Results and findings of the 2021 Image Similarity Challenge

Zoë Papakipos[1]   zoep@fb.com
Giorgos Tolias[2]   toliageo@fel.cvut.cz
Tomas Jenicek[2]   tomas.jenicek@seznam.cz
Ed Pizzi[1]   edpizzi@fb.com
Shuhei Yokoo[3]   bepemgdlp@gmail.com
Wenhao Wang[4]   wangwenhao0716@gmail.com
Yifan Sun[4]   sunyifan01@baidu.com
Weipu Zhang[4]   424729227@qq.com
Yi Yang[4]   yee.i.yang@gmail.com
Sanjay Addicam[5]   addicam.sanjay@gmail.com
Sergio Manuel Papadakis[6]   ocom2.00@gmail.com
Cristian Canton Ferrer[1]   ccanton@fb.com
Ondřej Chum[2]   chum@cmp.felk.cvut.cz
Matthijs Douze[1]   matthijs@fb.com

[1] Meta AI, [2] Czech Technical University in Prague, [3] DeNA, Japan,

[4] Baidu Research, [5] Chandler, Arizona USA, [6] San Carlos de Bariloche, Argentina

Editors: Douwe Kiela, Marco Ciccone, Barbara Caputo

Abstract

The 2021 Image Similarity Challenge introduced a dataset to serve as a benchmark to evaluate image copy detection methods. There were 200 participants to the competition. This paper presents a quantitative and qualitative analysis of the top submissions. It appears that the most difficult image transformations involve either severe image crops or overlaying onto unrelated images, combined with local pixel perturbations. The key algorithmic elements in the winning submissions are: training on strong augmentations, self-supervised learning, score normalization, explicit overlay detection, and global descriptor matching followed by pairwise image comparison.

Keywords: Competition, image copy detection

1. Introduction

The Image Similarity Challenge organized in 2021 aimed to assess the efficacy of image copy detection algorithms using a large dataset with robust image edits. The challenge design aimed to reflect practical requirements for large-scale copy detection systems, where most queries do not match references in the dataset, and it is important to efficiently separate copies from non-copies.

The challenge consisted of two tracks: a descriptor track, and an unconstrained matching track. In the descriptor track, participants provide descriptor vectors in $\mathcal{R}^{256}$ for each image in the dataset, and matching is performed using L2 distances between the vectors. In the matching track, any matching techniques can be used, including pairwise image comparisons. The challenge organizers created and released the DISC21 dataset for use in
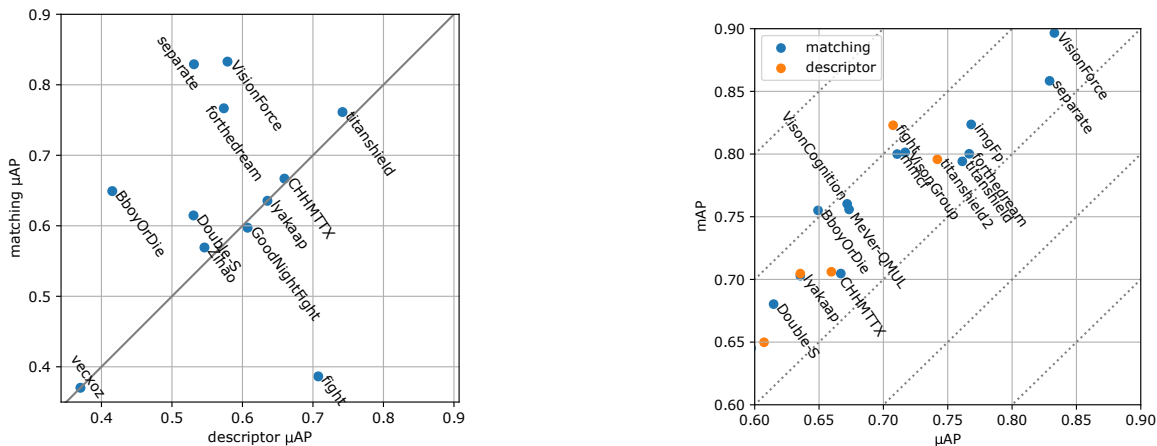
Figure 1: Comparison of the ISC submissions on two axes: Left: descriptor vs. matching track performance, Right: μAP vs. mean AP (mAP).

this challenge and beyond, as a benchmark for image copy detection. DISC21 includes a large reference set of images and a smaller set of query images, where the goal is to find the subset of matches between the two. See (Douze et al., 2021) for more details about the creation of the dataset. The challenge drew over 200 participants in its final phase, including strong solutions. The main takeaways for image copy detection from this challenge include the following. (1) Strong and non-standard image augmentations that mimic typical cases of image copies are very beneficial in the training. (2) Self-supervised learning by instance-discrimination is crucial, not only for pre-training, but also as the main training task. (3) Score normalization, either explicitly in the matching track, or implicitly by descriptor processing in the descriptor track, has a significant impact. (4) Explicit overlay detection is a task-tailored approach that has proven useful. (5) The use of regional representation and matching is able to significantly improve copy detection performance compared to global descriptor approaches.

