

An End-to-End Vision Transformer Approach for Image Copy Detection

Jiahe Steven Lee^{1,2} Wynne Hsu¹ Mong Li Lee²

¹Institute of Data Science ²Centre for Trusted Internet and Community
National University of Singapore
leejiahe@u.nus.edu, {whsu, leeml}@comp.nus.edu.sg

Abstract

Image copy detection is one of the pivotal tools to safeguard online information integrity. The challenge lies in determining whether a query image is an edited copy, which necessitates the identification of candidate source images through a retrieval process. The process requires discriminative features comprising of both global descriptors that are designed to be augmentation-invariant and local descriptors that can capture salient foreground objects to assess whether a query image is an edited copy of some source reference image. This work describes an end-to-end solution that leverage a Vision Transformer model to learn such discriminative features and perform implicit matching between the query image and the reference image. Experimental results on two benchmark datasets demonstrate that the proposed solution outperforms state-of-the-art methods. Case studies illustrate the effectiveness of our approach in matching reference images from which the query images have been copy-edited.

1. Introduction

Social media platforms have facilitated the creation of online communities with open interactions. However, these platforms have been exploited to spread misinformation and harmful content at an unprecedented scale and sophistication. Billions of images are being uploaded to these platforms each day, including images that have been intentionally edited to elicit specific response. Figure 1 shows an example of a copy-edit image of the Ukraine-Russia war that was shared on Twitter. To combat the dissemination of potentially offensive content, a scalable content moderation solution is needed to enable faster responses before it spirals out of control and becomes widespread.

Preserving information integrity in social media is critical. Content moderation on photo-sharing platforms and social networks often involves removing misinformation [7] and offensive memes [10] to maintain the credibility and integrity of the information shared. A scalable content mod-

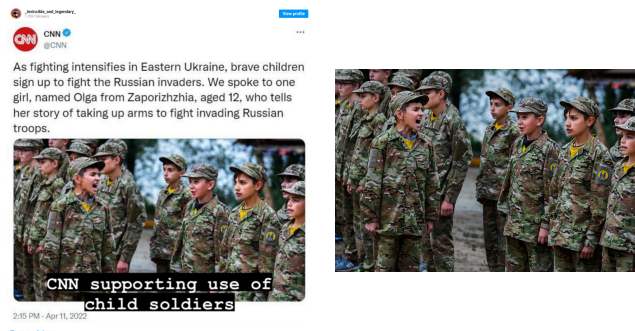


Figure 1. Real-world example where the image purportedly depicts child soldiers being recruited to fight in the war. It overlaid the official Twitter handle of news broadcaster CNN to give authenticity. In reality, the image was taken at a summer camp before the Ukraine-Russia war, where children were taught basic battlefield skills¹.

eration solution is needed to enable faster responses to mitigate the dissemination of potentially offensive content before it spirals out of control and becomes widespread. The task of copy-edit detection aims to identify images that have been edited from some original source and is vital in the content moderation process.

Algorithms for detecting copy-edit images involve a two-step "retrieve-and-match" process [4]. In the retrieval step, a search is carried out on a repository of source or reference images to identify potential candidates that may match the modified query images. The matching step gives a score if the queried image is an edited copy to any of the retrieved images. In scenarios where the query image has undergone substantial modifications, finding a matching reference image is non-trivial. Conventional near-duplicate detection algorithms that focus on image-level comparison have difficulty with instances where the original image has been overlaid with another image or embedded within different contexts, such as social media news feeds.

¹<https://factcheck.afp.com/doc.afp.com.328N282>

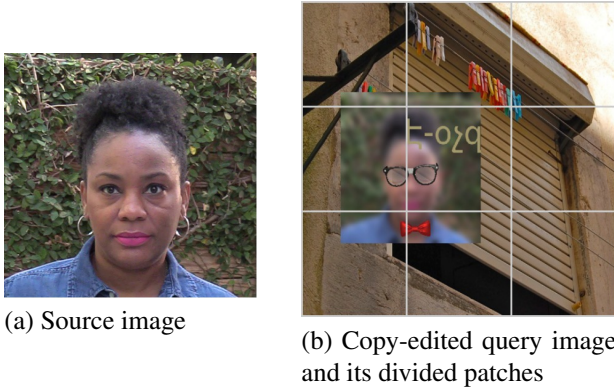


Figure 2. Example source image that has been copy-edited and overlaid onto another image.

