IID-Net: Image Inpainting Detection Network via Neural Architecture Search and Attention

Haiwei Wu, *Student Member, IEEE*, and Jiantao Zhou, *Senior Member, IEEE*

***Abstract*—Deep learning (DL) has demonstrated its powerful capabilities in the field of image inpainting, which could produce visually plausible results. Meanwhile, the malicious use of advanced image inpainting tools (e.g. removing key objects to report fake news, erasing visible copyright watermarks, etc.) has led to increasing threats to the reliability of image data. To fight against the inpainting forgeries (not only DL-based but also traditional ones), in this work, we propose a novel end-to-end Image Inpainting Detection Network (IID-Net), to detect the inpainted regions at pixel accuracy. The proposed IID-Net consists of three sub-blocks: the enhancement block, the extraction block and the decision block. Specifically, the enhancement block aims to enhance the inpainting traces by using hierarchically combined special layers. The extraction block, automatically designed by Neural Architecture Search (NAS) algorithm, is targeted to extract features for the actual inpainting detection tasks. To further optimize the extracted latent features, we integrate global and local attention modules in the decision block, where the global attention reduces the intra-class differences by measuring the similarity of global features, while the local attention strengthens the consistency of local features. Furthermore, we thoroughly study the generalizability of our IID-Net, and find that different training data could result in vastly different generalization capability. By carefully examining 10 popular inpainting methods, we identify that the IID-Net trained on only one specific deep inpainting method exhibits desirable generalizability; namely, the obtained IID-Net can accurately detect and localize inpainting manipulations for various unseen inpainting methods as well. Extensive experimental results are presented to validate the superiority of the proposed IID-Net, compared with the state-of-the-art competitors. Our results would suggest that common artifacts are shared across diverse image inpainting methods. Finally, we build a public inpainting dataset of 10K image pairs for future research in this area.**

***Index Terms*—Inpainting forensics, generalizability, deep neural networks.**

I. Introduction

IMAGE inpainting is to fill the missing region of an image with plausible contents. It has a wide range of applications in the field of image processing and computer vision, e.g., repairing damaged photos and removing unwanted objects. Nevertheless, image inpainting techniques might also be exploited maliciously to alter and delete contents, making them powerful tools for creating forged images. The trust issues and security concerns regarding the malicious use of image inpainting techniques have been attracting increasing attention in recent years; for instance, using inpainted images in court as evidence, removing key objects to report fake news, erasing visible copyright watermarks, just to name a few. The situation becomes even worse when the deep learning (DL)-based inpainting methods have become prevalent. As shown in Fig. 1 (a)-(b), a malicious attacker can very easily change the facial content or erase the key objects/watermarks by using the latest DL-based inpainting methods through their online website [1] or open resources. Therefore, it is imperative to study how to accurately detect and locate the inpainted regions for fighting against the inpainting forgeries.

The detection and localization of processed regions have always been a hot research topic in the field of information forensics. Many methods were proposed to detect the forged regions through their specific artifacts, e.g., compression artifacts [4], noise pattern [5], color consistencies [6], EXIF consistencies [7], and copy-move traces [8]. However, few researches have been done on the detection of inpainting manipulations, especially the latest DL-based ones. As mentioned in [9], the inpainting manipulations are more sophisticated and complex than some other forgeries (e.g., copy-move forgery) because the inpainting operations could produce non-continuous contents. Furthermore, DL-based inpainting schemes are capable of creating completely novel semantic contents, which impose great challenges for the detection task. The pioneering study on deep inpainting detection was conducted by Li and Huang [10], showing that it is feasible to train a deep model for detecting specific deep inpainting artifacts if the inpainting scheme is known. However, with the rapid progress of DL-based inpainting, it is very challenging to know the employed inpainting scheme for a given image; sometimes more than one inpainting schemes could be adopted to edit one single image. It is therefore very desirable if we can find a generalizable forensic approach for detecting various inpainting

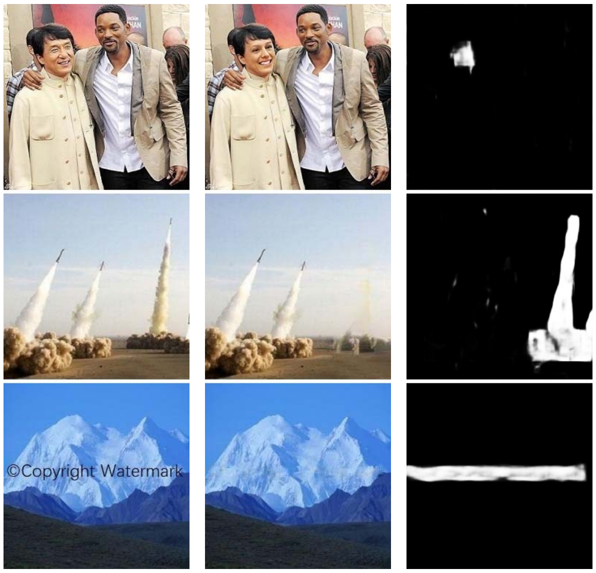
Manuscript received February 2, 2021; revised April 1, 2021 and April 16, 2021; accepted April 17, 2021. Date of publication April 22, 2021; date of current version March 9, 2022. This work was supported in part by the Macau Science and Technology Development Fund under Grant SKL-IOTSC-2021-2023, Grant 077/2018/A2, and Grant 0060/2019/A1; in part by the Research Committee at the University of Macau under Grant MYRG2018-00029-FST and Grant MYRG2019-00023-FST; in part by the Natural Science Foundation of China under Grant 61971476; and in part by the Alibaba Group through Alibaba Innovative Research Program. This article was recommended by Associate Editor H. Li. *(Corresponding author: Jiantao Zhou.)*

The authors are with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Taipa, Macau, and also with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Taipa, Macau (e-mail: yc07912@umac.mo; jtzhou@umac.mo).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3075039.

Digital Object Identifier 10.1109/TCSVT.2021.3075039

[1] https://www.nvidia.com/research/inpainting/



(a) Original (b) Forgery (c) Detection

Fig. 1. (a) The original images; (b) The forged images where the key objects/watermarks are removed/replaced by the DL-based inpainting methods [1]–[3] respectively; and (c) The output of IID-Net by using (b) as input.

manipulations, not only traditional inpainting schemes but also DL-based ones. This problem, though challenging, seems to be viable because nowadays convolutional neural networks usually produce common artifacts when generating images, as discovered in [11].

In this work, we tackle the challenge of providing a forensic solution that can generalize well to accurately detect various unseen inpainting manipulations. More specifically, we propose a novel end-to-end Image Inpainting Detection Network (IID-Net), to detect the inpainted regions at pixel accuracy. The proposed IID-Net consists of three sub-blocks: the enhancement block, the extraction block and the decision block. The enhancement block aims to enhance the inpainting traces by using hierarchically combined special layers. The extraction block, automatically designed by Neural Architecture Search (NAS) algorithm, is targeted to extract features for the actual inpainting detection tasks. In order to further optimize the extracted latent features, we integrate global and local attention modules in the decision block, where the global attention reduces the intra-class differences by measuring the similarity of global features, while the local attention strengthens the consistency of local features. Furthermore, we thoroughly study the generalizability of our IID-Net, and find that different training data could result in vastly different generalization capability. By carefully examining 10 popular inpainting methods, we identify that the IID-Net trained on one specific deep inpainting method exhibits desirable generalizability, namely, the obtained IID-Net can accurately detect and localize inpainting manipulations for unseen (not only DL-based but also traditional) inpainting methods as well. Extensive experimental results are presented to validate the superiority of the proposed IID-Net, compared with the state-of-the-art competitors. Our results would suggest that common artifacts are shared across diverse image inpainting methods. Finally, we build a public inpainting dataset of 10K image pairs for future research in this area. An example of the detection result of IID-Net is shown in Fig. 1 (c), which is the direct output of our model *without* any post-processing by using Fig. 1 (b) as input. Here we would like to emphasize that none of the original images in Fig. 1 (a) or the corresponding inpainting methods [1]–[3] were involved during the training of IID-Net.

Our major contributions can be summarized as follows:

- We propose the IID-Net, a novel end-to-end network for the image inpainting detection, where the NAS algorithm is used for designing appropriate network architecture and newly proposed attention modules are incorporated to further optimize latent features.
- We construct a diverse-inpainting test dataset with 10K images, based on 10 different inpainting methods, each contributing 1000 images. Among them, six (GC [12], CA [13], SH [14], EC [2], LB [3] and RN [15]) are DL-based, and the remaining (TE [16], NS [17], PM [18], and SG [19]) are traditional ones. This could serve as a publicly accessible dataset for standardized comparisons of inpainting detection approaches.
- Our IID-Net achieves much better detection performance in comparison with several state-of-the-art methods [10], [20], [21] over the diverse-inpainting dataset.
- We show that the forensic model trained on a specific deep inpainting method exhibits excellent generalizable detection capability to other inpainting methods, no matter DL-based or traditional ones. This validates that common detectable traces are left by various inpainting manipulations.

The rest of this paper is organized as follows. Section II reviews the related works on inpainting methods, inpainting forensics, as well as NAS. Section III presents our proposed IID-Net. Experimental results are given in Section IV and Section V concludes.

