SAFL-Net: Semantic-Agnostic Feature Learning Network with Auxiliary Plugins for Image Manipulation Detection

Zhihao Sun[1,2], Haoran Jiang[3], Danding Wang[1,2]*, Xirong Li[4], Juan Cao[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]School of Mathematics Science, University of Chinese Academy of Sciences
[4]MoE Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
{sunzhihao21s,wangdanding,caojuan}@ict.ac.cn,
jianghaoran21@mails.ucas.ac.cn, xirong@ruc.edu.cn

Abstract

Since image editing methods in real world scenarios cannot be exhausted, generalization is a core challenge for image manipulation detection, which could be severely weakened by semantically related features. In this paper we propose SAFL-Net, which constrains a feature extractor to learn semantic-agnostic features by designing specific modules with corresponding auxiliary tasks. Applying constraints directly to the features extracted by the encoder helps it learn semantic-agnostic manipulation trace features, which prevents the biases related to semantic information within the limited training data and improves generalization capabilities. The consistency of auxiliary boundary prediction task and original region prediction task is guaranteed by a feature transformation structure. Experiments on various public datasets and comparisons in multiple dimensions demonstrate that SAFL-Net is effective for image manipulation detection.

1. Introduction

The proliferation of novel image editing techniques has greatly enriched our visual world. However, these techniques have also brought significant challenges to the authenticity and security of graphic media content. To address these issues, image manipulation detection methods have been proposed to identify the specific regions that have been modified. These techniques are crucial for enabling us to distinguish between virtual and authentic components of rich multimedia content.

The majority of tampering operations take place in regions that exhibit strong correlations with semantic properties. However, the limited and biased nature of the available training data means that this correlation is insufficient to accurately represent the distribution of real-world scenes. For example, a dataset where tampered areas are concentrated in human regions can cause the detection model to display significant semantic association, leading to incorrect predictions, as illustrated by the regions marked with a green box in Figure 1. Consequently, this semantic correlation can impact the learning of tampering trace features, even though it may improve the model's fit to the training data. When detecting tampering in an unseen scene, it is essential to identify evidence of tampering rather than rely on the probabilistic semantic distribution inherent in this constrained training data. Therefore, the most critical issue for generalization is to learn semantic-agnostic features.

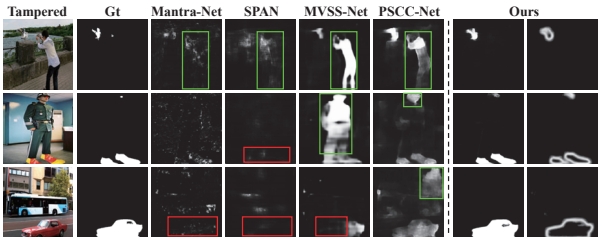

Figure 1. Some cases of the phenomenon exhibit methods encountering noticeable semantic-related false alarms, as indicated by the green boxes. Additionally, there are instances of significant missed detection, as denoted by the red boxes.

To enhance the generalization ability of semantic-agnostic tampering trace features, existing methods have focused on restricting the semantic information of the input by utilizing hand-crafted feature extraction modules [14, 26, 25, 10, 13], or by removing the guidance of semantic masks through the conversion of the task and changing the structure of the segmentation network [19, 28, 2, 22].

*Corresponding author.

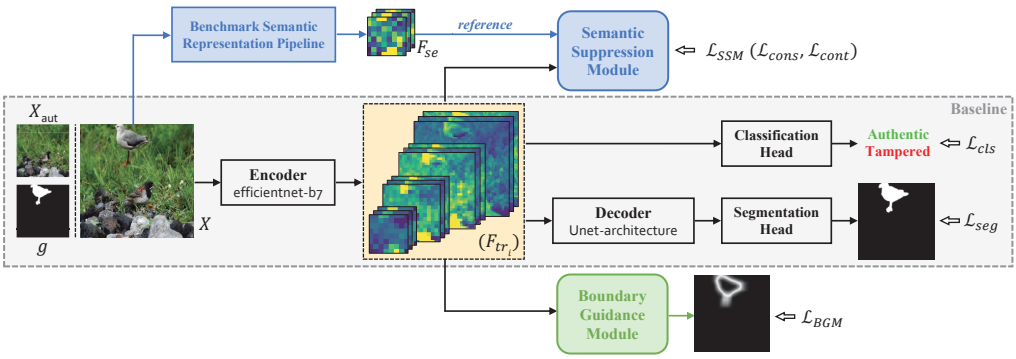

Figure 2. An overview of SAFL-Net. Based on baseline, the plug-and-play SSM and BGM achieve semantic-agnostic feature learning.

The effective extraction of specific features enables the conversion of the RGB space, which contains rich semantic content, into an underlying noise space or high-frequency space that is unrelated to semantics. However, the absolute noise and high-frequency information are easily erased or changed by post-processing operations, such as compression and blurring, and much available knowledge is also removed. Redesigning the network structure and converting the task can break the semantic segmentation supervision. However, when the auxiliary task conflicts with the original task, balancing the two tasks becomes a difficult problem.

