PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization

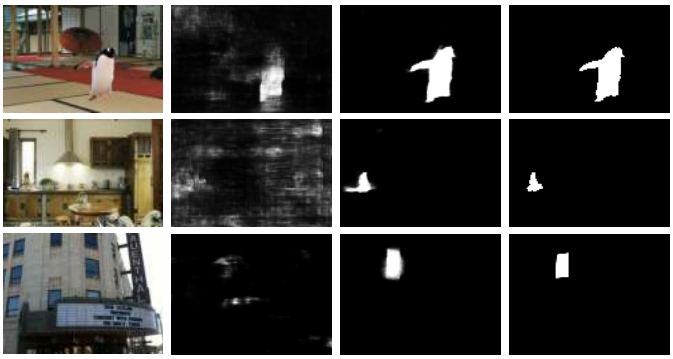
Xiaohong Liu, Yaojie Liu, Jun Chen, *Senior Member, IEEE*, and Xiaoming Liu, *Senior Member, IEEE*

***Abstract*— To defend against manipulation of image content, such as splicing, copy-move, and removal, we develop a Progressive Spatio-Channel Correlation Network (PSCC-Net) to detect and localize image manipulations. PSCC-Net processes the image in a two-path procedure: a top-down path that extracts local and global features and a bottom-up path that detects whether the input image is manipulated, and estimates its manipulation masks at multiple scales, where each mask is conditioned on the previous one. Different from the conventional encoder-decoder and no-pooling structures, PSCC-Net leverages features at different scales with dense cross-connections to produce manipulation masks in a coarse-to-fine fashion. Moreover, a Spatio-Channel Correlation Module (SCCM) captures both spatial and channel-wise correlations in the bottom-up path, which endows features with holistic cues, enabling the network to cope with a wide range of manipulation attacks. Thanks to the light-weight backbone and progressive mechanism, PSCC-Net can process 1, 080P images at 50+ FPS. Extensive experiments demonstrate the superiority of PSCC-Net over the state-of-the-art methods on both detection and localization. Codes and models are available at https://github.com/proteus1991/PSCC-Net.**

***Index Terms*— Image manipulation detection and localization, progressive mechanism, attention mechanism.**

I. Introduction

SEEING *is believing?* Not anymore. Recent advances on image manipulation techniques [1]–[4] enable easy editing of raw images, such as removing unwanted objects [5]–[8], face swapping [2], attribute changing [9], *etc*. Although such techniques are neutral, malicious attackers may utilize them to create deceitful content to propagate false information, *e.g.*, fake news [10], insurance fraud [11], and Deepfake [12]–[15]. Thus, concerns of the adverse impact on social media and even real-world systems have been raised [16], [17]. To alleviate the concerns, it is crucial to develop reliable models to expose the manipulated images. While being used in machine and systems, the model is required to, at a minimal, distinguish manipulated images from pristine ones, where the objective is to *detect*. While being used for human's viewing, the model is further required to estimate tampered areas in forged images, where the objective is to *localize*.



(a) Manipulated (b) SPAN [18] (c) PSCC-Net (d) GT

Fig. 1. Examples of image manipulation localization. Three examples are splicing, copy-move, and removal manipulations respectively. With novel designs of progressive mechanism and correlation module, our method demonstrates robust and accurate estimation at different scales and types.

Generally, image manipulation consists of the content-dependent process and content-independent process. The former includes splicing, copy-move, and removal, as shown in Fig. 1. Both splicing and copy-move are content-copying forgeries, where the splicing content is from a different donor image while the copy-move content is from the target image *per se*. Removal takes out certain objects from the target image and performs refilling via inpainting. Often, the content-dependent process follows the semantic arrangement in the target image, *e.g.*, placing a car on the road and replacing one face with another, which makes the resulting image visually "authentic" and indistinguishable from the pristine one. However, based on image/camera trace analysis [19], [20], subtle patterns can still be revealed to indicate the manipulation. On the other hand, the content-independent process includes global modifications such as brightness/contrast change, blurring, noising and image compression. They barely create any disinformation, but their resultant noise may undermine the analysis of image/camera traces and potentially hide the discrepancy between the manipulated and pristine areas.

Manuscript received 20 December 2021; revised 16 May 2022 and 22 June 2022; accepted 30 June 2022. Date of publication 7 July 2022; date of current version 28 October 2022. This work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Agreement HR00112090131. This work was conducted when Xiaohong Liu was a Visiting Scholar at Michigan State University. This article was recommended by Associate Editor X. Li. *(Corresponding author: Xiaoming Liu.)*

Xiaohong Liu was with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada. He is now with John Hopcroft Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiaohongliu@sjtu.edu.cn).

Yaojie Liu was with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA. He is now with Google Research, Mountain View, CA 94043 USA (e-mail: liuyaoj1@msu.edu).

Jun Chen is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada (e-mail: chenjun@mcmaster.ca).

Xiaoming Liu is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: liuxm@msu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3189545.

Digital Object Identifier 10.1109/TCSVT.2022.3189545

To defend against manipulations, many image manipulation detection and localization (IMDL) methods have been proposed in the past. In the early stages, methods are designed to handle a single type of manipulation. In recent years, works [11], [13], [18], [21]–[26] are proposed to build generic IMDL models for *multiple* manipulation types. However, there are still 3 major unsolved problems for IMDL:

A. Scale Variation

The forged area varies in sizes. Most prior works neglect the importance of scale variations and encounter difficulty when detecting forged areas of different sizes. Both the conventional encoder-decoder [23], [24] and no-pooling [11], [18] structures have difficulties in leveraging local and global features jointly, thus can only handle a limited scale variation.

B. Image Correlation

Manipulated regions can best be determined when compared to pristine regions, especially for splicing attacks. A naive learning of mapping from the manipulated image to manipulation mask may lead to an overfitting to the specific attack type in training. In contrast, considering the image spatial correlation can lead to a more generalized localization solution. Yet, such correlation is mostly neglected in prior works.

C. Detection

In principle, manipulation detection and localization are highly relevant tasks, where the detection score can be simply derived from the response of the predicted manipulation mask, *i.e.*, at least one part of the forged image has high response while no part of the pristine one does. However, most prior works assume the *existence of manipulation* in all input images. As a result, this could cause many false alarms on pristine images and make the detection unreliable.

To address the above issues, we propose a novel Progressive Spatio-Channel Correlation Network (PSCC-Net), as in Fig. 2. PSCC-Net consists of a top-down path and a bottom-up path. In the top-down path, a backbone encoder first extracts the local and global features from an input image. We adopt the network structure of [27] as our encoder, whose dense connections among different scales facilitate information exchange. In the bottom-up path, we leverage the learned features to estimate 4 manipulation masks from small scales to large ones, where each mask serves as a prior in the next-scale estimation. Thanks to such a design, the final mask is estimated in a coarse-to-fine fashion, harvesting both the local and global information. This design enables a potential speed-up by terminating the bottom-up mask estimation, if the intermediate mask is satisfactory. Moreover, rather than investigating the response of predicted manipulation masks, we feed the learned features into a detection head to produce the score for binary classification.

To exploit image correlation, we propose a Spatio-Channel Correlation Module (SCCM) that grasps both spatial and channel-wise correlations at each bottom-up step. The spatial correlation aggregates the global context among local features. As the response from different channels might be associated with the same class (*e.g.*, manipulated or pristine), the channel-wise correlation computes the similarity among feature maps to enhance the representation in interest areas. Given the light-weight design of the encoder, PSCC-Net can process $1,080$P at $50+$ FPS. Our proposed approach demonstrates a superior manipulation localization on several benchmarks. In addition, we show that the recent IMDL methods encounter difficulty in distinguishing manipulated images from pristine ones. By explicitly introducing a detection head, our method achieves the state of the art (SOTA) on manipulation detection.

We summarize the contributions of this work as follows:

- A new PSCC-Net is proposed that performs favorably on manipulation detection and enables progressive improvement of manipulation localization in a coarse-to-fine fashion;
- A novel SCCM module is designed to capture the spatial and channel-wise correlations for better generalization. SCCM avoids the use of massive annotated data to pre-train our feature extractor;
- The SOTA results for both image manipulation detection and localization are successively achieved.

II. Related Work

A. Image Manipulation Detection

Image manipulation detection aims to distinguish manipulated images from pristine ones via image-level binary classification. There are two major approaches for this detection: the implicit manner [10], [28] and the explicit manner [29]. The former obtains the detection score by the statistics (*e.g.*, average [10] or maximum [28] value) of the predicted manipulation mask, and the latter explicitly outputs the score from a dedicated classification module. Recent works [11], [18] focus on pixel-level manipulation localization but neglect the importance of image-level detection. Instead, this work leverages both manipulated and pristine images in training and jointly considers detection and localization of image manipulation.

B. Image Manipulation Localization

Early works propose to localize the manipulation of one specific type, *e.g.,* splicing [10], [19], [30]–[36], copy-move [28], [29], [37]–[40], removal [41]–[44], and the content-preserved process [24], [45]. Although most methods perform well on detecting that specific forgery type, they fall short in handling real-world cases, where usually the forgery type is unknown in advance and various types of forgery might be utilized in manipulation. In the related problem of face anti-spoofing, researchers also study how to localize the facial pixels covered with various spoof mediums [46].

