SUPPLEMENTARY MATERIAL
TOWARDS IMAGE COPY DETECTION AT E-COMMERCE SCALE

1. ENSEMBLE MATCH ALGORITHM

Algorithm 1 Ensemble Match Algorithm

Require: $im1, im2, algos$

Ensure: $algos \in [\textbf{\textit{SIFT}}, \textbf{\textit{SuperGlue}}, \textbf{\textit{LoFTR}}]$

$[im1, im2] = \textbf{\textit{PreProcessImage}}([im1, im2])$

$allkeypoints \leftarrow List()$

for $algo \in algos$: **do**

$keypoints = \textbf{\textit{FindKeyPoints}}(algo, im1, im2)$

$allkeypoints \leftarrow allkeypoints + keypoints$

end for

$inliers = \textbf{\textit{FindInliers}}(allkeypoints)$

return $keypoints, inliers$

2. ADDITIONAL RESULTS

2.1. Data Augmentation

We show the ablation study on data augmentations used during DTML training in Table 1.

Table 1. Performance evaluation on EPID-easy using different data augmentation settings

Transforms	μAP	R@P90
basic	0.70	0.41
basic + intermediate	0.75	0.52
basic + intermediate + advance	**0.86**	**0.73**

2.2. Training Parameters

We show the effect of γ, $margin_s$ and $margin_d$ parameters used in dual triplet loss formulation on copy detection performance on EPID-easy dataset in Table 2 and Table 3.

2.3. Ensemble Model

We study the performance of the ensemble configurations as shown in Table 4. We use the EnsembleMatch + MDE (img) evaluation setting on EPID-difficult and change only the combinations of local feature matchers within the ensemble. We can clearly observe that different combinations of local feature matchers have varying performance regimes and show additive improvements over standalone models.

Table 2. Impact of γ on DTML performance

γ	μAP	R@P90
0	0.75	0.46
0.2	0.82	0.70
0.4	**0.86**	**0.73**
0.6	0.72	0.43
0.8	0.69	0.38
1.0	0.51	0.18

Table 3. Impact of triplet margins on DTML performance

$margin_s$	$margin_d$	μAP	R@P90
0.01	0.05	0.79	0.62
0.01	0.1	0.76	0.46
0.05	0.1	**0.86**	**0.73**
0.05	0.5	0.80	0.69
0.1	0.05	0.79	0.68

Table 4. Ablation study of different combinations of the EnsembleMatch

EnsembleMatch	μAP	R@P80
SIFT + SuperGlue	0.71	0.55
SIFT + LoFTR	0.76	0.65
SuperGlue + LoFTR	0.75	0.63
SIFT + SuperGlue + LoFTR	**0.88**	**0.85**

2.4. Latency Measurements

We test model latency on 100 randomly sampled image pairs and report the averaged results in Table 5. To standardize the latency estimates, the image pairs are resized to (640, 480) which matches the resolution of phone clicked images and preserves sufficient local details to detect and match key-

points. For global embedding models, we report the time taken to compute embeddings for an image pair and estimate similarity score using vector dot product. For image matching model, latency is the time taken to extract and match keypoints. SIFT utilizes the standard CPU implementations for measurements. For LoFTR and SuperGlue, the measurements are performed on an NVIDIA T4 GPUs. The latencies largely depend on the image pair size. In practice, the claim image resolutions do range widely depending on the handset's camera quality.

Table 5. Latency measurements

Method	Timing (ms)
ResNet50 (simCLR/SSCD/DTML)	14
DINO ViT-B/8	18
SIFT	152
SuperGlue	128
LoFTR	172
EnsembleMatch	450
EnsembleMatch + MDE (img+tab)	455

2.5. Qualitative Results

Figures 1 and 2 highlights qualitative performance of the match algorithms EPID datasets. The performance of the individual models appears to be inconsistent across simple as well as challenging variations as discussed, hence it is difficult to index the match based on any one algorithm. EnsembleMatch plays out on the strengths of each match algorithm and results in superior quality matches and inliers. EnsembleMatch consistently estimates more correct matches and fewer mismatches, successfully coping with repeated texture, large viewpoint, and illumination changes. We further observe that, applying outlier filtering on the aggregated matches using a robust estimator (RANSAC or MAGSAC), removes noisy matches detected by individual models and outputs more consistent matches. Interestingly, the task of copy detection in our usecase is governed based on matches from both the foreground subjects with product artefacts such as damage, orientations, and written text as well as the background scenes to successfully ascertain a duplicate image. EnsembleMatch displays that with a combination of models each with their individual strengths, it can decently handle copy detection tasks in more complex scenarios.

We also qualitatively highlight the Top10 image retrieval results for sample queries from EPID-easy in Figure 3.

2.6. CopyDays Results

Figure 4 shows the top-10 retrieval outputs (on right) for each sample probe image (on left). We observe that embeddings from our DTML method is robust to most of the commonly used copy attacks to produce near duplicates. We also highlight some of our model failures (last two rows) in detecting severe attacks involving strong image transformations.