II. Related Works

A. Inpainting Methods

Image inpainting provides a means for the reconstruction of missing regions, and has been studied for decades (see [16]–[19], [22]–[28] and references therein). Bertalmio *et al.* [17] introduced an approach that uses ideas from classical fluid dynamics to propagate isophote lines continuously from the exterior into the missing regions. Based on the fast matching method for level set applications, Telea [16] proposed a simple and fast inpainting algorithm by propagating an image smoothness estimator along the image gradient. More recently, Huang *et al.* [19] showed that image inpainting can be substantially improved by automatically guiding the low-level synthesis algorithm using mid-level structural analysis of the known region. Herling and Broll [18] presented a combined pixel-based approach that not only allows for even faster inpainting, but also improves the overall image quality significantly. However, as these texture synthesis based inpainting methods essentially assumed that the missing region shares the same structural features with the known one, they cannot create novel contents for the challenging cases where the missing region involves complex structures (e.g., faces) and high-level semantics [10], [13].

To address these limitations, many DL models have been proposed for image inpainting in recent years. By utilizing large-scale datasets to learn semantic representations of images, DL-based inpainting methods are able to generate completely novel contents and achieve the state-of-the-art inpainting performance. Pathak *et al.* [29] pioneered the research in this direction by training deep generative adversarial networks for inpainting large holes in images. However, the proposed networks cannot satisfactorily maintain global consistency and tend to produce severe visual artifacts. Iizuka *et al.* [30] designed a generative network with two context discriminators to encourage global and local consistency. Instead of merely using the features of latent layers, some works [13], [14] introduced attention mechanism, which jointly uses the existing features to estimate the missing features. To further improve the attention mechanism, Wang *et al.* [31] suggested a multi-stage image contextual attention learning strategy to deal with the rich background information flexibly while avoiding abuse them. Meanwhile, several works [1], [12] adopted partial or gated convolutions to reduce the color discrepancy and blurriness, where the convolutions are masked, renormalized, and operated only on the known region. With the recent trend of using two-stage networks, Nazeri *et al.* [2] and Wu *et al.* [3] respectively proposed to use edge/LBP generator at the first stage, followed by a second image completion network to further improve the inpainting performance. Besides, the inpainting techniques can be utilized for facilitating the occluded face recognition systems [32], [33]. In particular, Ge *et al.* [32] proposed the inpainting based identify-diversity GAN to improve the capacity of well-trained face recognizers on identifying occluded faces. Li *et al.* [33] designed an inpainting guided de-occlusion distillation framework for efficient masked face recognition.

B. Inpainting Forensics

As the other side of the coin, many inpainting forensic methods [9], [34]–[45] have been proposed to fight against the malicious usage of inpainting manipulations. One common principle of these methods is to search similar blocks within a given image, where the blocks with high matching degrees are suspected to be forged. Specifically, Wu *et al.* [34] proposed a blind detection method based on zero-connectivity feature and fuzzy membership. Lin *et al.* [35] leveraged quantization table estimation to measure the inconsistency among images for detecting forged images. Further, Liang *et al.* [39] presented an efficient forgery detection algorithm which integrates central pixel mapping, greatest zero-connectivity component labeling and fragment splicing detection. More recently, Zhu *et al.* [43] built an encoder-decoder network that is supervised by a label matrix and weighted cross-entropy to capture the manipulation traces. Unfortunately, these forensic approaches can only detect exemplar-based inpainting manipulations, while not diffusion-based ones, as the latter type will not generate similar blocks in the inpainted regions [20]. To remedy this issue, Li *et al.* [20] suggested detecting diffusion-based inpainting by analyzing the local variance of image Laplacian along the isophote direction. In addition, to detect complicated combinations of forgeries (including inpainting), Wu *et al.* [21] proposed MT-Net, a more general forgery localization network, which first extracts image manipulation trace features and then identifies anomalous regions by assessing how different a local feature is from its reference features. However, for some challenging cases, e.g., when forged features dominate the image, MT-Net could fail completely.

Since DL-based inpainting methods can use learned high-level semantic information to generate more complex structures and even novel objects, they may leave completely different artifacts in the inpainted regions, causing very poor detection performance of the aforementioned forensic approaches [10], [46]. To improve the detection accuracy, Li and Huang [10] designed the HP-FCN, a DL-based method to locate the image regions manipulated by deep inpainting. A high-pass pre-filtering module is employed to suppress image contents and enhance the differences between the inpainted and untouched regions. Their experimental results showed that HP-FCN can effectively locate the inpainting forgeries when the training set created from the same inpainting method is available. It should be noted that the generalizability to unseen inpainting methods, though important in practice, has not been investigated in [10].

C. Neural Architecture Search (NAS)

The achievements of deep neural networks in various tasks depending on their exquisite architecture design, which requires a tremendous amount of domain knowledge and is usually time-consuming. Zoph and Le [47] introduced the NAS, an idea of using recurrent neural networks to search for an appropriate network architecture with the highest validation accuracy. Along this line, many works proposed to adopt advanced techniques to aid the search process, e.g., reinforcement learning [48], [49], evolution [50] and surrogate model [51]. However, searching and training thousands of models are almost infeasible for a single practitioner [52]. To significantly reduce the computational cost of the architecture search, the weight sharing mechanism [53] and one-shot NAS [52] were utilized. In this work, we adopt the one-shot NAS strategy as the backend search algorithm, due to its speediness and flexibility [54].

The core idea of the one-shot NAS is to use the same weights to evaluate different sampled architectures, thereby achieving the cost reduction of an order of magnitude. Specifically, instead of training several separate models, we can train a single model (the *one-shot model*) containing all the potential operations. Then at the evaluation stage, some operations' outputs are selectively zeroed out, in order to determine which operation contributes most to the prediction accuracy. It should be noted that the one-shot models are only used to rank different architectures in the search space; retraining the candidate model with the highest evaluation accuracy is still needed. For more details regarding the one-shot NAS, please refer to [52].

III. IID-Net

In this section, we present the details of the IID-Net for detecting the inpainting manipulations, not only for DL-based

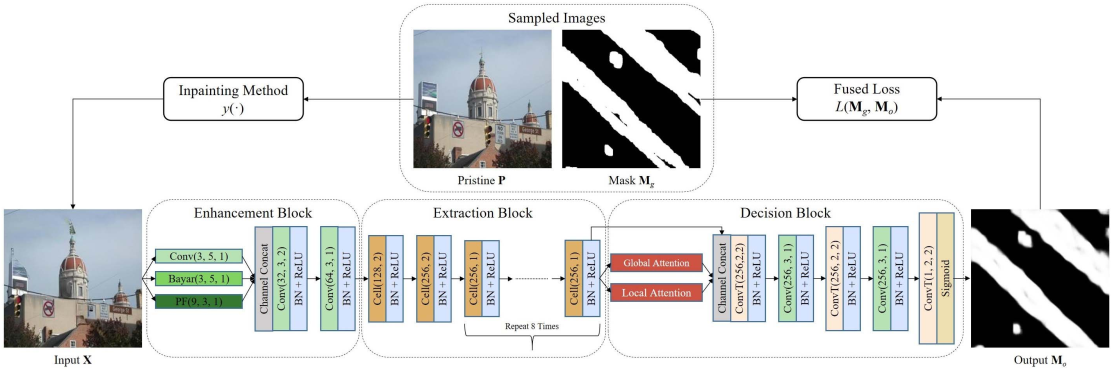


Fig. 2. The overview of our proposed IID-Net. The upper part shows how to generate images for training, while for inference, by directly using an image **X** as input, the detection result $\mathbf{M}_o$ can be obtained. $\mathbf{Conv}(f, k, s)$ means a convolution with f filters, kernel size k and stride s (similar for the remaining numbers in other layers). More details are given in Section III.

but also for traditional ones. The schematic diagram of the IID-Net is shown in Fig. 2. As can be seen, the IID-Net consists of three main blocks, namely, the enhancement block, the extraction block and the decision block. The enhancement block involves hierarchically combined input layers for enhancing the inpainting traces (see Section III-A). The following extraction block, composed of a series of cell units searched by one-shot NAS, is designed to extract high-level features that are suitable for distinguishing multiple forgeries (see Section III-B). Eventually, the decision block outputs the final detection result, with the assistance of global and local attention modules (see Section III-C).

At the training stage, we first sample a pristine 3-channel (RGB) color image $\mathbf{P} \in \mathbb{R}^{H \times W \times 3}$ and a corresponding binary mask $\mathbf{M}_g \in \{0, 1\}^{H \times W \times 1}$ (1's are assigned to the inpainted regions and 0's elsewhere). Then an input image **X** can be synthesized as

$$\mathbf{X} = \mathbf{P} \odot (1 - \mathbf{M}_g) + y(\mathbf{P} \odot (1 - \mathbf{M}_g)) \odot \mathbf{M}_g, \qquad (1)$$

where the operator $\odot$ means element-wise multiplication and the function $y(\cdot) : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 3}$ denotes the employed inpainting algorithm. The IID-Net $\mathcal{G} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 1}$ takes **X** as input, and outputs the predicted mask $\mathbf{M}_o$. Here, each element of $\mathbf{M}_o$ is a grayscale value in (0, 1), as we use the Sigmoid activation. A threshold (e.g., 0.5) could be employed to further binarize the output. During this process, the pair of $(\mathbf{M}_g, \mathbf{M}_o)$ are used by a fused loss function $\mathcal{L}$ to update the parameters of the network $\mathcal{G}$. At the inference stage, similar procedure can be performed to obtain the predicted inpainting mask.