Recent works attempt to tackle multiple forgeries in one model. J-LSTM [21] and H-LSTM [24] integrate the LSTM and CNN to capture the boundary-discriminative features. However, due to the patch-based design, both methods are time-consuming, and the size of detectable regions is limited by the preset patch size. RGB-N [23] adopts the steganalysis rich model [47] and Faster R-CNN [48], but it can only

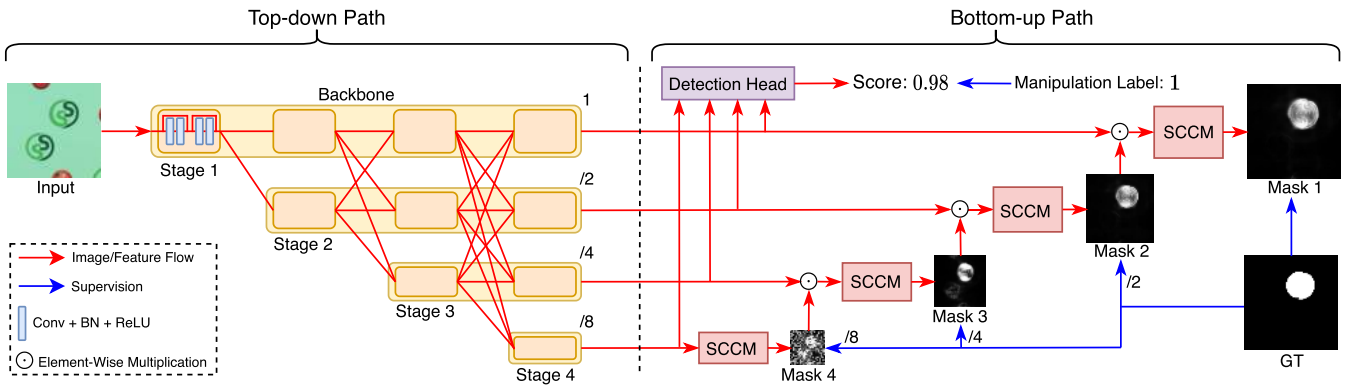


Fig. 2. The architecture of the proposed PSCC-Net. The detection score predicted by the detection head indicates if the input is manipulated or not. The accuracy of manipulation localization from *Mask 4* to *Mask 1* is gradually improved, *e.g.*, the prediction of *Mask 4* confuses the pasted (forged) region with the pristine (copied) one, while *Mask 1* effectively fixes it.

provide bounding boxes instead of segmentation masks. Later, ManTra-Net [11] learns features to distinguish 385 known manipulation types and treats the problem as anomaly detection. To learn the distinguishable features, auxiliary labeled data, such as camera sensors, are used. SPAN [18] extends ManTra-Net to further model the spatial correlation via local self-attention blocks and pyramid propagation. However, as the correlation is only considered in the local region, ManTra-Net and SPAN fail to take full advantage of the spatial correlation and consequently have limited generalizability. In this work, our PSCC-Net utilizes a progressive mechanism to improve the multi-scale feature representation and SCCM modules to better explore spatial and channel-wise correlations.

C. Progressive Mechanism

Progressive mechanism tackles a challenging task in a coarse-to-fine fashion. It has been widely adopted in many low-level and high-level vision tasks, such as denoising [5], [49], inpainting [50], super-resolution [51], [52], and object detection [53]–[56]. The pyramid structure is commonly utilized to build multi-scale features. In this work, we propose a densely connected pyramid structure that progressively refines the manipulation mask from small scales to large ones, where each predicted mask serves as a prior for the next-scale estimation.

D. Attention Mechanism

The pioneer work [57] proposes an attention mechanism to improve the feature representation with relatively low cost, which has been widely employed in various vision tasks [13], [29], [58]–[62]. According to the applied domain, the attention mechanism can be divided into two types: spatial attention [59] and channel-wise attention [58]. Recent works [63]–[65] take the benefit of both types to further improve the representation capability of DNN. These methods adopt separate schemes to explore the spatial and channel-wise attentions and thus require additional efforts to fuse them. In addition, due to memory limit, they can only apply to high-level features where the spatial size is small. In this work, a unified SCCM jointly explores the image correlation and discrepancy in both spatial domain and feature channels on the same features. Besides, owing to the dimensional reduction design, SCCM is able to adapt both low-level and high-level features with arbitrary sizes.

III. PSCC-Net

Our PSCC-Net enables the detection and localization of various types of manipulations. As compared to the image-level detection, the pixel-level localization is more difficult. Therefore, PSCC-Net pays special attention to tackling the localization problem. Indeed, since the features for detection and localization are jointly learned, improving the localization performance will naturally benefit detection.

A. Network Architecture

1) Top-Down Path: Most prior works use the conventional encoder-decoder [23], [24] and no-pooling structures [11], [18] to extract features. Since forged areas have various sizes, it is important to fuse local and global features to handle the scale variation. However, both structures extract features in a sequential pipeline and neglect feature fusion among different scales, and thus can only handle a limited scale variation. To address this issue, we adopt a light-weight backbone in [27], named HRNetV2p-W18. Following its default setting, the stage down-scaling ratio s is set to 2, and there are totally 4 stages.

Compared to encoder-decoder and no-pooling structures, the benefits of our backbone are two-fold. First, features from different scales are computed in parallel. Hence, dense connections among different scales enable effective information exchange, which is beneficial for handling scale variations. Second, since the local and global feature fusion is performed for every scale, each feature contains sufficient information to predict a manipulation mask at the corresponding scale. Therefore, this backbone is in line with our progressive mechanism, where the prediction of each mask should rely on all local and global features to improve its accuracy. Indeed, except the predicted mask on the last scale, the others serve as a prior for the next-scale mask prediction. After the top-down path, the manipulated features on 4 scales are extracted. Then,

we use the bottom-up path to perform manipulation detection and localization.

2) Bottom-Up Path: The bottom-up path in PSCC-Net estimates the detection score and the manipulation mask. Specifically, the detection score is predicted based on the extracted features from the top-down-path via a detection head [27], then the manipulation mask is generated through a progressive mechanism with full supervision. In particular, the coarse-to-fine progressive mechanism mimics how human tackles complicated problems in daily life.

We denote the input image as $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$. The extracted features at 4 scales are $\mathbf{F}_1 \in \mathbb{R}^{H\times W\times C}$, $\mathbf{F}_2 \in \mathbb{R}^{H/s\times W/s\times sC}$, $\mathbf{F}_3 \in \mathbb{R}^{H/s^2\times W/s^2\times s^2C}$ and $\mathbf{F}_4 \in \mathbb{R}^{H/s^3\times W/s^3\times s^3C}$, and their corresponding masks are denoted as $\mathbf{M}_1 \in \mathbb{R}^{H\times W}$, $\mathbf{M}_2 \in \mathbb{R}^{H/s\times W/s}$, $\mathbf{M}_3 \in \mathbb{R}^{H/s^2\times W/s^2}$ and $\mathbf{M}_4 \in \mathbb{R}^{H/s^3\times W/s^3}$. Here H, W, and C are the height, width, and channel number of the image/feature respectively. Formally, we have

$$\mathbf{M}_{n-1} = f_{n-1}(\tau(\mathbf{M}_n)\cdot\mathbf{F}_{n-1}), \quad n = 2, 3, 4, \tag{1}$$

where f_n denotes the SCCM on the nth scale, and τ is the upsampling operation (*e.g.*, the bilinear interpolation). Since $\mathbf{M}_4$ is the mask on the last scale, it can be directly expressed as $\mathbf{M}_4 = f_4(\mathbf{F}_4)$. For Scales 1-3, the feature on the current scale is associated with the upsampled mask from the previous scale for feature modulation. Then, the modulated feature is fed into SCCM to produce a manipulation mask.

To reduce the prediction difficulty, the proposed progressive mechanism avoids generating the mask at the finest scale directly. Instead, the mask on the coarsest scale is first predicted to locate the regions that are potentially forged based on current available information. The subsequent prediction on the finer scale can leverage the previous mask and pay more attention to those selected regions. This process continues until the generation of the manipulation mask at the finest scale, which serves as the final prediction. However, without explicit supervision on each scale, the intermediate masks might not follow the coarse-to-fine order. Therefore, full supervisions are applied on all scales to guide the mask estimation.

B. Spatio-Channel Correlation Module

Attention mechanisms are commonly used to modulate learned features according to their relative significance. As the final manipulation mask is binary, the localization can be considered as a pixel-level binary classification. Ideally, we expect the learned features on forged regions are similar to each other but distinct from those in pristine regions. In this case, a fundamental clustering method may suffice to produce an effective mask. Therefore, to better tackle manipulation localization, we propose a SCCM that employs the spatial attention to aggregate the pixel-level features based on their contextual correlations, and the channel-wise attention to consolidate the feature maps based on their channel correlations.