The proliferation of copy-edit images has become such a significant issue that Facebook recently launched the Image Similarity Challenge 2021 [6] to determine if a query image is an edited copy of a source image from a reference corpus of one million images. While the winning solutions [9, 22, 24] are effective at detecting copy-edit images where changes occur within individual patches, it cannot detect modifications that span multiple patches, as illustrated in Figure 2, where the woman’s face spans four non-overlapping patches.

To overcome this limitation, we propose an end-to-end Transformer-based solution called CEDetector to facilitate robust matching between query and source reference images. CEDetector extracts deep image descriptors that capture spatial and geometric information of image patches, as well as contextual details from neighbouring patches. Then it utilizes self-attention and cross-attention mechanisms in a vision transformer to perform implicit matching of these descriptors. We train CEDetector in a self-supervised manner to determine whether a given pair of images represents a copy-edit version of each other.

Experimental results on benchmark datasets show that CEDetector outperforms state-of-the-art methods and significantly increase the accuracy of copy-edit image detection. Case studies further validate the effectiveness of our approach in matching reference images even when substantial copy-editing has taken place.

2. Related Work

The top three copy-edit solutions in the Image Similarity Challenge 2021 [6] are D2LV [24], SEPARATE [9], and ImgFp [22]. D2LV [24] involves extensive preprocessing and an ensemble of ResNet models to extract compact feature vectors, which can be computationally expensive. The authors employ two types of matching: (a) global-patch matching where the query image is divided into patches,

and the similarity between the feature vectors of each query patch and the entire reference image is computed; (b) patch-global matching where the reference image is partitioned into patches, and the similarity between the feature vectors of each reference patch and the entire query image is computed. This method cannot handle modifications that span across different patches.

SEPARATE [9] reduces the query and reference images in the width dimension by half before combining them as one image. This combined image is then fed into a ViT to calculate the matching score as a binary classification task. However, this approach results in a loss of spatial resolution which leads to subpar performance in detecting image manipulations.

ImgFp [22] computes global matching scores by taking the similarity between the feature vectors of query and reference images, and local matching scores by using SIFT keypoints [16] and BoW [18]. Its applicability in real-world settings is limited due to extensive tuning for the SIFT matching threshold. EsViT [13] is a self-supervised Vision Transformer that extends the work of [2] by introducing a region-matching method to learn robust features from input images.

Asymmetrical Similarity Learning (ASL) [25] extends D2LV by introducing a distance-based metric learning approach in addition to global-patch and patch-global matching. This approach minimizes the distance between correct reference and query image feature vectors, and maximizes the distance between incorrect ones. Similar to D2LV, ASL’s performance suffers when the modification spans across different patches.

3. Proposed Approach

Given a query image q , reference image r and corpus of reference images, the goal is to determine the likelihood that q is an edited copy of r . Figure 3 gives an overview of our proposed CEDetector which aims to provide a robust detection of copy-edit images. We first obtain six patches from the query image. Our initial experiment shows that using six patches is a good balance covering sufficient diverse regions of an image while maintaining computational efficiency. Each patch x_q is divided into N non-overlapping partitions. We use a linear embedding to obtain a sequence of image tokens $e_q^1 \cdots e_q^N$. These image tokens encode a localized view by encapsulating the spatial and geometric information of the partitions.

We prepend a learnable embedding $e_q^{[CLS]}$ to the sequence of tokens to provide global-level information of the patch. The augmented sequence $[e_q^{[CLS]}, e_q^1 \cdots e_q^N]$ are fed into DINO [2], a self-supervised Vision Transformer (ViT) based on self-distillation pretraining, to obtain a sequence of tokens $[h_q^{[CLS]}, h_q^1; \cdots; h_q^N]$ which is aggregated to form

