Contour-Aware Contrastive Learning for Image Manipulation Localization

Qin Li, Chunfang Yu, Zhuohang Jiang, and Jizhe Zhou

***Abstract*—Image Manipulation Localization (IML), which aims to segment the forged region from a single image at pixel-level, is one of the cornerstone tasks in information security and multimedia forensic fields. Due to the residual artifacts or anomalies in image editing, both traditional and learning-based methods heavily rely on hand-crafted features, which may only work under limited circumstances. In this paper, we tackle this limitation by comparing the differences between the manipulated and authentic region automatically via Contrastive Learning. We propose a novel Contour-aware Contrastive Learning Network (CaCL-Net) based on the encoder-decoder architecture. On the encoder side, since the contour is foremost concerned in IML, we consider the image patches sampled along the manipulation contour are the hard examples and set them as the anchor. The patches of pure tampered and authentic pixels are set as positives and negatives respectively to conduct contrastive learning. The decoder then manages to specify the manipulated regions and restores the explicit contours of the manipulations through the proposed Contour Binary Cross-Entropy (CBCE) loss. Experiments show that the proposed CaCL-Net: (i) achieves state-of-the-art performance on four popular benchmarks without any extra synthesized training dataset; (ii) escalates the localization F_1 score, which indicates our network is far more robust and less prone to overfitting. Our source code is available at: Counter-Aware-Contrastive-Learning-for-IML.**

***Index Terms*—Information security, Image manipulation localization, Contrastive learning, Atrous convolution, Contour binary cross-entropy loss.**

I. Introduction

THE development of digital media techniques endows us with easy access to various image manipulations, such as copy-move and splicing. Manipulated images jeopardize the truth of the original context and are hard to be identified by the human eyes. Countless information security issues are then raised. For instance, under ulterior motivation, these manipulated images are intentionally leveraged to mislead netizens and spread rumors [1]. Carefully manipulated images can be used to compromise some identity authentication mechanisms based on facial recognition for security breaches, or to forge sensitive information for privacy invasions.

The Image Manipulation Localization (IML) algorithms are the key to pursuing genuineness by classifying and locating the tampered regions from the manipulated images. IML, or image forgery detection techniques, has long been heavily invested by the information security and multimedia forensic community. Manipulations or editions always leave residual artifacts or anomalies in images, and further, these artifacts or anomalies are inconsistent with the original authentic images under particular perspectives. Therefore, traditional methods rely on hand-crafted features to reveal such inconsistency, thereby locate the artifacts. For example, differences in the noise-level [2] and the broken JPEG-blocking [3] are commonly investigated in IML. Inspired by these hand-crafted methods, modern deep learning-based IML methods involve similar low-level clues as the additional features. For instance, RGB-N [4] utilizes SRM filter to extract the noise features as priors; ManTra-Net [5] proposes the z_1 score to localize the manipulation as the outliers; SPAN [6] extends the SRM filter by the BayarConv2D and focuses on the abnormal spatial relations of pixels. The performance of these methods depends on the adaptivity of the hand-crafted features against various manipulations. As a result, large self-collected or synthesized datasets containing copious manipulations are often used to achieve state-of-the-art performance. ManTra-Net engages a self-collected dataset that contains 385 manipulation types; RGB-N employs a large randomly synthesized dataset by MS-COCO dataset [4]. However, the additional dataset is not available to the public and further impedes a fair comparison. The roughly synthesized dataset is also different from the real-world cases.

Therefore, in this paper, we aim to rely on the data and build an IML model which completely discards the hand-crafted features. To achieve this goal, considering the essence of the IML task is capturing the artifacts, we propose a contour-aware contrastive learning method to adaptively extract the inconsistency between artifacts and authentic images.

As depicted in Figure 1, we first formulate the IML as a classic—encoder-decoder architecture. Then, we guide the encoder's feature generation process by self-supervised contrastive learning. Contrastive learning carries out pair-wise comparisons by crafting triple-tuples of **{anchor, positive, negative}**, which are image patches in IML. The learning objective optimizes the margin of pair-wise similarities to better distinguish positives from negatives regarding an anchor. In IML, the manipulated and the authentic patches are born to be the positives and negatives. For the anchor pick, first and foremost, we can directly localize the manipulation by delineating its contour. Besides, unlike image patches of pure manipulated or authentic regions, the contour patches contain both manipulated and authentic pixels, thereby are hard examples in contrastive learning. Thus, we select contour patches after a Fully Connected (FC) layer as the anchor, and the manipulated and authentic patches are respectively

Qin Li and Chunfnag Yu are with the Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China. E-mail: qli@sei.ecnu.edu.cn

Jizhe Zhou and Zhuohang Jiang are with the College of Computer Science, Sichuan University, Sichuan, China. Jizhe Zhou is the corresponding author E-mail: jzzhou@scu.edu.cn

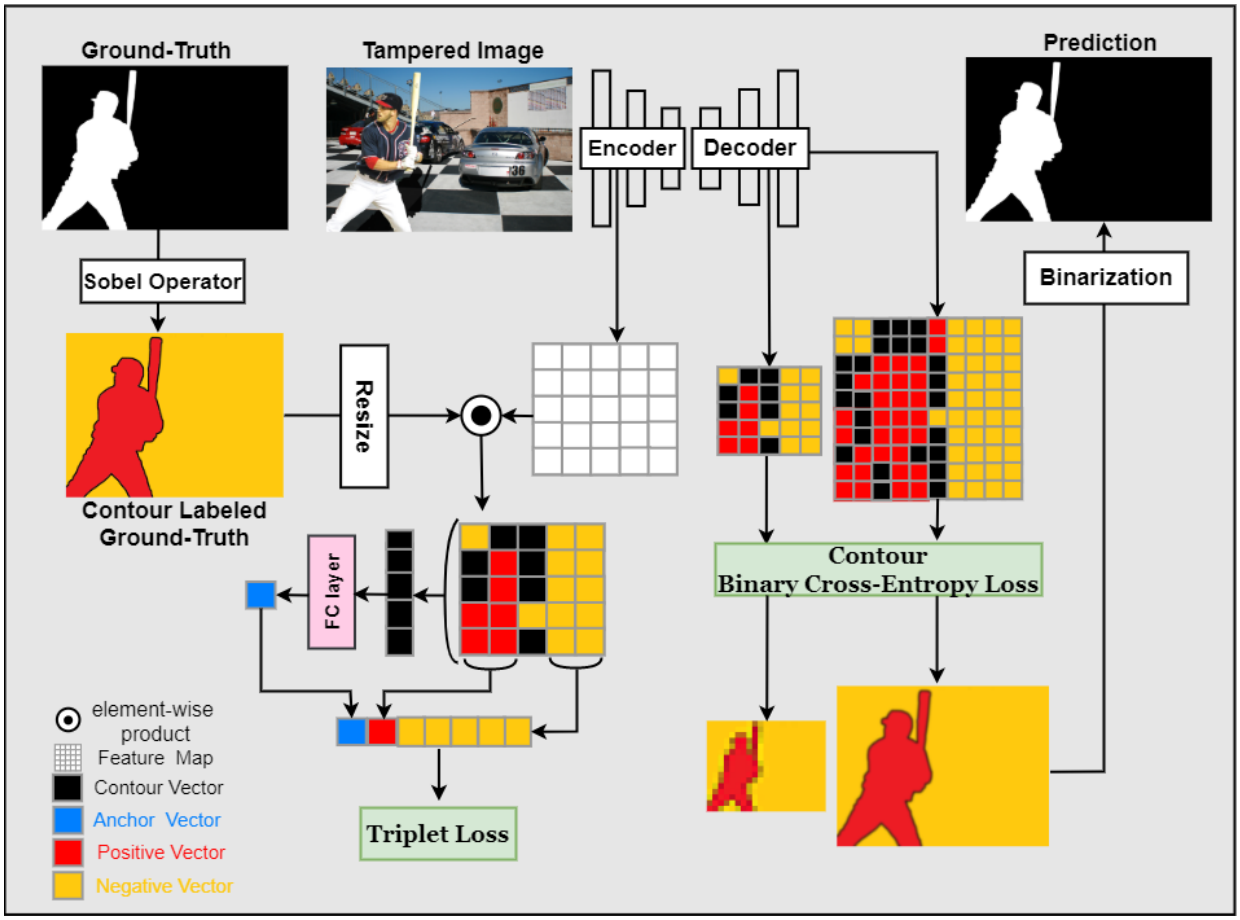


Fig. 1. The conceptual working pipeline of our CaCL-Net. The legend is in the left-down corner. CaCL-Net follows an encoder-decoder architecture. We re-render contour, manipulated and authentic pixels in the contour labeled ground truth with black, red and yellow, then resize it to product with the feature map. Contrastive learning uses the labeled featured map to compute their triplet loss. In the decoder, CBCE loss along with auxiliary classifiers guide the upsampling process. The output is also contour labeled, and we binarize it to match the ground truth.

