Inharmonious Region Localization with Auxiliary Style Feature

Penghao Wu
wupenghaocraig@sjtu.edu.cn
Li Niu*
ustcnewly@sjtu.edu.cn
Liqing Zhang
zhang-lq@cs.sjtu.edu.cn

MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
Shanghai, China

Abstract

With the prevalence of image editing techniques, users can create fantastic synthetic images, but the image quality may be compromised by the color/illumination discrepancy between the manipulated region and background. Inharmonious region localization aims to localize the inharmonious region in a synthetic image. In this work, we attempt to leverage auxiliary style feature to facilitate this task. Specifically, we propose a novel color mapping module and a style feature loss to extract discriminative style features containing task-relevant color/illumination information. Based on the extracted style features, we also propose a novel style voting module to guide the localization of inharmonious region. Moreover, we introduce semantic information into the style voting module to achieve further improvement. Our method surpasses the existing methods by a large margin on the benchmark dataset.

1 Introduction

With the wide application of photography and editing technology, people can easily create marvellous synthetic images with common image editing operations (*e.g.*, copy-paste, appearance adjustment). However, one serious problem of synthetic images is that the manipulated region may have inconsistent color and illumination characteristics with the background (see Fig. 1), making the whole image inharmonious and unrealistic. The inharmonious region localization task [21] aims to localize the inharmonious region, after which users can manually adjust the inharmonious region or utilize image harmonization techniques [8, 31] to harmonize the inharmonious region, yielding the images with higher quality and fidelity. Therefore, inharmonious region localization is indispensable for blind image harmonization [11], in which the inharmonious region mask is unavailable.

The first method on inharmonious region localization is DIRL [21], which mainly focuses on the backbone design to fuse multi-scale features and suppress redundant information, ignoring critical information related to color and illumination statistics, which is actually the essence of this task. Recent work MadisNet [22] transforms the image to another color-space to magnify domain discrepancy. Different from previous works, we aim to

*Corresponding author.

Figure 1: Examples of inharmonious images (1^{st} row) and their associated masks (2^{nd} row).

extract *style features* containing task-relevant color and illumination information with a style encoder, to help localize the inharmonious region. By dividing the image into inharmonious region and harmonious (background) region, we enforce intra-region coherence and inter-region divergence using a *style feature loss*, by pulling close the pixel-level style features within the same region while pushing apart those across different regions. Unlike the complex color mapping model in MadisNet [22], we design a simple yet effective *color mapping module* to manipulate the input image to help extract more discriminative style features.

To fully exploit the potential of the pixel-level style features extracted from the style encoder, we propose a novel *style voting module* and insert it into each decoder stage. Each decoder stage produces an auxiliary inharmonious region mask which is used to select harmonious pixels as voters. These voters need to vote for the pixels with similar style features. For each pixel, the total score it receives represents its probability of being harmonious. Furthermore, considering that semantically similar regions usually provide more reliable clues, we include extra semantic information to calculate weights, which are assigned to the scores given from each voter to each pixel. Finally, the weighted total scores of all pixels form the voting score map, which serves as the guidance for the next decoder stage.

In summary, our whole network consists of a style encoder and a UNet structure, which are linked together by style features. A color mapping module is placed in front of style encoder to help extract better style features. A style voting module leverages the style features to produce a style voting map to guide the UNet decoder. Because the semantic information in style voting module is expensive and optional, we refer to our base model without semantic information as AustNet (**Au**xiliary **St**yle feature) and the enhanced version with semantic information as AustNet-S. Following [21], we conduct experiments on the benchmark dataset iHarmony4 [8] and our method outperforms the state-of-the-art method MadisNet [22] by a large margin. Specifically, we improve the AP from 85.86% to 92.01% with our AustNet and to 93.01% with our AustNet-S compared with the SOTA results. Our main contributions include:

• We design a style encoder with a simple color mapping module to extract discriminative style feature to facilitate inharmonious region localization.

• We propose a novel style voting module based on extracted style features to provide guiding information for the decoder.

• We introduce semantic information into the weighting scheme in our style voting module, which can achieve further improvement.