This paper presents the main findings of the challenge. Section 2 presents an in-depth analysis of the results that were submitted. Section 3 describes what components the most successful participants used for their winning entries.

2. Analysis of the results

This section takes an outside view of the results without any insight into the methods used by the participants. The following analysis is based on the raw submission files to the final track. It includes results from participants that were disqualified (*e.g.* because they broke participation rules). For most results we included only the top submissions to improve the readability.
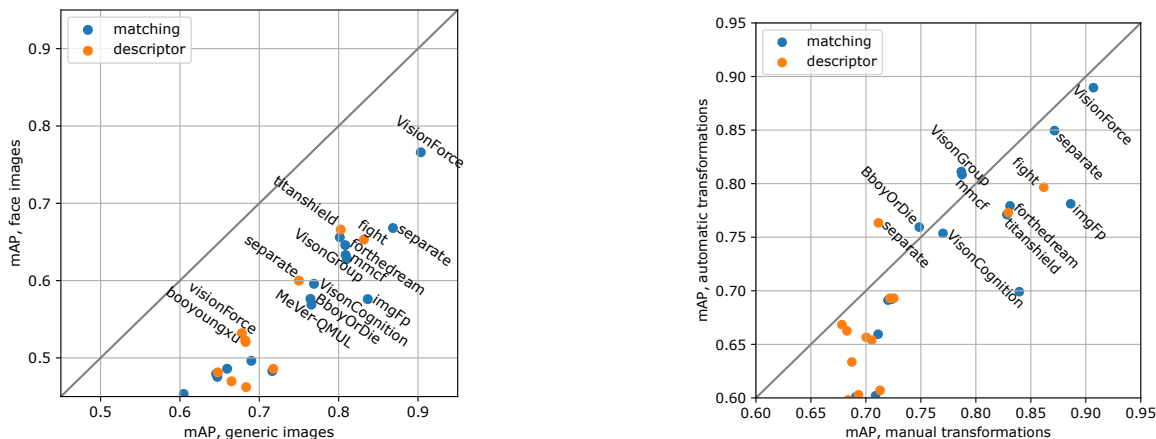
Figure 2: Results per broad data source. Left: reference images are face images (number of query images n=500) or generic images (n=9500), Right: transformations performed manually (n=4040) or automatically (n=5960).

2.1. High-level comparisons

The matching track is less constrained than the descriptor track; in fact a valid descriptor submission can be converted into a matching submission. Most participants to the descriptor track submitted to the matching track as well. We compare the submissions to assess the performance gain by the matching. Figure 1 (left) shows a maximum performance difference between the two track equal to 0.2 and almost equal performance for some submissions like teams TITANSHIELD, CHHMTX, LYAKAAP. There is one outlier case where the descriptor submission is better than the matching one.

Per-query comparison. The default evelution metric, namely μAP, considers all queries jointly. We provide a finer analysis of the results, with the following per-query measure of performance. We consider only the 10k queries that actually match one of the reference images and discard the 40k distractors, and compute the *average precision* for that query, which coincides with the inverse of the rank of the true positive result. Averaging this measure over the 10k queries results in the so-called mean average precision (mAP). The mAP can be computed on subsets of queries, in which case we indicate n, the number of query images of the subset.

Figure 1 (right) shows how mAP compares to μAP per submission. The two measures are not directly comparable, but a larger gap between mAP and μAP is a sign of ineffective score normalization; μAP is designed to evaluate how well matching scores are normalized across queries, see (Douze et al., 2021, Section 4.4). For example, it is possible that for the matching track, the VISIONFORCE team's score normalization is worse than that of SEPARATE.

Analysis per data source. The DISC21 dataset is built from two different sources and uses two different ways to apply the transformations.