the positives and negatives. Then, we leverage triplet loss to minimize the intra-class differences between the contour patches and manipulated image patches, and maximize the inter-class differences between the contour and authentic image patches. With the learned triplet loss, the encoder is aware of the pivotal contour pixels, thereby focus on the essential discrepancies between the manipulated artifacts and their authentic background.

However, contrastive learning is conducted patch-wise, but the IML requires pixel-wise manipulation masks. Closing the gap between patches and pixels is the other challenge when introducing contrastive learning into IML. In our previous work MSED [7], we find that the multi-scale features generated by different layers of the encoder extremely benefit the accuracy of the pixel-wise manipulation mask restoration. Inspired by this finding, we design auxiliary classifiers in each level of the decoder to guide each upsampling process through a Contour Binary Cross-Entropy (CBCE) loss. Unlike the previous methods merely using the Binary Cross-Entropy (BCE) loss at the final layer, the accuracy of contour pixels is measured and extra weighted through the CBCE loss. The auxiliary classifiers prevent the decoder from overfitting on the contour pixels.

As a whole, a hybrid loss consisting of a contrastive loss and multiple CBCE losses is constructed for our Contour-aware Contrastive Learning Network (CaCL-Net). Comprehensive experiments on public benchmarks validate that even without any self-collected training datasets, our model achieves comparable results against current data-cramming approaches. Also, CaCL-Net is more accurate to manipulations with tiny sizes or complex boundaries. Moreover, as the semi-supervised contrastive learning conducts massive comparisons between image patches, CaCL-Net is far more robust and less prone to overfitting on the relatively small benchmark test datasets. The direct evidence is that when switching the common Area Under Curve (AUC) metric to the cognate F_1 score, CaCL-Net exhibits a consistent performance between these two metrics, while existing methods incur a huge gap.

In a nutshell, our main contributions are:

- To the best of our knowledge, we are the first work that introduce contrastive learning in image manipulation localization without pre-training.
- We develop a contour-aware contrastive learning network through contrastive learning and contour binary cross-entropy loss, such that manipulation can be localized with a fine-delineation.

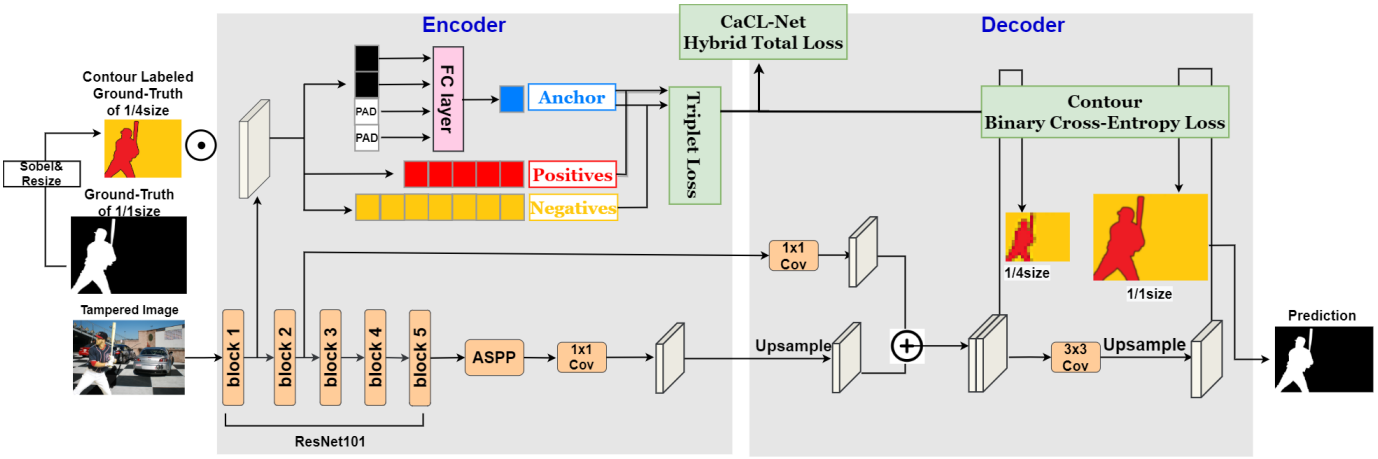


Fig. 2. The overview of proposed Contour-aware Contrastive Learning Network (CaCL-Net) for image manipulation localization. We give more details about the encoder-decoder network here. Cubes are feature maps, and rounded rectangles in honey color are convolution operations. Loss calculations are enclosed in light green. Others notations are the same with those in Figure 1. Different sizes of ground-truth are obtained through interpolation. In the encoder, the feature map after the first block of ResNet101 is used to produce contrastive pairs (**anchor, positive, negative**). "PAD" is the padding token we used to reshape the contour patches into the anchor through the Fully-Connected layer. In the decoder, the feature map after each upsampling is used to compute Contour Binary Cross-Entropy (CBCE) loss.

- Our network achieves state-of-the-art performance on benchmarks even without additional pre-training data and hand-crafted feature extraction.
- Our network escalates the F_1 score on all benchmarks, indicating higher robustness and better resistance against overfitting.

II. Related Work

Image Manipulation Localization. Research on IML based on handcrafted features detect the low-level manipulated artifacts within a tampered image, including Color Filter Artifacts (CFA) [8], local noise analysis [2], [9], [10], resampling [11], and double JPEG compression [3], [12], [13]. NOI(Noise Inconsistency) [2] models local noise by utilizing high-pass wavelet coefficients. CFA [8] approximates the camera filter array patterns through neighbouring pixels, which is used to generate tampering probability for each pixel. Amerini et al. [13] train model to distinguish among the single and double JPEG compressed images to localize the forgery regions. Bunk et al. [11] compute the radon transform of resampling features on overlapping image patches, and then create a heatmap through deep learning classifiers and Gaussian conditional random field model to locate tampered areas.

Recent deep learning-based work relies on an adaptive feature extraction framework, which shows promising performance in image manipulation detection. However, the main existing methods of IML are rather sensitive to different types of manipulations. This can limit those IML methods to only one specific manipulation technique, for instance, copy-move [14], [15], splicing [16], [17], and removal [18]. More recent works, by contrast, have managed to break the limitations of the manipulation techniques. RGB-N [4] proposes a two-stream Faster R-CNN network and bilinearly captures RGB tampering artifacts and local noise feature inconsistencies to identify tampered regions. ManTra-Net [5] defines the forgery detection problem as a local anomaly detection task and proposes a z_1 score to assess the probability of manipulation artifacts. SPAN(Spatial Pyramid Attention Network) [6] first models the multi-dimensional spatial relationship between image patches by constructing a pyramid of local self-attention blocks. The latest advances probe in two directions. One direction employs the multi-view and multi-scale features yielded by the upsampling process of the encoder to restore the manipulated mask [19], [20]. Such multi-scale features are already proven effective in the semantic segmentation task by the U-net structure. The other direction manages to alter the Transformer structure such that the dense self-attention mechanism can serve the IML task [21].