Here we would like to emphasize that the inpainted regions indicated by the binary mask $\mathbf{M}_g$ can be of any shape and appear anywhere, which better reflects the true situation of forgery operations. The masks are selected from a dataset generated by [1], where some examples are given in Fig. 3. Compared with the HP-FCN [10] which only uses rectangular masks within a fixed range, our training preparation allows the network to learn more about the diversity of the inpainting forgery, leading to better detection accuracy.

Now we are ready to explain the aforementioned three main blocks in the IID-Net, and also the loss function for its optimization.

Fig. 3. Mask examples from [1], usually having < 50% holes.

A. Enhancement Block

Normally, the standard convolutional layer learns the features for representing the contents of input images, as opposed to learning the traces left by modifications [55]. Besides, as some traces are hidden in local noise distributions, RGB channels are not sufficient to tackle all the different cases of manipulations [56]. Therefore, we propose to suppress the contents of input images and enhance the inpainting traces by adding several pre-designed input layers. The potential input layers that can be incorporated include Steganalysis Rich Model (**SRM**) layer [56], Pre-Filtering (**PF**) layer [10], **Bayar** layer [55], convolution (**Conv**), and combinations of them.

More specifically, **SRM** layer utilizes the local noise distributions of the image to provide additional evidence [57]. For a 3-channel input **X**, **SRM** layer extracts the corresponding features Φ_s by using a $5 \times 5 \times 3$ kernel $\mathbf{W}_s$, namely,

$$\Phi_s(\mathbf{X}) = \mathbf{W}_s \otimes \mathbf{X}, \qquad (2)$$

where

$$\mathbf{W}_s = \frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \\ \times \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}; \\ \times \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad (3)$$

and $\otimes$ represents the convolutional operation.

PF layer is designed to get filtered residuals for enhancing inpainting traces, as the inpainted regions are more distinguishable from the pristine ones in the residual domain [10]. It is also found in [10] that the transition probability matrices (TPM) for pristine and inpainted patches in the pixel domain (without filtering) are very similar, while the TPM in the residual domain (with filtering) exhibit notable differences. This is mainly because the inpainted patches contain fewer high-frequency components, as the inpainting methods usually focus on producing visually realistic image contents, while ignoring the high-frequency noises inherently existing in the natural images. Thereby a high-pass filter (i.e. the **PF** layer) can be utilized to extract the features in the high-frequency domain for the subsequent forensic analysis. Practically, **PF** layer can be initialized with a 3 × 3 × 3 first-order derivative high-pass filter $\mathbf{W}_p$:

$$\mathbf{W}_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}; \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

The high-passed features Φ_p can then be obtained by

$$\Phi_p(\mathbf{X}) = \mathbf{W}_p \otimes \mathbf{X}. \quad (5)$$

It should be noted that the filter kernels of **PF** layer are set as learnable so that they can be fine-tuned during the learning.

Instead of relying on pre-determined kernels, we also incorporate the **Bayar** layer to adaptively learn low-level prediction residual features for detecting inpainting traces. It is implemented by adding specific constraints to the standard convolutional kernels. For simplicity, we use $\mathbf{W}_b^i$ to represent the ith ($i = 1, 2, 3$) channel of the weights $\mathbf{W}_b$ in the **Bayar** layer, and the central values of each channel $\mathbf{W}_b^i$ are denoted by a spatial index $(0, 0)$. Then the following constraints are enforced on each channel of $\mathbf{W}_b$ before each training iteration:

$$\begin{cases} \mathbf{W}_b^i(0,0) = -1 \\ \sum\limits_{m,n \neq 0} \mathbf{W}_b^i(m,n) = 1 \end{cases} \quad \text{for } i = 1, 2, 3. \quad (6)$$

Finally, the constrained features Φ_b can be obtained by

$$\Phi_b(\mathbf{X}) = \mathbf{W}_b \otimes \mathbf{X}. \quad (7)$$

To find an appropriate combination of these layers, we have conducted intensive experiments by evaluating the inpainting detection performance of various combinations. We have found that the combination of **Conv+Bayar+PF** gives the best detection performance. We thus use this kind of combination as our first layer in the enhancement block. Next, we concatenate the enhanced features in the channel dimension and use two standard convolutions to initially process the enhanced features (i.e., decrease in resolutions and increase in the number of channels) in preparation for the subsequent high-level features extraction. The experimental justifications of such combination will be provided in Section IV-D.

B. Extraction Block

After the enhancement block, we also design the extraction block to extract the high-level features for the inpainting detection. Instead of adopting the commonly-used ResNet [58] or DnCNN [59] as a backbone, we propose to use an adjustable cell and fine-tune it with the one-shot NAS, so as to better fit the requirement of the inpainting detection. In the following, we successively introduce the cell architecture, i.e., search space, and the search algorithm.

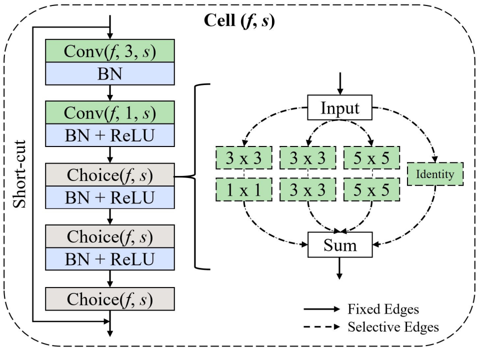


Fig. 4. Diagram of the cell unit used in our extraction block. Solid lines indicate fixed components while dashed lines represent selective components. f is the number of filters and s represents stride step.

1) Search Space: One of the core components of NAS is how to design a reasonable search space for the adjustable cell, as different architectures could lead to diversified results. To describe the search space symbolically, we adopt the notational convention that each cell is represented as a directed acyclic graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with N nodes. Each node $\mathbf{V}^{(i)}$ indicates the ith latent feature $\Phi^{(i)}(\mathbf{X})$ when using $\mathbf{X}$ as the input, and each edge $\mathbf{E}^{(i,j)}$ means a transformed operation $o^{(i,j)}(\cdot)$ chosen from a pre-defined operation pool

$$\mathcal{O} = \{o_k(\cdot), k = 1, \cdots, n\}, \quad (8)$$

which includes n candidate operations. For conveniently representing the selective edges in the search space, control parameters Λ are introduced:

$$\Lambda = \{\lambda^{(i,j)} | \lambda^{(i,j)} \in \{0, 1\}, i, j = 1, \cdots, N\}, \quad (9)$$

where 1 means the corresponding edge $\mathbf{E}^{(i,j)}$ is activated and 0 otherwise. Each latent feature in the graph can be calculated by using its predecessors, i.e.,

$$\Phi^{(j)}(\mathbf{X}) = \lambda^{(i,j)} o^{(i,j)}(\Phi^{(i)}(\mathbf{X})). \quad (10)$$

Compared with the traditional NAS, our search space is tailored to the inpainting detection and is mainly different in two aspects: 1) Selective operations in the pool $\mathcal{O}$ are pruned, remaining only three kinds of separable convolutions and identity transformation. This can reduce the computational cost while preserving the diversity of sampling models [54], [60], and 2) We limit the minimum number of transformations in the cell block, i.e., some edges are manually fixed. In particular, the cell is composed of 3×3 and 1×1 separable convolutions, where batch normalization (BN) [61] and ReLU [62] activation are embedded appropriately. A skip connection is introduced from beginning to end as a shortcut. As these operations have been proven to be very effective in helping the network learn feature representations [21], they are explicitly added to enhance the initial performance of the cell block. As for the remaining selective edges, we package them as choice blocks. The diagram of the cell architecture is shown in Fig. 4.

2) Search Algorithm: Similar to [47], [53], [54], we search the network architectures based on the one-shot NAS algorithm. Specifically, we first train a supernet containing all possible network architectures by enabling all selective edges in each choice block. Once the supernet is well-trained, we can sample a candidate architecture by simulating that each choice block contains only one kind of selective edge, and zeroing out the remaining ones. Recall that we have 3 choice blocks with 4 operations in the $\mathcal{O}$, and totally 10 cells in the network, which results in a search space with the complexity $4^{3\times 10} \approx 10^{18}$. Although the search space is huge, it is still possible to efficiently identify promising architectures from it. As the architectures are sampled independently from a fixed probability distribution [52], we can simulate the distribution through sampling a number of candidate architectures. To achieve a good tradeoff between the complexity and the effectiveness, we randomly screen 1000 architectures from Λ; a somewhat similar strategy was also adopted in [54]. These candidate architectures are then evaluated on a validation set mixed with multiple inpainting forgeries. Specifically, the validation set contains 5000 images generated by 5 randomly selected inpainting methods (details on inpainting methods are deferred to Section IV). This validation set naturally inherits the commonality of different inpainting methods, making the resulting architectures suitable for the inpainting detection task. The best-performing one among these candidates is chosen as the final architecture. Noted that the candidate architectures generally perform unsatisfactorily, as they have been trained with few steps; further fine-tuning is needed.