In contrast, we propose a method that constrains the feature extractor to learn semantic-agnostic features through modular auxiliary tasks, based on the common feature extraction network without any modification as the backbone. We directly restrict semantic information in the features with the assistance of benchmark semantic representation. Additionally, we leverage boundary supervision to identify inconsistencies in the features around the tampered boundary, and design a feature conversion structure to ensure the coherence of the auxiliary task and the primary task. We have conducted experiments on several widely used image tampering datasets, namely CASIA [5], Columbia [20], Coverage [24], NIST16 [7], and IMD20 [18]. Our results demonstrate that SAFL-Net outperforms existing state-of-the-art methods for tampering detection and localization. We contributes to the following key aspects:

- We propose SAFL-Net, a network with two auxiliary plugins for image manipulation detection and localization shown in Figure 2.
- We introduce plug-and-play Semantic Suppression Module and Boundary Guidance Module to directly constrain the feature extractor to learn semantic-agnostic feature.
- We conduct extensive experiments on various benchmarks which demonstrate that SAFL-Net achieves state-of-the-art tampering detection performance.

2. Related Work

In real-world scenarios, acquiring features with reliable trace evidence is a crucial yet challenging task, and such efforts may be significantly impeded by the issue of overfitting to the correlation between manipulation and specific semantic meaning in the training data. In the following section, we provide a brief summary of recent deep learning methods. Subsequently, we introduce our novel contributions in light of these advancements.

Various methods have proposed hand-crafted feature extraction modules to convert inputs from the RGB space, which contain abundant semantic content, into the underlying noise space or high-frequency space that is unrelated to semantics. This allows for the isolation of semantic information from the root. For example, Li and Huang [14] propose to supplement the network's forefront with trainable high-pass filters. Yang *et al*. [26] use BayarConv as the first convolution layer of their model. Recent methods have proposed more complex two-stream networks to fuse features from multiple views, fully utilizing the knowledge in the original RGB space. However, this approach risks reintroducing semantic information. Wu *et al*. [25] and Hu *et al*. [10] use both BayarConv and SRM as noise extractors. Chen *et al*. [2] use the RGB image and its noise counterpart generated by BayarConv as input. Wang *et al*. [22] extract high-frequency features from images and combine them with RGB features as multi-modal patch embeddings. To fully exploit all available knowledge in the original image while suppressing unreliable semantic distribution in limited training data, we use raw RGB image as the only input and introduce two plug-and-play auxiliary modules. These modules directly restrict semantic-related features and assist the encoder in mining higher-quality trace features for better performance.

Artifacts resulting from image editing operations and inconsistencies in features across adjacent local regions often manifest at the boundary of tampered regions. Leveraging such artifacts can help models extract subtle trace fea-

tures, resulting in better performance. For instance, Salloum *et al*. [19] propose a multi-task fully convolution network that predicts both the tampered region and its boundary. Zhou *et al*. [28] provide a branch for edge identification and refine features at multiple levels. Chen *et al*. [2] design an edge-supervised branch that learns edge information enhanced by the Sobel layer and edge residual block. Wang *et al*. [22] employ a boundary-sensitive contextual incoherence modeling module to detect pixel-level inconsistency and improve the sharpness of the predicted tampering masks. Instead of directly learning pixel-level artifacts on the boundary, we propose to use boundary supervision to guide the model to focus on subtle feature differences between authentic and tampered regions near the boundary. A feature conversion module transforms features into their differences, corresponding to region prediction and boundary prediction, respectively. This approach ensures the unification of the two tasks.

3. Methods

We denote the input as $X \in \mathbb{R}^{H\times W\times 3}$, the tampered region pixel-level annotation of X as $g \in \{0,1\}^{H\times W}$, and the image-level label indicates that whether the image has been altered with as $y \in \{0,1\}$ (authentic/tampered), where H, W and C are the height, width and channel of the image, respectively. The tampered trace features extracted from different block by encoder in baseline is $(F_{tr_i}), F_{tr_i} \in \mathbb{R}^{H_{si}\times W_{si}\times C_{si}}$, where $i \in \{1,2,3,4\}$ denotes the level of feature, and H_{si}, W_{si}, C_{si} are the height, width and channel of the feature, respectively. Let Seg denote the Unet architecture decoder and segmentation head to output probability of each pixel being manipulated, and Cls be the classification head to estimate probability of the image being altered.

In this section, we introduce SAFL-Net, consisting of Semantic Suppression Module (Section 3.1) and Boundary Guidance Module (Section 3.2), that achieves semantic-agnostic feature learning based on the baseline model. Figure 2 gives an overview of the framework. It is worth noting that the proposed method is designed as a plug-and-play approach, which only operates on the feature (F_{tr_i}) and does not require any modifications to the encoder and decoder.

3.1. Semantic Suppression Module

The strong association between image manipulation and semantic information can be easily learned with limited data, but it becomes unreliable in real-world scenarios, and may even mislead predictions of tampered regions. Previous research has acknowledged this issue but has only attempted to limit the learning of semantic information through indirect methods, such as extracting high-frequency or noise features. To address this, we propose a semantic suppression module, as illustrated in Figure 3, which more directly restricts semantic-related features.

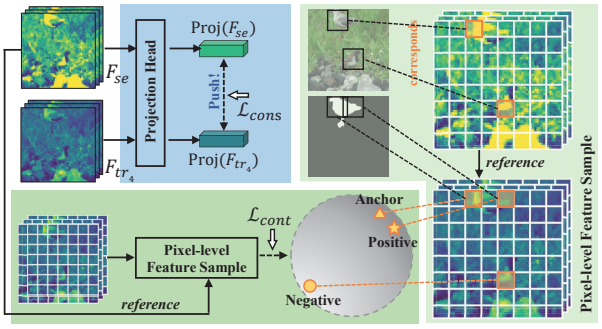

Figure 3. The details of Semantic Suppression Module. The module mainly consists of two parts: Image-level Constraint (in blue) and Patch-level Contrast (in green).