We illustrate the detailed structure of SCCM in Fig. 3, where the input feature $\mathbf{X}$ is of size $H \times W \times C$. Note that even though $\mathbf{X}$ is small (256×256), the size of its spatial correlation can be enormous ($65,536 \times 65,536$), easily exceeding the memory limit. Therefore, we use function h to reshape the

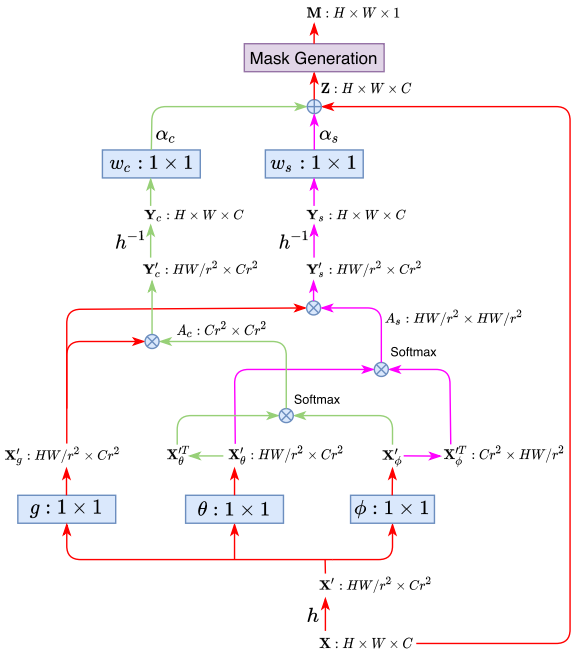


Fig. 3. The structure of SCCM. Here $\otimes$ represents the matrix multiplication and $\oplus$ the element-wise addition; the red arrow shows the common feature flows; the pink and green arrows show the feature flows of spatial and channel-wise attentions respectively.

input $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ to $\mathbf{X}' \in \mathbb{R}^{HW/r^2\times Cr^2}$, where each feature map is flattened to form a vector based on SCCM down-scaling ratio r. For instance, with $r = 4$, the size of spatial correlation is $4,096 \times 4,096$ instead of $65,536 \times 65,536$. Therefore, this operation preserves all feature information and avoids modeling the spatial correlation of potentially large size $HW \times HW$.

To build the spatial and channel-wise correlations, one may directly leverage $\mathbf{X}'$. However, additional flexibility could be achieved by introducing the embedded Gaussian function [59]. Therefore, we use the 1×1 convolution to build different functions g, θ, and ϕ to transform $\mathbf{X}'$ into new linear embeddings as $\mathbf{X}'_g = g(\mathbf{X}')$, $\mathbf{X}'_\theta = \theta(\mathbf{X}')$, and $\mathbf{X}'_\phi = \phi(\mathbf{X}')$, all with the same size as $\mathbf{X}'$. Subsequently, the spatial and channel-wise correlations (denoted as $\mathbf{A}_s \in \mathbb{R}^{HW/r^2\times HW/r^2}$ and $\mathbf{A}_c \in \mathbb{R}^{Cr^2\times Cr^2}$) of embedded features $\mathbf{X}'_\theta$ and $\mathbf{X}'_\phi$ are computed, and the Gaussian operation is implemented by Softmax function. In the end, the spatial and channel-wise attentions are realized by performing matrix multiplications $\mathbf{A}_s\mathbf{X}'_g$ and $\mathbf{X}'_g\mathbf{A}_c$, respectively. Unlike prior methods [63]–[65] that employ two attentions on *different* features, we apply both to the *same* linear embedding for mutual accommodation. Indeed, applying attentions in this way reduces the difficulty of subsequent fusion process, and also saves computational operations in SCCM. Specifically, the spatial attention can be formulated as:

$$\mathbf{Y}'_s = \mathbf{A}_s\mathbf{X}'_g = \text{softmax}(\mathbf{X}'_\theta\mathbf{X}'^T_\phi)\mathbf{X}'_g, \tag{2}$$

where $\mathbf{Y}'_s \in \mathbb{R}^{HW/r^2\times Cr^2}$ is the feature resulting from the application of spatial attention, and softmax$(\cdot)$ denotes the Softmax function. The element (i, j) in $\mathbf{A}_s$ indicates the similarity between the feature vectors in the ith row of $\mathbf{X}'_\theta$ and jth row of $\mathbf{X}'_\phi$. The more similar they are, the higher

correlation they have. This helps the network to learn feature representations for distinguishing forged regions from pristine ones and avoid overfitting to a specific attack type in training. Similarly, the channel-wise attention is expressed as:

$$\mathbf{Y}'_c = \mathbf{X}'_g \mathbf{A}_c = \mathbf{X}'_g \text{softmax}(\mathbf{X}'^T_\theta \mathbf{X}'_\phi), \tag{3}$$

where $\mathbf{Y}'_c \in \mathbb{R}^{HW/r^2 \times Cr^2}$ is the feature resulting from the application of channel-wise attention. The element (i, j) in $\mathbf{A}_c$ measures the similarity between the channel maps in the ith column of $\mathbf{X}'_\theta$ and jth column of $\mathbf{X}'_\phi$. Since the response from different channels might be associated with the same class, *e.g.*, manipulated or pristine, the channel-wise correlation aggregates feature maps based on their similarities to enhance the representation in forged regions.

We use h^{-1} to reshape $\mathbf{Y}'_s$ and $\mathbf{Y}'_c$ respectively back to $\mathbf{Y}_s$ and $\mathbf{Y}_c$ of size $H \times W \times C$. Further, two functions ω_s and ω_c are built by 1×1 convolution to improve their feature representations. The output features from ω_s and ω_c are complement to each other. As it is non-trivial to determine their relative significance, two learnable parameters α_s and α_c, both initialized as 1, are used for trade-off. The learned values of α_s and α_c can be found in supplementary. We also adopt the residual learning [66] to express the feature $\mathbf{Z}$ as:

$$\mathbf{Z} = \mathbf{X} + \alpha_s \cdot \omega_s(\mathbf{Y}_s) + \alpha_c \cdot \omega_c(\mathbf{Y}_c). \tag{4}$$

The final output of SCCM is a predicted mask with only one channel. To reduce the channel number in $\mathbf{Z}$, we employ a mask generation block with the sequential order of *Conv-ReLU-Conv-Sigmoid*, where *Conv* is a 3×3 convolution.

C. Loss Function

To train the PSCC-Net, we adopt the binary cross-entropy loss (L_{bce}) for both detection and localization tasks. The predicted detection score (s_d) is supervised by the ground-truth (GT) label (l_d) with 0 standing for pristine image and 1 for forged image. Moreover, full supervisions are applied on each predicted mask by downsampling the GT mask $\mathbf{G}_1$ to $\mathbf{G}_2$, $\mathbf{G}_3$, and $\mathbf{G}_4$ according to their corresponding sizes, with 0 standing for pristine pixel and 1 for forged pixel. The masks predicted through the progressive mechanism at different scales are considered to be of equal importance. Therefore, our final loss function $\hat{L}$ can be expressed as:

$$\hat{L} = L_{bce}(s_d, l_d) + \frac{1}{4}\sum_{m=1}^{4} L_{bce}(\mathbf{M}_m, \mathbf{G}_m). \tag{5}$$

D. Training Data Synthesis

Since there is no standard IMDL dataset for training, a synthetic dataset is built to train and validate our PSCC-Net. This dataset includes four categories 1) splicing, 2) copy-move, 3) removal, and 4) pristine classes. For splicing, following [34], [67], we use the MS COCO [68] to generate spliced images, where one annotated region is randomly selected per image, and pasted into a different image after several transformations. We adopt the same transformation as [34] including the scale, rotation, shift and luminance changes. Since the spliced region is not necessarily an object, we use the Bezier curve [69] to generate random contours, then fill them to produce splicing masks. We follow the same processes above but randomly select donor and target images in KCMI [70], VISION [71], and Dresden [72] that are commonly used to identify camera source [20], to generate additional spliced images as supplementary. For copy-move, the dataset from [28] is adopted. For removal, inspired by [34], [67], we adopt the SOTA inpainting method [6] to fill one annotated region that is randomly removed from each chosen MS COCO image. As to the pristine class, we simply select images from the MS COCO dataset.

(a) Content-aware splicing (b) Random-mask splicing (c) Copy-move (d) Removal

Fig. 4. Examples from our synthetic dataset. The generated images of different manipulation types and their ground-truth masks are demonstrated.

TABLE I

SUMMARY OF TEST DATASETS FOR OUR PRE-TRAINED AND FINE-TUNED MODELS (# STANDS FOR THE NUMBER OF IMAGES. ✔ AND ✗ INDICATE WHETHER OR NOT THE MANIPULATION TYPE IS INVOLVED)

Dataset	Pre-trained	Fine-tuned		Splicing	Copy-move	Removal
	# Test	# Train	# Test			
Columbia	180	–	–	✔	✗	✗
Coverage	100	75	25	✗	✔	✗
CASIA	6,044	5,123	921	✔	✔	✗
NIST	564	404	160	✔	✔	✔
IMD20	2,010	–	–	✔	✔	✔

In summary, we have 116,583 images in splicing class, 100,000 images in copy-move class, 78,246 images in removal class, and 81,910 images in pristine class, thus ∼0.38M in total. Examples of different manipulation types in our synthetic dataset are demonstrated in Fig 4. It should be emphasized that our training dataset is much smaller than that of MantraNet and SPAN, where massive annotated data (1.25M) is used to train their feature extractor, not to mention the large number of synthesized manipulations for training the rest of their networks.