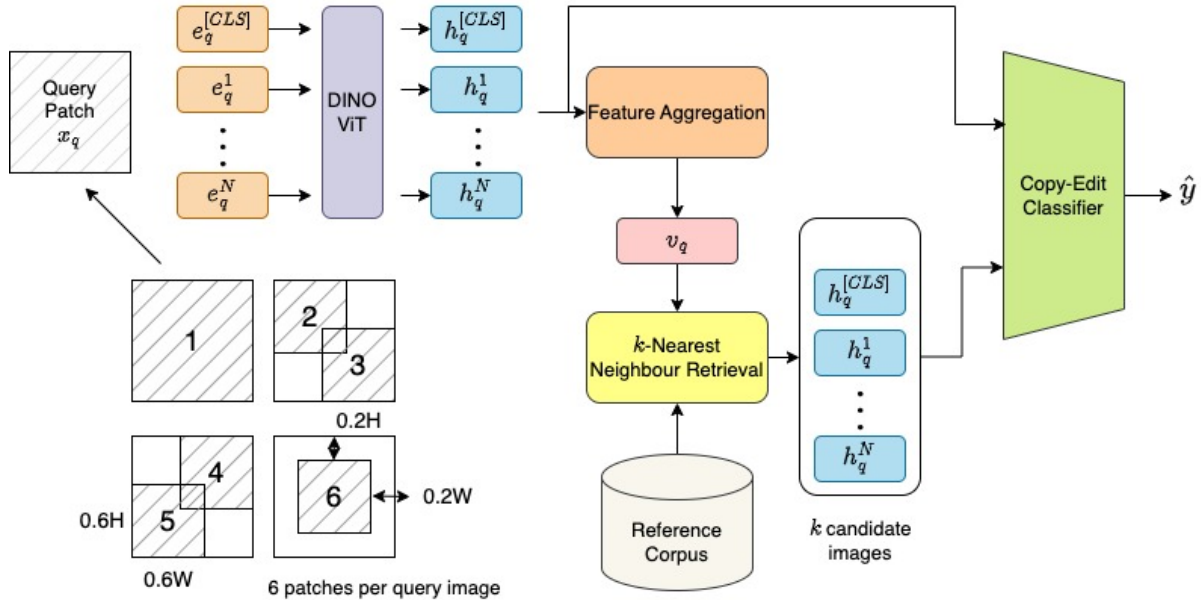


Figure 3. Overview of CEDetector.

the deep image descriptor v_q .

We use v_q to retrieve the corresponding descriptors of a set of candidate images that are most similar to the patch. Note that the reference corpus consists of the token sequence $[h_r^{[CLS]}; h_r^1; \dots; h_r^N]$ and aggregated deep image descriptor v_r of each reference image x_r obtained through DINO and feature aggregation.

For each retrieved image, we pass its token sequence, together with token sequence of the patch, to the Copy-Edit Classifier which outputs the likelihood that the patch is a copy-edit of the retrieved image. After processing all the six patches, we take the maximum score as the overall likelihood that the query image has been copy-edited.

3.1. Feature Aggregation

Using global feature vectors to retrieve candidate images has limitations since they are coarse-grained and may be sensitive to clutter or irrelevant objects. We overcome this limitation by extracting significant regionalized features that can help to identify images that have been partially edited or overlaid, including changes that are subtle or sophisticated.

We use ViT as it relies on self-attention mechanisms to capture long-range dependencies and relationships between different image regions. Unlike convolutional neural networks that process images through localized filters, ViT decomposes an image into a sequence of patches and use self-attention mechanisms to weigh and relate different parts of the image, irrespective of their position. This allows ViT to understand complex compositions and modifications in im-

ages, making it effective in discerning subtle discrepancies that indicate copy-edit manipulations.

Figure 4 shows the feature aggregation process. The self-attention mechanism in the last layer L of the DINO model highlights salient foreground regions by assigning an attention score to each partition of a patch. We perform an element-wise multiplication between the attention score of the CLS token $\alpha_L^{[CLS]}$ and h_L^i as follows:

$$\mathbf{u} = \alpha_L^{[CLS]} \otimes \mathbf{h}_L \quad (1)$$

where $\mathbf{h}_L = [h_L^1; \dots; h_L^N]$ is the output embeddings at L .

We apply GeM Pooling [20] followed by whitening [8] on \mathbf{u} to obtain the salient regional features. With this, we form the deep image descriptor $v_q = [z; u]$ where z is obtained from the projection of $h^{[CLS]}$.

3.2. Copy-Edit Classifier

After retrieving a set of candidate images, the next step is to classify whether the query patch has been copy-edited from any of these images. Figure 5 give the details of the Copy-Edit Classifier. Given the pair of query patch x_q and a candidate reference image r , we feed their token sequences h_q and h_r into the cross-attention layer to compute an affinity matrix between h_r and h_q :

$$\text{cross-attention}(h_r, h_q) = \frac{(h_q \cdot W_1) \cdot (h_r \cdot W_2)^T}{\sqrt{d}} \cdot (h_r \cdot W_3) \quad (2)$$

where W_1, W_2, W_3 are learnable weight matrices, d is the dimension of the weight matrix, and $(\cdot)^T$ is the transpose operator.

