0.35 determined empirically (Fig. 8). Tab. 2 (first column) reports the IoU metric between the predicted heatmap and the ground truth, both with and without the thresholding. Whilst most baselines are improved through use of our geometric alignment ($\mathcal{F}_A$) process, our ICN significantly exceeds baseline performances by at least 0.30.

Heatmap Interpretability is assessed against baseline methods via a crowd-sourced study on Amazon Mechanical Turk (MTurk). Participants see an original image, and an image subjected to both manipulation and benign transformation. The latter is annotated with the ground truth as a guide. The participants are shown a grid of heatmaps generated by 9 methods: ours, 4 baselines SSD, MantraNet, ErrAnalysis and ResNetConv, and 4 warp-corrected baselines pre-applying $\mathcal{F}_A$ for geometric alignment. Participants indicate which of the 9 heatmaps best summarizes the image modification; 200 such tasks are each annotated by 5 unique participants. Tab. 2 (final col.) presents the results, which favour our proposed method, even when the image pair are pre-aligned. There is preference for the visually simpler, thresholded heatmap, especially in presence of noisy transformations (Fig. 9).

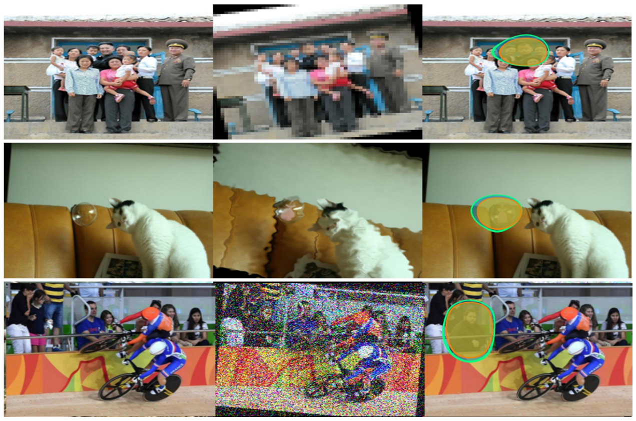


Figure 9. ICN heatmap results. Left col.: Original image. Middle col.: Manipulated image also subjected to benign transformation. Right col.: Heatmap output (green) ignoring benign transformation and highlighting manipulation (ground truth in yellow).

Method	Detection (AP)	Localization (IoU)
Full	**0.989**	**0.6543**
No Geo. Align. $\mathcal{F}_A$	0.860	0.4314
No Pred. Mod. $\mathcal{F}_P$	0.943	0.5652
Frozen Pred. Mod. $\mathcal{F}_P$	0.874	0.5333
Frozen Geo. Align. $\mathcal{F}_A$	0.929	0.5315
No $\mathcal{L}_C$	0.450	0.0000
No $\mathcal{L}_T$	0.955	0.1683

Table 3. Ablation study for the ICN exploring the effect of omitting loss terms, modules, or end-to-end training.

4.5.2 ICN Classifier Accuracy

We evaluate the performance of the 3-way classification by comparing each original image in the test set with: itself, benign transformed version of itself, manipulated version, manipulated as well as benign transformed and an entirely different image, chosen at random. The first two cases are expected to be classified as 'not manipulated', second two as 'manipulated' and the last one as 'distinct'. Fig. 7 (right) shows the AP achieved for each. A non-modified original-original pair is always correctly classified as not manipulated. Introduction of benign transformations reduces the accuracy slightly, as 3.6% of benign transforms are misclassified as manipulations. The most challenging case is queries that are both manipulated and benign transformed, with 11.5% being marked as not manipulated. Performance for each benign manipulation class is analysed in Fig. 7.