Since shallow feature are less likely to carry semantic information, while deeper feature learned by the encoder tend to capture richer semantic information, the semantic suppression module is specifically applied to the deepest feature F_{tr_4}, referred to as F_{tr} throughout this section.

3.1.1 Benchmark Semantic Representation

Before implementing the semantic suppression module, it is necessary to establish a concrete representation of semantics. In light of the significance of semantic segmentation tasks, we propose to define the features derived from a semantic segmentation model as the **benchmark semantic representation**, serving as the reference for limiting semantic information.

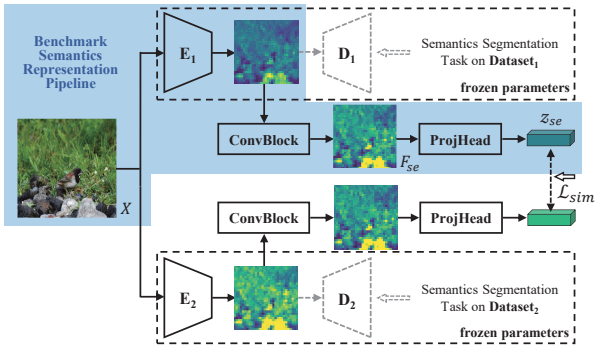

Figure 4. The details of Benchmark Semantic Representation.

Specifically, semantic features of inputs are extracted from a backbone encoder E_1, which is pretrained under semantic segmentation task, and the benchmark semantic representation is obtained by a $\mathrm{ConvBlock}$, denote as $F_{se} = \mathrm{ConvBlock}(E_1(X)) \in \mathbb{R}^{H_s\times W_s\times C_s}$ which has the same dimensions as tampered trace feature F_{tr}. Benchmark semantic representation F_{se} is then mapped to a low-

dimensional vector $\boldsymbol{z}_{se} = \mathrm{Proj}(F_{se})$ by a projection head Proj for further operation, which has been proved effective in previous work[11].

Additionally, in order to minimize the influence of the pretrained dataset and the encoder structure and to maintain semantic information as much as possible in F_{se}, we have designed a symmetric framework to pretrain $\mathrm{ConvBlock}$ and Proj as shown in Figure 4. Another backbone encoder, E_2, which is pretrained on a different dataset than E_1, along with its corresponding $\mathrm{ConvBlock}$ and Proj are integrated to obtain another benchmark semantic representation $\boldsymbol{z}_{se}^2$ for the same input X. The two benchmark semantic representations $\boldsymbol{z}_{se}^1$ and $\boldsymbol{z}_{se}^2$ are expected to be highly consistent through the optimization of the loss function

$$\mathcal{L}_{sim}(X) = \mathrm{Sim}(\boldsymbol{z}_{se}^1, \boldsymbol{z}_{se}^2), \tag{1}$$

where Sim indicates cosine similarity. Eventually, the benchmark semantic representation F_{se} and vector $\boldsymbol{z}_{se}$ can be obtained via pretrained E_1 and trained $\mathrm{ConvBlock}$, Proj (blue in Figure 4). The parameters of pretrained E_1, E_2 are always frozen.

3.1.2 Image-level Constraint

Based on the feature F_{tr}, a vector z_{tr} can be obtained through Proj with the same structure as previously introduced, which has a global receptive field. The aim is to ensure that this traces feature vector z_{tr}accurately reflects the information pertaining to tampering traces, while remaining independent of any abnormal semantic information present in the image content.

As shown blue in Figure 3, we calculate the similarity between z_{tr} and z_{se}, and minimize the similarity by

$$\mathcal{L}_{cons}(X) = \mathrm{Sim}(\boldsymbol{z}_{se}, \boldsymbol{z}_{tr}). \tag{2}$$

This image-level constraint imposes semantic restrictions on tampering features from a global perspective.

3.1.3 Patch-level Contrast

We also introduce contrastive learning[8, 11, 23] in the semantic suppression module to reduce the sensitivity to the semantic information of the tamper region at pixel-level. Denote $F_{tr} = \{\boldsymbol{t}_{ij}\}, F_{se} = \{\boldsymbol{s}_{ij}\}$, $i = 1, \cdots, H_s, j = 1, \cdots, W_s$. The convolution operation will continuously expand the receptive field of the feature, thus the pixel-level vector $\boldsymbol{t}_{ij}$ and $\boldsymbol{s}_{ij}$, where contrastive learning is exactly implemented, correspond to a specific patch in the original input image. To distinguish whether the image patch corresponding to the vector $\boldsymbol{t}_{ij}$ is located in the tamper region, we define a subset $T = \{\boldsymbol{t}_k\} \subset F_{tr}, k = 1, \cdots, |T|$, which consists of the vectors whose corresponding image patch is in the tampering region. Specifically in the experiment, an image patch is regarded in the tampering region, if more than 95% of the pixels in the image patch are tampered with.