2 Related Work

2.1 Image Manipulation Detection

Image manipulation detection aims to detect and localize the manipulated or tampered image, which is somewhat similar to inharmonious region localization. In image manipulation detection, manipulation operations usually contain copy-move, removal, inpainting, and splicing. Traditional image manipulation methods heavily rely on prior information of manipulated images like noise patterns [23, 25] and JPEG compression artifacts [4, 5, 20]. Recently, deep learning approaches tackle the image manipulation detection task by comparing local patches [2, 27, 29], extracting forgery features [3, 33, 36, 39], and adversarial learning [18]. However, the discrepancy between color or illumination statistics, which is the main focus of inharmonious region localization, is not specifically studied in these methods.

2.2 Image Harmonization

Image harmonization [1, 6, 8, 10, 14, 15, 31] aims to adjust the foreground of a composite image in terms of color and illumination characteristics, to make the foreground compatible with the background, which can be deemed as a successor task of inharmonious region localization. Tsai *et al.* [31] proposed the first convolutional neural network for image harmonization. A spatially separated attention module S^2AM was proposed in [11] to process the features of foreground and background differently. Domain translation was adopted in [8, 9] to translate the inharmonious foreground to the background domain. Images are decomposed into illumination map and reflectance map in [13, 14] to tackle the harmonization problem. Semantic information was utilized in [30] to assist with the harmonization process.

Although image harmonization methods have shown impressive performance, most of them require the ground-truth foreground mask, which are not always available in real-world applications. For blind image harmonization without provided foreground mask, S^2AM predicts the inharmonious foreground mask as an auxiliary task, but it is not the main focus and the quality is very low. Therefore, the importance of inharmonious region localization is significant and the quality of the predicted mask is crucial for blind image harmonization.

2.3 Inharmonious Region Localization

Inharmonious region localization aims to localize the inharmonious region which is incompatible with the background due to distinctive color and illumination statistics. DIRL [21] is the first method working on inharmonious region localization, which develops an effective way to merge multi-scale features and use mask guided attention module to localize the inharmonious region. However, the discrepancy about color and illumination information was not fully exploited in [21]. MadisNet [22] uses HDRNet [12] to map the input color space to magnify the domain discrepancy, while we propose to use a simple linear color mapping module to better extract style features. Moreover, we leverage the style features in two aspects, which has never been explored before. We are also the first to introduce semantic information into inharmonious region localization task.

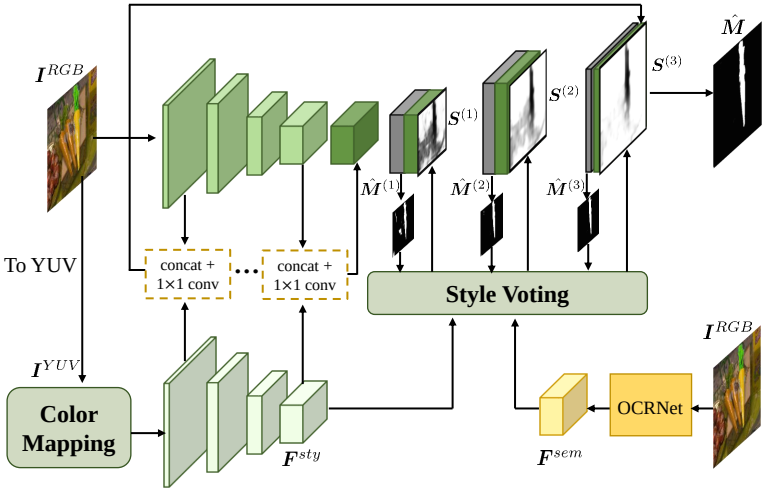

Figure 2: Our network structure is comprised of a style encoder and a UNet [28] structure. We insert a color mapping module in front of style encoder to help extract better style features $\boldsymbol{F}^{sty}$. The style voting module takes in $\boldsymbol{F}^{sty}$, the semantic features $\boldsymbol{F}^{sem}$ from pretrained segmentation network (*e.g.*, OCRNet [37]), and the auxiliary inharmonious region mask $\hat{\boldsymbol{M}}^{(k)}$ from each decoder stage, producing a style voting map $\boldsymbol{S}^{(k)}$.

3 Methodology

Given a synthetic RGB image $\boldsymbol{I}^{RGB}$, our goal is estimating a binary mask $\hat{\boldsymbol{M}}$ to localize the inharmonious region. As shown in Fig. 2, our whole framework is comprised of a UNet [28] and a style encoder. For UNet, we adopt ResNet34 [16] as the encoder, which takes the RGB image $\boldsymbol{I}^{RGB}$ as input and extracts multi-scale feature maps. For the style encoder, we first convert RGB image $\boldsymbol{I}^{RGB}$ to YUV image $\boldsymbol{I}^{YUV}$ (see Sec 3.1), and then apply a color mapping module to map it to a new color space that allows us to extract more discriminative style features $\boldsymbol{F}^{sty}$.