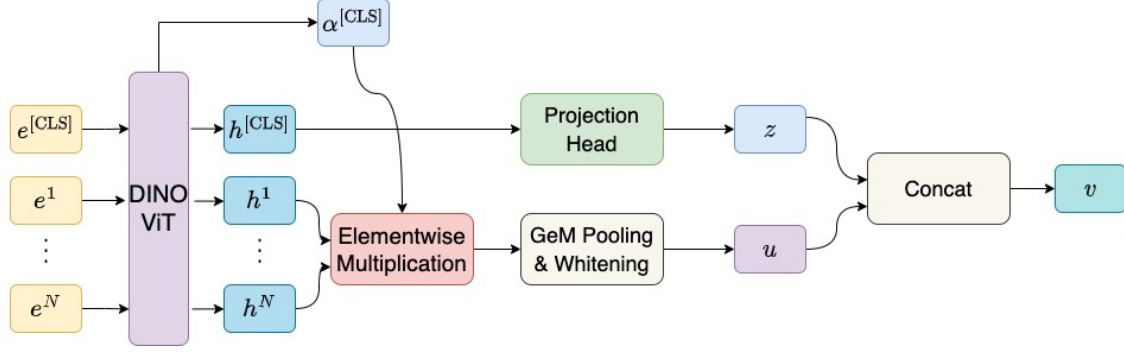


Figure 4. Details of Feature Aggregation.

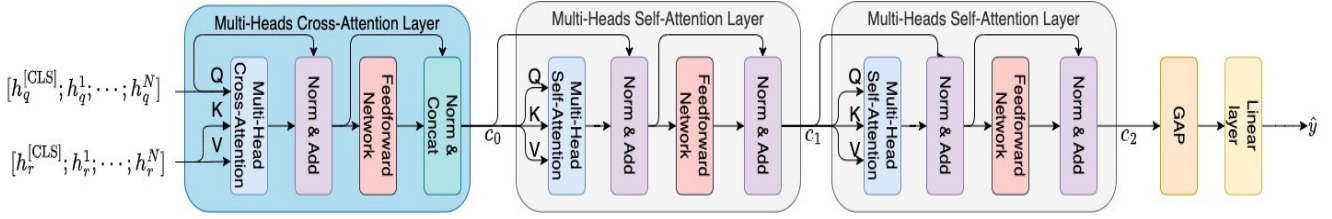


Figure 5. Details of Copy-Edit Classifier.

Cross-attention between the embeddings enable the matching between the query and candidate reference images. Output C_0 from the cross-attention layer is fed into the first layer of the first multi-head self-attention block. The output C_M from the last layer of the second multi-head self-attention block is passed to global averaging pooling layer [15], followed by linear layer. We apply a Sigmoid function to obtain the likelihood \hat{y} that the query image is an edited copy of the reference image.

4. Model Training

We adopt a joint training approach for our CEDetector for the retrieval and matching phases. Similar to COCO-LM [17], we employ self-supervised learning to train a token-level DINO Vision Transformer and an image-level Copy-Edit Classifier. This allows learning of fine-grained differentiation features in the matching phase and enhances the model’s ability to fine-tune across both retrieval and matching tasks, leading to improved performance.

A pair of images (x_i, x_j) is said to be a positive sample if x_j is obtained by applying some sequence of transformations on x_i . Otherwise, the pair is a negative sample if x_j is the result of sequence of transformations on some image $x_k \neq x_i$. Let z_i and z_j be the projections of the $h^{[CLS]}$ tokens of x_i and x_j respectively. We employ the contrastive loss function in SimCLR [3] to maximise the agreement by pulling positive samples close while pushing negative samples away in the embedding space.