Randomly select an anchor $\boldsymbol{t}_k \in T$, a positive set $P(\boldsymbol{t}_k)$ and a negative set $N(\boldsymbol{t}_k)$ is necessary for contrastive loss

$$\begin{aligned}&\mathcal{L}_{cont}(\boldsymbol{t}_k; \tau) \\ &\quad = -\frac{1}{|P(\boldsymbol{t}_k)|} \sum_{\boldsymbol{t}_p \in P(\boldsymbol{t}_k)} \log \frac{\exp(\boldsymbol{t}_k \cdot \boldsymbol{t}_p / \tau)}{\sum_{\boldsymbol{t}_a \in P(\boldsymbol{t}_k) \cup N(\boldsymbol{t}_k)} \exp(\boldsymbol{t}_k \cdot \boldsymbol{t}_a / \tau)},\end{aligned} \tag{3}$$

where $\cdot$ denotes inner product equivalent to cosine similarity between two L_2-normalized vectors, $\tau > 0$ is a temperature hyper-parameter.

Positive set. We expect that the tempered trace features of the tamper region are only related to the local pixels rather than the semantic information of the image content, which means $\boldsymbol{t}_k, \boldsymbol{t}_p \in T$ should be pulled together. Therefore other vectors in the tamper region are selected as positive set $P(\boldsymbol{t}_k) = T \backslash \{\boldsymbol{t}_k\}$.

Negative set. Similarly, the tampered trace features of the untampered region are also expected only related to the local pixels rather than the semantic information of the image content. Further, suppose a tampered vector $\boldsymbol{t}_p \in T$ and an untampered vector $\boldsymbol{t}_{ij} \in F_{tr} \backslash T$, if their corresponding benchmark semantic feature $\boldsymbol{s}_p$ and $\boldsymbol{s}_{ij}$ have a significant degree of similarity, that is, they may have similar semantic information. However, the high similarity is not what we want between a tampered vector $\boldsymbol{t}_p$ and an untampered vector $\boldsymbol{t}_{ij}$, instead, the two vectors need to be pushed away to suppress semantic information. Concretely, the set of benchmark semantic vectors corresponding to the vectors in T is denoted as T_s. For each positive vector $\boldsymbol{t}_p \in P(\boldsymbol{t}_k)$, we select a negative vector $\boldsymbol{t}_n$ by

$$\begin{aligned}&\boldsymbol{t}_n = \boldsymbol{t}_{ij}, \\ &\text{s.t.} \quad (i,j) = \arg\max_{(i,j)} \{\boldsymbol{s}_p \cdot \boldsymbol{s}_{ij} | \boldsymbol{s}_{ij} \in F_{se} \backslash T_s\}.\end{aligned} \tag{4}$$

Then obtain $N(\boldsymbol{t}_k) = \{\boldsymbol{t}_n\}, n = 1, \cdots, |P(\boldsymbol{t}_k)|$.

Combining the above two part, the training objective is

$$\mathcal{L}_{SSM} = (1 - \alpha)\mathcal{L}_{cons} + \alpha \mathcal{L}_{cont}, \tag{5}$$

where α are weighting factor used to balance the two parts in $\mathcal{L}_{SSM}$. By default, we set $\alpha = 0.5$.

3.2. Boundary Guidance Module

The boundaries of tampered regions often leave detectable traces, making it challenging to conceal tampering. Previous works have attempted to guide the attention of model to these boundary traces by treating boundary prediction as an auxiliary task. However, these two tasks serve distinct objectives in feature learning. The former focuses on artifacts and manipulation traces at boundaries, while the

latter facilitates exploration contrasting differences between tampered and authentic regions. This making it challenging for the model to independently transform and unify them [2]. Consequently, as an auxiliary task, the former fails to guarantee providing gains to the latter. To better utilize boundary information, we have two goals: 1) to transform and unify the two tasks in a reasonable manner, and 2) to not only mine the traces on the boundary but also to guide the model to focus on subtle feature differences in relatively large regions inside and outside the boundary.

3.2.1 Differential Block

We design a differential block to achieve a reasonable transformation between the region prediction and the boundary prediction task. We illustrate our idea for the transformation through Figure 5. f_{tr} represents the feature map that focuses on tampered traces that are semantically irrelevant. Ideally, there should be a noticeable difference in features between tampered and authentic region. f_{bd} denotes the feature used for boundary prediction, where there are distinctive features in the boundary. The transformation between f_{tr} and f_{bd} can be achieved using the convolution kernel as shown in the Figure 5, i.e., $f_{bd} = \mathrm{Conv}(f_{tr})$. With this, the conflict between the two tasks is resolved, and learning the boundary tampering features does not negatively affect the learning of tampering features, as the boundary supervision is transformed by the differential block to guide the encoder to mine subtle differences in regions inside and outside the boundary.

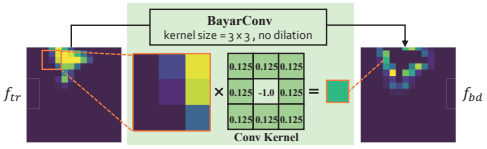

Figure 5. Illustration of the transformation process.

In practical situations, the feature maps are not as intuitive and orderly as shown above, and a fixed parameter convolution kernel, which plays a differential role, cannot be designed reasonably and effectively. This inspired us to use trainable but constrained BayarConv [1] as the core component of the transformation module. The purpose is to learn the differences between the features of the current convolution operation window's central position and its surroundings, while the trainable parameters of the BayarConv allow for greater flexibility in the module's performance.