Face images vs. generic images. DISC21 is built from two data sources, namely generic images from YFCC100M (Thomee et al., 2016) that contain no images of people and 5%
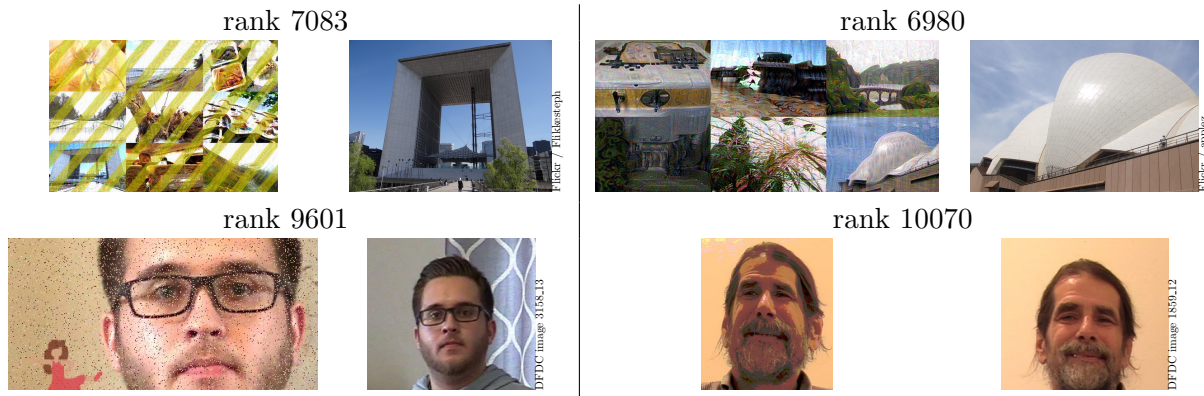
rank 7083   rank 6980

rank 9601   rank 10070

Figure 3: Example false positive matches (pairs of query images and detected references images) from SEPARATE's matching track submission. The ranks are in the ranked list of (query, database) pairs. The smaller the ranking of a false positive is, the more it harms the evaluation metric.

is made up of face images from the DFDC challenge (Dolhansky et al., 2020). The plot in Figure 2 (left) shows that the scores for queries of face images is lower than that for generic images for all submissions. Figure 3 shows that false positive results often depict the same object/face from a sightly different viewpoint, which is a mismatch from a copy detection point of view and forms a very challenging case. There are more pairs of images with such small variations in the images of faces, and they are more likely to be returned as results.

Manual vs. automatic transformations. The image manipulations are either performed manually or via a series of carefully calibrated automatic transformations. Figure 2 (right) compares the performance of the submissions depending on the type of transformations. It appears that the manual transformations are generally easier than the automatic ones.

2.2. Analysis per transformation

The automatically generated transformations were built by applying 2 to 6 transformations, in different steps, to all images. The random sampling of transformations was calibrated on the baseline matching methods at our disposal.

Marginalized mAP measurements. Assessing the impact of each transformation type is not easy, because (1) there are only 132 query images that are produced with a single transformation, (2) the intensity of most transformations depends on random parameters that are different between query images and (3) the impact on a retrieval measure like AP depends on the image content. Therefore, there are not enough observations, *i.e.* query images, to measure the impact of each transformation precisely. To mitigate this, we group the query images that have common transformation characteristics and compute the mAP within these groups. This marginalizes over the transformations of the sequence.

Analysis per number of transformations. Since the focus of DISC21 is on difficult queries, the sampling was tuned to favor a large number of transformations, *e.g.* 2340 query images are transformed with 4 steps, vs. 132 with a single one. The transformations are grouped
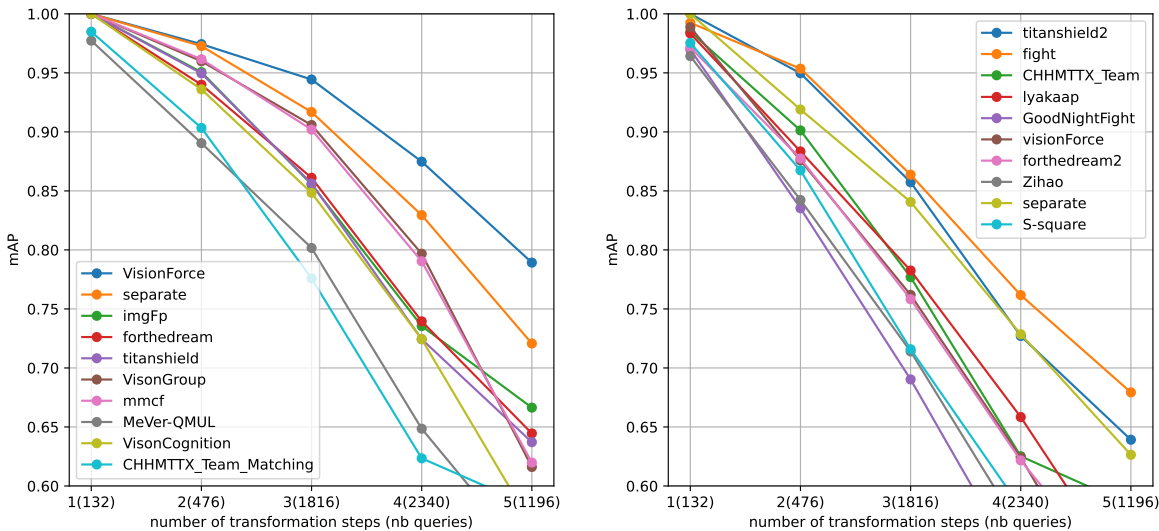
Figure 4: Performance (mAP) per number of transformations applied to create the query images, in the matching track (left) and the descriptor track (right). The labels on the x-axis also indicate on how many images this mAP was computed.