As it is inefficient to train all manipulated images in one epoch, we uniformly sample 0.025M images per class to form a 0.1M dataset on-the-fly for training in each epoch. In addition, we also build a validation set that contains 4×100 images. The size of synthetic images are all set to 256×256.

IV. EXPERIMENTS

A. Experimental Setup

1) Test Data: We evaluate the manipulation localization on four standard test datasets: Columbia [73], Coverage [38],

CASIA [74] and NIST16 [75], and one real-world dataset: IMD20 [76]. To finetune PSCC-Net, we follow the same training/testing split on Coverage, CASIA, and NIST16 as in [18], [23] for fair comparisons. Specifically, Columbia [73] is a splicing dataset of 180 images. Coverage [38] is a copy-move dataset of 100 images; for fine-tuning, it is split into 75/25 for training and testing. CASIA [74] (v1.0 + v2.0) includes both splicing and copy-move; for fine-tuning, 5,123 images from v2.0 is adopted for training, and 921 images from v1.0 is for testing. NIST16 [75] has 564 images, involving all three manipulations; for fine-tuning, 404 images are used for training and 160 for testing. IMD20 [76] consists of 2,010 real-life manipulated images collected from Internet, and involves all three manipulations as well. We summarize the manipulation types for each test dataset and the number of images for evaluating our pre-trained and fine-tuned models in Tab. I.

As the manipulation detection is not considered by recent works, there is no standard dataset for benchmarking. Since CASIA is the only test dataset in here that corresponds each manipulated image to its pristine image, we use both forged and pristine images and define an evaluation protocol for detection. This dataset is named CASIA-D and consists of 1,842 images with 50% forged and 50% pristine.

2) Metrics: To quantify the localization performance, following previous works [11], [18], we use pixel-level Area Under Curve (AUC) and F1 score on manipulation masks. To evaluate the detection performance, we use image-level AUC and F1 score, Equal Error Rate (EER), and True Positive Rate at 1% false positive rate ($TPR_{1\%}$). Since binary masks and detection scores are required to compute F1 scores, we adopt the EER threshold to binarize them.

3) Implementation Details: PSCC-Net is end-to-end trainable and light-weighted. Its top-down path and bottom-up path have 2.0 and 1.6 Million (M) parameters. In the bottom-up path, the detection head has 0.9 M and the rest part (for localization) has only 0.7 M parameters. In comparison, the ManTra-Net [11] and SPAN [18] have 3.8 and 3.7 M parameters, respectively. Implemented by PyTorch, our model is trained with GeForce GTX 1080Ti. We initialize our backbone with ImageNet pre-trained weights, and optimize the whole model by Adam [77] with a batch size of 10 and an initial learning rate of $2e$-4. The learning rate is halved every 5 epochs and the total training period is 25 epochs.

Our network can take arbitrary-size images as input. To avoid performance degradation caused by size mismatch between training (*e.g.*, 256×256) and testing data (*e.g.*, $4,000 \times 3,000$), at the end of top-down path, we resample the extracted features from the first to the last scales respectively into fixed sizes 256×256, 128×128, 64×64, and 32×32, where the ratio r in SCCM is set to 4, 2, 2, and 1 respectively to reduce the computational burden. The produced masks are resampled back to the same size as the input image for localization evaluation.

B. Comparisons on Localization

The compared IMDL methods include J-LSTM [21], H-LSTM [24], RGB-N [23], ManTra-Net [11], and SPAN [18] where SPAN has reported the SOTA performance on localization. Following the evaluation protocol defined in SPAN [18], we compare the localization performance using two models: 1) the pre-trained model is trained on the synthetic dataset and evaluated on the *full* test datasets, and 2) the fine-tuned model is the pre-trained model further fine-tuned on the training split of test datasets and evaluated on their *test split*. The pre-trained model is to show the generalization ability of each method, and the fine-tuned model is to manifest their localization performance while the domain discrepancy has been greatly alleviated. Note that the reported results of all compared methods are either from their original papers or by running their public codes.

TABLE II
LOCALIZATION AUC (%) OF PRE-TRAINED MODELS

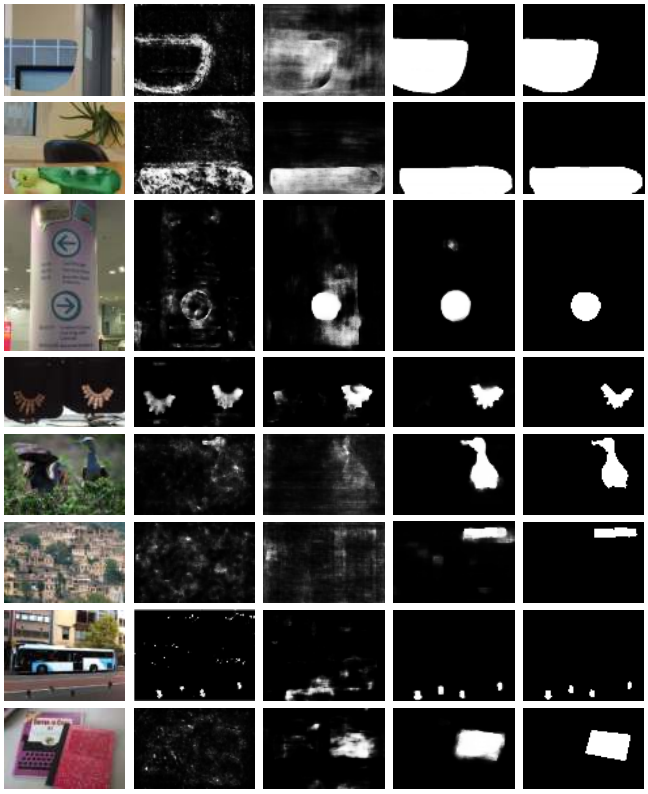
Method	Columbia	Coverage	CASIA	NIST16	IMD20
ManTra-Net [11]	82.4	81.9	81.7	79.5	74.8
SPAN [18]	93.6	**92.2**	79.7	84.0	75.0
PSCC-Net	**98.2**	84.7	**82.9**	**85.5**	**80.6**

TABLE III
EVALUATION OF THE FINE-TUNED MODELS. LOCALIZATION AUC/F1S ARE REPORTED (IN %). MANTRA-NET IS NOT SHOWN HERE AS IT HAS ONLY DEVELOPED THE PRE-TRAINED MODEL

Method	Coverage	CASIA	NIST16
J-LSTM [21]	61.4 / -	- / -	76.4 / -
H-LSTM [24]	71.2 / -	- / -	79.4 / -
RGB-N [23]	81.7 / 43.7	79.5 / 40.8	93.7 / 72.2
SPAN [18]	93.7 / 55.8	83.8 / 38.2	96.1 / 58.2
PSCC-Net	**94.1 / 72.3**	**87.5 / 55.4**	**99.1 / 74.2**

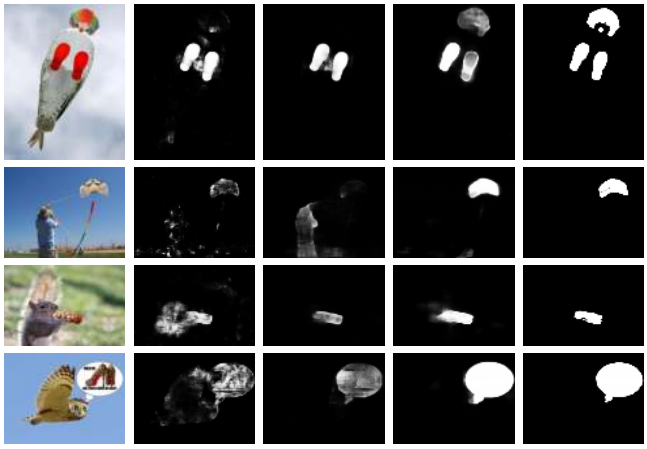
1) Pre-Trained Model: We choose the best pre-trained model based on the performance on our validation set. Tab. II shows the localization performance of pre-trained models for different methods on four standard datasets and one real-world dataset under pixel-level AUC. The pre-trained PSCC-Net achieves the best localization performance on Columbia, CASIA, NIST16, and IMD20, and ranks the second on Coverage. The most significant performance gain is achieved while tackling real-life manipulated images (5.6% ↑). This validates that the PSCC-Net has the best generalization ability as compared to the others. We fail to achieve the best performance on Coverage, despite surpassing ManTra-Net 2.8% under AUC. The reason might be the imperfection of our training data for the case, where the copied object is intentionally moved to cover a pristine object with similar appearance. Indeed, by fine-tuning the pre-trained model on Coverage, PSCC-Net achieves the 0.4% gain over SPAN under AUC (Tab. III).