4.5.3 Ablation Study

We perform ablations to determine the impact of each component of the ICN. Results are shown in Tab. 3. Both detection and localization performance are significantly dependent on $\mathcal{F}_A$, dropping by 13% and 0.22, respectively when it is omitted. In the 'no $\mathcal{F}_P$' experiment, the prediction module $\mathcal{F}_P$ is replaced by a ResNet50 architecture $f_R(.)$, and $z = f_R(q') - f_R(I)$ (c.f. Fig.2). Training with either $\mathcal{F}_P$ or $\mathcal{F}_A$ frozen also reduces both detection and localization performances. Finally, we train ICN using just one of the two losses by changing the weights of total loss $\mathcal{L}(.) = w_c\mathcal{L}_C(.) + w_t\mathcal{L}_T(.)$ from $w_c = w_t = 0.5$ to $w_c = 1, w_t = 0$ and $w_c = 0, w_t = 1$. While $\mathcal{L}_C(.)$ alone still yields good detection performance with AP of 95.5%, localization IoU suffers significantly; both tasks are required to train the ICN.

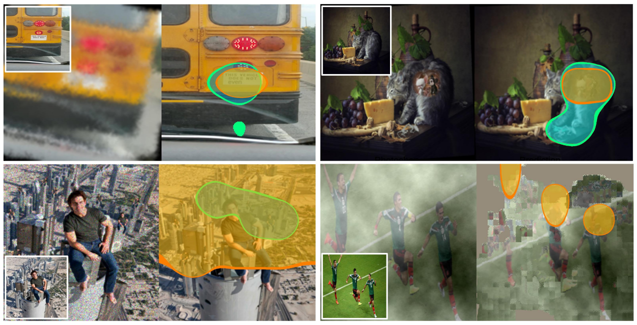


Figure 10. ICN Limitations. Top-left: Spurious detections due to benign degradation; Top-right: Mismatch due to annotators missed the skeletal mouse in the ground-truth; Bot-left: Heatmap unable to separate many small manipulations. Bot-right: Missed detections due to poor geometric alignment.

4.6. Limitations

Fig. 10 illustrates failure cases of the ICN. If the degradation is very severe it will not be ignored by the model's trained invariance, and spurious additional detection (top-left) or complete absence of detection (bot-right) may occur. The 7×7 heatmap activations are of insufficient resolution to separate many small manipulations (bot-right). Inaccurate ground-truth (all 3 annotators missed the skeletal mouse) gives an artificially low IoU (top-right).

5. Conclusion

We presented an Image Comparator Network (ICN) for visually comparing a pair of images in order to detect and localize manipulated regions. The ICN, when combined with a robust near-duplicate search, enables users to match images circulating 'in the wild' to a trusted database of original images. Given a query and matched original, the ICN visualizes areas of manipulation as a 'heatmap'. The heatmap ignores artifacts due to benign transformations that commonly occur as images are reshared. We train and evaluate using a novel ground-truth annotation collected over PSBattles [16], and validate heatmap interpretability via a user study. The ICN classifier enables fast false positive rejection from the near-duplicate image search. Future work could expand the dataset to include generative content, and extend heatmap detection to video.

Acknowledgement

We thank Andy Parsons, Leonard Rosenthol, Bill Marino and the Adobe Content Authenticity Initiative (CAI) for discussions. This work was supported by Adobe Research and EPSRC DECaDE Grant Ref EP/T022485/1.

References

[1] S. Baba, L. Krekor, T. Arif, and Z. Shaaban. Watermarking scheme for copyright protection of digital images. *IJCSNS*, 9(4), 2019.

[2] J. Buchner. Imagehash. https://pypi.org/project/ImageHash/, 2020.

[3] Content Authenticity Initiative (CAI). Setting the standard for content attribution. Technical report, Adobe Inc., 2020.

[4] Z. Cao, M. Long, J. Wang, and P. S. Yu. Hashnet: Deep learning to hash by continuation. In *Proc. CVPR*, pages 5608–5617, 2017.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. ICML*, pages 1597–1607, 2020.