C. Decision Block

The role of the decision block is to transform the learned high-level features into low-level discriminative information, i.e., inpainting detection results. Apparently, at the pixel level, the detection results can be divided into two classes: positive class (inpainted pixel) and negative class (pristine pixel). During this process, misclassified pixels may be generated to form inaccurate detection, due to the ineffectiveness of convolutional neural networks in modeling long-term feature correlations [13]. To track this problem, many attention modules have been proposed and used recently in the decision phase of networks [3], [13], [14]. The main idea of the attention modules is simple but very efficient, which is to optimize specific features with the assistance of other features. Along this line, we propose to integrate novel attention modules (i.e., global attention and local attention) into the decision block for better generating the detection results. The global attention aims to reduce the number of misclassified pixels through a very effective technique in the classification task: minimizing the intra-class variance. Motivated by the observations that generally surrounding pixels are of the same class as the center pixel, we use local attention to improve the consistency of features within a specific region for generating more accurate detection results. The procedure of the global and local attentions is shown in Fig. 5, and the details are given below. For a more intuitive understanding, the feature maps before and after passing through the attention modules are visualized in Fig. 6. It can be seen that the attentions do help the model optimize the feature maps, so as to make the classification task easier. The rest of the decision block is mainly composed of three transposed convolutions (ConvT), where the BN [61], ReLU [62], and Sigmoid activation are embedded appropriately.

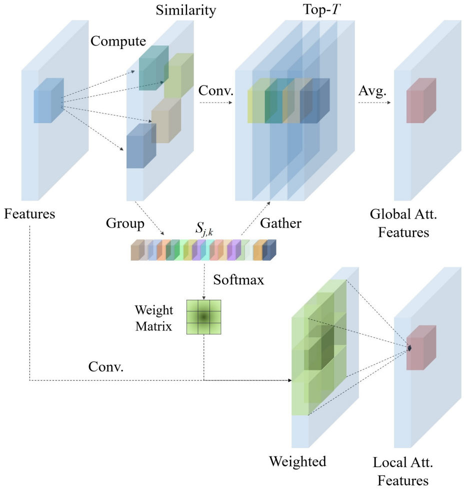
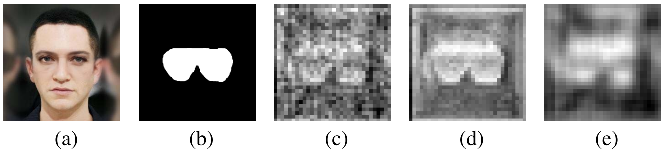


Fig. 5. Illustration of the global attention and local attention.



(a) (b) (c) (d) (e)

Fig. 6. Visualization of the feature maps output by the proposed global and local attention modules. (a) Input; (b) ground-truth inpainting mask; (c) feature map before passing any attention modules; (d) feature map after passing the global attention module; and (e) feature map after passing the local attention module.

1) Gloabl Attention: The global attention is motivated by an essence method for improving classification performance: reducing intra-group distances. Practically, we re-generate each feature with its several most similar features, so as to reduce the differences within the same class. Let $\Phi(\mathbf{X})$ be the feature map of the latent layer in the decision block when using $\mathbf{X}$ as the input. We extract all 1×1 patches $\{\mathbf{P}_j\}_{j=1}^{K}$ from $\Phi(\mathbf{X})$ and group them into a set $\mathcal{P}$. For each patch $\mathbf{P}_j \in \mathcal{P}$, its intra-cosine similarities within $\mathcal{P}$ can be computed as

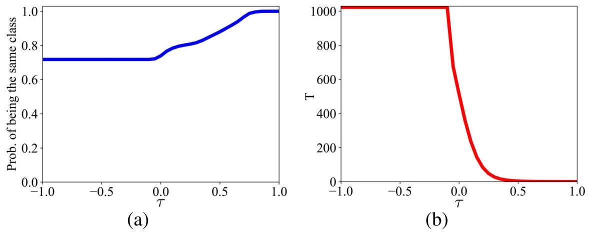
$$S_{j,k} = \left\langle \frac{\mathbf{P}_j}{||\mathbf{P}_j||}, \frac{\mathbf{P}_k}{||\mathbf{P}_k||} \right\rangle, \quad \mathbf{P}_k \in \mathcal{P}. \tag{11}$$

Upon computing all $S_{j,k}$'s, we can set a similarity threshold τ to select the top-T most similar patches for $\mathbf{P}_j$ from $\mathcal{P}$. Let $\mathcal{N} = \{n_1, \ldots, n_T\}$ record all the indexes of these top-T most similar patches. Then we have

$$\mathcal{N} = \Big\{ k | S_{j,k} \geq \tau \Big\}. \tag{12}$$

In practice, the process of similarity search can be conducted via a modified convolutional layer to reduce the computation burden caused by loop operations, as explained in [3], [14]. We then propose to update each $\mathbf{P}_j \in \mathcal{P}$ via the average of its corresponding top-T most similar patches:

$$\mathbf{P}_j^* = \frac{1}{T} \sum_{k \in \mathcal{N}} \mathbf{P}_k. \tag{13}$$



(a) (b)

Fig. 7. The relationship (a) between τ and the probability that the selected patches are in the same class; and (b) between τ and T.

Therefore, the updated $\mathbf{P}_j^*$ will increase the intra-class similarity along with the training processes, benefiting the ultimate inpainting detection task.

However, there are two potential problems when applying this global attention mechanism: 1) how to make sure that the top-T most similar patches belong to the same class; and 2) how to set an appropriate value for the parameter T (or equivalently the threshold τ), since the proportion of the inpainted and untouched regions could be varying. To answer these two questions, we take a data-driven approach by analyzing the statistics of 1000 images that are randomly selected from the training dataset. In Fig. 7 (a), we plot the relationship between the employed threshold τ and the probability that the corresponding selected top-T patches belong to the same class. As can be seen, with the increasing τ, the probability of belonging to the same class tends to increase as well. Meanwhile, in Fig. 7 (b), we also show how the value T varies with respect to the threshold τ. It can be observed that when τ is relatively significant, increasing τ leads to a decrease of T. In other words, when τ is very large, then the number of selected similar patches would be very small. Therefore, it is crucial to set the threshold τ appropriately by balancing two factors: 1) the probability of belonging to the same class should be high enough, and 2) the number of selected patches T should be sufficiently large as well. According to the above experiments, we empirically set $\tau = 0.5$, which corresponds to the case that the probability of belonging to the same class is around 0.9 and $T = 5.3$.

2) Local Attention: Inspired by the observation that adjacent pixels (features) are often highly correlated, we now propose a local attention module to better maintain the local consistency. Similar to the process of the global attention, we update each feature with its surrounding features in a weighted manner. To reflect the local correlation, the surrounding features in a small local window are exploited. Specifically, we define a weight matrix $\mathbf{W}_l$ of size $m \times m$, where $m = 5$, and convolve it with the patch $\mathbf{P}_j$ to obtain the updated feature $\mathbf{P}_j^*$. Namely,

$$\mathbf{P}_j^* = \mathbf{W}_l \otimes \mathbf{P}_j. \tag{14}$$

To determine an appropriate weight matrix $\mathbf{W}_l$ for the generalizable inpainting detection, we again adopt a data-driven approach by exploiting the 1000 training images used in the global attention. We calculate the *average* similarity matrix $\bar{\mathbf{S}}$ of size $m \times m$ over these 1000 images,[2] and have

$$\bar{\mathbf{S}} = \begin{bmatrix} .314 & .351 & .419 & .345 & .308 \\ .337 & .408 & .525 & .403 & .334 \\ .385 & .479 & 1 & .478 & .384 \\ .366 & .404 & .525 & .407 & .336 \\ .312 & .347 & .421 & .352 & .315 \end{bmatrix}. \tag{15}$$

where each element is computed according to (11). Upon having the similarity matrix $\bar{\mathbf{S}}$, the weight matrix $\mathbf{W}_l$ can be naturally determined by transforming $\bar{\mathbf{S}}$ via a softmax activation [63]:

$$\begin{aligned} \mathbf{W}_l &= \frac{\exp(\bar{\mathbf{S}})}{\sum \exp(\bar{\mathbf{S}})} \\ &= \begin{bmatrix} .0360 & .0374 & .0400 & .0371 & .0358 \\ .0368 & .0395 & .0445 & .0393 & .0367 \\ .0386 & .0424 & .0715 & .0424 & .0386 \\ .0368 & .0394 & .0444 & .0395 & .0368 \\ .0359 & .0372 & .0400 & .0374 & .0360 \end{bmatrix}. \end{aligned} \tag{16}$$

D. Fused Loss Function

We use the binary cross-entropy (BCE) loss to supervise the training of IID-Net, as its objective is to detect the inpainted/pristine region, which is essentially a binary classification task. More specifically, for a pair of ground-truth and predicted inpainting masks $(\mathbf{M}_g, \mathbf{M}_o)$, the BCE loss can be defined as

$$\begin{aligned} \mathcal{L}_B(\mathbf{M}_g, \mathbf{M}_o) = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\Big(\mathbf{M}_g(i,j)\log \mathbf{M}_o(i,j) \\ + (1-\mathbf{M}_g(i,j))\log(1-\mathbf{M}_o(i,j))\Big), \end{aligned} \tag{17}$$

where $\mathbf{M}_g(i,j)$ (similarly for $\mathbf{M}_o(i,j)$) denotes the (i,j)th element of $\mathbf{M}_g$ with a resolution $H \times W$.