We use the normalized temperature-scaled cross entropy

NT-Xent in our SimCLR loss function:

$$\text{NT-Xent}(i, j) = -\log \frac{\exp(z_i \cdot z_j / \tau)}{\sum_{k=1}^{2B} \mathbb{I}_{[k \neq i]} \exp(z_i \cdot z_k / \tau)} \quad (3)$$

where τ is the temperature hyperparameter and B is the number of images in the mini-batch. Then we have

$$\mathcal{L}_{\text{SimCLR}} = \frac{1}{2B} \sum_{i=1}^B [\text{NX-Xent}(i, j) + \text{NT-Xent}(j, i)] \quad (4)$$

To increase the distance between positive samples and its nearest neighbouring negative samples, we use a regularizer term based on the Kozachenko-Leonenko differential entropy estimator [12]:

$$\mathcal{L}_{\text{KL}} = \frac{1}{B} \sum_{i=1}^B \log(\min_{i \neq k} \|z_i - z_k\|_2) \quad (5)$$

The final contrastive loss function used in CEDetector framework for training DINO is given by:

$$\mathcal{L}_{\text{contrast}} = \mathcal{L}_{\text{SimCLR}} + \lambda \mathcal{L}_{\text{KL}} \quad (6)$$

In addition, we apply multi-similarity loss in deep metric learning [26] to learn an embedding space where salient regions in positive samples are projected close to each other while regions in negative samples are projected away from

each other, expressed as follows:

$$\mathcal{L}_{\text{MSL}} = \frac{1}{B} \sum_{i=1}^B \left[\frac{1}{\alpha} \log \left(1 + \sum_{k \in \mathcal{P}_i} \exp(-\alpha(u_i \cdot u_k - \gamma)) \right) + \frac{1}{\beta} \log \left(1 + \sum_{j \in \mathcal{N}_i} \exp(\beta(u_i \cdot u_j - \gamma)) \right) \right] \quad (7)$$

where u_i, u_j, u_k are the token embeddings obtained in Equation 1 for images x_i, x_j, x_k respectively, $\mathcal{P}_i, \mathcal{N}_i$ are positive and negative samples, α, β are weights applied to positive and negative samples, and γ is the margin.

The overall loss function can be expressed as

$$\mathcal{L} = \mathcal{L}_{\text{Contrast}} + \mathcal{L}_{\text{MSL}} + \mathcal{L}_{\text{BCE}} \quad (8)$$

where \mathcal{L}_{BCE} is the binary cross entropy loss given by

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{B} \sum_{i=1}^B (y_i \cdot \log(\hat{y}_i)) + ((1 - y_i) \cdot \log(\hat{y}_i)) \quad (9)$$

where \hat{y}_i is the probability that x_i is copy-edited image, y_i is 1 if x_i is indeed a copy-edit, and 0 otherwise.

The binary cross entropy loss explicitly trains the Copy-Edit Classifier to obtain the likelihood \hat{y} . We use stop gradient to decouple the gradient backpropagation to DINO to ensure training stability. In our experiments, we set the temperature τ used in NT-Xent to 0.025, and entropy weight λ in Equation (6) to 0.5. The hyperparameters α, β, γ in Equation (7) are set to 2, 50, 1 respectively.

5. Performance Study

We implement our proposed CEDetector in PyTorch and carry out experiments on 4 NVIDIA-A100 GPU to evaluate its effectiveness in detecting copy-edit images. We utilize DeepSpeed-stage 2 optimizer [21] to aid multi-GPU optimization. We use Adam optimizer [11], with a learning rate of 0.00002 and a batch size of 64 with 30 epochs. The following datasets are used:

- Image Similarity Challenge 2021 dataset (ISC) [6]. This dataset consists of a reference set of 1 million images, a training set of 1 million images, and a testing query set of 50,000 images, out of which 40,000 are distractor images.
- Negative Distractor for Edited Copy dataset (NDEC) [25]. This dataset focuses on hard negative distractor images that are visually similar to samples in the reference set but are not copy-edits. The reference set is the same 1 million images as the ISC reference set. The query set has 49,252 images, out of 5009 are copy edits. Among the remaining 44,243 distractors, 24,252 are hard negative.

We use a Vision Transformer that has been pretrained on ImageNet with a patch size of 16 to initialize the weights

of DINO-ViT. We use a resolution of 224x224 for training, and increase the resolution to 384x384 during testing to enable the model to process more fine-grained details in the images. The multi-head cross-attention and self-attention blocks in the Copy-Edit Classifier have the same parameters as Transformer Encoder blocks [23].