Above mentioned methods are trained end-to-end and demonstrate the effectiveness and robustness to localize manipulating artifacts with diverse tampering techniques. Similarly, our proposed CaCL-Net is also designed for IML without the limitations of manipulation techniques, and localize the entire tampered pixels within a single image.

Contrastive Learning. Contrastive learning is emerging in semi-supervised and unsupervised visual representation learning [22]–[25]. [22] employs contrastive learning in the kinship field. Resemblance directly related to kinship is maximized, while the confounding patterns are excluded by the contrastive loss. Contrastive learning on the scene graph is introduced by [23]. Graphical contrastive loss compares positive relationships against hard negative relationships to disambiguate instances relations. [24] adopts contrastive learning to disentangle factor variations of face generation by comparing pairs of generated images under different combinations of factor variations. [25] introduces patch-wise contrastive learning to handle the unpaired image translation as a classification task. As far as we know, we are the first to introduce contrastive learning into IML. We follow the common triplet loss to conduct our contrastive learning.

Atrous Convolution. Atrous convolution [26], [27] is proposed for the image segmentation task and brought to use in

the encoder-decoder architecture by [28]. Atrous convolution supersedes the ordinary convolution kernels with atrous ones to preserve the spatial relations between pixels. Most recent works [6], [14] on the IML field have already noticed the superiority of atrous convolution and employed it in their study. Like [6], in CaCL-Net, we adopt the atrous convolution network built by [28] as the base model for feature extraction.

III. Proposed Method

The framework of our proposed Contour-aware Contrastive Learning Network (CaCL-Net) is illustrated in Figure 2. In the encoder, the contour region, manipulated region, and authentic region are extracted and correspondingly set as the **anchor, positive, negative** to implement contrastive learning. In the decoder, we develop a Contour Binary Cross-Entropy (CBCE) loss to recover the pixel-wise information of manipulated boundaries. We provide a detailed introduction in the following.

A. Encoder

1) Basic Encoder of Atrous Convolution: The encoder utilizes atrous convolution [27] to extract features for arbitrary input resolution. We select the standard ResNet101 [29] as the backbone and replace the original striding by atrous convolution in the last few blocks. Following DeepLabv3+ [28], we use $outputstride$ to denote the ratio of spatial resolution of input image to the final output resolution before fully connected layer or global pooling. For more intensive feature extraction, we also adopt $outputstride$ = 16 here. In the last two blocks of the ResNet101, we adopt atrous convolution with dilated rate $rate$ = 2 and $rate$ = 4. The Atrous Spatial Pyramid Pooling (ASPP block is likewise applied as that in [28]) to probe high-level contextual features at multiple scales. The atrous rates are 1, 6, 12, and 18 in our work. We then attach a 1 × 1 convolution layer after the ASPP block to flatten the feature map. Afterwards, the encoded feature of size (64*64) is delivered to the downstream decoder.

2) Contrastive Learning: Varying sizes of artifacts and diverse manipulating techniques make it difficult to craft decent representations based on priory knowledge. Hence, we employ contrastive learning [25], [30] to endow the feature encoding process with insights into the differences brought in by manipulation operations. Contrastive learning dedicates to parameterizing the manifold on which the margin of positives and negatives are maximized. The objective function of contrastive learning is defined as:

$$score(f(x), f(x^+)) >> score(f(x), f(x^-)) \quad (1)$$

where x donates an anchor, and x^+, x^- donate positive and negative samples respectively. f represents the function that need to learn. $score$ is the similarity between the anchor and different types of samples.

Before approaching f in equation (1) through contrastive learning, we first probe the best representation for the input of f, namely, the feature extraction method of x, x^+, and x^-. Some previous works [31] truncate the deep CNNs at different layers and reveal that the earlier truncated networks provide better features for forgery detection. Besides, the early truncated network has a shallow layout, small receptive fields and a large feature map. As contour patches contain doctored and authentic pixels simultaneously, a large feature map with small receptive fields contributes to diminish the number of authentic pixels in a contour patch. The contour patches are to be classified as manipulated, therefore less authentic pixels in a contour patch signifies less noises and better contrasts. Hence, instead of intuitively adopting the output of the last encoder layer, we argue that the feature map generated by earlier blocks of ResNet101 is more suitable for both IML and contrastive learning. We divide the ResNet101 into convolution blocks as in their paper [29], and explore the feature maps yielded by each ResNet101 block. As expected, the experimental results verify the feature map after the first block to be the most suitable one. For sake of simplicity, this feature map is referred as the "1st feature map" in the later parts. We provide more detailed information about the selection of the feature map in Section 4.6.

As shown in Figure 1, Sobel Operator generates contour labeled ground truth masks, where the contour pixels are additionally identified. We then shrink the contour labeled ground truth mask into the same size as the feature map after the first block. The production between the shrunken mask and the feature map assigns contour, positive and negative labels to every vector in the feature map. However, it is impulsive to directly designate a contour vector as an anchor without knowing its noise level. Illustrated in Figure 2, instead of computing the noise-level in image patches regarding every contour vector, we concatenate the contour vectors (black squares) and impose an extra Fully Connected (FC) layer to squeeze noises brought by post-processing of the datasets, and transform the concatenated feature vector into a fixed size of z. This simple yet effective FC layer projects the raw features of contour points into a compact space where the learned feature vector of size z characterizes all the contour points. We set a large enough input size with padding to handle the variant number of contour vectors in images. The feature vector after the FC layer is set as the unique anchor (blue squares) of an image. The number of triplet-tuples in an image is therefore significantly reduced and the hard examples critical to the contrastive learning outcomes are still preserved. This unique anchor shares similar effectiveness as the bound {anchor, positive} pairs on image translation [25] or face recognition fields [32]. Unique anchor greatly contributes to the convergence of the triplet loss function. More details of anchor selection are discussed in Section 4.6.

We adopt the "fixed-in-order" policy to choose the rest positives (red squares) and negatives (yellow squares) in Figure 2. This policy selects anchors, positives, and negatives in sequence. With the settled anchor, the manipulated and authentic points on the 1st feature map are the positives and negatives respectively. As negatives are the last to fix, one pair of {*anchor, positive*} assesses multiple randomly-chose negatives. After determining the triple-tuples, we adopt the triplet loss [32] for contrastive learning. The detailed triplet

loss function is:

$$L_{tri}(A,P,N) = \frac{1}{mn}\sum_i\sum_j l_{tri}(a,p_i,n_j)$$
$$s.t. \quad (2)$$
$$a = \Phi(a_1,a_2,a_3,...,a_k)$$
$$l_{tri}(a,p_i,n_j) = ||a-p_i||_2^2 - ||a-n_j||_2^2$$

where a_k, p_i and n_j are feature vectors corresponding to the image patches of the contours (A), tampered regions (P) and authentic regions (N). i, j, k indicate the index of each image patches. m, n are the total number of selected positives and negatives for training. Φ is learning parameter of the FC layer used to characterize the contour patches. a is the anchor. L_{tri} is the triplet loss we adopted, which is used to supervise the feature extraction of the encoder.

B. Decoder

1) Basic Decoder of Multiple Upsampling: We follow the basic design of [28] in decoder. As demonstrated in Figure 2, the decoder adopts multiple upsamplings to recover the spatial information of image patches. The encoder output is first bilinearly upsampled by a factor of 4 and then concatenated with the corresponding low-level features from the second block in the ResNet101 backbone that captured by another 1 × 1 convolution. We apply a few 3 × 3 convolutions to additionally refine the features after the concatenation. Then another bilinear upsampling by a factor of 4 is conducted, after which, the feature map for IML is the same size as the input image.