However, in most of the inpainting-based forgeries, the inpainted regions are relatively smaller than the pristine ones, resulting in a class imbalance problem caused by the above loss function. Such imbalance would lead to a serious problem that the trained model tends to more likely classify the samples as pristine. To address this issue, we propose to incorporate the focal loss [64] into the BCE loss, forming a fused loss function. The idea of the focal loss is to add a modulating term to the standard cross entropy loss, so as to focus learning on hard examples and down-weight the numerous easy negatives. Typically, an α-balanced variant of the focal loss can be defined as:

$$\begin{aligned} &\mathcal{L}_F(\mathbf{M}_g, \mathbf{M}_o) \\ &= -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\Big(\alpha(1-\mathbf{M}_o(i,j))^{\gamma}\mathbf{M}_g(i,j)\log \mathbf{M}_o(i,j) \\ &\quad + (1-\alpha)(\mathbf{M}_o(i,j))^{\gamma}(1-\mathbf{M}_g(i,j))\log(1-\mathbf{M}_o(i,j))\Big). \end{aligned} \tag{18}$$

In particular, γ is a focusing parameter that can smoothly adjust the rate at which easy examples are down-weighted.

[2] We also have tried other settings of m, e.g., $m = 3$ and 7. Experimental results show that the setting of $m = 5$ gives slightly better results.

Clearly, when $\gamma = 0$, the focal loss is equivalent to the cross-entropy loss, and as γ becomes larger, the effect of the modulating factor is likewise increased. We evaluate different choices of $\gamma \in \{1, 2, 3\}$, and empirically find that $\gamma = 2$ works best in the experiments. Also, α is the weight assigned to the rare class for further adjusting the imbalance classes. Recall that the proportion of the inpainting regions (positive class) in each image is roughly between $0 \sim 50\%$. We hence set $\alpha = 0.75$ to balance the rare class. In fact, we also test other settings of $\alpha \in \{0.65, 0.85\}$, and observe similar performance.

Thus, the fused loss function can be written as

$$\mathcal{L}(\mathbf{M}_g, \mathbf{M}_o) = \mathcal{L}_B(\mathbf{M}_g, \mathbf{M}_o) + \mathcal{L}_F(\mathbf{M}_g, \mathbf{M}_o). \tag{19}$$

Instead of directly using the above fused loss function as the objective for optimizing IID-Net, we propose to apply a median filter to $\mathbf{M}_o$ before calculating the loss functions (17) and (18). Median filter is a non-linear statistical filter, often used to remove impulse noises. The intuition behind applying median filter to $\mathbf{M}_o$ is that we hardly tamper with only one or two isolated pixels in reality; namely, the inpainting area is usually continuous within a certain area. Hence, median filtering is a natural choice for "denoising" the isolated regions, which could boost the inpainting detection performance. Finally, the total loss function of IID-Net can be expressed as:

$$\mathcal{L}(\mathbf{M}_g, \mathbf{M}_o) = \mathcal{L}_B(\mathbf{M}_g, \mathbf{F} \otimes \mathbf{M}_o) + \mathcal{L}_F(\mathbf{M}_g, \mathbf{F} \otimes \mathbf{M}_o), \tag{20}$$

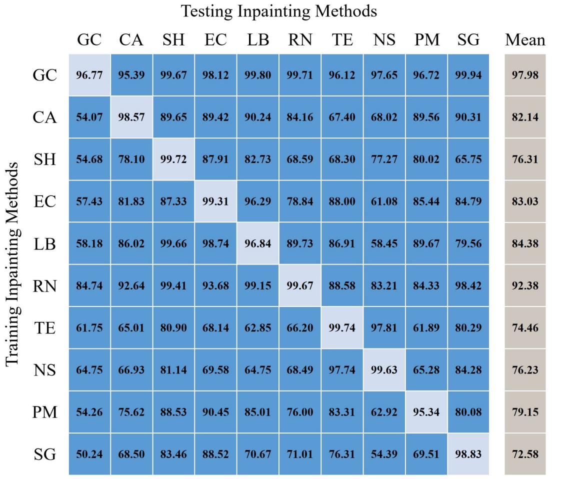
where $\mathbf{F}$ is a standard 3×3 median filter kernel.

IV. Training Data Selection and Experimental Results

The proposed IID-Net is implemented using the PyTorch framework. The training is performed on a desktop equipped with an Intel(R) Xeon(R) Gold 6130 CPU and three GTX 2080 GPUs. Adam [65] with default parameters is adopted as the optimizer. We set the batch size to 24 and have 2000 batches per epoch. The Area Under the receiver operating characteristic Curve (AUC, in the percentage format, if not otherwise stated) in the pixel domain, and the F1 score are used as the evaluation criteria. We train the network in an end-to-end manner with an initial learning rate 1e-4. The learning rate will be halved if the loss $\mathcal{L}$ fails to decrease for 10 epochs until the convergence. All the images used in the training phase are cropped to a size of 256 × 256, while there is no size limit for the inference phase. The average inference time of our model is 0.1724 seconds for a 256 × 256 RGB image. To embrace the concept of reproducible research, the code of our paper is available at **https://github.com/HighwayWu/InpaintingForensics**.

A. Training Data Selection and Generalizability Evaluation

The training data selection is crucial to the success of the IID-Net, especially for the generalizability to unseen inpainting approaches. For the generation of training data, `Places` [66] (JPEG lossy compression) and `Dresden` [67] (NEF lossless compression) datasets are used as base images $\mathbf{P}$, and the masks $\mathbf{M}_g$ are randomly sampled from [1]. The training dataset contains a total of 48K images (around 3 Gigabyte), half of which are randomly sampled from `Places` and the remaining half are randomly selected from `Dresden`. It should be noted that we keep the inpainting method $y(\cdot)$ unchanged when generating the training dataset, and regenerate the entire training dataset if $y(\cdot)$ is changed. In other words, we only use the training dataset generated by *one* inpainting method at a time in the actual training process. As for test images, we further introduce additional datasets, `CelebA` [68] and `ImageNet` [69], to increase the data diversity. Besides, we randomly generate a series of basic shapes, e.g., rectangles, circles, ellipses, and polylines, as additional test masks, which can locate at any position. These additional masks occupy approximately the same proportions as the masks generated from [1]. Regarding the inpainting methods, we here consider totally 10 representative ones, among which 6 are DL-based ones proposed in recent years, namely, GC [12], CA [13], SH [14], EC [2], LB [3] and RN [15]. The remaining 4 methods are traditional (non DL-based), which include TE [16], NS [17], PM [18], and SG [19]. Though TE and NS were published before 2005, they have been included into the OpenCV extension package as the built-in default inpainting methods. This implies that these two methods are widely used, and the results based on them would be very meaningful. A brief introduction of these inpainting methods has already been presented in Section II-A. Specifically, we build up a test dataset of 10K pairs of inpainted images and the corresponding ground-truth masks, where a variety of image categories and mask shapes are incorporated. In addition, each of the 10 aforementioned inpainting methods contributes 1000 inpainted images. The whole test dataset is downloadable from **https://github.com/HighwayWu/InpaintingForensics**, serving as a useful resource of our research community for fighting against the inpainting-based forgeries. Here we emphasize that the training dataset and test dataset have no overlap.



	GC	CA	SH	EC	LB	RN	TE	NS	PM	SG	Mean
GC	96.77	95.39	99.67	98.12	99.80	99.71	96.12	97.65	96.72	99.94	97.98
CA	54.07	98.57	89.65	89.42	90.24	84.16	67.40	68.02	89.56	90.31	82.14
SH	54.68	78.10	99.72	87.91	82.73	68.59	68.30	77.27	80.02	65.75	76.31
EC	57.43	81.83	87.33	99.31	96.29	78.84	88.00	61.08	85.44	84.79	83.03
LB	58.18	86.02	99.66	98.74	96.84	89.73	86.91	58.45	89.67	79.56	84.38
RN	84.74	92.64	99.41	93.68	99.15	99.67	88.58	83.21	84.33	98.42	92.38
TE	61.75	65.01	80.90	68.14	62.85	66.20	99.74	97.81	61.89	80.29	74.46
NS	64.75	66.93	81.14	69.58	64.75	68.49	97.74	99.63	65.28	84.28	76.23
PM	54.26	75.62	88.53	90.45	85.01	76.00	83.31	62.92	95.34	80.08	79.15
SG	50.24	68.50	83.46	88.52	70.67	71.01	76.31	54.39	69.51	98.83	72.58

Fig. 8. Inpainting detection performance of IID-Net. Each column shows the test results (AUC values) of the 10 models trained with different training datasets, and evaluated on the same test dataset resulting from one particular inpainting method.