We train our CEDetector model by augmenting the images on-the-fly using 35 types of transformations from the Augly library [19] and the Albumentations library [1] to emulate possible copy-edits scenarios in the real world. For an image x drawn from the image corpus, a transformation is drawn uniformly without replacement from a set of transformation techniques. We apply a sequence of transformations to obtain an augmented image x' . This allows the model to see multiple version of augmented images, thereby enabling the model to be augmentation-invariant. As inspired by Next Sentence Prediction, [5], we create the positive label by pairing x with x' 50% of the time and negative label by pairing x with $x'' \neq x'$ 50% of the time. With this, we train the CEDetector in a self-supervised manner.

We use the metrics micro average precision (μAP), also known as the area under the precision-recall curve [6], and recall when precision is at 90% (R@P90) as our evaluation metrics. The precision-recall curve plots precision (positive predictive value) on the y-axis against recall (sensitivity or true positive rate) on the x-axis at various classification thresholds. μAP can be computed by summing up the areas of the rectangles formed under the curve.

5.1. Sensitivity Experiments

We observe that a copy-edit detector has to be robust against a variety of augmentations as typically seen in real-world scenarios. We first carry out sensitivity experiments on the ISC dataset to determine the optimal number of transformations needed to train the CEDetector.

Table 1(a) shows the effect on the performance of CEDetector as we vary the number of transformations applied to one image. We see that applying four transformations on an image yields the optimal performance. When only one or two transformations are applied, the images may not be sufficiently transformed to emulate the complex augmentations seen in real-world augmented images, leading to poor results. Conversely, when six transformations are applied, the images may become excessively transformed to the point where they no longer resemble real-world augmented images, resulting in a drop in performance.

We also carry out experiments to determine the optimal number of candidate images to retrieve for comparison and matching with the query image. This parameter balances the trade-off between efficiency and accuracy of the matching process. A larger k value would result in a slower matching process while a smaller k value may not retrieve the correct reference image within its candidate set leading

Table 1. Sensitivity experiments on ISC dataset.

#Trans	μ AP	R@P90	#Cand	μ AP	R@P90
1	0.170	0.137	1	0.637	0.588
2	0.376	0.325	2	0.727	0.689
3	0.691	0.624	5	0.811	0.774
4	0.854	0.803	10	0.854	0.803
5	0.832	0.786	15	0.844	0.795
6	0.633	0.592	20	0.829	0.781

(a) Effect of number of transformations

(b) Effect of number of candidate images retrieved

to decreased accuracy. Table 1(b) shows that the optimal value for k is 10. Retrieving only one candidate image for matching is clearly not sufficient while retrieving 20 images increases the number of false positive, lowering the matching accuracy.

5.2. Comparative Study

We first compare CEDetector with the top three results published in the Image Similarity Challenge 2021: D2LV [24], SEPARATE [9], ImgFp [22]. Table 2(a) shows that CEDetector outperforms all these methods without the need for extensive pre-processing or heavily engineered models. Note CEDetector effectively utilizes both global- and local-level information as it infers from ViT [CLS] token and ViT patch embeddings.

We also compare CEDetector with state-of-the-art ASL [25] and EsViT [14] on the more difficult NDEC dataset. Table 2(b) shows that the μ AP for CEDetector exceeds ASL by 5.14%. Both ASL and D2LV utilise compact dense vectors to compute the similarity metric which may not capture fine-grained information that provide crucial details necessary for matching the images. CEDetector is able to circumvent this issue by incorporating both global and local descriptors when calculating the matching score. Note that there is no published R@P90 results for D2LV, EsViT and ASL on the NDEC dataset.

5.3. Ablation Study

We examine the effect of the various components in overall loss function \mathcal{L} on the performance of CEDetector. Table 3 shows the results. The largest drop in performance occurs when we use the multi-similarity loss \mathcal{L}_{MSL} only. This suggests that information provided by the local descriptors of salient regions may not be sufficient. When we use \mathcal{L}_{SimCLR} only, we achieve a μ AP of 0.743 and R@P90 of 0.693, highlighting that SimCLR is able to produce augmentation-invariant global descriptors. The results are improved when we use $\mathcal{L}_{Contrast}$ only because the regularizer term, Kozachenko-Leonenko differential entropy estimator, ensures that the deep image descriptors are pro-

Table 2. Results of comparative experiments.

Methods	μ AP	R@P90
CEDetector	0.854	0.803
D2LV	0.832	0.731
SEPARATE	0.829	0.792
ImgFp	0.768	0.672

(a) ISC dataset

Methods	μ AP	R@P90
CEDetector	0.691	0.683
D2LV	0.588	n/a
EsViT	0.456	n/a
ASL	0.642	n/a

n/a - published results not available

(b) NDEC dataset

Table 3. Ablation study on the ISC dataset.