As shown in Figure 6, different from the original version of BayarConv, we use dilated convolution and set dilation $scale_i$ for different level of F_{tr_i}, then match the Conv_{scale_i} of the corresponding scale, the Differential Block DB_i is formed. We can perform a reasonable transformation between two feature maps for different tasks by

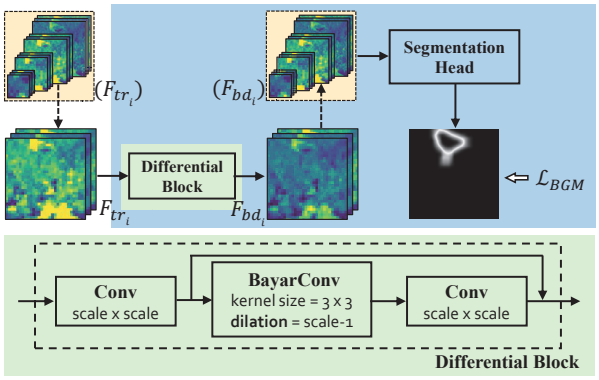

Figure 6. The details of Boundary Guidance Module.

$F_{bd_i} = \mathrm{DB}_i(F_{tr_i})$. Different scale settings let the Differential Block has different perception ranges on feature maps of different scales. This is to maximize the impact of boundary supervision over a larger area. The constraints of BayarConv are

$$\begin{cases} \omega_k^{\mathrm{BayarConv}}(0,0) = 1, \\ \sum_{(m,n)} \omega_k^{\mathrm{BayarConv}}(m,n) = 0, \end{cases} \tag{6}$$

where spatial index $(0,0)$ denotes the central value of the convolution filter. Since we guide the encoder to pay attention to the subtle feature differences in local regions through boundary supervision, the receptive field of high-level features is limited to a certain extent, and the learning of semantically irrelevant features is further completed.

3.2.2 Soft Boundary Supervision

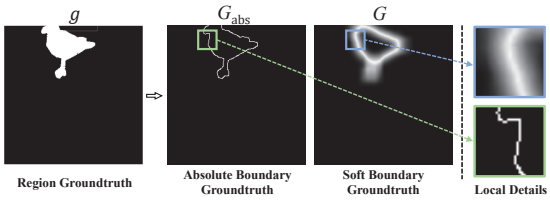

Figure 7. Illustration of the extracted soft boundary groundtruth.

Previous methods [19, 28, 2] utilize absolute boundary groundtruth G_{abs} extracted by edge detection methods. In contrast, we employ a sliding window W to measure the distance of each pixel from the absolute boundary, rather than a binary label indicating whether the pixel is a boundary or not, and thereby obtain soft boundary groundtruth G,

$$G_{i,j} = 1 - \frac{\left|\sum_{(m,n)\in W_{ij}} 1 - 2\sum_{(m,n)\in W_{ij}} g_{m,n}\right|}{\sum_{(m,n)\in W_{ij}} 1}, \tag{7}$$

where W_{ij} denotes the window centered on (i, j), and $g_{m,n}$ is the value at (m, n) in the pixel-level manipulation annotation. The soft boundary supervision, illustrated in Figure 7, is more compatible with the differential block as it provides a more reasonable expression of the boundary concept and significantly increases the number of positive pixel-level samples, which is beneficial for segmentation tasks.

The commonly used binary cross-entropy (BCE) loss function is unable to provide soft supervision and does not account for the varying significance of different pixels, so we adopt pixel-wise weighted BCE loss and combined L1 loss for boundary prediction, denoted as

$$\mathcal{L}_{BGM} = \sum_{(i,j)} (\mathcal{L}_{bce}(S_{i,j}, bin(G_{i,j})) \cdot w_{i,j} + \mathcal{L}_{l1}(S_{i,j}, G_{i,j})) \tag{8}$$

where $S = \mathrm{Seg}^{\mathrm{bd}}(X) \in \mathbb{R}^{H \times W}$. To obtain a pixel-wise weight for each pixel $G_{i,j}$ in G, we binarize the boundary map G, and then add a small constant γ to the resulting binary map to obtain a soft label $w_{i,j} = G_{i,j} + \gamma$.

3.3. Learning Objective

In our method's baseline, we employ the BCE loss function for manipulation localization and detection, referred to as $\mathcal{L}_{seg}$ and $\mathcal{L}_{cls}$ respectively. By combining all components, we obtain the overall learning objective given by

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{SSM} + \lambda_3 \mathcal{L}_{BGM}, \tag{9}$$

where $\lambda_1, \lambda_2, \lambda_3$ are weighting factors that are used to balance the different modules. In our experiments, we set the default values for λ_1, λ_2, and λ_3 to be 0.2, 0.5, and 0.5 respectively. It is worth noting that authentic images is only used for calculating the detection loss, $\mathcal{L}_{cls}$.

4. Experiments

We synthesize the experimental setups and evaluation metrics used by the latest state-of-the-art method, ObjectFormer [22], in order to comprehensively evaluate and make equitable comparisons. Furthermore, we conduct detailed ablation experiments on the settings related to the proposed plug-ins and provide necessary visualizations to illustrate the role of each component more intuitively.

4.1. Experimental Setup

Synthesized Dataset. We modify existing public datasets to create a rich dataset consisting of four manipulation types and three subsets: 1) ProDEFACTO, obtained by post-processing the original DEFACTO [17], which includes a large amount of data, including splicing (*spli.*), copy-move (*cpmv.*), and inpainting (*inpa.*), and is built based on MS COCO [15]. 2) PSBattles [9], gathered from a large community of image manipulation enthusiasts, provides a more refined form of manipulation (*ps.*) that is more valuable in real scenes and more difficult to detect. 3) Authentic images, collected from MS COCO and PSBattles corresponding to all selected tampered images. The existence of semantically related tampering operations in the dataset ensures that the pre-training data is more aligned with real-world scenarios. Moreover, this characteristic of the dataset serves as evidence of our method's ability to learn semantic-agnostic features.