In Fig. 8, we report the inpainting detection performance of our proposed IID-Net, where the 10K-sized test dataset is used. More specifically, for each column, we present the results of 10 models trained with different training datasets

TABLE I

QUANTITATIVE COMPARISONS BY USING AUC AND F1 AS CRITICS. FOR EACH COLUMN, THE HIGHEST VALUE IS HIGHLIGHTED IN **BLACK**, AND GRAY VALUE MEANS THAT THE INPAINTING METHODS USED IN THE TEST DATASET ARE USED IN THE TRAINING, I.E., NOT TESTING GENERALIZATION. "–" IN "RETRAIN" COLUMN INDICATES THAT THE MODELS ARE OFFICIALLY RELEASED WITHOUT RETRAINING

Models	Retrain	Critics	Test Dataset											Mean
			DL-based Inpainting						Traditional Inpainting					
			GC	CA	SH	EC	LB	RN	TE	NS	LR	PM	SG	
LDI	-	AUC	47.63	40.91	49.20	42.26	44.70	46.23	56.88	71.75	47.97	49.93	52.17	49.97
MT-Net	-	AUC	73.29	82.06	93.99	89.18	92.75	85.00	97.50	99.12	98.82	93.74	96.33	91.07
MT-Net	GC	AUC	96.31	75.44	73.58	61.92	62.27	87.38	90.93	89.03	97.12	90.59	86.09	82.79
HP-FCN	-	AUC	50.18	50.22	55.84	50.02	50.67	50.01	63.30	60.53	62.59	48.02	50.26	53.79
HP-FCN	GC	AUC	96.65	87.50	98.14	74.51	96.51	96.59	92.27	97.18	99.18	98.64	99.78	94.26
IID-Net	GC	AUC	96.77	**95.39**	**99.67**	**98.12**	**99.80**	**99.71**	96.12	97.65	**99.79**	**99.54**	**99.94**	**98.41**
LDI	-	F1	15.08	2.30	6.26	3.62	7.83	14.17	16.96	25.02	3.03	2.65	15.92	10.26
MT-Net	-	F1	14.17	28.80	72.63	67.55	60.14	35.22	82.31	90.67	81.35	49.98	66.93	59.07
MT-Net	GC	F1	92.10	19.02	32.78	10.62	2.38	10.80	**83.23**	86.75	27.37	13.11	45.84	30.17
HP-FCN	-	F1	0.04	0.22	0.38	0.05	0.42	1.98	0.14	0.61	0.92	0.08	0.01	0.44
HP-FCN	GC	F1	76.93	35.75	81.43	8.57	55.78	56.58	41.05	44.13	50.91	24.66	73.55	51.76
IID-Net	GC	F1	83.61	**81.46**	**94.13**	**87.95**	**96.14**	**94.41**	82.47	85.27	**87.28**	**75.74**	**94.78**	**87.57**

(including GC, CA, SH,...), where the test set resulting from one particular inpainting method is fixed. Take the first column for example. The 10 trained models are evaluated on the same test dataset prepared by using GC as the inpainting method. Similarly, for the second column, these 10 trained models are evaluated on the same test dataset prepared by using CA as the inpainting method. In other words, the comparison among these 10 models is still fair enough, as the test dataset in each column keeps the same. This also implies that there is no need to enforce the same source images and masks for the test dataset of different columns.

The diagonal AUC values in Fig.7 represent the detection results when the inpainting methods at the training and testing stages are the same, i.e., the scenario where the utilized inpainting method is known. In such a scenario, IID-Net achieves very desirable AUC performance (> 95%) for all cases. Meanwhile, the off-diagonal elements in Fig. 8 demonstrate the generalizability of IID-Net to unknown inpainting methods. It can be observed that the generalizability of IID-Net is vastly different when different training data are utilized. The best generalizability is achieved when GC is adopted at the training phase, with an average AUC 97.98%. We conjecture that it may be because GC incorporates multiple inpainting characteristics, enabling the trained model with good generalizability. However, the in-depth understanding needs to be combined with the interpretability of neural networks, which is beyond the scope of this paper. In fact, the networks trained on DL-based inpainting methods usually have more favorable generalizability than the ones trained on classic inpainting methods. It is safe to conclude that the DL-based and traditional inpainting algorithms leave somewhat common detectable traces that can be distinguished from untouched images. As the IID-Net trained with GC achieves the best generalizability, the training data generated with GC will be adopted in the following evaluations.

B. Quantitative Comparisons

For the comparison purpose, we adopt three state-of-the-art inpainting forensic approaches (i.e., LDI [20], MT-Net [21] and HP-FCN [10]) to detect the aforementioned inpainting methods, together with a newly proposed traditional inpainting scheme LR [70]. LDI is a traditional forensic approach that designs discriminative features for identifying the inpainted regions and uses post-processing for refining the detection results. MT-Net uses the powerful learning ability of neural networks to classify anomalous features of an input image, and attains good generalizability to various conventional manipulation types, including inpainting operations. HP-FCN is a high-pass fully convolutional network for locating the forged regions generated by deep inpainting. For fairness, we compare our proposed IID-Net not only with the pre-trained models officially released by competitors, but also with the models *retrained* on our training dataset. More specifically, we retrain the competitors' models based on their open-source codes, and strictly follow their training procedures, e.g., using the same batch size, learning rate, training epochs and strategies.

The quantitative comparisons in terms of the AUC value and F1 score (higher are better) in the pixel domain are presented in Table I. As can be observed, the detection performance of LDI on the traditional inpainting methods is relatively better than on the DL-based ones. On average, the AUC value is 49.97%, which is close to random guessing. This phenomenon is probably due to the fact that the manually designed features are not reliable, especially for unseen inpainting approaches. In contrast, the learning-based detection methods (MT-Net, HP-FCN, and IID-Net) achieve much better AUC performance. More specifically, the original MT-Net obtains 91.07% mean AUC value (and 59.07% F1 score), meaning that the pre-trained MT-Net is already able to (relatively) accurately detect the inpainted regions created by various inpainting algorithms. Surprisingly, the retrained MT-Net achieves a bit worse AUC/F1 performance (82.79%/30.17%), compared to the pre-trained model. This may be because the network architecture of MT-Net is specially designed according to their original training dataset. Furthermore, the pre-trained HP-FCN performs unsatisfactorily (53.79% in AUC and 0.44% in F1), mainly because this model is overfitted with a specific inpainting method and the fixed inpainting mask. The retained HP-FCN performs

TABLE II

Ablation Studies on Test Dataset by Using AUC as the Critic

Enhancement Block	**Conv**	✓														
	Bayar		✓													
	PF			✓												
	SRM				✓											
	Conv+Bayar					✓										
	Conv+Bayar+PF						✓			✓	✓	✓	✓	✓	✓	✓
	Conv+Bayar+PF+SRM							✓								
	Conv+Conv+Conv+Conv								✓							
Extraction Block	w/ Random NAS	✓	✓	✓	✓	✓	✓	✓	✓							
	w/ Fixed Kernels									✓						
	w/ Sub-optimal NAS										✓	✓	✓	✓	✓	✓
Decision Block	w/o Att.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓					
	Global Att.											✓				
	Local Att.												✓			
	Global & Local Att.													✓	✓	✓
Loss	w/o Filtering	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	w/o Focal Loss														✓	
	Filtering & Focal															✓
Number of Trainable Parameters ($\times 10^3$)		4598	4598	4599	4598	4598	4601	4603	4603	4601	4601	5191	5191	5781	5781	5781
Mean AUC		89.78	90.60	90.43	89.61	90.38	91.43	90.72	90.11	92.76	96.28	97.14	97.55	97.84	97.71	**97.98**

much better, with 94.26% AUC and 51.76% F1. Thanks to the adopted NAS algorithm for the architecture search and the global/local attention mechanisms, our proposed IID-Net leads to very accurate and consistent inpainting detection and localization, with 98.41% AUC and 87.57% F1.

C. Qualitative Comparisons

In addition to the quantitative comparisons, we also compare different models qualitatively, as shown in Fig. 9. More specifically, Fig. 9 gives several representative examples of using inpainting as a powerful tool to remove objects or even change the semantic meaning of an image. Due to the space limit, only the best-performing version of each competing model is shown (i.e., the pre-trained MT-Net and the retrained HP-FCN). It can be seen that LDI only performs relatively well in detecting the NS-based inpainting manipulations (the seventh row); but its detection performance degrades severely for other deep and traditional inpainting algorithms. For the pre-trained MT-Net, it can locate the forged regions well in some test datasets; but cannot achieve a consistent performance across all test datasets (e.g. see the second, third and fourth rows). The retrained HP-FCN generally can produce pretty good detection results; but inaccurate, broken or blurred detection results can be observed (the fourth, seventh and tenth rows). Besides, we also compare our model with a state-of-the-art general deepfake localization method, MAM [71]. As can be observed, MAM performs poorly for inpainting forensics tasks, which may be due to the gap between the traces of inpainting and deepfake. Compared with these models, our proposed IID-Net can learn more reasonable high-level semantics and generate a more precise predicted mask, primarily thanks to the carefully designed architectures as well as attention modules.

D. Ablation Studies

We now conduct the ablation studies of our proposed model by analyzing how each component (e.g., **Bayar/PF** layers, the NAS, and the attention modules) in the blocks contributes to the final inpainting detection results. To this end, we first prohibit the use of additional components in each block, and then evaluate the performance of different retrained models with appropriate settings. The obtained results are shown in Table II.

For the enhancement block, the pre-designed input layers (e.g., **SRM** [56], **PF** [10] and **Bayar** [55] layers) lead to better performance comparing with the traditional convolution (**Conv**). This is mainly because these input layers can enhance the inpainting traces, providing additional evidence for the subsequent detection. Among these input layers, **SRM** layer gives the worst performance, possibly because it uses fixed weights and cannot better adapt to generalizable inpainting traces. In addition, the combination of **Conv+Bayar+PF** could offer the best detection performance, and that is the reason for adopting such a combination as the input layer in our IID-Net.

For evaluating the performance of NAS, we initialize the extraction block with three different architectures: an architecture designed by standard heuristics (i.e., all the choice block are fixed with 3 × 3 separable convolutions), a random architecture and the sub-optimal architecture sampled from Λ (see (9)), respectively. It is found that the performance of the randomly sampled architecture is relatively poor (90.72% AUC), as it is likely to contain inappropriate structures, such as redundant identity transforms. The heuristic architecture is barely satisfactory (92.76% AUC), mainly because the structure is not specifically designed for inpainting detection. While the sub-optimal architecture sampled from Λ by using the NAS strategy performs much better (96.28% AUC). This implies that NAS can better design network structures, which are crucial for the inpainting detection task and its generalizability.