Methods	μ AP	R@P90
\mathcal{L}_{MSL} only	0.624	0.583
\mathcal{L}_{SimCLR} only	0.743	0.693
$\mathcal{L}_{contrast}$ only	0.775	0.748
$\mathcal{L}_{contrast} + \mathcal{L}_{MSL}$	0.808	0.788
$\mathcal{L}_{contrast} + \mathcal{L}_{MSL} + \mathcal{L}_{BCE}$	0.854	0.803

jected uniformly in the hypersphere.

Using $\mathcal{L}_{Contrast} + \mathcal{L}_{MSL}$ leads to further improvements in both μ AP and R@P90 indicating that both global and local descriptors are important in capturing augmentation-invariant information in an image. The best performance is achieved when we include the binary cross-entropy loss \mathcal{L}_{BCE} that helps in identifying distinguishable features to better classify whether an image is a copy edit.

6. Case Studies

We showcase sample images to provide insights into the performance of CEDetector and potential areas for improvement. Figure 6 shows a query image from ISC dataset and the six patches obtained. For each patch, we give the top-1 retrieved reference image and the corresponding score. We see that query patch 2 is able to accentuate the eagle in the source image, and the retrieved image corresponds to the actual source. This enables the copy-edit classifier to correctly classify that the query image is a copy-edit. Figure 7 illustrates a challenging scenario where CEDetector incorrectly identifies as copy-edit. This is because the source image has been severely cropped and embedded within a small box in the query image. The small box containing the source image is disproportionately smaller compared to the patch size used by CEDetector, making it difficult for the patches to focus on the correct region. This



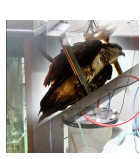
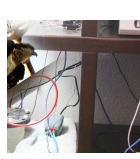




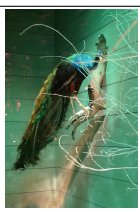





Query image 1	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6
						
Source image	Retrieved 1	Retrieved 2	Retrieved 3	Retrieved 4	Retrieved 5	Retrieved 6
						
Score	0.114	0.944	0.214	0.095	0.074	0.181

Figure 6. Sample query image from ISC dataset that has been correctly classified.

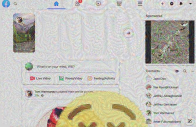
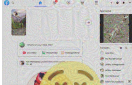


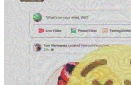
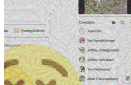
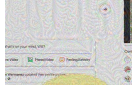
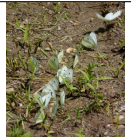

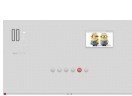




Query image 3	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6
						
Source image	Retrieved 1	Retrieved 2	Retrieved 3	Retrieved 4	Retrieved 5	Retrieved 6
						
Score	0.078	0.061	0.076	0.085	0.054	0.082

Figure 7. Sample query image from ISC dataset that has been incorrectly classified.








Query image 4	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6
						
Source image	Retrieved 1	Retrieved 2	Retrieved 3	Retrieved 4	Retrieved 5	Retrieved 6
						
Score	0.614	0.098	0.122	0.112	0.933	0.919

Figure 8. Sample query image from NDEC dataset that has been correctly classified.






















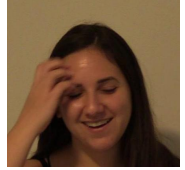






Query image 5	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6
						
Source image	Retrieved 1	Retrieved 2	Retrieved 3	Retrieved 4	Retrieved 5	Retrieved 6
						
Score	0.122	0.173	0.180	0.833	0.233	0.945
Query image 6	Patch 1	Patch 2	Patch 3	Patch 4	Patch 5	Patch 6
						
Source image	Retrieved 1	Retrieved 2	Retrieved 3	Retrieved 4	Retrieved 5	Retrieved 6
						
Score	0.127	0.084	0.241	0.098	0.082	0.104

Figure 9. Sample query images from NDEC dataset that have been incorrectly classified.

leads to incorrect retrieval of the reference image, and consequently misclassification.