In the UNet decoder, each decoder stage predicts an auxiliary inharmonious region mask. Moreover, we insert a style voting module after each decoder stage to guide the mask estimation in a coarse-to-fine manner and output the final mask $\hat{\boldsymbol{M}}$. By taking the k-th decoder stage as an example, its output inharmonious region mask $\hat{\boldsymbol{M}}^{(k)}$ and the style features $\boldsymbol{F}^{sty}$ are sent into the style voting module to produce a voting score map $\boldsymbol{S}^{(k)}$. Additionally, we merge the multi-scale features from UNet encoder with the multi-scale features from style encoder via concatenation and 1×1 convolution, leading to aggregated encoder features. Then, the voting score map $\boldsymbol{S}^{(k)}$ is concatenated with aggregated encoder features and delivered to the next decoder stage. Finally, the last decoder stage outputs the final mask $\hat{\boldsymbol{M}}$. Next, we detail our style encoder in Sec. 3.1 and style voting module in Sec. 3.2.

3.1 Style Encoder

We adopt ResNet34 [16] as the backbone of style encoder to extract multi-scale features. We refer to the feature map from the last layer as style feature map, which contains task-relevant

color and illumination information. We first introduce the loss to regulate the style features, and then introduce the color mapping module which can help extract better style features.

3.1.1 Style Feature Loss

The extracted style feature map $\boldsymbol{F}^{sty}$ is expected to contain task-relevant color and illumination information, which could distinguish the inharmonious region from the background. Therefore, the main goal of style feature loss is to enlarge intra-region coherence and inter-region divergence. Specifically, we use s_{inter} to denote the average cosine similarity between pixel-level style features across different regions (inharmonious region and harmonious region), and s_{intra} to denote the average cosine similarity between pixel-level style features within the same region (either inharmonious or harmonious region). By denoting the ground-truth inharmonious region mask as $\boldsymbol{M}$, we define a set of inter-region pixel pairs $\mathcal{P}_{inter} = \{(p_1, p_2) | M_{p_1} \neq M_{p_2}\}$, in which p is a 2D position and M_p is the p-th entry in $\boldsymbol{M}$. Similarly, we define a set of intra-region pixel pairs $\mathcal{P}_{intra} = \{(p_1, p_2) | M_{p_1} = M_{p_2}\}$. By using $cos(\cdot, \cdot)$ to denote the cosine similarity between two feature vectors, s_{inter} and s_{intra} are calculated as follows,

$$s_{inter} = \frac{1}{|\mathcal{P}_{inter}|} \sum_{(p_1,p_2)\in\mathcal{P}_{inter}} cos(\boldsymbol{F}^{sty}_{p_1}, \boldsymbol{F}^{sty}_{p_2}), s_{intra} = \frac{1}{|\mathcal{P}_{intra}|} \sum_{(p_1,p_2)\in\mathcal{P}_{intra}} cos(\boldsymbol{F}^{sty}_{p_1}, \boldsymbol{F}^{sty}_{p_2}). \quad (1)$$

We adopt the triplet loss ℓ_{sty} to enforce s_{inter} to be smaller than s_{intra} by a margin m: $\ell_{sty} = \max(s_{inter} - s_{intra} + m, 0)$, where m is set to 0.5 via cross-validation. In this way, we pull close the pixel-level style features within the same region while pushing apart them across different regions.

3.1.2 Color Mapping Module

To extract more discriminative style features, we insert a simple color mapping module in front of the style encoder, which converts the input image to another color space using linear color transformation. By jointly training color mapping module and style encoder supervised by the style feature loss, the intra-region coherence and inter-region divergence could be enlarged in the new color space.

Prior to using our color mapping module, we first convert the input RGB image $\boldsymbol{I}^{RGB} \in \mathbb{R}^{H\times W\times 3}$ to a YUV image $\boldsymbol{I}^{YUV} \in \mathbb{R}^{H\times W\times 3}$. Compared with the correlated RGB color space, YUV is a decorrelated color space, in which the luminance channel (Y channel) encodes the intensity of light and the chrominance channels (U, V channels) encode the color information. Since the inharmony is caused by color/illumination discrepancy, a YUV image may exhibit the discrepancy in a better way. Moreover, we can also transform each channel independently in the decorrelated color space.