3. EPID-DIFFICULT

The data augmentations used to generate EPID-difficult are listed in Table 6. Figure 5 shows sample images from EPID-easy and EPID-difficult dataset.

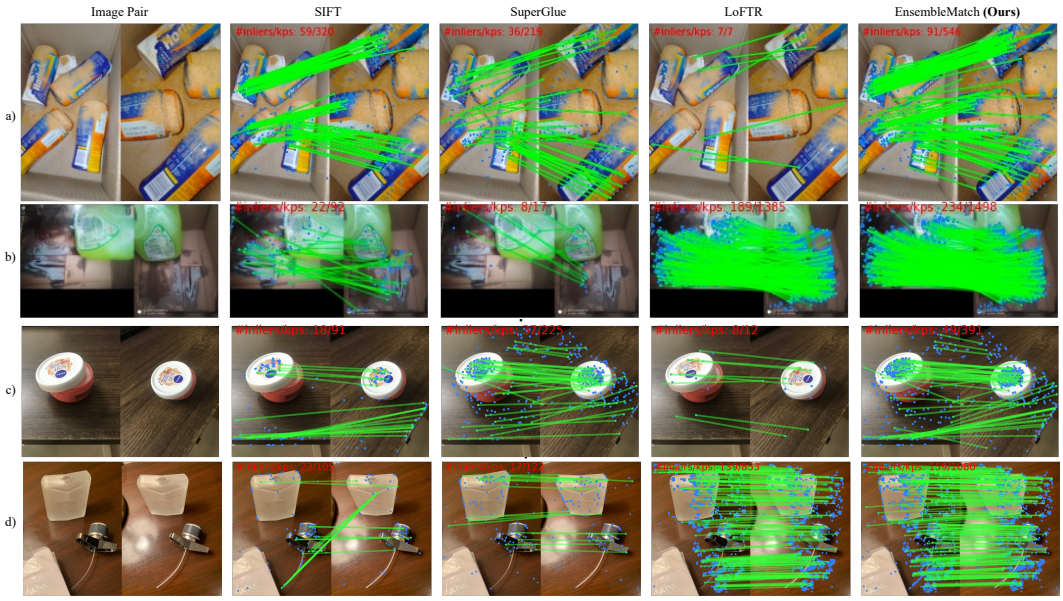

Fig. 1. Qualitative comparison on challenging multi-view images from EPID-Easy. Here, the reference image is the multi-view instance of the product from the same scene. LoFTR is inconsistent in detecting keypoints within image pairs with slight viewpoint change (d) coupled with changes to scale (a). (b) represents the case where the query image is a picture of picture, SuperGlue and SIFT detects fewer high quality keypoints. SIFT and LoFTR detects fewer keypoints (c,d) due to large viewpoint changes. EnsembleMatch is able to produce more consistent matches with its match aggregation and filtering logic.

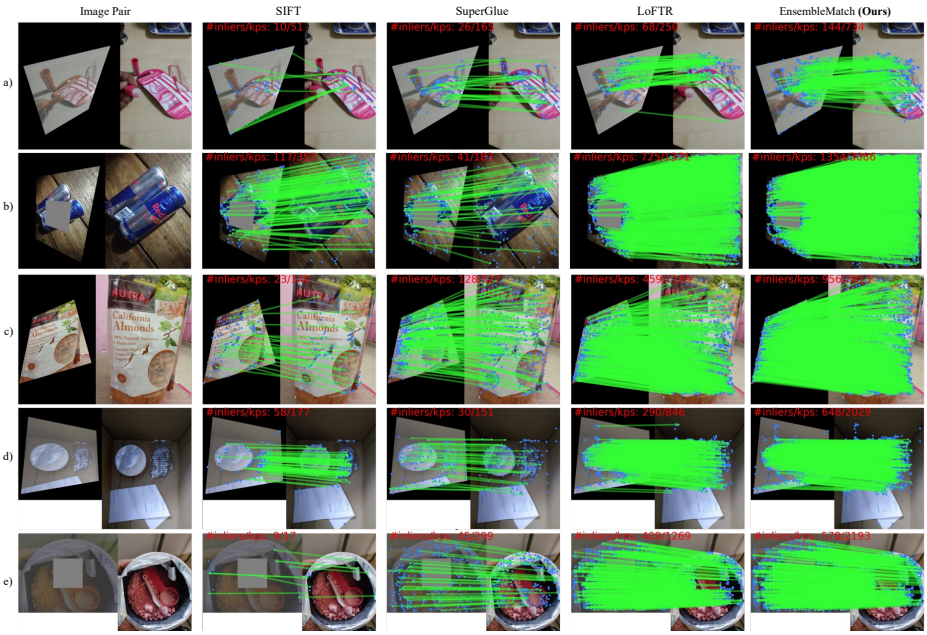

Fig. 2. Qualitative comparison on EPID-difficult dataset. The first column shows the input image pair and subsequent columns shows the performance of match algorithms. SIFT is unable to detect good matches under illumination (a,e) and high perspective transforms (c). SuperGlue detects coarse keypoints while LoFTR detects dense points in the co-visible region in challenging cases with cutout overlays, perspective and illumination changes. EnsembleMatch finds robust inliers from aggregated keypoints of individual methods.