2) Fine-Tuned Model: The network weights of the pre-trained model are used to initiate the fine-tuned models that will be trained on the *training split* of Coverage, CASIA, and NIST16 datasets, respectively. The training strategy for fine-tuned models is the same as the one for pre-trained model, except setting the initial learning rate to $1e$-4. We evaluate the fine-tuned models of different methods in Tab. III. For AUC, PSCC-Net surpasses baselines in all cases (over 2.4% to SPAN on average). As for F1 score, our model outperforms them



(a) Manipulated (b) ManTra-Net (c) SPAN (d) PSCC-Net (e) GT

Fig. 5. Qualitative localization evaluations on four standard test datasets. From top to bottom, our PSCC-Net is compared to SOTAs on Columbia, Coverage, CASIA, and NIST16 datasets respectively, each with two images.



(a) Manipulated (b) ManTra-Net (c) SPAN (d) PSCC-Net (e) GT

Fig. 6. Qualitative localization evaluations on the IMD20 real-life dataset.

with a large margin (over 16.6% to SPAN on average). This validates the effectiveness of our overall network design.

3) Qualitative Comparisons: We provide qualitative evaluations of manipulation localization on four standard test datasets and one real-life dataset shown in Fig. 5 and Fig. 6, respectively, where the best available model for each method is used to produce manipulation masks. Compared to ManTra-Net [11] and SPAN [18], the predicted masks from our PSCC-Net achieve the best performance in terms of higher prediction accuracy (*e.g.*, the 1*st* row in Fig. 5) and fewer false alarms (*e.g.*, the 6th row in Fig. 5). In addition, the proposed method is less sensitive to the scale variation. Both large (*e.g.*, the 5th row in Fig. 5) and small (*e.g.*, the 7th row in Fig. 5) manipulations can be localized effectively. On the real-life dataset, PSCC-Net still performs much better than the other two (*e.g.*, the 2th row in Fig. 6), which demonstrates its good generalization ability.

TABLE IV

DETECTION EVALUATION ON CASIA-D, ALL REPORTED IN %

Method	AUC ↑	F1 ↑	EER ↓	$TPR_{1\%}$ ↑
ManTra-Net [11]	59.94	56.69	43.21	5.43
SPAN [18]	67.33	63.48	36.47	5.54
PSCC-Net†	74.40	66.88	33.21	28.37
PSCC-Net	**99.65**	**97.12**	**2.83**	**95.65**

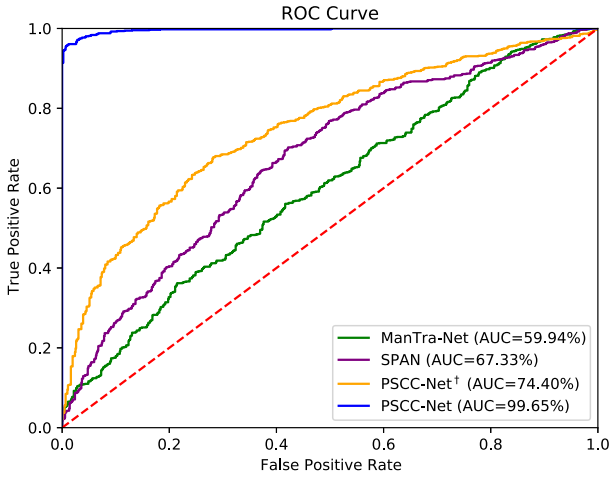
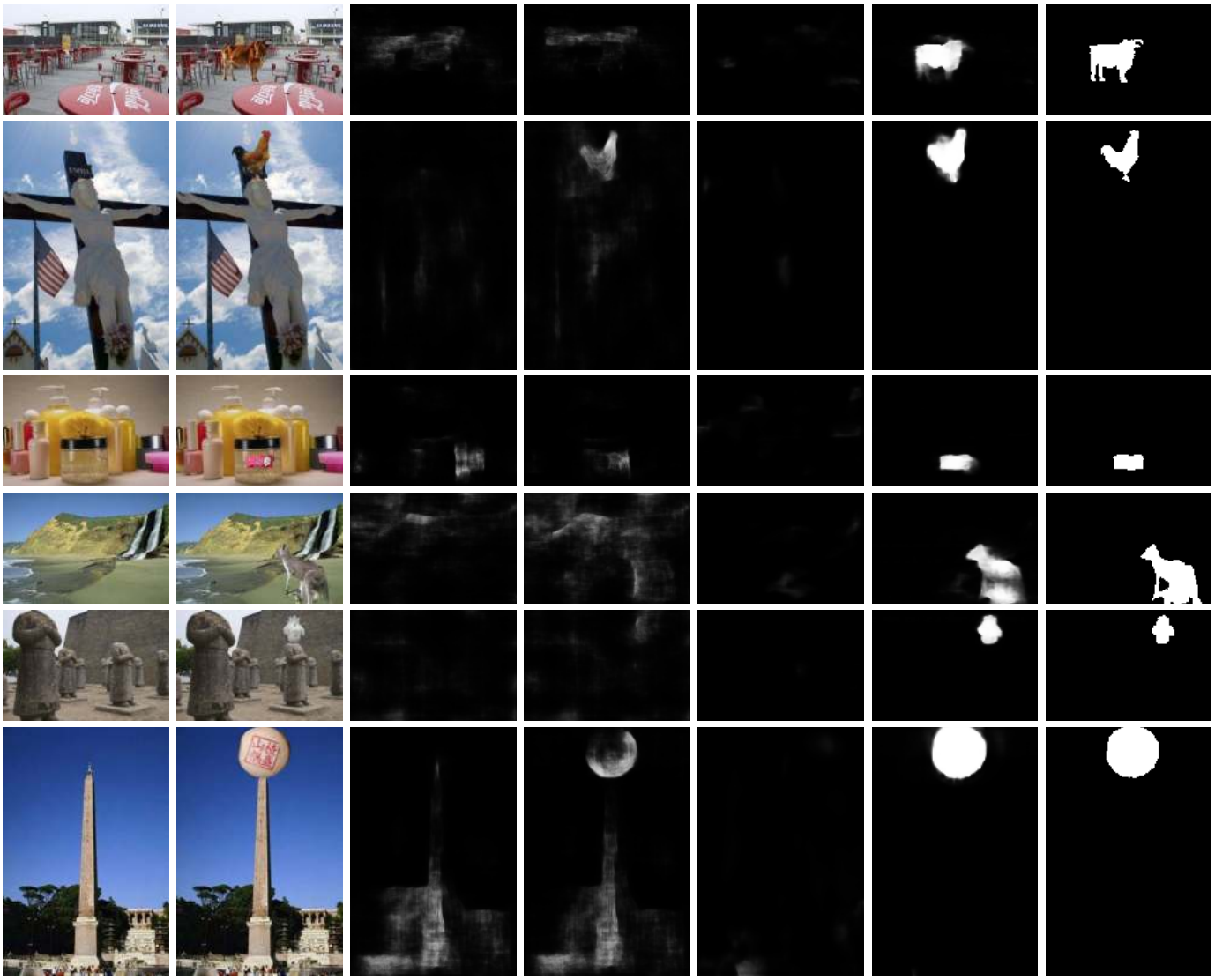


Fig. 7. ROC of different methods for detection. Our detection head successfully alleviates the influence of false alarms in pristine images, thus achieves the best result.

C. Comparisons on Detection

Since ManTra-Net and SPAN are the best performing baselines in the localization evaluation, and ManTra-Net does not develop the fine-tuned model, we choose to use the pre-trained model for detection evaluation, in order to make comparisons to both of them. Although these two baselines make no direct attempt to perform detection, their estimated manipulation masks can be leveraged for this purpose. As such, we simply regard the average of the mask as their scores. For fair comparisons, we build a variant that adopts the same averaging strategy to calculate this score, denoted as PSCC-Net†. In Tab. IV, owing to our well-predicted manipulation masks, the PSCC-Net† achieves the best detection performance on all used metrics. Moreover, we depict the corresponding Receiver Operating Characteristic (ROC) curve in Fig. 7. It is evident that the detection performance can be dramatically improved by introducing a tailored head. With a favorable detection, the IMDL methods can be more efficient. That is, detection is performed before localization, and only the detected forgery is passed for localization. Our network design is compatible with this efficiency consideration as the detection head is placed at the beginning of the bottom-up path. The qualitative evaluations of manipulation detection are demonstrated in Fig. 8, where the predicted masks from SPAN [18] and our PSCC-Net on both pristine and manipulation images are



(a) Pristine image (b) Manipulated image (c) Pristine mask [18] (d) Manip. mask [18] (e) Our pristine mask (f) Our manip. mask (g) GT manip. mask

Fig. 8. Qualitative detection evaluations on CASIA-D. Since GT pristine masks are blank, they are not shown here for clarity.

compared. Without the *existence-of-manipulation* assumption, for pristine images, the corresponding predicted masks from our PSCC-Net are nearly blank. However, the ones from SPAN suffer severe false alarms in most cases. As for the relevant manipulated images, the proposed method localizes the forged regions more accurately.