[6] J. Collomosse, T. Bui, A. Brown, J. Sheridan, A. Green, M. Bell, J. Fawcett, J. Higgins, and O. Thereaux. ARCHANGEL: Trusted archives of digital public documents. In *Proc. ACM Doc.Eng*, 2018.

[7] IPTC Council. Social media sites photo metadata test results. http://embeddedmetadata.org/social-media-test-results.php, 2020.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009.

[9] P. Devi, M. Venkatesan, and K. Duraiswamy. A fragile watermarking scheme for image authentication with tamper localization using integer wavelet transform. *J. Computer Science*, 5(11):831–837, 2019.

[10] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer. The deepfake detection challenge (DFDC) dataset. *CoRR*, abs/2006.07397, 2020.

[11] J. Aythora et al. Multi-stakeholder media provenance management to counter synthetic media risks in news publishing. In *Proc. Intl. Broadcasting Convention (IBC)*, 2020.

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[13] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *Proc. ECCV*, pages 241–257, 2016.

[14] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, pages 1735–1742, 2006.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016.

[16] S. Heller, L. Rossetto, and H. Schuldt. The PS-Battles Dataset – an Image Collection for Image Manipulation Detection. *CoRR*, abs/1804.04866, 2018.

[17] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[18] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proc. ICLR*, 2019.

[19] S. Jenni and P. Favaro. Self-supervised feature learning by learning to spot artifacts. In *Proc. CVPR*, 2018.

[20] J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[21] S. Gilani K. Hameed, A. Mumtax. Digital image watermarking in the wavelet transform domain. *WASET*, 13:86–89, 2006.

[22] F. Khelifi and A. Bouridane. Perceptual video hashing for content identification and authentication. *IEEE TCSVT*, 1(29), 2017.

[23] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convnets. In *Proc. NIPS*, 2012.

[24] Z. Lenyk and J. Park. Microsoft vision model resnet-50 combines web-scale data and multi-task learning to achieve state of the art. https://pypi.org/project/microsoftvision/, 2021.

[25] Q. Li, Z. Sun, R. He, and T. Tan. Deep supervised discrete hashing. In *Proc. NeurIPS*, pages 2482–2491, 2017.

[26] Y. Li, M-C. Ching, and S. Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *Proc. IEEE WIFS*, 2018.

[27] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In *Proc. CVPR*, pages 2064–2072, 2016.

[28] M.Huh, A. Liu, A. Owens, and A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proc. ECCV*, 2018.

[29] T. Pan. Digital-content-based identification: Similarity hashing for content identification in decentralized environments. In *Proc. Blockchain for Science*, 2019.

[30] D. Profrock, M. Schlauweg, and E. Muller. Content-based watermarking by geometric wrapping and feature- based image segmentation. In *Proc. SITIS*, pages 572–581, 2006.

[31] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, pages 402–419. Springer, 2020.

[32] P. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision Image Understanding (CVIU)*, 78(1):138–156, 2000.

[33] J. Wang, H. T. Shen, J. Song, and J. Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2004.

[34] S-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. Efros. Detecting photoshopped faces by scripting photoshop. In *Proc. ICCV*, 2019.

[35] S-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proc. CVPR*, 2020.

[36] W. Wang, J. Dong, and T. Tan. Tampered region localization of digital color images based on jpeg compression noise. In *International Workshop on Digital Watermarking*, pages 120–133. Springer, 2010.

[37] Y. Wu, W. AbdAlmageed, and P. Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proc. CVPR*, pages 9543–9552, 2019.

[38] N. Yu, L. Davis, and M. Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[39] L. Yuan, T. Wang, X. Zhang, F. Tay, Z. Jie, W. Liu, and J. Feng. Central similarity quantization for efficient image and video retrieval. In *Proc. CVPR*, pages 3083–3092, 2020.

[40] C. Zauner. Implementation and benchmarking of perceptual image hash functions. Master's thesis, Upper Austria University of Applied Sciences, Hagenberg, 2010.

[41] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In *Proc. AAAI*, 2016.