Our color mapping module applies a simple convolutional block (3×3 and 7×7 convolutions) to the input YUV image to produce linear transformation parameters $\boldsymbol{P} = [\boldsymbol{A}, \boldsymbol{B}] \in \mathbb{R}^{H\times W\times 6}$, where $\boldsymbol{A} \in \mathbb{R}^{H\times W\times 3}$ and $\boldsymbol{B} \in \mathbb{R}^{H\times W\times 3}$. The linear color transformation is both position-specific and channel-specific. Formally, the value $I^{YUV}_{c,p}$ in the c-th channel at 2D position p of $\boldsymbol{I}^{YUV}$ will be mapped to a new value in the following way:

$$\hat{I}_{c,p} = A_{c,p} \times I^{YUV}_{c,p} + B_{c,p}, \quad (2)$$

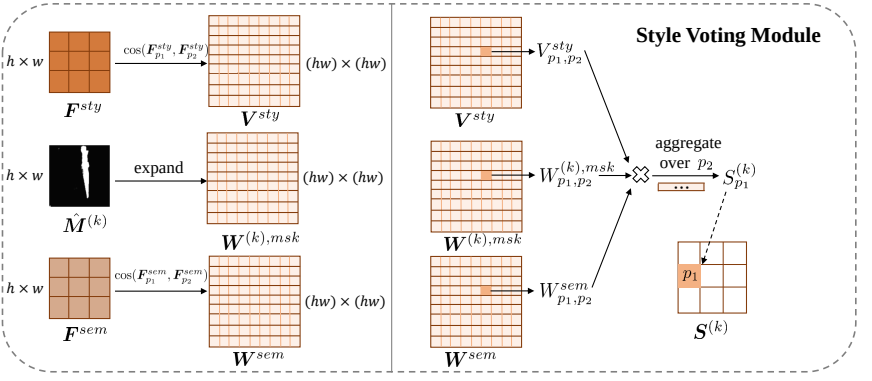

Figure 3: Details of the style voting module. The left part shows constructions of style similarity matrix $\boldsymbol{V}^{sty}$, harmonious weight matrix $\hat{\boldsymbol{M}}^{(k)}$, and semantic similarity matrix $\boldsymbol{W}_{sem}$. The right part illustrates calculations of the final voting score $S^{(k)}_{p_1}$ for a specific pixel in $\boldsymbol{S}^{(k)}$.

where $A_{c,p}$ and $B_{c,p}$ are the transformation parameters corresponding to the channel c and 2D location p. After the mapping process, we can get an image $\hat{\boldsymbol{I}} \in \mathbb{R}^{H\times W\times 3}$ with each entry being $\hat{I}_{c,p}$. Then, the style encoder takes in $\hat{\boldsymbol{I}}$ and outputs a style feature map $\boldsymbol{F}^{sty} \in \mathbb{R}^{h\times w\times C}$, in which $h = \frac{H}{8}$ and $w = \frac{W}{8}$.

3.2 Style Voting Module

Based on the extracted style feature map $\boldsymbol{F}^{sty} \in \mathbb{R}^{h\times w\times C}$, we design a novel style voting module, which produces a voting score map to indicate the harmonious pixels. We insert it after each decoder stage. By taking the k-th decoder stage as an example, we use its output inharmonious region mask to select harmonious pixels as voters. The voters need to vote for similar pixels based on style features. After all the votings, we calculate the total score each pixel receives from all voters to be the final score for it, which indicates how likely this pixel is harmonious. Finally, we obtain a voting score map formed by the final scores of all pixels, which highlights the harmonious regions. The process of voting is visualized in Fig. 3.

Our style voting module takes the style feature map $\boldsymbol{F}^{sty} \in \mathbb{R}^{h\times w\times C}$, and calculates a style similarity matrix $\boldsymbol{V}^{sty} \in \mathcal{R}^{(h\times w)\times(h\times w)}$ with the (p_1, p_2)-th entry $V^{sty}_{p_1,p_2}$ being the cosine similarity between the p_1-th pixel-level style feature and the p_2-th pixel-level style feature:

$$V^{sty}_{p_1,p_2} = cos(\boldsymbol{F}^{sty}_{p_1}, \boldsymbol{F}^{sty}_{p_2}), \tag{3}$$

in which $V^{sty}_{p_1,p_2}$ can be viewed as the score that the p_1-th pixel receives from the p_2-th voter.