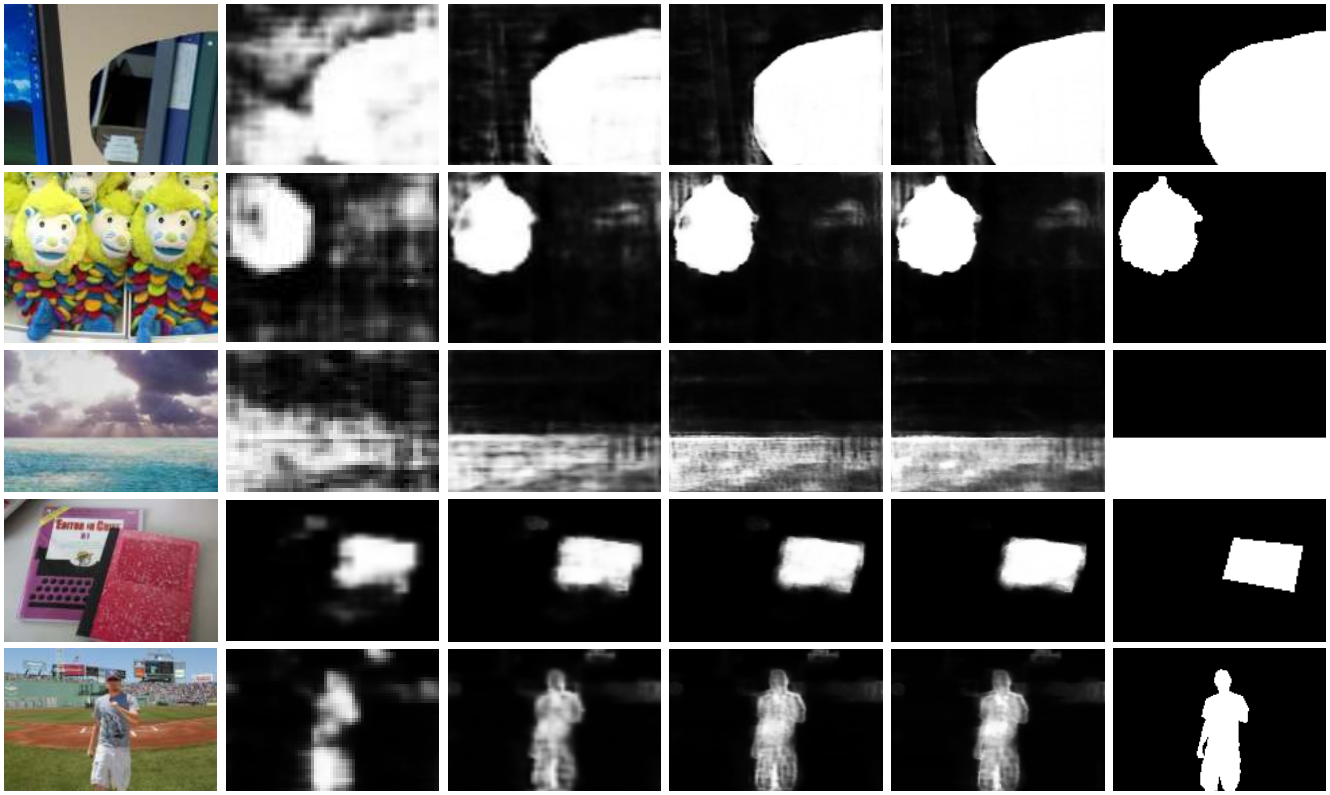
D. Visualization of SCCM

To provide insights into SCCM, we visualize the spatial response map for forged and pristine pixels in $\mathbf{M}_3$, by examining its spatial correlation represented in $\mathbf{A}_s$. After interpolation, each row of $\mathbf{A}_s$ is associated with one pixel (*e.g.*, P_1) in the test image, and its grayscale spatial response map can be obtained by reshaping this row vector from $1 \times HW$ to $H \times W$ (*e.g.*, P_1 response). In Fig. 9 (a), spliced, copy-moved, inpainted, and authentic images are shown from top to bottom respectively, each with one example. We select 3 representative pixels for each image and annotate as P_1, P_2, and P_3. For manipulated images, P_1 and P_2 are from forged regions, and P_3 is from pristine regions; as for the authentic image, all pixels are pristine. We project their grayscale spatial response maps into *Jet* color map and overlay them on the manipulated image as in Figs. 9 (c-e). It can be seen that for manipulated images, the spatial response maps of P_1 and P_2 have high values in forged regions and low values in pristine regions at most cases, but the map of P_3 retains low values in all regions including the one providing the copied content (*e.g.*, the P_3 response in the 2nd row of Fig. 9 (e)). As for the authentic image, the spatial response maps of all selected pixels retain low values consistently. This visualization indicates that the features in forged regions are successfully clustered together, thus justifies the effectiveness of spatial attention in SCCM.

For channel-wise correlation $\mathbf{A}_c$, it is hard to provide a comprehensible visualization. Instead, we choose to visualize one channel of $\mathbf{Y}_c$ and compare it to the same channel of $\mathbf{X}$ to see if any region is enhanced. We visualize the 1st channel of $\mathbf{X}$ and $\mathbf{Y}_c$ in Figs. 9 (f,g). Indeed, the forged region in $\mathbf{Y}_c$ is consolidated compared to the one in $\mathbf{X}$, and if the forged region does not exist (*i.e.,* in the case of authentic images), no region

(a) Test Image (b) GT manip. mask (c) P_1 response (d) P_2 response (e) P_3 response (f) 1st channel in $\mathbf{X}$ (g) 1st channel in $\mathbf{Y}_c$

Fig. 9. Visualization of spatial and channel-wise attentions in SCCM. From top to bottom, we show the spliced, copy-moved, inpainted, and authentic images respectively. For each test image, we show its GT manipulation mask, 3 spatial response maps (one for each selected pixel), and the 1st channel map in $\mathbf{X}$ and $\mathbf{Y}_c$. Zoom in for details.



(a) Manipulated image (b) Mask 4 (c) Mask 3 (d) Mask 2 (e) Mask 1 (f) GT manip. mask

Fig. 10. Visualization of predicted manipulation masks from *Scale* 4 to *Scale* 1. From top to bottom, manipulated images are from Columbia, Coverage, CASIA, NIST16, and IMD20 respectively. All predicted masks are from our pre-trained model.

is enhanced. This proves the effectiveness of channel-wise attention in SCCM.

E. Visualization of Predicted Manipulation Masks on Different Scales

The proposed PSCC-Net utilizes a progressive mechanism to reduce the prediction difficulty by avoiding generating the mask from the finest scale directly. Instead, the mask on the coarsest scale is first predicted to locate the regions that are potentially forged based on the current available information. The subsequent prediction on the finer scale can leverage the previous mask and pay more attention to those selected regions. This process repeatedly performs until generating the manipulation mask at the finest scale as our final prediction.

Here, we visualize the performance improvement of manipulation localization from the Scale 4 to Scale 1. In Fig. 10, *Mask 4*, *Mask 3*, and *Mask 2* are the variants that truncate the original model after generating manipulation masks on the 4th, 3rd, and 2nd scales, and *Mask 1* is the output of

TABLE V
ABLATION STUDY OF PSCC-NET (AUC/F1 IN %). THE RUNTIME IS REPORTED IN PROPORTION TO THAT OF ORIGINAL PSCC-NET

Variants	Columbia	Coverage	CASIA	NIST16	Runtime
Mask 4	93.34 / 79.22	82.99 / 44.23	81.49 / 31.69	84.15 / 30.55	**0.63**
Mask 3	98.08 / 92.41	83.48 / 47.29	82.55 / 34.64	85.25 / 33.55	0.75
Mask 2	98.18 / 93.32	84.44 / 49.08	82.78 / 35.59	85.38 / 34.94	0.88
w/o CA+SA	85.78 / 70.32	79.95 / 43.27	79.26 / 31.06	79.58 / 31.73	0.84
w/o SA	90.70 / 75.68	80.56 / 43.50	79.51 / 31.08	83.49 / 32.34	0.92
w/o CA	94.50 / 85.34	82.16 / 45.04	82.63 / 35.97	84.65 / 33.42	0.92
w/o FS	97.54 / 91.30	84.06 / 48.23	81.88 / 34.33	84.45 / 34.21	1.04
PSCC-Net	**98.19 / 93.45**	**84.65 / 49.78**	**82.93 / 36.27**	**85.47 / 35.73**	1.00

TABLE VI
ROBUSTNESS ANALYSIS OF LOCALIZATION WITH RESPECT TO VARIOUS DISTORTIONS. PIXEL-LEVEL AUCS ARE REPORTED (IN %)

Distortion	Resize $0.78\times$	Resize $0.25\times$	GSBlur $k=3$	GSBlur $k=15$	GSNoise $\sigma=3$	GSNoise $\sigma=15$	JPEGComp $q=100$	JPEGComp $q=50$	Mixed	w/o distortion
Columbia										
ManTraNet [11]	71.66	68.64	67.72	62.88	68.22	54.97	75.00	59.37	60.47	77.95
SPAN [18]	89.99	69.08	78.97	67.70	75.11	65.80	93.32	74.62	62.54	93.60
PSCC-Net	**93.40**	**78.41**	**84.18**	**73.24**	**82.64**	**74.35**	**97.97**	**89.11**	**72.69**	**98.19**
NIST16										
ManTraNet [11]	77.43	75.52	77.46	74.55	67.41	58.55	77.91	74.38	64.82	78.05
SPAN [18]	83.24	80.32	83.10	79.15	75.17	67.28	83.59	80.68	68.36	83.95
PSCC-Net	**85.29**	**85.01**	**85.38**	**79.93**	**78.42**	**76.65**	**85.40**	**85.37**	**73.93**	**85.47**

the original model. It can be seen that benefiting to the proposed progressive mechanism, the localization performance is gradually improved from *Mask 4* to *Mask 1* in terms of lower false alarms and clearer boundaries. More discussions about quantitative comparisons and terminating PSCC-Net earlier for runtime saving can be found in Sec. IV-F.