in classes (geometric, overlay, *etc.*) and at most one transformation per class is applied. All the query images with 5 transformations include an adversarial attack step.

Figure 4 shows the impact of the number transformations on mAP. We observe that every additional transformation causes a larger performance drop than the previously added one. The impact of a transformation is not independent of that of others: for an image that is already hard to recognize, one additional transformation degrades the retrieval more than if it is applied to the original image. Hence, the gap between methods is more significant for images with many transformations (more than 0.2 mAP) than for those with one (below 0.05 mAP).

2.3. Penalty analysis

To further analyze the impact of each transformation, we use a *penalty analysis* based on a simplistic model: for a given submission, we associate a fixed penalty to each of the 31 transformations, $(P_t)_{t=1..31}$. For a query image that undergoes transformation steps $(t_1, t_2, ..., t_N)$, we model the resulting AP for that query as:

$$\text{AP} = 1 - \sum_{i=1}^{N} P_{t_i} \tag{1}$$

This model is very rough, but its advantage is that the (P_t) values can be estimated easily in the least squares sense from the AP measurements.

Table 1 shows that in general the hardest transformations are when the source image is inserted on top of an unrelated image. This is hard to match with global descriptors in the descriptor track.

Geometric transformations are generally quite mild. Some submissions have particularly low performance on specific transformations, *e.g.* VISIONFORCE has low performance on the vertical flip (vflip) transformation, and teams CHHMTTX and GoodNightFight struggle with rotations. It is likely that these geometric transformations were not included at training time for these submissions. The clip transformation seems to have a *positive* impact on on the retrieval accuracy. This could be because the descriptor extraction at inference time often benefits from a stronger cropping than what is applied by default, See (Touvron et al., 2019).

The penalties for the matching track are generally less severe than for the descriptor track. When using the full scale of matching techniques, some submissions like VISION-FORCE or VISIONGROUP become quite insensitive to geometric transformations where a large fraction of the image detail is removed.

The crop and overlay transformations are the hardest ones to handle, even in the matching track. Strong matching submissions like the VISIONFORCE one are able to retrieve an image half of the cases if only 6-12% of the original image surface remains. Similarly, if the image is overlaid over another one, representing less than 20% of the image surface, the method can recover it almost 80% of the time.

3. Top-ranked methods

The methods of top-ranked teams for the two tracks are presented. We refer to top-ranked methods simply as *methods* and to top-ranked participants as *participants* in the following. The major components that are common among methods are identified and used to structure this section; method details are provided per component while similarities and differences are discussed. The top three methods, starting from the top-ranked for the matching and descriptor track are denoted by VISIONFORCE-mt1 (Wang et al., 2021a), SEPARATE-mt2 (Jeon, 2021), ImgFp-mt3 (Sun et al., 2021), and LYAKAAP-dt1 (Yokoo, 2021), S-SQUARED-dt2 (Papadakis and Addicam, 2021), and VISIONFORCE-dt3 (Wang et al., 2021b), respectively.

Deep backbone and classical features. All submissions rely on a neural net to analyze the images. We report the architecture here and discuss the training approach in the next subsection.

VISIONFORCE-mt1 uses all three ResNet-50, ResNet-152, and ResNet50-IBN as backbones followed by GeM pooling (Radenović et al., 2019), combined with WaveBlock (Wang et al., 2022), and finally append a final projector module that consists of linear and non-liner layers and increases the dimensionality to 2048. WaveBlock can be seen as a type of augmentation method at the feature level. VISIONFORCE-dt3 from the same team and exploits the same backbones for both tracks. SEPARATE-mt2 uses ViT (Dosovitskiy et al., 2021) ("vit_large_patch16_384") to map an image to a global descriptor for the first ranking stage, and another ViT backbone ("vit_large_patch16_224") that receives an image pair in the form of a horizontally concatenated image as input and outputs a binary prediction for matching or non-matching. ImgFp-mt3 uses EsViT (Li et al., 2021) with Swin-B transformer (Liu et al., 2021), adjusted as follows. Global average pooling is performed on the feature maps of each of the last blocks whose number of channels is $[512, 512, 1024, 1024]$, respectively. The outputs are concatenated and a fully connected layer is used to generate a 256-D global descriptor. This is the only method using classical local features too, in particular SIFT descriptors (Lowe, 2004).