Suppose that the output inharmonious region mask from the k-th decoder stage is $\hat{\boldsymbol{M}}^{(k)}$, we resize $\hat{\boldsymbol{M}}^{(k)}$ to $h \times w$. Then, we select harmonious pixels as voters by assigning weight $(1 - \hat{M}^{(k)}_p)$ to the p-th pixel, because $\hat{M}^{(k)}_p$ is the p-th entry in $\hat{\boldsymbol{M}}^{(k)}$ and small $\hat{M}^{(k)}_p$ indicates reliable harmonious pixels. We replicate $\hat{\boldsymbol{M}}^{(k)}$ for $(h \times w)$ times and arrive at the weight matrix $\boldsymbol{W}^{(k),msk} \in \mathcal{R}^{(h\times w)\times(h\times w)}$. Based on the weight matrix, the weighted total score that

the p_1-th pixel receives can be represented by

$$S_{p_1}^{(k)} = \sum_{p_2} W_{p_1,p_2}^{(k),msk} V_{p_1,p_2}^{sty}. \tag{4}$$

The scores of all pixels form a voting score map $\mathbf{S}^{(k)} \in \mathcal{R}^{h\times w}$, which serves as prior information to guide the UNet decoder. As shown in Fig. 2, we concatenate the voting score map with the encoder features from two encoders, which are delivered to the next decoder stage.

Note that our style voting module is flexible in the weighting scheme. Besides $\boldsymbol{W}^{(k),msk}$, we can design other types of weights assigned to the voters, to select the voters which satisfy the expected property. Next, we will describe how to introduce auxiliary semantic information into the weighting scheme.

3.2.1 Semantic Guided Voting

Intuitively, the objects of similar semantic categories are prone to share similar color or illumination characteristics. Hence, it would be easier to judge whether an object is harmonious or inharmonious by comparing it with other objects of similar semantic categories. For example, given an synthetic image with multiple zebras, we could tell which zebra is a composite foreground cut from another image by comparing each zebra with other zebras. Therefore, we explore the effect of bringing the semantic information into our voting process. When a voter is voting for a pixel, we assign a higher weight to this voter if they are semantically similar and a lower weight otherwise.

To achieve this goal, we feed the input image to a pretrained semantic segmentation network OCRNet [37] and extract the feature map from the last layer as the semantic feature map $\boldsymbol{F}^{sem}$. After resizing $\boldsymbol{F}^{sem}$ to $h \times w$, we calculate the semantic similarity matrix $\boldsymbol{W}^{sem} \in \mathcal{R}^{(h\times w)\times(h\times w)}$. Each entry W_{p_1,p_2}^{sem} is the cosines similarity between each pair of pixel-level semantic features, in a similar way to (3). We multiply $\boldsymbol{W}^{sem}$ with $\boldsymbol{W}^{(k),msk}$ as the new weight matrix, in which case the weighted total score each pixel receives can be calculated as

$$S_{p_1}^{(k)} = \sum_{p_2} W_{p_1,p_2}^{(k),msk} W_{p_1,p_2}^{sem} V_{p_1,p_2}^{sty}. \tag{5}$$

Then, we can replace $S_{p_1}^{(k)}$ in (4) with that in (5) when constructing the voting score map.

3.3 Loss Function

Following [21, 26], our mask-related loss consists of three parts: 1) the binary cross entropy loss ℓ_{bce}; 2) the structural similarity loss ℓ_{ssim}; 3) the intersection over union loss ℓ_{iou}. Besides the final estimated mask $\hat{\boldsymbol{M}}$, we also supervise the auxiliary masks output from each decoder stage. Recall that we also have the style feature loss ℓ_{sty} to supervise the style encoder (see Section 3.1.1). Therefore, the total loss can be expressed as

$$\mathcal{L} = \ell_{sty} + \ell_{bce+ssim+iou} + \sum_{k=1}^{K} \ell_{bce+ssim+iou}^{(k)}, \tag{6}$$

where $\ell_{bce+ssim+iou}$ is the sum of three mask-related losses. The upperscript (k) indicates the output mask from the k-th decoder stage, and the number K of decoder stages is 3.