2) Contour Binary Cross-Entropy Loss: Existing methods of IML commonly adopt Cross-Entropy (CE) or Binary Cross-Entropy (BCE) loss to classify the manipulated pixels against authentic ones. Besides the contours are blurred by the contrastive learning in our encoder, we observe that most of the existing methods have similar trouble capturing accurate manipulation boundaries. We blame on the over-plain BCE loss which blends the most concerned contour pixels with the other manipulated pixels. To obtain more accurate manipulation localization, we develop a Contour Binary Cross-Entropy (CBCE) loss to explicitly recover the contour details through assigning extra weights to the contour pixels.

$$L_{CBCE}(F,G) = \frac{1}{p}\sum_i l_{CBCE}(f_i,g_i)$$
$$s.t.$$
$$l_{CBCE}(\Psi(f_i),g_i)\begin{cases}\mu * |\Psi(f_i)-g_i|, & f_i \in C\\ (1-\mu)*l_{BCE}(\Psi(f_i),g_i), & f_i \notin C\end{cases}$$

$$l_{BCE}(x,y) = -[g_i log(\Psi(f_i)) + (1-g_i)log(1-\Psi(f_i))] \quad (3)$$

f_i is a pixel in the prediction (F) yielded by the upsampling. g_i is the manipulation label corresponding to f_i in ground truth (G). We slightly abuse the notation of i to indicate the index of each pixel. p is the number of pixels. $\Psi(f_i)$ is the predicted label of f_i trough the upsampling process. g_i and $\Psi(x)$ are binary valued. C denotes the set of manipulated contour pixels. μ is the weight.

The contour pixels are far less than the number of the manipulated pixels. To avoid overfitting on the contour pixels, we employ auxiliary classifiers [33] to calculate CBCE loss in each upsampling process. We shrink the ground truth to the same size as the feature map after each upsampling. Since ground truth masks are binary value, we binarize our prediction by equal treating anchor pixels and the manipulated ones. The value of μ directly confines the CBCE loss. We find that, to some extent, larger μ benefits the final IML accuracy. The assessment of μ is detailed in Section 4.6.

C. Loss Function

To sum up, CaCL-Net has a hybird total loss denoted as:

$$L_{total} = \omega_{tri} * L_{tri} + \sum_i \omega_i * L_{CBCE}^{upsample(i)}, i = 1,2 \quad (4)$$

We slightly abuse the notation of i here. $L_{CBCE}^{upsample(i)}$ indicates the CBCE loss after the i-th upsampling process in the decoder. The computation of the triplet loss L_{tri} and CBCE loss $L_{CBCE}^{upsample(i)}$ is listed in equation (2) and (3).

IV. Experiments and Discussions

In this section, we first illustrate four standard image manipulation datasets and then demonstrate the experimental evaluation on these benchmarks. The comparisons against other State-of-The-Art (SoTA) methods and ablation study clearly measure the effectiveness and robustness of our proposed method.

A. Datasets

For fair comparisons, we train and evaluate our CaCL-Net on four widely-used datasets, Columbia [34], CASIA [35], Coverage [36], NIST16 [37].

The **Columbia** [34] provides only spliced images. It contains 180 manipulated images and corresponding masks, which is divided into training and test set in a ratio of 3 to 1. Besides, the provided edge masks have been changed into binary ground-truth.

The **CASIA** [35], a varied and large-scale dataset, consists of spliced and copy-moved images with binary ground-truth. CASIA 1.0 has 921 samples, which is used for testing while CASIA 2.0 owns 5123, which is for training. Image enhancement methods, such as Gaussian Blurring, are also adopted to hide manipulated traces.

The **Coverage** [36] is a small size dataset with only 100 copy-moved images and ground-truth. We also divide it into 75%-25% for training and evaluation.

The **NIST16** [37] is a standard real-word manipulated images with high resolution. It consists of 564 spliced, copy-moved and removed images and we select randomly select 404 samples for training and the rest for test.

The **synthesized pre-training datasets** have been widely applied in SoTA methods. For instance, RGB-N [4] uses COCO [38] to construct a 42k forged dataset. ManTra-Net

[5] applies spliced dataset [39], copy-moved dataset [15], the Dresden [40]-based synthesized removal dataset and enhancement dataset. SPAN [6] also uses these four manipulated datasets for pre-training and four benchmarks for fine-tuning. Different with above SoTA methods, our CaCL-Net is trained only on four standard datasets without any synthetic dataset for pre-training.

B. Implementation Details

To train our proposed model, we follow the training protocols in [28] and fine-tune partial parameters. In detail, we set the batch size to 4 on each dataset. The crop size is 512 × 512. We adopt Stochastic Gradient Descent (SGD) optimizer under the "poly" policy with the initial learning rate of 0.007, weight decay of 5×10^{-4}, and momentum of 0.9. Our proposed model is trained end-to-end without staged pre-training of each component. The parameters, therefore, are applicable in the whole network. Moreover, the weight of triplet loss (ω_{tri} in equation (4)) is 0.01, and each CBCE (ω_i in equation (4)) is 1.0. In CBCE, the weight of contour (μ in equation (3)) is 0.9. We provided more detailed information about the selection of weights in Section 4.6.

C. Evaluation Metrics

Established Metrics. In state-of-the-art studies, AUC (Area Under the Curve) is the foremost concern when assessing an IML model. The curve in the AUC metric commonly refers to the Receiver Operator Characteristic (ROC) Curve, which shows the model performance variation regarding the correctly classified positive examples and the incorrectly classified negative examples. We simply denote the calculation of AUC with respect to the True Negatives (TN), True Positives (TP), False Positives (FP), and False Negatives (FN) here:

$$AUC \propto \frac{TP}{TP+FN} * (1 - \frac{FP}{FP+TN}) \tag{5}$$

The manipulated regions in an image are usually tiny. Thereby, the number of tampered pixels and the number of authentic pixels in a manipulated image is extremely unbalanced. Due to such a large skew in the class distribution, although AUC works well on many unbalanced classification problems, it still trends to an over-optimistic estimation of IML models' performance. Besides, AUC averages over all possible thresholds, but we can only apply a fixed threshold for IML in real-life scenarios. In other words, a high AUC value justifies that if we can find the optimal threshold, the model will perform well on IML tasks; but it doesn't assure whether the optimal threshold has been correctly found by the model. Consequently, as shown in the second row of Figure 5, state-of-the-art models with a high AUC value may own a disaster behavior on manipulation localization. Some most recent studies [19], [41] have also noticed this optimal threshold finding problem like us, but they did not offer an insight solution. In this paper, we speculate such a disaster behavior shall blame on the over-fitting caused by large-scale pre-training, and the AUC metric is not sufficient to dispose of such over-fitting. Therefore, we further provide a new set of metrics on IML models' performances.

Our Metrics. Considering the unbalanced nature of IML and the potential over-fitting issue, we here advocate grounding on the **consistency between F_1 score and AUC value to evaluate an IML model**. Although F_1 score is measured by most of the existing works, the consistency between F_1 score and the AUC value is always neglected. In specific, Precision-Recall (PR) curves, or the equivalent F_1 score, have been cited as an alternative for tasks with a large skew in the class distribution. F_1 score can expose differences between algorithms that are NOT distinguishable under the AUC value. Besides, according to (5) and (6), the differences between these two metrics can reflect the over-fitting potential of the model. Therefore, a well-learned IML model shall have similar and high-enough values on both F_1 score and AUC metrics. As shown in Table I, all current studies own a huge gap between F_1 score and AUC values, indicating their performances are not accurate as expected in manipulation localization and the potential to be over-fitted on unseen images.

$$F_1 \propto \frac{TP}{TP+FN} * \frac{TP}{FP+TP} \tag{6}$$

We apply pixel-level F_1 score and Area Under the receiver operating characteristic Curve (AUC) as evaluation metrics. We generate all TP, TN, FP, and FN through the pixel-level confusion matrix and compute the F_1 score and AUC for each epoch over the whole datasets. The best model is picked according to the highest AUC score. The F_1 score and AUC are evaluated at the validation of each epoch.