F. Runtime Analysis and Ablation Study

In Tab. V, we test several variants of PSCC-Net to justify the network design, where all variants are pre-trained on our synthetic dataset. Average AUC/F1s are reported (in %), and the runtime (in proportion) is relative to that of PSCC-Net. Our full model takes $0.019s$ to process one $1,080$P image, whereas ManTra-Net and SPAN take $0.208s$ and $0.161s$, respectively. Moreover, as shown in Fig. 2, terminating the PSCC-Net earlier on *Mask 4*, *Mask 3* or *Mask 2* in inference time is feasible and will not interfere the prediction of manipulation mask at that scale. From our experiments, terminating the prediction on *Mask 4* can shorten the runtime to $0.012s$, *i.e.*, $\sim$ 37% additional saving. Though nonessential for research datasets, this time-saving is significant and economical in practical applications, *e.g.*, 14.6 million photos are uploaded to Facebook *per hour*.[1]

The comparisons of *Mask 4*, *Mask 3*, *Mask 2*, and the original PSCC-Net demonstrate the gradual improvement in performance, which is a clear manifestation of our progressive mechanism. Since *Mask 3* already performs well under AUC and F1 scores, it is a good stopping point for mask prediction.

We also build several variants for SCCM, including the ones without spatial and channel-wise attentions (*w/o SA+CA*), without spatial attention (*w/o SA*), without channel-wise attention (*w/o CA*), and without feature sharing (*w/o FS*), which obtains embeddings from different θ and ϕ functions to compute spatial and channel-wise similarities. The comparisons illustrate that both SA and CA outperform the baseline (*w/o SA+CA*), and the performance gain acquired from SA is more than that from CA. In addition, feature sharing not only slightly reduces the runtime, but also enables mutual accommodation between these two attentions to help SCCM achieve better results than the one employing different features (*i.e., w/o FS*).

G. Robustness Analysis

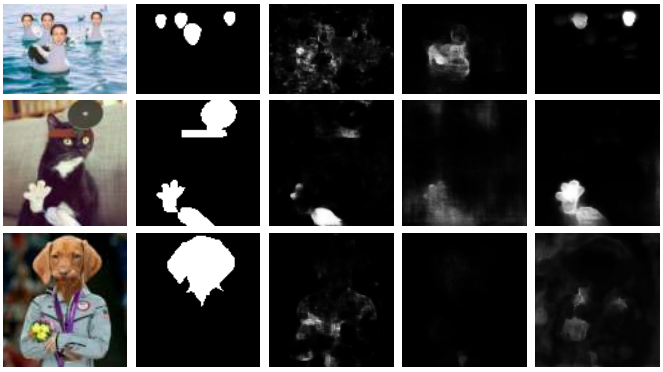
To analyze the robustness of PSCC-Net for localization, we follow the distortion settings in [18] to degrade the raw manipulated images from Columbia and NIST16. These distortions include resizing images to a different scale (*Resize*), applying Gaussian blur with kernel size k (*GSBlur*), adding Gaussian noise with standard deviation σ (*GSNoise*), and performing JPEG compression with quality factor q (*JPEGComp*). In addition, we introduce a mixed version of the aforementioned distortions (*Mixed*), where the resizing scale, kernel size k, standard deviation σ, quality factor q are randomly selected from the intervals $[0.25, 0.78]$, $[3, 15]$, $[3, 15]$, and $[50, 100]$, respectively. Tab. VI shows the robustness

[1]https://www.pingdom.com/blog/social-media-in-2017/

TABLE VII

ROBUSTNESS ANALYSIS OF DETECTION FOR PSCC-NET WITH RESPECT TO VARIOUS DISTORTIONS ON CASIA-D. IMAGE-LEVEL AUCS AND F1 SCORES ARE REPORTED (IN %)

Distortion	AUC	F1
Resize $0.78\times$	95.48	91.46
Resize $0.25\times$	74.86	70.47
GSBlur $k = 3$	92.93	87.55
GSBlur $k = 15$	88.59	83.56
GSNoise $\sigma = 3$	89.78	83.37
GSNoise $\sigma = 15$	85.50	80.87
JPEGComp $q = 100$	99.44	96.47
JPEGComp $q = 50$	99.45	96.47
Mixed	87.55	83.51
w/o distortion	**99.65**	**97.12**



(a) Manipulated (b) GT (c) ManTra-Net (d) SPAN (e) PSCC-Net

Fig. 11. Failure cases. Zoom in for details.

analysis of localization under pixel-level AUC with pre-trained models. The PSCC-Net is more robust than ManTra-Net and SPAN under all distortions. It is worth noting that resizing is commonly performed when uploading images to social media. Indeed, benefiting from the operation that resamples the manipulation features into the fixed sizes, the impact of resizing to PSCC-Net is the least as compared to the others.

We also analyze the detection robustness of PSCC-Net with respect to various distortions on CASIA-D. In Tab. VII, it can be seen that our PSCC-Net is quite robust for detection, especially in the case where the JPEG compression is performed.

H. Limitations

PSCC-Net enables us to detect and localize various types of manipulations. As compared to image-level detection, the pixel-level localization is more challenging, especially while dealing with real-life manipulated images. Here we demonstrate some failure cases on IMD20 [76].

In Fig. 11, it is clear that for real-life manipulated images, the forged regions may have diverse sizes and shapes. In the first row, we show a specific case where the same pattern is copied several times but with different scales. Despite our method fails to localize all forged regions, it is less sensitive to scale variation as compared to ManTra-Net [11] and SPAN [18], owing to our tailored network design. In addition, our method may fail to localize the whole forged regions or only localize part of them in some cases (*e.g.*, the last two rows). One possible reason is that some manipulation traces are elaborately removed by fabricators. Indeed, the compared IMDL methods also have difficulty to tackle these manipulated images. Note that even in theses cases, our PSCC-Net still performs relatively better than the SOTAs [11], [18] for image manipulation localization (*e.g.*, see the 2rd row).

V. CONCLUSION

In this work, a novel PSCC-Net is proposed to meet the challenge of advanced image manipulation techniques. We employ a progressive mechanism to predict the manipulation mask on all backbone scales, where each mask serves as a prior to help predict the next-scale mask. Moreover, a SCCM is designed to perform spatial and channel-wise attentions on extracted features, which provides holistic information to make our model more generalized to manipulation attacks. Extensive experiments demonstrate that our PSCC-Net outperforms the SOTA methods on both detection and localization. For future work, we will develop techniques for estimating the uncertainty of predicted manipulation masks to further improve the IMDL performance.

REFERENCES

[1] B. Li, X. Qi, T. Lukasiewicz, and P. H. S. Torr, "ManiGAN: Text-guided image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7880–7889.

[2] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5549–5558.

[3] H. Dhamo *et al.*, "Semantic image manipulation using scene graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5213–5222.

[4] X. Liu *et al.*, "Open-Edit: Open-domain image manipulation with open-vocabulary instructions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 89–106.

[5] J. Li, F. He, L. Zhang, B. Du, and D. Tao, "Progressive reconstruction of visual structure for image inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5962–5971.

[6] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7760–7768.

[7] A. Lahiri, A. K. Jain, S. Agrawal, P. Mitra, and P. K. Biswas, "Prior guided GAN based semantic inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13696–13705.

[8] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 1–17.

[9] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9243–9252.

[10] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 101–117.

[11] Y. Wu, W. Abdalmageed, and P. Natarajan, "ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9543–9552.

[12] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Inf. Fusion*, vol. 64, pp. 131–148, Dec. 2020.

[13] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain, "On the detection of digital face manipulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5781–5790.

[14] J. Yang, S. Xiao, A. Li, W. Lu, X. Gao, and Y. Li, "MSTA-Net: Forgery detection by generating manipulation trace based on multi-scale self-texture attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4854–4866, Dec. 2021.

[15] J. Hu, X. Liao, W. Wang, and Z. Qin, "Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1089–1102, Apr. 2021.

[16] T. Thomson, D. Angus, and P. Dootson. (Nov. 4, 2020). Seeing no longer means believing. In Daily. [Online]. Available: https://indaily.com.au/opinion/2020/11/04/seeing-should-not-mean-believing/

[17] A. Willingham. (Oct. 2020). Is that video real? CNN. [Online]. Available: https://www.cnn.com/interactive/2020/10/us/manipulated-media-tech-fake-news-trnd/

[18] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "SPAN: Spatial pyramid attention network for image manipulation localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 312–328.