transformation	type	nb of queries	VisionForce	separate	imgFp	forthedream	titanshield	VisonGroup	mmcf	MeVer-QMUL	VisonCognition	CHHMTTX	titanshield2	fight	CHHMTTX	lyakaap	GoodNightFight	visionForce	forthedream2	Zihao	separate	S-square
			Matching track										Descriptor track									
change-aspect_ratio	G	474	-0.02	0.01	0.04	0.00	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02	-0.01	-0.02	-0.01	-0.02	-0.00	-0.01	0.02	-0.00	-0.01
hflip	G	498	-0.05	-0.03	0.09	-0.01	-0.03	-0.04	-0.04	-0.02	-0.03	-0.03	-0.03	-0.01	-0.03	-0.01	-0.02	-0.03	-0.04	-0.00	-0.02	-0.01
pad_square	G	494	-0.01	-0.02	0.00	-0.01	-0.01	-0.02	-0.02	-0.02	-0.02	0.00	-0.00	-0.01	-0.03	-0.01	-0.02	-0.03	-0.02	0.04	-0.02	-0.12
perspective_transform	G	450	0.04	0.06	0.13	0.07	0.07	0.07	0.06	0.10	0.06	0.11	0.07	0.08	0.11	0.09	0.12	0.06	0.08	0.09	0.10	-0.12
vflip	G	474	-0.01	-0.01	0.18	0.03	-0.02	-0.01	0.00	0.05	0.10	0.00	-0.02	0.03	0.00	0.04	0.00	0.36	0.03	-0.01	-0.01	-0.01
pixelization	L	677	0.01	0.01	-0.00	0.01	0.01	-0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.04	0.00	-0.01	0.01	0.02	0.03	0.01
overlay-text	O	1737	0.02	0.01	0.04	0.03	0.01	0.02	0.03	0.05	0.04	0.05	0.02	0.04	0.05	0.02	0.05	0.05	0.04	0.06	0.03	0.05
apply-pil_filter	L	631	0.03	0.04	0.06	0.04	0.06	0.02	0.02	0.02	0.02	0.04	0.06	0.04	0.05	0.01	0.03	0.04	0.03	0.03	0.04	0.05
encoding-quality	L	644	0.04	0.01	0.03	0.02	0.01	0.01	0.01	0.01	0.00	0.04	0.01	0.00	0.04	-0.01	0.03	0.03	0.00	0.02	0.00	-0.01
apply-ig_filter	P	1066	0.01	0.01	0.03	0.03	0.06	0.04	0.04	0.02	0.05	0.03	0.06	0.06	0.03	0.04	0.03	0.03	0.03	0.01	0.03	0.03
overlay-emoji	O	1753	0.03	0.02	0.07	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.05	0.06	0.07	0.04	0.06	0.06	0.09	0.05	0.06	0.08
grayscale	P	1018	0.01	0.03	0.05	0.07	0.08	0.10	0.11	0.14	0.07	0.06	0.08	0.06	0.06	0.05	0.06	0.06	0.07	0.05	0.06	0.07
rotate	G	417	0.07	0.08	0.08	0.08	0.05	0.10	0.04	0.14	0.11	0.32	0.05	0.07	0.32	0.11	0.33	0.18	0.13	0.13	0.15	0.15
saturation	P	1021	0.02	0.03	0.03	0.05	0.03	0.03	0.04	0.04	0.04	0.06	0.03	0.02	0.06	0.04	0.06	0.02	0.06	0.04	0.05	0.03
shuffle_pixels	L	643	0.04	0.06	0.03	0.06	0.06	0.06	0.06	0.05	0.06	0.09	0.06	0.03	0.09	0.05	0.08	0.08	0.06	0.06	0.06	0.02
clip_image-size	G	7161	-0.10	-0.13	-0.16	-0.17	-0.17	-0.14	-0.15	-0.16	-0.16	-0.16	-0.17	-0.13	-0.16	-0.12	-0.13	-0.14	-0.14	-0.12	-0.12	-0.12
random-noise	L	621	0.06	0.11	0.07	0.13	0.10	0.10	0.10	0.06	0.07	0.10	0.10	0.06	0.10	0.10	0.08	0.02	0.15	0.06	0.10	0.07
brightness	P	1009	0.05	0.09	0.04	0.11	0.07	0.07	0.07	0.06	0.07	0.10	0.07	0.05	0.10	0.09	0.08	0.09	0.06	0.07	0.10	0.07
legofy	L	633	0.06	0.10	0.11	0.11	0.06	0.11	0.11	0.07	0.12	0.07	0.06	0.04	0.07	0.08	0.06	0.07	0.10	0.06	0.07	0.05
apply_ar-effect	O	1049	0.07	0.09	0.04	0.12	0.19	0.14	0.14	0.08	0.07	0.12	0.19	0.09	0.12	0.10	0.11	0.08	0.14	0.12	0.07	0.17
convert_color	P	1871	0.15	0.14	0.17	0.17	0.21	0.16	0.14	0.14	0.15	0.14	0.21	0.13	0.14	0.15	0.11	0.15	0.13	0.14	0.11	0.13
overlay-stripes	O	1735	0.16	0.15	0.29	0.26	0.19	0.19	0.22	0.22	0.16	0.21	0.18	0.20	0.21	0.22	0.17	0.26	0.30	0.10	0.15	0.25
blur	L	648	0.13	0.31	0.28	0.22	0.18	0.19	0.20	0.23	0.24	0.17	0.18	0.19	0.17	0.19	0.12	0.24	0.23	0.32	0.23	0.17
adversarial_attack	A	1201	-0.04	-0.01	-0.08	-0.05	-0.09	0.04	0.04	-0.01	-0.02	-0.08	-0.08	-0.03	-0.08	0.02	-0.07	-0.03	-0.04	0.10	-0.02	-0.01
overlay-onto_screenshot	I	438	0.12	0.20	0.21	0.20	0.38	0.15	0.15	0.53	0.22	0.52	0.38	0.19	0.52	0.35	0.73	0.42	0.32	0.47	0.28	0.33
crop	G	468	0.24	0.26	0.26	0.45	0.36	0.33	0.34	0.51	0.34	0.63	0.35	0.20	0.63	0.42	0.64	0.37	0.32	0.57	0.49	0.58
overlay-onto_image	I	470	0.07	0.17	0.22	0.16	0.33	0.18	0.14	0.53	0.20	0.54	0.33	0.32	0.54	0.43	0.88	0.20	0.18	0.81	0.20	0.54
overlay-blurred_mask	I	479	0.08	0.20	0.38	0.19	0.43	0.24	0.23	0.51	0.30	0.50	0.43	0.41	0.50	0.43	0.76	0.49	0.24	0.79	0.27	0.74
collage	I	454	0.04	0.05	0.08	0.39	0.49	0.11	0.10	0.45	0.56	0.43	0.49	0.27	0.42	0.65	0.76	0.42	0.62	0.72	0.63	0.71