Partition	**Negative**	**Positive**	*spli.*	*cpmv.*	*inpa.*	*ps.*
Training	17,554	30,000	5,000	5,000	5,000	15,000
Testing	3,747	4,000	1,000	1,000	1,000	1,000

Table 1. The statistics of the synthesized dataset.

Dataset	**Negative**	**Positive**	*spli.*	*cpmv.*	*inpa.*	*ps.*
Columbia [20]	183	180	180	-	-	-
Coverage [24]	100	100	-	100	-	-
CASIAv1 [5]	800	920	459	461	-	-
CASIAv2 [5]	7,491	5,063	3,235	1,828	-	-
NIST16 [7]	-	564	288	68	208	-
IMD20 [18]	414	2,010	-	-	-	2,010

Table 2. The statistics of the selected public datasets.

We partition the synthetic dataset into a training set and a test set. *Training* set is utilized for pre-training our model, whereas *Testing* set is utilized to conduct ablation experiments. Table 1 provides detailed statistical information regarding this dataset.

Public Datasets. We select five public datasets to evaluate the performance of our model. Table 2 shows the number of authentic and tampered images, as well as the number of samples of different tampering types in each dataset. To ensure fair comparisons with previous work [10, 16, 22], we use the same training/testing splits for fine-tuning the model.

Evaluation Metrics. To evaluate both image manipulation detection and localization, we employ metrics at both pixel-level and image-level. To locate forged regions, we utilize pixel-level metrics such as Area Under Curve (AUC) and F1 score, which is the harmonic mean of precision and recall scores. For detection, we report image-level AUC, sensitivity, specificity, and F1 score to measure the miss detection rate and false alarm rate. In order to calculate the F1 score, we use binarized prediction masks and labels with a fixed threshold of 0.5 in the ablation experiments. To ensure a fair comparison with state-of-the-art methods, we employ a certain strategy as introduced in [22] to select the best threshold.

Implementation. We implement SAFL-Net using PyTorch and trained it with an NVIDIA A100 GPU. The input size is set to 512×512, and we use the pre-trained Efficientnet-b7 [21] as the backbone, which is pre-trained with ImageNet [3]. AdamW is used for optimization with a

Setup (Seg+)	**Colombia**	**Coverage**	**CASIA**	**NIST16**	**IMD20**
0: Cls#0	90.3	74.1	73.8	72.4	83.1
1: Cls#1 (Baseline)	90.2	81.6	77.7	79.6	84.6
2: Cls#1+bdSup#0	91.9	81.7	72.5	77.4	79.7
3: Cls#1+BGM#1	93.4	84.6	80.1	83.2	87.3
4: Cls#1+BGM#2	95.5	88.2	81.9	83.9	89.9
5: Cls#1+BGM#2+SSM#1	97.1	93.8	86.1	86.0	93.1
6: Cls#1+BGM#2+SSM#2	96.9	93.5	90.9	88.8	96.5

Table 3. Ablation study of different modules (pixel-level AUC).

Method	**Data**	**Colombia**	**Coverage**	**CASIA**	**NIST16**	**IMD20**
ManTra-Net	64K	82.4	91.0	81.7	79.5	74.8
SPAN	96K	93.6	92.2	79.7	84.0	75.0
MVSS-Net	12K	87.0	87.8	-	78.8	81.5
MVSS-Net++	12K	80.7	88.1	-	78.4	81.3
CL-Net	100K	94.5	82.3	81.6	84.7	-
PSCC-Net	100K	**98.2**	84.7	82.9	85.5	80.6
ObjectFormer	62K	95.5	92.8	84.3	87.2	82.1
Ours	48K	96.9	**93.5**	**90.9**	**88.8**	**96.5**

Table 4. Performance of pretrained models at pixel-level.

learning rate of 0.0002. In the pre-training stage, we train the complete model for 25 epochs with a batch size of 16, and the learning rate is decayed by a factor of 2 every 5 epochs. We only apply flipping as the data augmentation technique during training.

4.2. Ablation Study

To assess the effectiveness of each component in enhancing generalization, we conduct ablation studies on public datasets that the model has not encountered during training. Setups of the ablation study are listed in Table 3.

Effect of Classification Head. Several of the methods compared in this study [25, 10, 27] do not include a specific classification head for image-level prediction. For these models, image-level supervision can only be performed by adopting a strategy such as averaging the pixel-level outputs to calculate the image-level prediction score (Cls#0). However, this approach may cause the model to be overly conservative and biased towards negative predictions, resulting in a higher specificity and abnormal sensitivity. To address this issue and ensure performance in image-level detection task, we introduce a dedicated classification head (Cls#1) that effectively balances specificity and sensitivity, resulting in significant improvements in pixel-level performance. We consider this configuration as our baseline (Setup#1).

Effect of BGM. Previous works such as MFCN [19] and GSR-Net [28] have used tampered boundary prediction as an auxiliary task (bdSup#0). However, since region segmentation and edge detection are inherently different tasks, it is challenging to find a suitable balance between them [4]. The results in Setup#2 demonstrate the existence of this difference and the negative impact it can have.