D. Comparisons with SoTA methods.

To validate the effectiveness of our proposed CaCL-Net, we conduct a series of comparisons with baselines on benchmarks and list the F_1 score and AUC results in Table I. As shown clearly in this table, our method achieves state-of-the-art performance compared with existing methods in both F_1 score and AUC. Note that except our CaCL-Net, all the other methods use a considerable synthetic dataset for training and four benchmarks for fine-tuning.

Fig. 3. CaCL-Net detection results on copy-moved images on CASIA. (a) represents manipulated images with flipping manipulation and similar patterns within an image which are difficult to localization. (b) denotes the other types of copy-moved images.

TABLE I
F_1 SCORE (%) AND AUC (%) COMPARISONS BETWEEN OUR PROPOSED METHOD AND BASELINES ON BENCHMARKS.

Method	pre-train	fine-tune	NIST16		CASIA		Coverage		Columbia	
			F_1 score ↑	AUC ↑	F_1 score↑	AUC↑	F_1 score↑	AUC↑	F_1 score↑	AUC↑
ELA [12]	×	×	23.6	42.9	21.4	61.3	22.2	58.3	47.0	58.1
NOI1 [2]	×	×	28.5	48.7	26.3	61.2	26.9	58.7	57.4	54.6
CFA1 [8]	×	×	17.4	50.1	20.7	52.2	19.0	48.5	46.7	72.0
J-LSTM [10]	✓	✓	-	76.4	-	-	-	61.4	-	-
ManTra [5]	✓	✓	-	79.5	-	81.7	-	81.9	-	82.4
RGB-N [4]	✓	✓	72.2	93.7	40.8	79.5	43.7	81.7	69.7	85.8
SPAN (1) [6]	✓	×	29.0	83.6	33.6	81.4	53.5	91.2	81.5	93.6
SPAN (2) [6]	✓	✓	58.2	96.1	38.2	**83.8**	55.8	93.7	-	-
MVSS-Net [19]	✓	✓	73.7	-	75.3	44.5	82.4	27.1	70.3	72.6
MVSS-Net++ [20]	✓	✓	71.5	-	77.1	44.3	83.2	28.8	73.1	79.2
ObjectFormer [21]	✓	✓	82.4	**99.6**	88.2	57.9	95.7	75.8	-	-
MSED [7]	×	✓	96.0	96.2	74.7	67.8	95.1	96.1	94.6	94.5
CaCL-Net	×	×	**96.1**	96.2	**76.8**	73.8	**96.1**	**97.8**	**95.0**	**95.3**

SPAN (1) is under the pre-training setup while SPAN (2) is under the fine-tuning.
MVSS-Net++ and MVSS-Net share the same network structure but adopt different training parameters.
'-' denotes that the result is not available in the literature.
'↑' indicates that the higher value is better.

As illustrated in Table I, CaCL-Net outperforms other methods in F_1 score, especially on NIST16. For AUC, without any pre-training and fine-tuning process, CaCL-Net achieves outstanding performance on the datasets except CASIA. By investigating the compositions of different datasets and the predicted masks, we notice CASIA contains many copy-moved images that is manipulated in a data-augmentation manner, see the left Figure 3. A certain part of the image is flipped, shifted or rotated to construct the manipulation. However, **(i)** such augmentation-styled image manipulations are in fact the same operations adopted by the ResNet101 to boost the robustness. Therefore, the common backbone networks are not sensitive to such augmentation-styled manipulation. **(ii)** The copy-moved and pasted regions belong to the same image, and share very similar patterns. Thereby weaken the contrastive learning efficacy. Moreover, such kind of manipulation is rather rare in real life, and not contained in other three benchmarks. Apart from this augmentation-styled copy-move, as seen from the right side of Figure 3, our CaCL-Net is still very efficacious for the other common copy-moved images.

E. Qualitative Analysis

Lateral Qualitative Comparisons with SoTA Methods. We conduct longitudinal qualitative comparisons of different components among the CaCL-Net in Figure 4. The first two rows are forged images and corresponding ground-truth. The next three rows are results of the base model (atrous encoder-decoder) with Binary Cross-Entropy (BCE) loss, the base model with contrastive learning loss, and the base model combining contrastive learning with Contour Binary Cross-Entropy (CBCE) loss. The leftmost two columns of Figure 4 vividly demonstrate us the efficacy of the contrastive learning. The base model totally fails these cases, but contrastive learning keenly catches the clue of manipulations. The left column of the Coverage dataset and cases in NIST16 dataset present active examples of refining the delicate contour of roughly localized artifacts through CBCE loss. Particularly for the red traffic cone(NIST16) and yellow bell pepper (Coverage) cases, in the last row, the CBCE loss yields much accurate contour comparing with the BCE loss whose results are in the last but one row.

Longitudinal Qualitative Comparisons with variants of CaCL-Net. Figure 5 shows some qualitative comparisons between our proposed CaCL-Net and ManTra-Net on four benchmarks. We pick forged images with complex manipulated boundaries. The first and third rows are the test images and their ground-truth. The second row is the output of ManTra-Net [5], and the last row is the results of CaCL-Net. Figure 5 shows that our proposed method produces better predictions. Particularly on the leftmost four columns, ManTra-Net fails in these tests with trivial artifacts localization, but CaCL-Net identifies accuracy contours and clear locations of the manipulations.

F_1 score and Corresponding AUC. The F_1 score and AUC are common metrics applied to evaluate the real model performance when the size of classification results is extremely unbalanced. The manipulated pixels are often the minority with respect to the background pixels. Therefore, both metrics are suitable in assessing the model performance.

As defined in equation (5) and (6), the difference between F_1 and AUC lies in the true negatives (authentic pixels) and true positives (manipulated pixels). AUC considers how many authentic pixels claimed are real authentic ones, while F_1 investigates how many manipulated pixels claimed are real manipulated ones. Therefore, they are the two sides of the same coin. For an individual method, its F_1 and AUC values shall be consistent on one dataset. Meaning its F_1 and AUC values shall close and proportional to each other.

Hence, a good IML model must have appropriate F_1 and AUC values at the same time, even if AUC is the value of an integral while F_1 score is a pixel-wise measurement. As shown in Table I, there is a huge gap between F_1 score and AUC in most SoTA methods, which indicates the classification result is unsatisfying. Various auxiliary means are employed to improve the performance, for instance, extra pre-training data. The underlying reason is the weakness of hand-crafted