4 Experiments

4.1 Experimental Setting

We conduct our experiments on the image harmonization dataset iHarmony4 [8] following [21]. This dataset contains inharmonious-harmonious image pairs and the corresponding inharmonious region masks. The iHarmony4 dataset [8] is comprised of four sub-datasets: HAdobe5K, HCOCO, HFlickr, and HDay2Night. Following [21], we only choose image pairs with area of inharmonious region smaller than 50% of the whole image to avoid ambiguity, resulting in 64255 training images and 7237 testing images.

Our model is implemented based on the Pytorch framework. Our optimizer is Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay=1e-4. We use the cosineannealing learning rate scheduler. Our model is trained for 250 epochs in total on 4 GeForce GTX TITAN X GPUs with batch size 24.

We adopt the same evaluation metrics including Average Precision (AP), F_1 score, and Intersection over Union (IoU) following [21].

4.2 Comparison with the State-of-the-art

Besides DIRL [21] and MadisNet [22] working on inharmonious region localization, we also choose popular methods from three closely related fields for comparison. The first group is semantic segmentation networks including UNet [28], DeepLabv3 [7], HRNet-OCR [37], SegFormer [35]. The second group is image manipulation localization methods including MantraNet [34], MAGritte [19], SPAN [17]. The third group is salient object detection methods including F3Net [32], GATENet [38], MINet [24]. For fair comparison, we also use ResNet34 for ResNet-based models. We choose HRNet30 for HRNet-OCR and SegFormer-B3 for SegFormer for comparable model size.

4.2.1 Quantitative Comparison

Evaluation results of all methods are listed in Table 1. We can see that our method achieves the best overall results and outperforms the strongest baseline MadisNet [22] by a large margin. We also report the detailed results on four subdatasets, based on which our improvement mainly comes from HCOCO and Hday2night. We will report the comparison of computational complexity in the Supplementary.

4.2.2 Qualitative Comparison

To visually compare our method with others, we show some visualization results of our method and the well-behaved baselines in Fig. 4. It can be seen that our method can successfully localize the inharmonious region, even in some challenging cases. More results and analyses can be found in the Supplementary.

4.3 Ablation Studies

In this section, we conduct comprehensive ablation studies to verify the effectiveness of our design, which are summarized in Table 2. The first row only contains a simple encoder-decoder branch with RGB image as input. In row 2, we change the input image to YUV color space and observe performance improvement, which shows that suitable color space is

Method	All			HCOCO			HAdobe5k			HFlickr			Hday2night		
	AP	F1	IoU	AP	F1	IoU	AP	F1	IoU	AP	F1	IoU	AP	F1	IoU
UNet	74.90	0.6717	64.74	68.11	0.5869	56.57	89.26	0.8380	80.85	80.72	0.7683	74.58	35.74	0.2362	19.60
DeepLabv3	75.69	0.6902	66.01	69.09	0.6070	58.21	90.20	0.8591	81.56	80.01	0.7698	74.91	35.87	0.2550	21.38
HRNet-OCR	75.33	0.6765	65.49	68.89	0.5981	57.69	89.63	0.8387	80.98	79.62	0.7489	74.55	34.98	0.2477	21.34
SegFormer	78.05	0.7249	66.55	72.46	0.6578	58.78	89.43	0.8531	80.44	85.19	0.7986	75.02	45.16	0.3856	32.75
MantraNet	64.22	0.5691	50.31	56.55	0.4811	41.04	81.07	0.7510	68.50	67.52	0.6302	58.51	28.88	0.2019	16.71
MAGritte	71.16	0.6907	60.14	64.75	0.6058	51.77	85.50	0.8630	76.36	75.02	0.7725	70.25	31.20	0.2549	17.05
SPAN	65.94	0.5850	54.27	58.41	0.4906	45.07	82.57	0.7786	72.49	69.22	0.6510	62.20	29.58	0.2171	19.41
F3Net	61.46	0.5506	47.48	54.17	0.4703	40.03	74.31	0.6944	60.08	72.53	0.6582	59.31	30.08	0.2563	20.83
GATENet	62.43	0.5296	46.33	55.07	0.4568	38.89	75.19	0.6634	59.18	74.13	0.6256	57.51	30.98	0.2174	19.38
MINet	77.51	0.6822	63.04	71.74	0.6022	55.79	89.58	0.8379	77.23	83.86	0.7761	72.51	37.82	0.2710	19.38
DIRL	80.02	0.7317	67.85	74.25	0.6701	60.85	92.16	0.8801	84.02	84.21	0.7786	73.21	38.74	0.2396	20.11
MadisNet(UNet)	81.15	0.7372	67.28	79.02	0.7108	63.31	88.31	0.8219	77.41	79.24	0.7182	68.12	49.60	0.3851	32.52
MadisNet(DIRL)	85.86	0.8022	74.44	83.78	0.7741	70.50	**92.45**	**0.8850**	**84.75**	85.65	**0.8032**	**75.49**	57.40	0.4672	40.47
AustNet	92.20	0.8453	79.63	95.11	0.8866	83.30	89.01	0.8047	76.55	87.72	0.7777	72.61	74.01	0.5554	51.31
AustNet-S	**93.01**	**0.8571**	**80.96**	**95.92**	**0.8963**	**84.61**	89.38	0.8113	76.93	**88.21**	0.8012	75.16	**84.10**	**0.6438**	**60.47**