We propose the Boundary Guidance Module (BGM#1), which includes the Difference Block to establish a transformation bridge between the two tasks. This transformation structure guides the model to focus not only on tampering artifacts along the boundary, but also on discovering subtle feature differences inside and outside the boundary through boundary guidance. The experimental results in Setup#3 demonstrate the effectiveness of this module. Moreover, we develop a loss function $\mathcal{L}_{BGM}$ tailored to the proposed soft boundary supervision (BGM#2), which is used in Setup#4.

Effect of SSM. The SSM module exerts direct constraints on the core tampering feature extractor by leveraging benchmark semantic features. From a global feature perspective, the module constrains the extracted information from the feature map to deviate from its natural semantic information via the image constraint loss $\mathcal{L}_{cons}$ (SSM#1). Furthermore, the module enhances local feature learning through contrastive learning loss $\mathcal{L}_{cont}$ (SSM#2), which further stresses semantic-agnostic learning through a positive-negative sample selection strategy.

Comparing Setup#6 with Setup#5 demonstrates the effectiveness of $\mathcal{L}_{cont}$ in pixel-level localization tasks. Overall, the SSM module provides the most significant improvement for *ps.* tampering type, suggesting that the module effectively suppresses the semantic-related feature learning to extract tampering artifact feature.

4.3. Comparison with the State-of-the-art

We compare our method with other state-of-the-art methods under two settings: 1) pretraining on an integrated dataset and evaluating on other test datasets. 2) fine-tuning the pretrained model on the training split of test datasets and evaluating on their test split.

Methods to compare. We compare our method with various models as described following: Mantra-Net [25], which extracts trace features using SRM and BayarConv and localize the forgery with a anomaly detection network. RGB-N [29], which leverages RGB stream and noise stream to independently identify tampering features and noise inconsistencies in an image. SPAN [10], which adopts the feature extractor of Mantra-Net and introduces pyramid spatial attention architecture. MVSS-Net [2] and MVSS-Net++ [4], which consist of boundary supervised branch and noise sensitive branch, and dual attention [6] is adopted to fuse features from them. CL-Net [27], which propose a novel representation learning approach based on contrastive learning. PSCC-Net [16], which introduces a network with dense cross-connections to leverage features at different scales. ObjectFormer [22], which combines RGB features and high-frequency features as patch embeddings, and uses transformer architecture. We adopt a consistent experimental setup and utilize the results reported in their respective papers.

Pretrained models. The pixel-level localization performance of the pretrained model is presented in Table 4. SAFL-Net achieves state-of-the-art performance on most datasets, particularly on the real-world dataset IMD20, where it achieves 96.5, outperforming ObjectFormer by 14.4. On the Columbia dataset, we surpass CL-Net and ObjectFormer by 2.4 and 1.4, respectively, but falls 1.3 behind PSCC-Net. We hypothesize that this might be because the synthesized training data used by PSCC-Net closely resembles the distribution of the Columbia dataset [22]. This can be further verified by the results on other datasets, which show that SAFL-Net outperforms PSCC-Net. It is worth noting that we achieve good results using less pre-training data compared to other methods except MVSS-Net.

Method	AUC	F1
ManTra-Net	59.94	56.69
SPAN	67.33	63.48
PSCC-Net	99.65	97.12
ObjectFormer	**99.70**	97.34
Ours	99.48	**98.37**

Table 5. Performance of pretrained models at image-level.

We also evaluate our model on CASIA-D, introduced by [10], to demonstrate the image-level detection performance, and the results are listed in Table 5. Our AUC metric lags behind ObjectFormer by 0.22, but our F1 metric surpasses ObjectFormer by 1.03, achieving state-of-the-art performance. This highlights the effectiveness of BGM and SSM in capturing manipulation artifacts.

Method	Coverage		CASIA		NIST16	
	AUC	F1	AUC	F1	AUC	F1
RGB-N	81.7	43.7	79.5	40.8	93.7	72.2
SPAN	93.7	55.8	83.8	38.2	96.1	58.2
MVSS-Net	84.9	50.4	87.7	52.2	94.2	81.4
CL-Net	85.7	51.2	89.5	58.4	98.5	82.3
PSCC-Net	94.1	72.3	87.5	55.4	99.6	81.9
ObjectFormer	95.7	75.8	88.2	57.9	99.6	82.4
Ours	**97.0**	**80.3**	**90.8**	**74.0**	**99.7**	**87.9**

Table 6. Performance of fine-tuned models at pixel-level.

Fine-tuned models. Following the approach in [22], we performed fine-tuning of the pretrained models on specific datasets, and the results are presented in Table 6. The significant improvements in both AUC and F1 score demonstrate that SAFL-Net is capable of learning tampering features without being affected by semantic information.

4.4. Visualization

The visualization of outputs predicted by various methods and SAFL-Net are presented in Figure 1. Three high-quality manipulated images that closely resemble real-world scenarios are selected for this analysis, obtained from PSBattles, IMD20 and NIST16, respectively. False alarms in pixel-level with significant semantic association are identified and marked by green boxes. For manipulated images that are more complex and closer to real-world scenes, traditional methods relying on handcrafted features like Mantra-Net and SPAN are unable to perform well, while models with advanced and complex structures like MVSS-Net suffer from significant semantic-related false alarms. SAFL-Net learns semantic-agnostic features which enables it to accurately locate the tampered regions and reduce false alarms in regions without forgery.