For further improving the performance, we embed the pre-designed attention modules in the decision block. From Table II, we can also observe that both global and local attention modules indeed can bring positive improvements. This is because attention modules can further optimize high-level features, boosting the eventual inpainting detection performance.

Finally, instead of directly using the loss function $\mathcal{L}$ given in (19) for optimization, we propose to incorporate median filtering operations in the loss function, as shown in (20). We notice from Table II that slight improvements can be brought by such refined loss function. We can also see that the focal loss term $\mathcal{L}_F$ can enhance the performance by adjusting the imbalance classes.

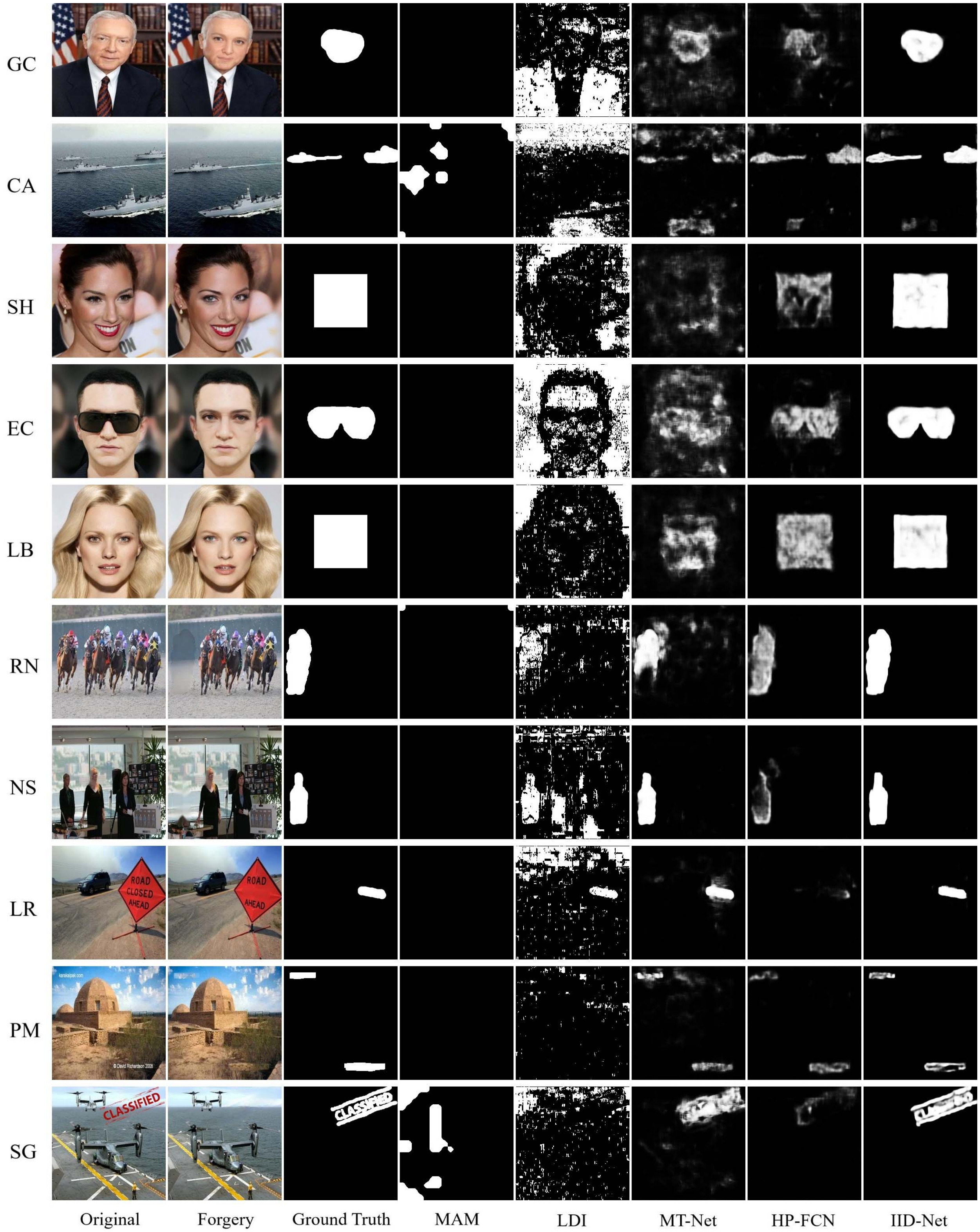


Fig. 9. Qualitative comparisons for detection of inpainting forgeries. For each row, the images from left to right are original, forgery (input), ground-truth, detection result (output) generated by MAM [71], LDI [20], MT-Net [21], HP-FCN [10] and our IID-Net, respectively. The forged images from top to bottom are inpainted by GC [12], CA [13], SH [14], EC [2], LB [3], RN [15], NS [17], LR [70], PM [18], and SG [19], respectively.

E. Robustness Evaluations

We would also like to evaluate the robustness of our IID-Net in detecting inpainting manipulations. This is very critical in real-world detection scenarios, because many post-processing operations, such as noise addition, resizing, and/or compression, could be applied to potentially hide the inpainting traces. To this end, we apply these post-processing operations with different types and magnitudes to the test datasets and report

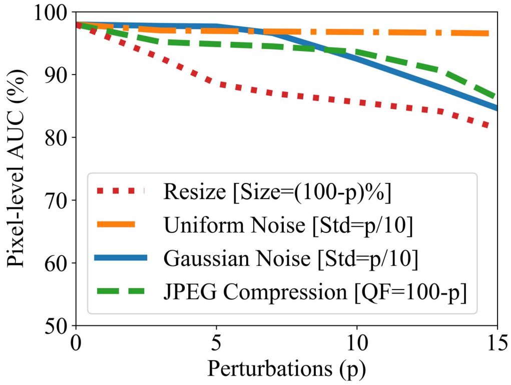


Fig. 10. Robustness evaluation of IID-Net against additive noise, resizing and compression.

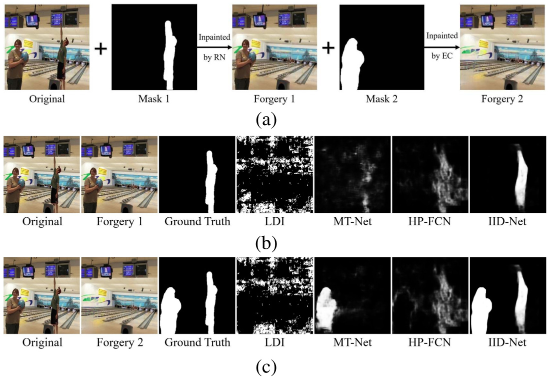


Fig. 11. Example of multiple inpainting manipulations. (a) Generation of Multiple Inpainting; (b) Detection results when “Forgery 1” is used as input; and (c) Detection results when “Forgery 2” is used as input.

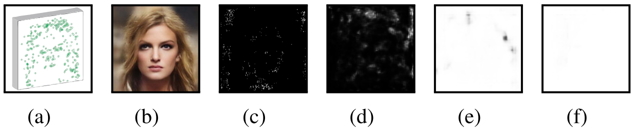


Fig. 12. Detection results of a fully DL-based synthesis. (a) The input SIFT features. (b) The image synthesized from (a). (c)-(f) The detection results of LDI [20], MT-Net [21], HP-FCN [10] and our IID-Net by using (b) as input. (A border is added for better viewing.).

the statistical detection results in Fig. 10. Here, we utilize a unified parameter p for different cases, e.g., for uniform and Gaussian noises, $p/10$ represents the standard deviation, while $100 - p$ stands for the QF employed in the JPEG compression. It is observed that the overall performance is good when the intensity of perturbations is relatively low, e.g., the performance is almost unchanged when performing JPEG compression with a quality factor of 95. With the increase of the perturbation intensity, the performance gradually drops. Such phenomenon agrees with the observations from [11], [21], [72]. The robustness evaluation results indicate that our IID-Net exhibits desirable robustness against the perturbations with small or medium magnitudes. Of course, when the perturbation intensity becomes further larger, the inpainting evidence will be destroyed, causing severe detection errors. But meanwhile, strong perturbations also lead to severely degraded images, which deviates the purpose of performing inpainting. Finally, data augmentation at the training phase by considering various distortions could be a viable solution for improving the robustness.

F. Challenging Cases

Before ending this section, we further evaluate the performance of our proposed IID-Net and other competing schemes under several challenging cases. One particular challenge arises when multiple regions in a single image are manipulated differently, e.g., by different inpainting algorithms. As indicated in [21], MT-Net would fail in such cases. To this end, we give an example by first inpainting an original image with a mask via RN [15], and the inpainting result is called “Forgery 1”. We then perform inpainting again on top of “Forgery 1” according to another mask via EC [2], and generate the “Forgery 2”. The two-round inpainting process is shown in Fig. 11 (a).

We now examine the inpainting detection performance of IID-Net and the competitors (MT-Net [21], LDI [20] and HP-FCN [10]) by using “Forgery 1” and “Forgery 2” as inputs, respectively. The detection results of these methods are demonstrated in Fig. 11 (b)-(c). As can be observed, LDI fails completely in both “Forgery 1” and “Forgery 2”. For MT-Net, it can only detect one of these inpainted regions at certain accuracy, while missing the other one. This phenomenon is consistent with the observation in [21]. One possibility is that the addition of the second type of inpainting changes the distribution of anomalous features, thereby affecting the discriminative capability of MT-Net. HP-FCN can detect both rounds of inpainting manipulations, but with severe detection errors. In contrast, our proposed IID-Net gives a much more accurate detection result not only in a single inpainting case, but also in multiple inpainting cases. We also have tested some other examples with different inpainting methods and more original images; similar conclusions can be drawn.