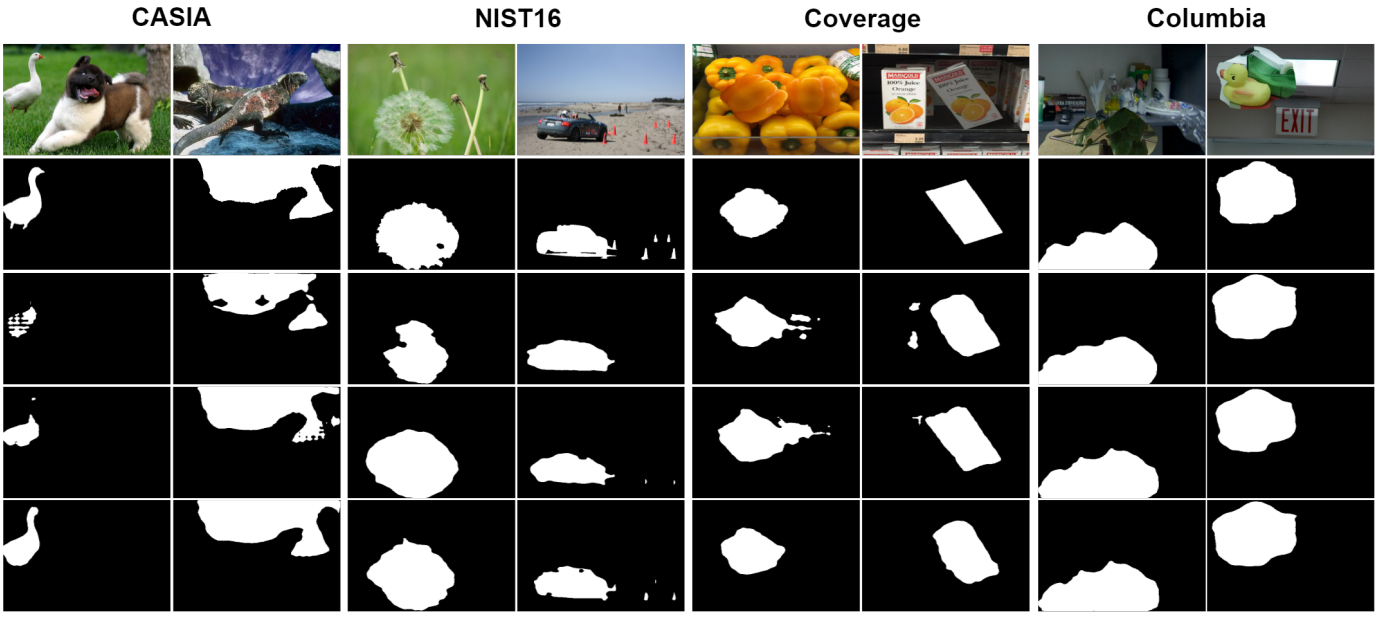


Fig. 4. Prediction comparisons of CaCL-Net variants on Columbia, Coverage, NIST16, CASIA datasets. From top to bottom: forged images, ground-truth masks, baseline model with Binary Cross-Entropy (CBCE) loss at the end of decoder, attaching contrastive learning in the encoder, and combining contrastive learning in the encoder and CBCE loss in the decoder.

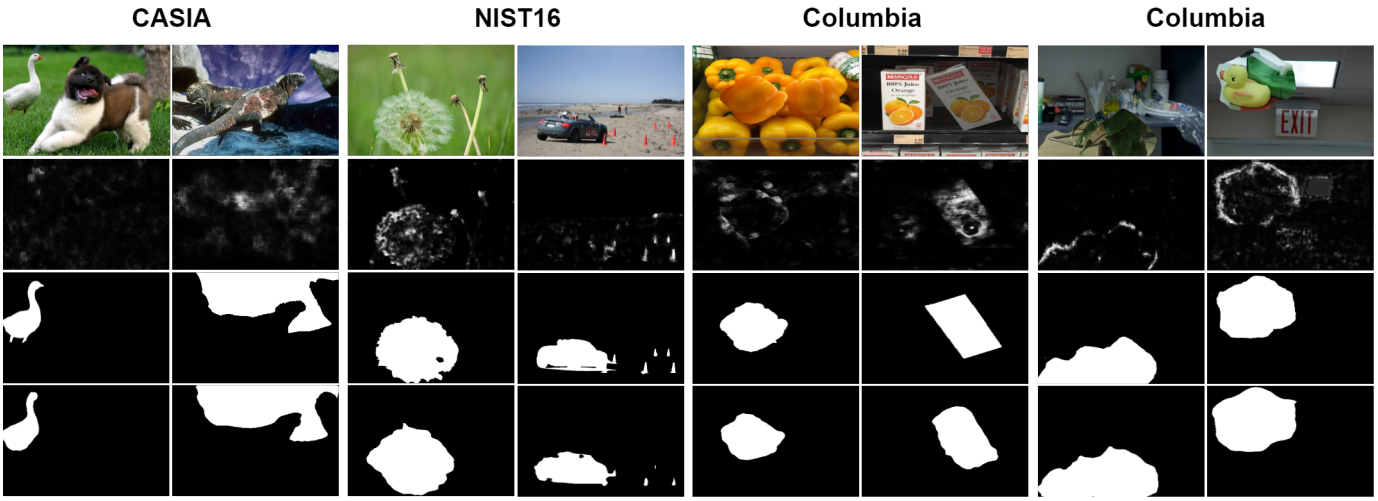


Fig. 5. Comparison of CaCL-Net prediction results on Columbia, Coverage, NIST16, CASIA datasets, with ManTra-Net predictions. From top to bottom: Manipulated Image, ManTra-Net prediction, Ground-truth mask, CaCL-Net prediction. The ManTra-Net predictions are obtained through the pre-trained model provided on GitHub.

features, which make models confuse to distinguish between manipulated and authentic pixels. In addition, large synthesized dataset for pre-training and benchmarks for fine-tuning may cause overfitting. To address this problem, CaCL-Net leverages unlabeled information in the benchmark dataset itself to carry out vast comparisons. From the experimental results, CaCL shares consistent and similar performance between the measurements, which indicates our model has higher robustness and less overfitting.

Since the best F_1 score and best AUC are commonly achieved by different sets of parameters, we further conduct a consistency investigation on our model. In Table II, we check the F_1 score when CaCL-Net having its best AUC value. We compute F_1 score using the same set of parameter which attains best AUC value for CaCL-Net. The result is in the middle row of Table II. Clearly, comparing with the last row, the two F_1 scores are quite close, and our CaCL-Net gains a consistent behavior across measurements.

F. Robustness Validation

Cross Experiments for Overfitting. As we stated in the Introduction, our mainly contribution regarding the performance of current IML tasks is double folded: training with benchmark dataset and better robustness against overfitting.

TABLE II
CORRESPONDING F_1 SCORE UNDER THE SET OF PARAMETERS OF BEST AUC MODEL. BEST F_1 IS THE MODEL PERFORMANCE ADOPTING THE SET OF PARAMETER BENEFITING ITS F_1 SCORE MOST.

Metric	NIST16	CASIA	Coverage	Columbia
best AUC	96.2	73.8	97.8	95.3
F_1 under best AUC	94.8	75.0	95.8	94.9
best F_1	96.1	76.8	96.1	95.0

These two merits are self-contradict at the first glance. Because except CASIA, the other datasets are sufficiently small, thereby, intuitively speaking, benchmark training and testing under the same dataset will surely raising the concern of overfitting. However, our self-supervised contrastive learning paradigm automatically crafts massive contrastive pairs as training samples, therefore the training sample size are boosted at least one or two orders of magnitude. As a result, CaCL-Net can overcome the overfitting issue caused by training under the tiny benchmark training split. Besides, we have limited the training epoch to 70, early stopping is proved to be effective in mitigating the overfitting as well [42]. In this section, to validate the robustness of our CaCL-Net, we here conduct a series of cross experiments on benchmarks. Models that are trained one dataset is tested on other datasets to verify the contrasts-based CaCL-Net dedicates to characterizing the common clues of image manipulation rather than focusing on some non-sense to benefit the metrics. We pick the models with the highest AUC value on different datasets and test the performance on the other datasets except the training set resource. The results are listed in Table III.

TABLE III
F_1 SCORE (%) RESULTS FOR ROBUSTNESS VALIDATION.

MethodDataSet	NIST16↑	CASIA	Coverage	Columbia
ELA [12]	23.6	21.4	22.2	47.0
NOI1 [2]	28.5	26.3	26.9	57.4
CFA1 [8]	17.4	20.7	19.0	46.7
RGB-N [4]	**72.2**	40.8	43.7	69.7
SPAN (1) [6]	29.0	33.6	53.5	**81.5**
SPAN (2) [6]	58.2	38.2	55.8	-
	Our CaCL-Net			
TrainTest	NIST16	CASIA↑	Coverage↑	Columbia
NIST16	\	54.5	53.4	53.6
CASIA	69.6	\	**59.6**	66.9
Coverage	58.1	54.6	\	52.2
Columbia	68.9	**68.9**	58.7	\

SPAN (1) is under the pre-training setup while SPAN (2) is under the fine-tuning.
'-' denotes that the result is not available in the literature.
'\' denotes the model is trained and tested in the same dataset leading to trivial results for robustness validation.
'↑' indicates the higher value is better.