Table 1: Relative penalties for all transformations. The transformations are classified into 5 types: G=geometric, L=local pixel transformation, O=source image is partially occluded, P=photometric transformation, I=insertion on another image.

LYAKAAP-dt1 uses EfficientNetv2 with GeM pooling and reduce the dimensionality of the final descriptor by a linear layer with batch norm that is followed by l2 normalization. S-SQUARED-dt2 uses multiple backbones: EfficientNetV2 1, EfficientNetV2 s, EfficientNet b5, and NfNet 11. Each backbone is followed by GeM pooling (Radenović et al., 2019) and a linear layer to reduce the dimensionality and L2 normalization.

Training approaches. There were various training approaches for the given architectures, often decomposed into several phases that we describe here.

Pre-training on external data. External datasets are used by all participants in the pre-training stage: either an existing pre-trained network is used or the participants performed the pre-training themselves. ImageNet is used in all cases, with supervised learning for SEPARATE-mt2, LYAKAAP-dt1, and S-SQUARED-dt2, and with unsupervised learning for VISIONFORCE-mt1, IMGFP-mt3, and VISIONFORCE-dt3.

Training augmentations. All methods compose an augmentation set that is richer than the conventional augmentation used to train classifiers, with transformations that mimic the task of copy detection. Such examples are more extreme geometric and photometric transformations and image/text/emoji overlays. VISIONFORCE-mt1 validates the impact of the enriched augmentations, which is quantified to be significant. This is not surprising, as the copy detection task is close to self-supervised learning, where data augmentation is the only source of intra-class variablity (Dosovitskiy et al., 2014; Chen et al., 2020).