Another challenging case is that the whole image is completely regenerated by a DL-based network, i.e., the whole image is inpainted. For instance, a recent work [73] showed that an image can be reconstructed from Scale Invariant Feature Transform (SIFT) descriptors. Since the synthesis is totally generated by DL-based networks, it can also be regarded as a “global inpainting”, i.e., the ground-truth should be fully positive (white). We demonstrate the inpainting detection results of different methods in Fig. 12. It can be noticed that MT-Net can hardly locate the inpainted regions, mainly because its working principle relies on finding the anomalous features relative to the dominating ones. However, such relative dominance does not hold when the synthesis is completely composed of “anomalous” features. Fortunately, this limitation does not exist in our model, and the IID-Net gives almost perfect detection result even under this challenging case, as shown in Fig. 12 (f). The results of LDI and HP-FCN are also presented for comparison purposes.

Furthermore, we also present the localization results in Fig. 13 when pristine images without inpainting are inputted. In these cases, LDI and MT-Net generate relatively poor results, while HP-FCN and IID-Net can make predictions

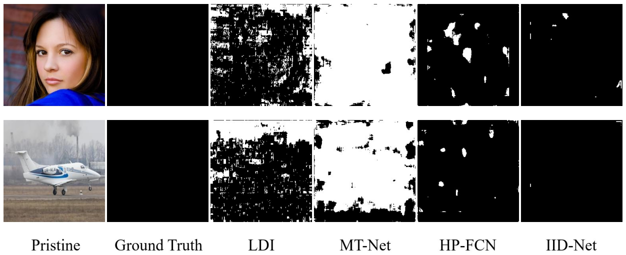


Fig. 13. Detection results of pristine images by thresholding at 0.1 for better visualization of the false alarm ratio.

almost perfectly. Here we use a lowered threshold (0.1) to binarize the results for better visualizations of the differences among competing methods, in terms of false alarm ratio.

V. Conclusion

In this paper, we propose the IID-Net, a novel DL-based forensic model for the detection of various image inpainting manipulations. The proposed model is designed with the assistance of the NAS algorithm and the embedded attention modules to optimize the latent high-level features. Experimental results are provided to not only demonstrate the superiority of our model against state-of-the-art competitors, but also verify that common artifacts are shared across diverse DL-based and traditional inpainting methods. This allows the forensic approaches to generalize well from one inpainting method to unseen ones without extensive retraining.

References

[1] G. L. Liu, F. A. Reda, K. J. Shih, T. C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 85–100.

[2] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edge-connect: Generative image inpainting with adversarial edge learning," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Oct. 2019, pp. 1–10.

[3] H. Wu, J. Zhou, and Y. Li, "Deep generative model for image inpainting with local binary pattern learning and spatial attention," 2020, *arXiv:2009.01031*. [Online]. Available: http://arxiv.org/abs/2009.01031

[4] 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.

[5] 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.

[6] Y. Fan, P. Carre, and C. Fernandez-Maloigne, "Image splicing detection with local illumination estimation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2940–2944.

[7] 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.*, 2018, pp. 101–117.

[8] Y. Li and J. Zhou, "Fast and effective image copy-move forgery detection via hierarchical feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1307–1322, May 2019.

[9] D. T. Trung, A. Beghdadi, and M.-C. Larabi, "Blind inpainting forgery detection," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2014, pp. 1019–1023.

[10] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8301–8310.

[11] S. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8695–8704.

[12] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4471–4480.

[13] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.

[14] Z. Y. Yan, X. M. Li, M. Li, W. M. Zuo, and S. G. Shan, "Shift-Net: Image inpainting via deep feature rearrangement," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–17.

[15] T. Yu *et al.*, "Region normalization for image inpainting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12733–12740.

[16] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[17] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. 355–362.

[18] J. Herling and W. Broll, "High-quality real-time video inpainting-with PixMix," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 6, pp. 866–879, Jun. 2014.

[19] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–10, Jul. 2014.

[20] H. Li, W. Luo, and J. Huang, "Localization of diffusion-based inpainting in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3050–3064, Dec. 2017.

[21] 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.

[22] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn.*, 2000, pp. 417–424.

[23] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, Aug. 2001.

[24] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 2001, pp. 341–346.

[25] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.

[26] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[27] D. Ding, S. Ram, and J. J. Rodriguez, "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1705–1719, Apr. 2019.

[28] J. Liu, S. Yang, Y. Fang, and Z. Guo, "Structure-guided image inpainting using homography transformation," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3252–3265, Dec. 2018.

[29] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[30] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, Jul. 2017.

[31] N. Wang, J. Li, L. Zhang, and B. Du, "MUSICAL: Multi-scale image contextual attention learning for inpainting," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3748–3754.

[32] S. Ge, C. Li, S. Zhao, and D. Zeng, "Occluded face recognition in the wild by identity-diversity inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3387–3397, Oct. 2020.

[33] C. Li, S. Ge, D. Zhang, and J. Li, "Look through masks: Towards masked face recognition with de-occlusion distillation," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 3016–3024.

[34] Q. Wu, S.-J. Sun, W. Zhu, G.-H. Li, and D. Tu, "Detection of digital doctoring in exemplar-based inpainted images," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2008, pp. 1222–1226.

[35] G.-S. Lin, M.-K. Chang, and Y.-L. Chen, "A passive-blind forgery detection scheme based on content-adaptive quantization table estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 421–434, Apr. 2011.

[36] I.-C. Chang, J. C. Yu, and C.-C. Chang, "A forgery detection algorithm for exemplar-based inpainting images using multi-region relation," *Image Vis. Comput.*, vol. 31, no. 1, pp. 57–71, Jan. 2013.

[37] K. Bacchuwar and K. Ramakrishnan, "A jump patch-block match algorithm for multiple forgery detection," in *Proc. Int. Mutli-Conf. Automat., Comput., Commun., Control Compressed Sens. (iMac4s)*, Mar. 2013, pp. 723–728.

[38] S. Bian, W. Luo, and J. Huang, "Exposing fake bit rate videos and estimating original bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2144–2154, Dec. 2014.

[39] Z. Liang, G. Yang, X. Ding, and L. Li, "An efficient forgery detection algorithm for object removal by exemplar-based image inpainting," *J. Vis. Commun. Image Represent.*, vol. 30, pp. 75–85, Jul. 2015.

[40] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic detection of object-based forgery in advanced video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2138–2151, Nov. 2016.

[41] F. J. Huang, X. C. Qu, H. J. Kim, and J. W. Huang, "Reversible data hiding in JPEG images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1610–1621, Sep. 2015.

[42] C. Feng, Z. Xu, S. Jia, W. Zhang, and Y. Xu, "Motion-adaptive frame deletion detection for digital video forensics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2543–2554, Dec. 2017.

[43] 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.

[44] 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.

[45] 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, Mar. 2021.

[46] T. Zhu *et al.*, "Forensic detection based on color label and oriented texture feature," in *Proc. Int. Conf. Brain Inspired Cogn. Syst.* Cham, Switzerland: Springer, 2019, pp. 383–395.

[47] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[48] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–18.

[49] M. Guo, Z. Zhong, W. Wu, D. Lin, and J. Yan, "IRLAS: Inverse reinforcement learning for architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9021–9029.

[50] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4780–4789.

[51] C. Xi *et al.*, "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 19–35.

[52] G. Bender, P. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 550–559.

[53] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–13.

[54] M. Guo, Y. Yang, R. Xu, Z. Liu, and D. Lin, "When NAS meets robustness: In search of robust architectures against adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 631–640.

[55] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2691–2706, Nov. 2018.

[56] 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. 1907–1915.

[57] 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.

[58] 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.

[59] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[60] S. Xie, A. Kirillov, R. Girshick, and K. He, "Exploring randomly wired neural networks for image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1284–1293.

[61] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[62] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[63] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, "Deep learning," *Nature*, vol. 1, no. 2, pp. 180–184, 2016.

[64] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[66] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[67] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *J. Digit. Forensic Pract.*, vol. 3, nos. 2–4, pp. 150–159, Dec. 2010.

[68] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: http://arxiv.org/abs/1710.10196

[69] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[70] Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang, "Patch-based image inpainting via two-stage low rank approximation," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 6, pp. 2023–2036, Jun. 2018.

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

[72] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[73] H. Wu and J. Zhou, "Privacy leakage of SIFT features via deep generative model based image reconstruction," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2973–2985, 2021.

Haiwei Wu (Student Member, IEEE) received the B.S. and M.S. degrees in computer science from the University of Macau, Macau, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Faculty of Science and Technology. His research interests include multimedia security, image processing, and machine learning.

Jiantao Zhou (Senior Member, IEEE) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the M.Phil. degree from the Department of Radio Engineering, Southeast University, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, in 2009. He held various research positions at the University of Illinois at Urbana-Champaign, The Hong Kong University of Science and Technology, and McMaster University. He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, and also the Interim Head of the newly established Centre for Artificial Intelligence and Robotics. He holds four granted U.S. patents and two granted Chinese patents. His research interests include multimedia security and forensics, multimedia signal processing, artificial intelligence, and big data. He has coauthored two papers that received the Best Paper Award from the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award from the IEEE International Conference on Multimedia and Expo in 2016. He is also an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.