Figure 8 shows a sample query image in the NDEC dataset where CEDetector has successfully identified as copy-edit. We see that two patches (Patch 5 and Patch 6) of query image 4 are matched to the source image, resulting in a correct classification. Figure 9 highlights two NDEC query images that was incorrectly classified by CEDetector. Query image 5 is a difficult case as there are several source images with similar view. The highest scoring retrieved image (Retrieved 6) is not the source image, but it is an image taken from a different perspective of the source image. Although the copy-edit classifier correctly determines that it is a copy-edit, however, the identified reference image is not the source image. As such, we consider this as an incorrect classification. Query image 6 is an incorrect classification by CEDetector. Here, the copy-edit classifier says the query image is not a copy-edit. Closer examination reveals that the embedded reference image in query image 6 has been

subjected to an adversarial attack transformation which can significantly degrade retrieval performance.

7. Conclusion

We have described a robust solution to the complex image copy-edit detection problem. The proposed solution leverages deep image descriptors, allowing relevant features to be extracted and matched to identify manipulated images. Experiment results on ISC and NDEC datasets shows that CEDetector is able to outperform state-of-the-art methods, even in complex scenarios with subtle or sophisticated alterations. CEDetector has the potential to combat image-based misinformation and enhance the integrity of digital media. Future work includes using adaptive patch sizes to better handle small embedded reference images and withstand adversarial attack transformation.

Acknowledgments. This work is supported by the Ministry of Education, Singapore, under its MOE AcRF TIER 3 Grant (MOE-MOET32022-0001).

References

- [1] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020. arXiv:1809.06839 [cs]. 5
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *arXiv:2104.14294 [cs]*, 2021. 00026 arXiv: 2104.14294. 2
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv:2002.05709 [cs, stat]*, 2020. 00646 arXiv: 2002.05709. 4
- [4] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew. Deep learning for instance retrieval: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(06):7270–7292, 2023. 1
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, 2019. arXiv: 1810.04805. 5
- [6] Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, Ondřej Chum, and Cristian Canton Ferrer. The 2021 Image Similarity Dataset and Challenge, 2022. Number: arXiv:2106.09672 arXiv:2106.09672 [cs]. 2, 5
- [7] Facebook. Using AI to detect COVID-19 misinformation and exploitative content. <https://ai.facebook.com/blog/using-ai-to-detect-covid-19-misinformation-and-exploitative-content>, 2020. 1
- [8] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end Learning of Deep Visual Representations for Image Retrieval, 2017. arXiv:1610.07940 [cs]. 3
- [9] SeungKee Jeon. 2nd Place Solution to Facebook AI Image Similarity Challenge : Matching Track. page 2, 2021. 2, 6
- [10] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes, 2021. arXiv:2005.04790 [cs]. 1
- [11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5
- [12] L F Kozachenko and N N Leonenko. Sample Estimate of the Entropy of a Random Vector. page 9, 1987. 4
- [13] 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, 2021. 2
- [14] 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. *arXiv:2106.09785 [cs]*, 2021. 00001 arXiv: 2106.09785. 6
- [15] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network, 2013. 4
- [16] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2
- [17] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. Cocolm: Correcting and contrasting text sequences for language model pretraining, 2021. 4
- [18] Eva Mohamedano, Kevin McGuinness, Noel E. O’Connor, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Bags of Local Convolutional Features for Scalable Instance Search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 327–331, New York New York USA, 2016. ACM. 2
- [19] Zoe Papakipos and Joanna Bitton. AugLy: Data Augmentations for Robustness, 2022. arXiv:2201.06494 [cs]. 5
- [20] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN Image Retrieval with No Human Annotation, 2018. arXiv:1711.02512 [cs]. 3
- [21] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. 5
- [22] 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, 2021. Number: arXiv:2112.02373 arXiv:2112.02373 [cs]. 2, 6
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, 2017. 23314 arXiv: 1706.03762. 5
- [24] Wenhao Wang, Yifan Sun, Weipu Zhang, and Yi Yang. D²LV: A Data-Driven and Local-Verification Approach for Image Copy Detection, 2021. Number: arXiv:2111.07090 arXiv:2111.07090 [cs]. 2, 6

- [25] Wenhao Wang, Yifan Sun, and Yi Yang. A Benchmark and Asymmetrical-Similarity Learning for Practical Image Copy Detection, 2022. arXiv:2205.12358 [cs]. [2](#), [5](#), [6](#)
- [26] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning, 2020. arXiv:1904.06627 [cs]. [4](#)