Training on ISC training set. The main training is performed on the provided training set, with augmentations that mimic the query attacks. This step is performed in a self-supervised way for all participants since the training set is not labeled. This process follows the concept of instance discrimination (Wu et al., 2018) where each image in the training set forms its own class, and any of its augmentations belongs to that class. If not mentioned otherwise, the training optimizes a backbone network to generate a global image descriptor.

Deep metric learning is used by VISIONFORCE-mt1 with a combined classification and triplet loss, for which hard samples are mined. SEPARATE-mt2 trains with SimCLR (Chen et al., 2020) where an augmented image is matched to the original one using InfoNCE loss. Similarly, IMGFP-mt3 uses a triplet loss. LYAKAAP-dt1 uses a contrastive loss and cross-batch memory (Wang et al., 2020) where one augmentation of the training image is performed with the enriched augmentation set and the other augmentation of the same image with the conventional augmentation set. S-SQUARED-dt2 uses the ArcFace loss (Deng et al., 2019) and additionally combine ImageNet with the ISC training set in this step. The large output space raises challenges in the training, handled by gradually increasing the number of classes in the training. IMGFP-mt3 uses triplet loss with hard-negative mining combined with cross-entropy loss.

Fine-tuning on ISC query/reference set. The competition rules allow training using the provided labels in Phase I, *i.e.* ground-truth that defines the correspondences between the provided 5k queries that are not distractors and the reference images. Only IMGFP-mt3 and LYAKAAP-dt1 perform such a fine-tuning process, which is shown to noticeably boost the performance. All other participants rely on their own augmentations applied to the training images in order to mimic query transformations, as described in Section 3.

Sub-image region detection and feature extraction. Detecting regions appears to be important for the matching track. VISIONFORCE-mt1 uses a fixed set of crops, regions detected by Selective Search (Uijlings et al., 2013), and regions detected by YOLOv5 (Jocher,

2020) trained to detect overlays of other images or emojis. Overlays of the former are used in further processing, while overlays of the latter are ignored. ImgFp-mt3 trains a pasted-image detector to obtain crops of possibly overlaid images during inference. Similarly to the approach of VisionForce-mt1, positive examples are synthetically created, but standard uninformative overlaid images such as emojis are considered negatives.

Only VisionForce-dt3 uses region detection for the descriptor track; *i.e.* the same YOLOv5 is used as in VisionForce-mt1.

Feature extraction. VisionForce-mt1 feeds the whole image but also each crop to the backbone and obtains a descriptor per case. Note that 33 backbones are used for the whole image, but only 3 of them are used for the region descriptors. This is done both for query and reference images. Separate-mt2 feeds the whole image or the concatenated one to ViT. ImgFp-mt3 uses the whole reference and query image as input to the backbone, and additionally the region, if any, provided by the overlay detector on the query image. Moreover, SIFT is used for local feature detection and descriptor extraction on all images.

Methods for the descriptor track feed the whole input image to the backbone and obtain a global descriptor. An exception is VisionForce-dt3 that replaces the full image with the region, if any, that is obtained with the Yolo-based overlay detector.

Ensembles. Model and similarity combination is done in different ways. We point out the case of combining different backbones, *e.g.* different architectures or multiple training runs of the same architecture, representation from fixed geometric augmentations performed at test time, or global and local representation. Some methods use more than one of these ensemble types.

Backbone ensemble. S-squared-dt2 ensembles the representation of the different backbones by concatenation and dimensionality reduction with PCA. The backbones not only differ in terms of the architecture, but are also a result of training with a different number of training classes. In total, 7 backbones are ensembled. VisionForce-mt1 keeps the maximum similarity over 33 different backbones. The three previously mentioned backbone networks are pre-trained in a self-supervised way; each backbone is pre-trained with either BYOL (Grill et al., 2020) or Barlow-Twins (Zbontar et al., 2021). Additionally, each backbone is trained 11 times, each one with a different augmentation set. A special case, not directly fitting into this category, is Separate-mt2 who fuses the two ViT models, *i.e.* the one for single image to obtain descriptor and the one for the concatenated image pair to obtain a relevance confidence. If the reference image is top-ranked with descriptor similarity, then relevance confidence is used to re-rank.

Test-time augmentation ensemble. VisionForce-dt3 ensembles multi-resolution representations, simply by averaging and re-normalizing the descriptor obtained for input images at 4 different resolutions. ImgFp-mt3 horizontally flips the query and maintains the maximum similarity over the two query versions.