Table 1: Quantitative comparison with other methods on the iHarmony4 dataset including detailed results on each sub-dataset. All metrics are the larger, the better. The best method is marked in bold and the second best method is marked with underline.

#	Components					Evaluaion Metrics		
	Input	Color-mapping	Voting	ℓ_{sty}	Semantic	AP	F1	IoU
1	RGB					73.45	0.6330	56.76
2	YUV					76.28	0.6511	59.24
3	RGB+YUV					76.45	0.6573	59.73
4	RGB+YUV		only aux mask			77.32	0.6569	59.81
5	RGB+RGB		✓	✓		75.57	0.6685	60.24
6	RGB+YUV		✓	✓		79.10	0.6986	64.37
7	RGB+YUV		✓	✓	✓	79.56	0.7242	66.42
8	RGB+YUV	✓				86.17	0.7741	70.61
9	RGB+YUV	✓		✓		86.97	0.7826	71.98
10	RGB+RGB	✓		✓		76.78	0.6599	59.65
11	RGB+YUV	✓	✓	✓		92.01	0.8477	78.78
12	RGB+YUV	✓	✓	✓	✓	**93.01**	**0.8600**	**81.14**

Table 2: Ablation study of key components in our method.

beneficial for the inharmonious region localization task. For row 3, we simply add a style encoder without color mapping module or style feature loss, and concatenate the multi-scale encoder features from two encoders as in our method. This leads to minor improvement compared with row 2, implying that the style feature should be utilized in a better way.

Based on row 3, we add our style voting module in row 6, the performance gain proves the effectiveness of our style voting module. Adding semantic information in row 7 further boosts the result. We change the YUV color space in row 6 to RGB, and the obtained results in row 5 indicate the advantage of YUV color space. To ensure that the major improvement is not brought by predicting auxiliary masks in each decoder stage, we experiment to only predict auxiliary masks without style voting module in row 4.

In row 8-12, we conduct experiments with color mapping module. From row 8, we see that our color mapping module significantly advances the performance and adding style feature loss in row 9 brings further improvement. In row 10, we replace the input YUV image with RGB image and the performance drops sharply. The results in row 11 and 12 demonstrate the effectiveness of (semantic-guided) style voting map.

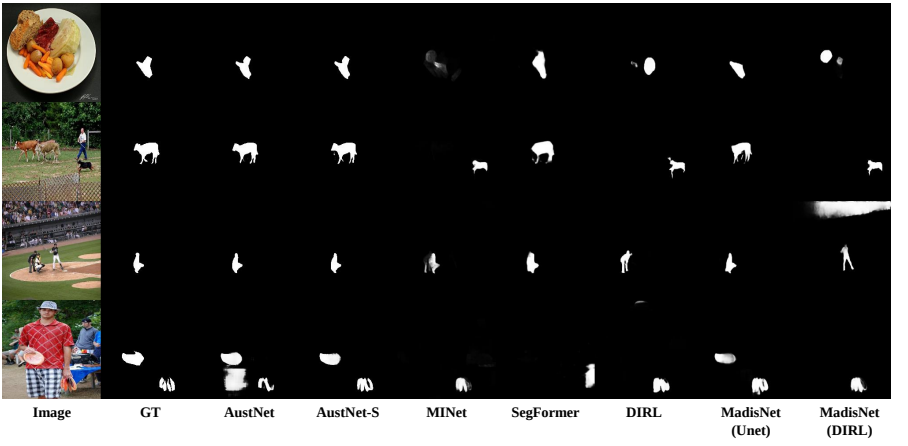

Figure 4: Qualitative comparison with baseline methods. GT is ground-truth mask.