From Table III, we can see our CaCL-Net achieves excellent performance even only training on a single small dataset, especially for the model training on CASIA and Columbia, which fully indicates the robustness of our CaCL-Net. Our CaCL-Net fully outperforms traditional IML methods, including ELA, NOI1 and CFA1, which depend on specific low-level manipulated traces. Although RGB-N and SPAN adopt massive self-collected datasets for pre-training, our CaCL-Net almost achieves state-of-the-art performance in the f_1 score, except NIST16 and Columbia.

We investigate the composition of different datasets and analyze the underlying reason for general performance on NIST16 and Columbia. We find that the forged artifacts of NIST16 mostly are some intact and independent objects, such as the person and the flower. Hence, the network learns more about the information of the object and image content rather than manipulated information. In addition, these two datasets are also much smaller than CASIA in size, hence, limit the performance of the model.

Experiments on Harsh and Unseen Images. Apart from the datasets mentioned above, we additionally adopt IMD2020 [43] to further validate the effectiveness and robustness of our proposed CaCL-Net. IMD2020 contains 2010 real-life manipulated images downloaded from the Internet and binary masks localizing the exact forged regions. We compare our CaCL-Net with baselines declared in [43] and list the results in Table IV.

TABLE IV
AUC (%) COMPARISONS ON IMD2020.

Method	AUC
NOI1 [2]	58.6
CFA1 [8]	48.7
BLK [44]	59.6
ADQ1 [45]	57.9
ManTra-Net [5]	74,8
CaCL-Net	71.0

As shown in Table IV, our CaCL-Net shows much better performance on IMD2020 compared to NOI1 [2], CFA1 [8], BLK [44], and ADQ1 [45]. Considering the massive synthetic pre-training dataset adopted by Mantra-Net, it is not surprising that Mantra-Net performs better. However, there is not a huge gap between us. Therefore, our CaCL-Net is quite effective and robust in image manipulation localization.

G. Ablation Study

Before going further, we first clarify the exact improvement of each contribution of CaCL-Net. As shown in Tables V and VI, it is clear that our baseline model is also good at image manipulation localization comparing with some SoTA methods from Table I on most datasets. Moreover, the proposed Contrastive Learning module (CL) and Contour Binary Cross-Entropy (CBCE) loss can further boost the performance. Below, we try to answer how CaCL-Net works by decomposing it into three sub-questions: *where* to conduct contrastive learning, *what* to compare in the contrastive learning, and *how* to assemble each component in the hybrid loss. These three sub-questions are studied in an asymptotic order to explore the specific promotion brought by each component of CaCL-Net.

Where to Conduct Contrastive Learning. The feature map of different layers produces different contrastive pairs, which is vital for the performance of contrastive learning. [31] observes that the shallow layers have a smaller receptive field and perform better for face forgery detection. Thus, to find the best location to conduct the proposed contrastive learning module, we select the encoded feature in different stages of the network and compare the results. In detail, we divide the

TABLE V
F_1 SCORE (%) COMPARISONS OF CACL-NET VARIANTS EVALUATED ON BENCHMARKS. FROM TOP TO BOTTOM ARE RESULTS OF BASELINE MODEL WITH BINARY CROSS-ENTROPY (BCE) LOSS AT THE END OF DECODER, ATTACHING CONTRASTIVE LEARNING IN THE ENCODER, AND COMBINING CONTRASTIVE LEARNING IN THE ENCODER AND CONTOUR BINARY CROSS-ENTROPY (CBCE) LOSS IN THE DECODER.

Methods	NIST16	CASIA	Coverage	Columbia
BCE	93.2	65.9	89.0	89.0
CL+BCE	94.3	72.8	92.3	92.0
CL+CBCE	**96.1**	**76.8**	**96.1**	**95.0**

TABLE VI
AUC (%) COMPARISONS OF CACL-NET VARIANTS EVALUATED ON BENCHMARKS. FROM TOP TO BOTTOM ARE RESULTS OF BASELINE MODEL WITH BINARY CROSS-ENTROPY (BCE) LOSS AT THE END OF DECODER, ATTACHING CONTRASTIVE LEARNING IN THE ENCODER, AND COMBINING CONTRASTIVE LEARNING IN THE ENCODER AND CONTOUR BINARY CROSS-ENTROPY (CBCE) LOSS IN THE DECODER.

Methods	NIST16	CASIA	Coverage	Columbia
BCE	90.9	63.5	93.7	91.4
CL+BCE	93.4	64.7	95.9	91.9
CL+CBCE	**96.2**	**73.8**	**97.8**	**95.3**

original ResNet101 to five stages as the original paper [29] and perform the contrastive framework at the end of each stage. As shown in Table VII, the earlier layer outperforms the deeper layer a lot, which also confirms the observation from [31]. One possible explanation is that the features in the earlier layers have the similar properties with the features from the traditional low-level feature filters and include numerous vital information.

TABLE VII
F_1 SCORE (%) AND AUC (%) COMPARISONS AFTER EACH BLOCK TO CONDUCT CONTRASTIVE LEARNING ON CASIA. FROM THE TOP TO BOTTOM ARE RESULTS TO CONDUCT CONTRASTIVE LEARNING AFTER ENCODER, AFTER THE SECOND BLOCK AND AFTER THE FIRST BLOCK OF THE RESNET101.

Methods	F_1 score	AUC
After Encoder	62.8	69.7
After ResNet (b2)	73.5	71.0
After ResNet (b1)	**76.8**	**73.8**

What to Compare in Contrastive Learning. The selection of anchors, positives and negatives is quite significant for model performance in contrastive learning. Hence, we explore several different selection strategies and their combinations. Intuitively, we can randomly select two patches in the forged region as anchor and positive, respectively. The Figure 6 is the triplet loss function trained under such an anchor scheme. Furthermore, randomly selecting a specific number of pixels in the corresponding regions and then combine them randomly or in a certain order makes a little effect. This is probably because the randomness of the selection gives rise to the unbalance of input. Differently, we compress the hard examples (contour pixels) as the anchor and randomly choose positive pixels and negative pixels in the forged and authentic region in sequence, which is stated in Section 3.1. This method works well in our task.

How to Assemble Each Component in the Hybrid Loss. Besides contrastive learning, we develop the CBCE loss in the decoder to focus on the contour pixels for the sharper boundaries of the manipulated artifacts. The total loss, therefore, comes from several parts. Different ratios of each loss produce diverse results. We explore different allocations of weight to maximize the F_1 score and AUC. Under this circumstance, we find the allocation scheme and adopt these parameters in the whole training.

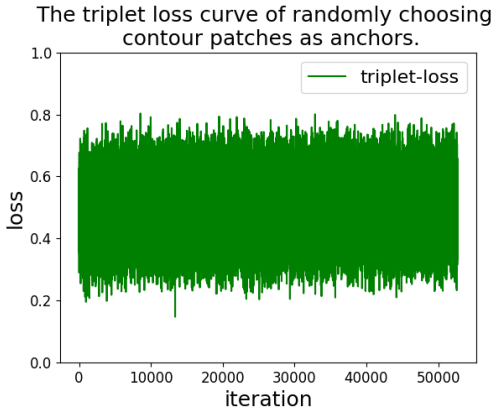


Fig. 6. The triplet loss curve of randomly choosing contour patches as anchors.

We first determine the inner-relation within Contour Binary Cross-Entropy (CBCE) loss. Here, we explore different ratios (μ and $1-\mu$ in equation (3) between contour pixels and the rest manipulated pixels based on the CASIA dataset. As shown in Table VIII, μ is the weight of the contour pixels. As μ increase, the performance is better until μ reach 0.90. We hence select 0.9 for further training and evaluation. For the sake of simplicity, to probe the proper value of ω_{tri}, we fix ω_i to 1.0 (ω_{tri} and ω_i are weights defined by equation (4). Similarly, we implement our exploration on the CASIA dataset. For the ratio between encoder and decoder ((ω_{tri} and ω_i in equation (4)), that means between contrastive loss and CBCE loss, the best weight of contrastive loss is 0.01 and CBCE is 1.0, as illustrated in Table IX.