Global/local ensemble. Matching track methods use an ensemble of global and local/regional processing. VisionForce-mt1 computes the similarity between the whole query image and each crop of a reference image and vice versa (reference versus query) and the maximum similarity is maintained. ImgFp-mt3 uses SIFT to estimate the *SIFT-score* by counting the number of correspondences that are formed between query descriptors and the closest descriptor among all reference images whose similarity is above a certain threshold, and satisfy the ratio test (Lowe, 2004). The SIFT-score is used for all images that appear in

the top similar images which are estimated in three different ways and then accumulated if an image appears in multiple top-ranked image shortlists. The three ways are (i) with CNN global descriptor from the full query, (ii) with CNN descriptor from the cropped query image according to the pasted-image detector, and (iii) with SIFT-score. The SIFT-score is estimated on the full query image in the first case, but in the detected region in the other two. In this way, information from the SIFT and CNN representations are fused.

Score normalization. Score normalization is shown to be useful in the baselines provided with the DISC2021 dataset (Douze et al., 2021). In particular, the similarity score is normalized *w.r.t.* the similarity between the query and images in the training set. This approach is used by VisionForce-mt1 in the matching track. All participants propose new ways to achieve a similar normalization in the descriptor track. Note that it is more challenging in that case, because any normalization needs to be applied a priori to the descriptor itself. All three methods try to move the query or reference image descriptor far from descriptors of the training set. The performance impact of descriptor normalization is significant for all participants.

Discussion. It turns out that top results are achieved with a variety of different approaches. The backbones that are used are either CNNs or ViTs; losses are either classification-based, pairwise, or both, while regional representations come from fixed regions, trained detectors, or even SIFT. As common winning components we identify score normalization, strong augmentations that mimic image copies, and ensembles. Ensembles are a common winning component for research competitions without computational complexity constraints. Note that the top matching method relies on up to 33 different backbones and multiple image regions that are represented separately. The memory that is required to store the representation of all references images is around 900Gb, which is two orders of magnitude greater than the 1Gb needed for the global descriptor track approaches. Achieving high performance with limited resources is a challenging task and an interesting future direction.

4. Conclusion

We organized the Image Similarity Challenge with the intention to introduce a benchmark for image copy detection and to push the state of the art in this field. The solutions from participants were of high quality, some of which introduce interesting new research directions. The main ingredients for the top submissions were careful tuning of data augmentation at training time, score normalization, explicit overlay detection and local-to-global comparison. We hope that this competition will spur more progress in the field of image copy detection, using the DISC21 dataset as a benchmark. A longer version of this analysis is on ArXiV (Papakipos et al., 2022).

Acknowledgments

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv*, 2020.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27:766–774, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

SeungKee Jeon. 2nd place solution to facebook ai image similarity challenge : Matching track. In *arXiv*, 2021.

Glenn Jocher. Yolov5. https://github.com/ultralytics/yolov5, 2020.

Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. In *arXiv*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

Sergio Manuel Papadakis and Sanjay Addicam. Producing augmentation-invariant embeddings from real-life imagery. In *arXiv*, 2021.

Zoë Papakipos, Giorgos Tolias, Tomas Jenicek, Ed Pizzi, Shuhei Yokoo, Wenhao Wang, Yifan Sun, Weipu Zhang, Yi Yang, Sanjay Addicam, Sergio Manuel Papadakis, Cristian Canton Ferrer, Ondrej Chum, and Matthijs Douze. Results and findings of the 2021 image similarity challenge, 2022.

Filip Radenović, Giorgos Tolias, and Ondrej Chum. Fine-tuning CNN image retrieval with no human annotation. *PAMI*, 2019.

Xinlong Sun, Yangyang Qin, Xuyuan Xu, Guoping Gong, Yang Fang, and Yexin Wang. 3rd place: A global and local dual retrieval solution to facebook ai image similarity challenge. In *arXiv*, 2021.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: the new data in multimedia research. *Commun. ACM*, 59:64–73, 2016.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171, 2013.

Wenhao Wang, Yifan Sun, Weipu Zhang, and Yi Yang. D2lv: A data-driven and local-verification approach for image copy detection. In *arXiv*, 2021a.

Wenhao Wang, Weipu Zhang, Yifan Sun, and Yi Yang. Bag of tricks and a strong baseline for image copy detection. In *arXiv*, 2021b.

Wenhao Wang, Fang Zhao, Shengcai Liao, and Ling Shao. Attentive waveblock: Complementarity-enhanced mutual networks for unsupervised domain adaptation in person re-identification and beyond. *IEEE Transactions on Image Processing*, 31:1532–1544, 2022.

Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proc. CVPR*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

Shuhei Yokoo. Contrastive learning with large memory bank and negative embedding subtraction for accurate copy detection. In *arXiv*, 2021.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proc. ICML*, volume 139, pages 12310–12320. PMLR, 2021.