4.4 Experiments on Multiple Inharmonious Regions

Images in the iHarmony4 [8] dataset mainly contain a single inharmonious region, but in real-life scenario, it is possible that there are several separate inharmonious regions in one image and each inharmonious region may also be different in terms of color and illumination. To investigate the ability of our model to detect multiple inharmonious regions, we build a set of test images with multiple disjoint inharmonious regions based on the HCOCO subset of iHarmony4. Specifically, real images in HCOCO may have different inharmonious image pairs with different manipulated foregrounds. Thus, we combine these inharmonious images corresponding to a single real image to construct a test set with multiple inharmonious regions. This test set contains 19482 images in total, with the number of inharmonious regions ranging from 2 to 9. We compare our AustNet and AustNet-S with the strongest baseline MadisNet [22] on this test set. The detailed quantitative results and visualization results are left to the Supplementary.

5 Conclusions

In this work, we focus on the essence of inharmonious region localization task and extract discriminative style features. Centering around the style features, we have proposed a novel color mapping module and a novel style voting model to help localize the inharmonious region. We have also verified the effectiveness of utilizing semantic information in the voting process. Our method significantly outperforms the existing methods.

Acknowledgement

The work was supported by the Shanghai Municipal Science and Technology Major/Key Project, China (2021SHZDZX0102, 20511100300) and National Natural Science Foundation of China (Grant No. 61902247).

References

[1] Zhongyun Bao, Chengjiang Long, Gang Fu, Daquan Liu, Yuanzhen Li, Jiaming Wu, and Chunxia Xiao. Deep image-based illumination harmonization. In *CVPR*, 2022.

[2] Jawadul H. Bappy, Amit K. Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and B.S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *ICCV*, 2017.

[3] 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.

[4] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7 (3):1003–1017, 2012.

[5] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *ICASSP*, 2011.

[6] Junyan Cao, Wenyan Cong, Li Niu, Jianfu Zhang, and Liqing Zhang. Deep image harmonization by bridging the reality gap. In *BMVC*, 2022.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[8] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *CVPR*, 2020.

[9] Wenyan Cong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. Bargainnet: Background-guided domain translation for image harmonization. In *ICME*, 2021.

[10] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. In *CVPR*, 2022.

[11] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Transactions on Image Processing*, 29: 4759–4771, 2020.

[12] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118, 2017.

[13] Zonghui Guo, Dongsheng Guo, Haiyong Zheng, Zhaorui Gu, Bing Zheng, and Junyu Dong. Image harmonization with transformer. In *ICCV*, 2021.

[14] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *CVPR*, 2021.

[15] Yucheng Hang, Bin Xia, Wenming Yang, and Qingmin Liao. Scs-co: Self-consistent style contrastive learning for image harmonization. In *CVPR*, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *ECCV*, 2020.

[18] Kotaro Kikuchi, Kota Yamaguchi, Edgar Simo-Serra, and Tetsunori Kobayashi. Regularized adversarial training for single-shot virtual try-on. In *ICCV Workshops*, 2019.

[19] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. In *NeurIPS*, 2019.

[20] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821–1829, 2009.

[21] Jing Liang, Li Niu, and Liqing Zhang. Inharmonious region localization. In *ICME*, 2021.

[22] Jing Liang, Li Niu, Penghao Wu, Fengjun Guo, and Teng Long. Inharmonious region localization by magnifying domain discrepancy. In *AAAI*, 2022.

[23] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009.

[24] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, 2020.

[25] Chi-Man Pun, Bo Liu, and Xiaochen Yuan. Multi-scale noise estimation for image splicing forgery detection. *J. Vis. Commun. Image Represent.*, 38:195–206, 2016.

[26] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019.

[27] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *WIFS*, 2016.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[29] Paolo Rota, Enver Sangineto, Valentina Conotter, and Christopher Pramerdorfer. Bad teacher or unruly student: Can deep learning say something in image forensics analysis? In *ICPR*, 2016.

[30] Konstantin Sofiiuk, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. In *WACV*, 2021.

[31] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017.

[32] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: Fusion, feedback and focus for salient object detection. In *AAAI*, 2020.

[33] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, 2019.

[34] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, 2019.

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 2021.

[36] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao. Constrained r-cnn: A general image manipulation detection model. In *ICME*, 2020.

[37] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020.

[38] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, 2020.

[39] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *CVPR*, 2018.