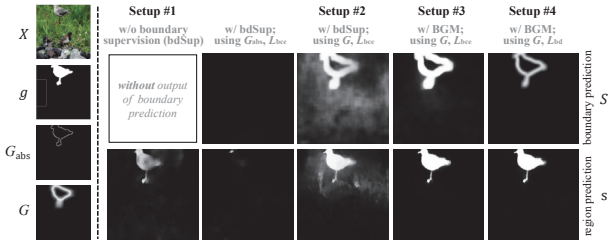

Figure 8. Ablation visualization on various setups about BGM.

To better exploit the role of BGM, we specifically design a soft boundary supervision G and a corresponding loss function $\mathcal{L}_{BGM}$. To highlight the performance gain achieved with this approach compared to using absolute boundary mask G_{abs} and BCE loss $\mathcal{L}_{bce}$, we visualize the outputs under various settings, as depicted in Figure 8.

4.5. Robustness Evaluation

Distortion	SPAN	MVSS-Net	PSCC-Net	ObjectFormer	Ours
no distortion	83.95	78.82	85.47	87.18	**88.79**
Resize (0.78×)	83.23	78.32	85.29	87.17	**88.39**
Resize (0.25×)	80.32	77.54	85.01	86.33	**86.92**
GaussianBlur (k3)	83.10	78.60	85.38	85.97	**88.13**
GaussianBlur (k15)	79.15	75.81	79.93	80.26	**87.68**
Compress (q100)	83.59	78.84	85.40	86.37	**88.56**
Compress (q50)	80.68	78.84	85.37	86.24	**88.07**

Table 7. Performance on NIST16 dataset under various distortions.

The robustness is also of vital significance due to the inevitable various post processing operations when the images spreads. We evaluate the robustness of our model by applying different image distortion methods to the raw images from the NIST16 dataset, following the approach in [22]. The manipulation localization performance (AUC score) of our pretrained models and other methods on these corrupted data is compared, and the results are reported in Table 7. Our SAFL-Net shows superior robustness against various distortion techniques compared to other methods.

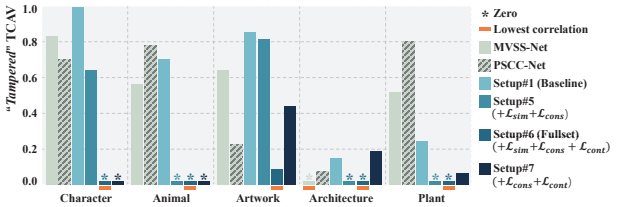

Figure 9. Degree of correlation between concepts and "tampered".

4.6. Statistics of Semantic Correlation

To quantify the correlation between model predictions and specific semantics within tampered regions, we employ TCAV [12] for statistical analysis. TCAV serves as an interpretation tool that can quantify the extent of correlation between user-defined concepts and specific predictions. We select several frequently encountered semantics in image tampering datasets as concepts for analysis. These concepts include character, animal, artwork, architecture, and plant. In this study, high-level concepts are defined using sets of example input images sourced from CASIA. This choice is motivated by the substantial collection of tampered regions corresponding to selected semantic categories.

The ablation study and the comparison with others in Figure 9 demonstrates the ability of semantic-agnostic feature learning. Both the existing research and the baseline model without additional constraints exhibit strong correlations in predicting tampering, while our design within the SSM module consistently demonstrates effective semantic-agnostic feature learning capabilities.

5. Conclusion

We introduced SAFL-Net, a network with two auxiliary plugins for image manipulation detection and localization. To restrict the semantics of feature and improve the generalization performance of the model, Semantic Suppression Module restricts semantic information directly depending on the benchmark semantic representation. Besides, Boundary Guidance Module achieves a unification of the auxiliary boundary prediction task and the original region prediction task through a feature transformation structure. Moreover, it explores subtle difference on both sides of the tampered region boundary. Extensive experiments on different benchmarks demonstrate the effectiveness of the proposed modules and our framework.

6. Acknowledgements

The research is supported in part by the National Natural Science Foundation of China (62203425, 62172420), the Project of Chinese Academy of Sciences (E141020).

References

[1] Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.

[2] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14185–14193, 2021.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[5] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426. IEEE, 2013.

[6] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019.

[7] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72. IEEE, 2019.

[8] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[9] Silvan Heller, Luca Rossetto, and Heiko Schuldt. The ps-battles dataset-an image collection for image manipulation detection. *arXiv preprint arXiv:1804.04866*, 2018.

[10] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *European conference on computer vision*, pages 312–328. Springer, 2020.

[11] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[12] Been Kim et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[13] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. Cat-net: Compression artifact tracing net-

work for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 375–384, 2021.

[14] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[16] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[17] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc. Defacto: image and face manipulation dataset. In *2019 27Th european signal processing conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[18] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. Imd2020: a large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 71–80, 2020.

[19] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018.

[20] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[21] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[22] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2022.

[23] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021.

[24] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage—a novel database for copy-move forgery detection. In *2016 IEEE international conference on image processing (ICIP)*, pages 161–165. IEEE, 2016.

[25] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2019.

[26] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained r-cnn: A general image manipulation detection model. In *2020 IEEE International conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2020.

[27] Qilin Yin, Jinwei Wang, Wei Lu, and Xiangyang Luo. Contrastive learning based multi-task network for image manipulation detection. *Signal Processing*, 201:108709, 2022.

[28] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, Abhinav Shrivastava, Ser-Nam Lim, and Larry Davis. Generate, segment, and refine: Towards generic manipulation segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13058–13065, 2020.

[29] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1053–1061, 2018.