[19] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, May 2019.

[20] C. Chen, Z. Xiong, X. Liu, and F. Wu, "Camera trace erasing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2950–2959.

[21] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, "Exploiting spatial structure for localizing manipulated image regions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4970–4979.

[22] R. Salloum, Y. Ren, and C.-C. Jay Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, Feb. 2018.

[23] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1053–1061.

[24] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3286–3300, Jul. 2019.

[25] V. Asnani, X. Yin, T. Hassner, S. Liu, and X. Liu, "Proactive image manipulation detection," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 15386–15395.

[26] V. Asnani, X. Yin, T. Hassner, and X. Liu, "Reverse engineering of generative models: Inferring model hyperparameters from generated images," 2021, *arXiv:2106.07873*.

[27] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Apr. 2020.

[28] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy-move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 168–184.

[29] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4676–4685.

[30] S. Lyu, X. Pan, and X. Zhang, "Exposing region splicing forgeries with blind local noise estimation," *Int. J. Comput. Vis.*, vol. 110, no. 2, pp. 202–221, Nov. 2014.

[31] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[32] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli, "Localization of JPEG double compression through multi-domain convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1865–1871.

[33] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based CNN features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops. (CVPRW)*, Jul. 2017, pp. 1–10.

[34] 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," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1480–1502.

[35] V. V. Kniaz, V. Knyaz, and F. Remondino, "The point where reality meets fantasy: Mixed adversarial generators for image splice detection," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1–12.

[36] Y. Zhang, G. Zhu, L. Wu, S. Kwong, H. Zhang, and Y. Zhou, "Multi-task se-network for image splicing localization," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 32, no. 7, pp. 4828–4840, Oct. 2021.

[37] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy–move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2284–2297, Jul. 2015.

[38] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "COVERAGE—A novel database for copy-move forgery detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 161–165.

[39] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Image copy-move forgery detection via an end-to-end deep neural network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1907–1915.

[40] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A patchmatch-based dense-field algorithm for video copy–move detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 669–682, Feb. 2019.

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

[42] M. Aloraini, M. Sharifzadeh, and D. Schonfeld, "Sequential and patch analyses for object removal video forgery detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 917–930, May 2020.

[43] Q. Yang, D. Yu, Z. Zhang, Y. Yao, and L. Chen, "Spatiotemporal trident networks: Detection and localization of object removal tampering in video passive forensics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 4131–4144, Dec. 2020.

[44] H. Wu and J. Zhou, "IID-Net: Image inpainting detection network via neural architecture search and attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1172–1185, Apr. 2021.

[45] R. M. Joseph and A. Chithra, "Literature survey on image manipulation detection," *Int. Res. J. Eng. Technol.*, vol. 2, no. 4, p. 740, Jul. 2015.

[46] Y. Liu, J. Stehouwer, and X. Liu, "On disentangling spoof traces for generic face anti-spoofing," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 406–422.

[47] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 1–9.

[49] W. Ren *et al.*, "Gated fusion network for single image dehazing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3253–3261.

[50] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3106–3115.

[51] K. Jiang *et al.*, "Multi-scale progressive fusion network for single image deraining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8346–8355.

[52] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "FSRNet: End-to-end learning face super-resolution with facial priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2492–2501.

[53] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 714–722.

[54] J. Zhu, D. Li, T. Han, L. Tian, and Y. Shan, "ProgressFace: Scale-aware progressive learning for face detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 344–360.

[55] X. Song, K. Zhao, W.-S. C. H. Zhang, and J. Guo, "Progressive refinement network for occluded pedestrian detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 32–48.

[56] G. Brazil and X. Liu, "Pedestrian detection with autoregressive network phases," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7231–7240.

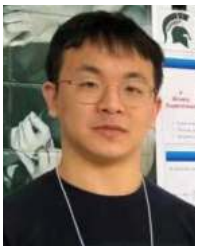
[57] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 1–11.

[58] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.

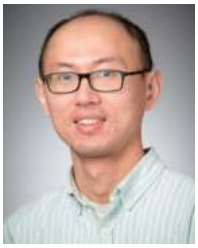
[59] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.

[60] X. Liu, Y. Ma, Z. Shi, and J. Chen, "GridDehazeNet: Attention-based multi-scale network for image dehazing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7314–7323.
[61] T. Isobe *et al.*, "Video super-resolution with temporal group attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8008–8017.
[62] S. Gong, X. Liu, and A. K. Jain, "Mitigating face recognition bias via group adaptive classifier," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3414–3424.
[63] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–14.
[64] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.
[65] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.
[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[67] Y. Liu, X. Zhu, X. Zhao, and Y. Cao, "Adversarial learning for constrained image splicing detection and localization based on atrous convolution," *IEEE Trans. Inf. Forensics Security*, vol. 19, no. 10, pp. 2551–2566, Oct. 2019.
[68] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740–755.
[69] M. E. Mortenson, *Mathematics for Computer Graphics Applications*. New York, NY, USA: Industrial Press, 1999.
[70] IEEE's Signal Processing Society. *Camera Model Identification*. Accessed: Jul. 12, 2022. [Online]. Available: https://www.kaggle.com/c/sp-society-camera-model-identification
[71] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, pp. 1–16, Dec. 2017.
[72] T. Gloe and R. Böhme, "The 'Dresden image Database' for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Mar. 2010, pp. 1584–1590.
[73] T.-T. Ng, J. Hsu, and S.-F. Chang. *Columbia Image Splicing Detection Evaluation Dataset*. Accessed: Jul. 12, 2022. [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm
[74] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *Proc. China Summit Int. Conf. Signal Inf. Process.*, Jul. 2013, pp. 422–426.
[75] (2016). *NIST: Nist Nimble 2016 Datasets*. [Online]. Available: https://www.nist.gov/itl/iad/mig/
[76] A. Novozamsky, B. Mahdian, and S. Saic, "IMD2020: A large-scale annotated dataset tailored for detecting manipulated images," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Mar. 2020, pp. 71–80.
[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Dec. 2015, pp. 1–15.

Xiaohong Liu received the B.E. degree in communication engineering from Southwest Jiaotong University, Chengdu, China, in 2014, the M.A.Sc. degree in electrical and computer engineering from the University of Ottawa, Ottawa, ON, Canada, in 2016, and the Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 2021. He is currently a Tenure-Track Assistant Professor with the John Hopcroft Center, Shanghai Jiao Tong University, Shanghai, China. His research interests include image/video restoration and image segmentation. He was a recipient of the Ontario Graduate Scholarship in 2019, NSERC Alexander Graham Bell Canada Graduate Scholarship-Doctoral, and Borealis AI Global Fellowship Award in 2020. He is a Reviewer of several IEEE journals, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.



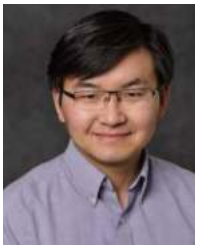
Yaojie Liu received the B.S. degree in communication engineering from the University of Electronic Science and Technology of China in 2014, the M.S. degree in computer science from Ohio State University in 2016, and the Ph.D. degree in computer science and engineering from Michigan State University in 2021. He is currently a Research Scientist at Google Research. His research areas of interest are security of face biometric systems (e.g., face anti-spoofing, digital manipulation attack, adversarial attack), 3D face modeling, face representation and analysis, XAI, image systhesis, and multi-model modeling.



Jun Chen (Senior Member, IEEE) received the B.E. degree in communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2004 and 2006, respectively.

From September 2005 to July 2006, he was a Post-Doctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana, IL, USA, and a Post-Doctoral Fellow with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, from July 2006 to August 2007. Since September 2007, he has been with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he is currently a Professor. His research interests include information theory, machine learning, wireless communications, and signal processing.

Dr. Chen was a recipient of the Josef Raviv Memorial Post-Doctoral Fellowship in 2006, the Early Researcher Award from the Province of Ontario in 2010, the IBM Faculty Award in 2010, the ICC Best Paper Award in 2020, and the JSPS Invitational Fellowship in 2021. He held the title of the Barber-Gennum Chair of information technology from 2008 to 2013 and the title of the Joseph Ip Distinguished Engineering Fellow from 2016 to 2018. He served as an Editor for IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2020 to 2021. He is currently an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY.



Xiaoming Liu (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2004. He is currently a MSU Foundation Professor with the Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan. Before joining MSU, in Fall 2012, he was a Research Scientist at General Electric (GE) Global Research. His research interests include computer vision, machine learning, and biometrics. As the coauthor, he was a recipient of Best Industry Related Paper Award Runner-up at ICPR 2014, Best Student Paper Award at WACV 2012 and 2014, and Best Poster Award at BMVC 2015. He has been the Area Chair for numerous conferences, including CVPR, ECCV, ICCV, NeurIPS, and ICLR. He is the Program Chair of WACV 2018, BTAS 2018, AVSS 2022, and the General Chair of FG 2023. He is also an Associate Editor of the *Pattern Recognition* and the IEEE TRANSACTIONS ON IMAGE PROCESSING. He has authored more than 160 scientific publications, and has filed 29 U.S. patents.