Fig. 3. Illustration of Top10 image retrievals using our proposed method. Note that, near-duplicates (highlighted in green) appear precisely among the top retrievals, followed by visually similar images.

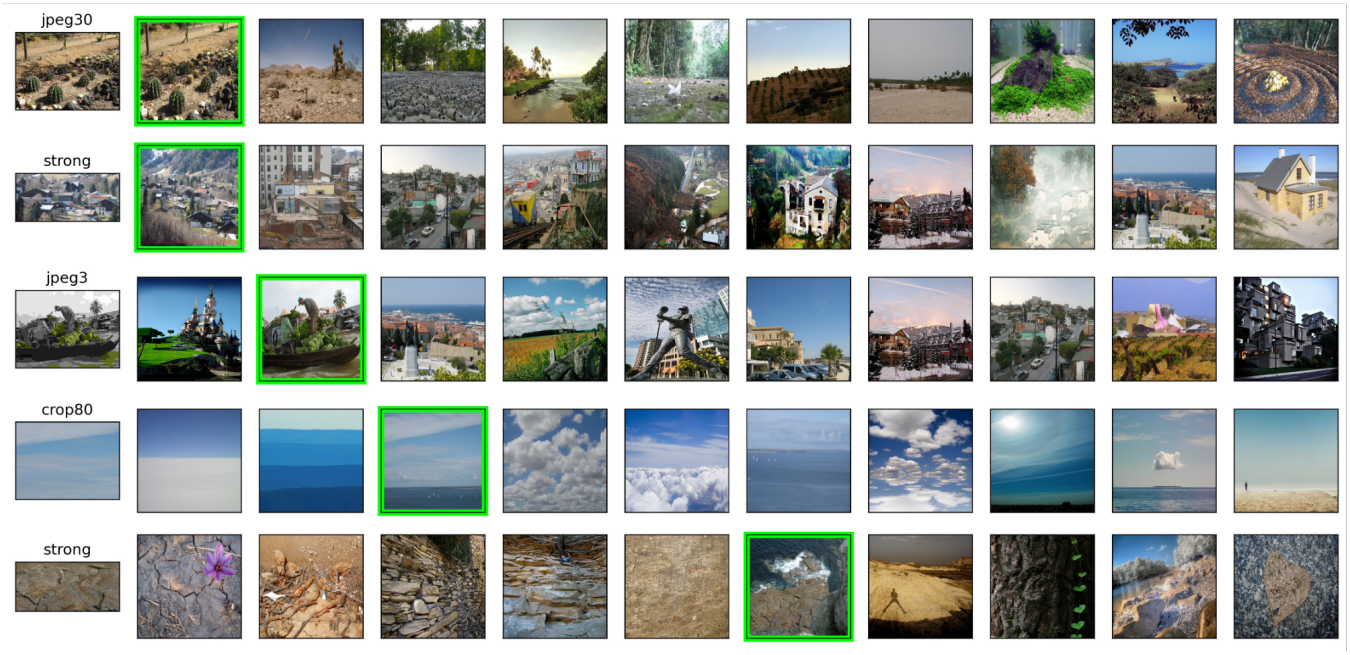

Fig. 4. Image retrieval results for sample probe images (left) from Copydays ranked by their similarity score using DTML. The original/source image is highlighted in green.

Table 6. Data augmentations used in creating positive pairs for online triplet formulation and for generating EPID-difficult dataset.

Transformation	Description	Parameter Setting
RandColorJitter	Randomly change the brightness, contrast, saturation and hue of an image with probability of p.	$p = 0.5, jitter\_strength = 0.8$
RandNoise	Adds random Gaussian noise to the image with $mean$ and var.	$p = 0.5, mean = 0, var = 0.001$
RandEncodingQuality	Randomly changes the JPEG encoding quality level.	$p = 0.5, quality \in [0, 100]$
RandOpacity	Randomly alters the opacity of an image.	$p = 0.5, level \in [0, 1]$
RandCutOut	Fill one or more rectangular areas in an image using a fill mode.	$p = 0.5, size \in [.2, .8], pos = center$
RandScreenshot	Overlay the image onto a screenshot template.	$p = 0.5$
RandPerspective	Apply a perspective transform to the image so it looks like it was taken from different viewpoint	$p = 0.5, distortion\_scale = 0.75$

Fig. 5. Sample input images from EPID-easy (top row) and corresponding transformed images in EPID-difficult (bottom row)