TABLE VIII
F_1 SCORE (%) AND AUC (%) COMPARISONS UNDER VARIOUS μ ON CASIA. μ REPRESENTS THE WEIGHT OF THE CONTOUR PIXELS. NOTE THAT THE TEST MODEL IS A BASIC ENCODER-DECODER STRUCTURE WITH A CONTOUR BINARY CROSS-ENTROPY (CBCE) LOSS JUST AFTER THE SECOND UPSAMPLING PROCESS.

μ	0.50	0.55	0.60	0.70	0.80	0.90	0.95
F_1	70.5	71.2	71.6	71.8	72.7	**73.1**	72.5
AUC	68.9	70.9	70.4	70.3	71.7	**72.1**	70.3

TABLE IX
F_1 SCORE (%) AND AUC (%) COMPARISONS UNDER VARIOUS ω_{tri} ON CASIA. ω_{tri} REPRESENTS THE WEIGHT OF TRIPLET LOSS. NOTE THAT THE TEST MODEL COMBINES CONTRASTIVE LEARNING IN THE ENCODER AND MULTIPLE CONTOUR BINARY CROSS-ENTROPY (CBCE) LOSS IN THE DECODER.

ω_{tri}	0.500	0.300	0.100	0.050	0.010	0.005	0.001
F_1	71.3	71.0	70.1	73.1	**76.8**	71.7	72.2
AUC	69.7	71.6	70.6	72.3	**73.8**	70.6	70.6

V. Conclusion

In this paper, we propose a novel Contour-aware Contrastive Learning Network (CaCL-Net) to localize image manipulation. In detail, CaCL-Net uses the contour as anchor and adopts the contrastive learning framework in the early layer of the backbone network to maximize the difference between tampered regions and authentic background as well as minimize the difference among the tampered pixels. Besides, we also apply the proposed Contour Binary Cross-Entropy (CBCE) loss in the decoder to further focus on the contour pixels of the manipulated regions. The extensive experimental results demonstrate that the proposed CaCL-Net achieves state-of-the-art performance on standard datasets even without extra pre-training. Furthermore, the proposed method can accurately localize the tampered boundaries and tiny-sized manipulations.

Acknowledgment

This work was funded by the National Key Research and Development Program under Grant No.2020AAA0107800.

References

[1] J. Zhou, C.-M. Pun, and Y. Tong, "News image steganography: A novel architecture facilitates the fake news identification," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2020, pp. 235–238.

[2] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, 2009.

[3] J. Park, D. Cho, W. Ahn, and H. K. Lee, *Double JPEG Detection in Mixed JPEG Quality Factors Using Deep Convolutional Neural Network: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part V*, 2018.

[4] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1053–1061.

[5] Y. Wu, W. AbdAlmageed, and P. Natarajan, "Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9535–9544.

[6] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "Span: Spatial pyramid attention network forimage manipulation localization," 2020.

[7] C. Yu, J. Zhou, and Q. Li, "Multi-supervised encoder-decoder for image forgery localization," *Electronics*, vol. 10, no. 18, p. 2255, 2021.

[8] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of cfa artifacts," *IEEE Transactions on Information Forensics & Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[9] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.

[10] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath, "Exploiting spatial structure for localizing manipulated image regions," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4980–4989.

[11] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson, "Detection and localization of image forgeries using resampling features and deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1881–1889.

[12] N. Krawetz, "A picture's worth..." *Hacker Factor Solutions*, vol. 6, no. 2, 2007.

[13] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli, "Localization of jpeg double compression through multi-domain convolutional neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1865–1871.

[14] A. Islam, C. Long, A. Basharat, and A. Hoogs, "Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4675–4684.

[15] Y. Wu, W. Abdalmageed, and P. Natarajan, "Busternet: Detecting copy-move image forgery with source/target localization," in *European Conference on Computer Vision (ECCV)*, 2018.

[16] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," 2018.

[17] R. Salloum, Y. Ren, and C. C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (mfcn)," *Journal of Visual Communication & Image Representation*, vol. 51, no. feb., pp. 201–209, 2017.

[18] X. Zhu, Y. Qian, X. Zhao, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Processing: Image Communication*, vol. 67, pp. 90–99, 2018.

[19] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 185–14 193.

[20] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, "Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[21] J. Wang, Z. Wu, J. Chen, X. Han, A. Shrivastava, S.-N. Lim, and Y.-G. Jiang, "Objectformer for image manipulation detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2364–2373.

[22] H. Dibeklioglu, "Visual transformation aided contrastive learning for video-based kinship verification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[23] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] Y. Deng, J. Yang, D. Chen, F. Wen, and X. Tong, "Disentangled and controllable face image generation via 3d imitative-contrastive learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[25] T. Park, A. Efros, R. Zhang, and J.-Y. Zhu, *Contrastive Learning for Unpaired Image-to-Image Translation*, 11 2020, pp. 319–345.

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR. arXiv*, 12 2014.

[27] G. Papandreou and I. Kokkinos, "Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection," *Eprint Arxiv*, 2014.

[28] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] A. Anand, "Contrastive self-supervised learning," 2020, https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html.

[31] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," 2020.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[33] X. Cun and C.-M. Pun, "Defocus blur detection via depth distillation," 2020.

[34] T. Ng, T, J. Hsu, and S. Chang, "Columbia image splicing detection evaluation dataset," in *DVMM lab. Columbia Univ CalPhotos Digit Libr*, 2009.

[35] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 422–426.

[36] B. Wen, Y. Zhu, R. Subramanian, T. T. Ng, and S. Winkler, "Coverage — a novel database for copy-move forgery detection," in *IEEE International Conference on Image Processing*, 2016.

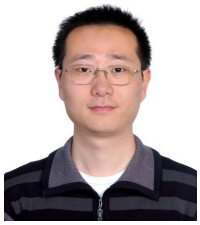
[37] "Nist: Nist nimble 2016 datasets," https://www.nist.gov/itl/iad/mig/, 2016.

[38] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick, "Microsoft coco: Common objects in context," 2014.

[39] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection," 10 2017, pp. 1480–1502.

[40] T. Gloe and R. Bohme, "The dresden image database for benchmarking digital image forensics," *Journal of digital forensic practice*, vol. 3, no. 2/4, pp. 150–159, 1 2010.

[41] J. Zhou and C.-M. Pun, "Personal privacy protection via irrelevant faces tracking and pixelation in video live streaming," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1088–1103, 2020.
[42] L. Prechelt, *Early Stopping — But When?* Springer Berlin Heidelberg, 2012.
[43] A. Novozamsky, B. Mahdian, and S. Saic, "Imd2020: A large-scale annotated dataset tailored for detecting manipulated images," in *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, March 2020, pp. 71–80.
[44] W. Li, Y. Yuan, and N. Yu, "Passive detection of doctored jpeg image via block artifact grid extraction," *Signal Processing*, vol. 89, no. 9, pp. 1821–1829, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168409001315
[45] Z. L. A, J. H. B, X. T. A, and C. K. T. B, "Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.



Qin Li received the Ph.D. degree from East China Normal University in 2011. He is currently an Associate Professor with the Software Engineering Institute, East China Normal University, Shanghai, China. His research interests include Trustworthy Artificial Intelligence and Formal Modeling and Verification on Safety and Security of Cyber-physical Systems.

Chunfang Yu Chunfang Yu received the M.S. degree from East China Normal University in 2022. Her research direction includes Computer Vision and Software Engineering.

Zhuohang Jiang is an undergraduate student at Sichuan University, majoring in Computer Science and Technology. His research direction includes Computer Vision,Knowledge graph and security and privacy.

Jizhe Zhou received his Ph.D. degree in Computer and Information Science from the University of Macau in 2021. He is currently the associated research professor in the College of Computer Science, University of Sichuan. His current research interests include video semantics, security and privacy, multimedia analysis.