IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received December 10, 2019, accepted December 30, 2019, date of publication January 3, 2020, date of current version January 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963745

Constrained Image Splicing Detection and Localization With Attention-Aware Encoder-Decoder and Atrous Convolution

YAQI LIU AND XIANFENG ZHAO, (Senior Member, IEEE)
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100093, China

Corresponding author: Xianfeng Zhao (zhaoxianfeng@iie.ac.cn)

This work was supported by the National Key Technology R&D Program under Grant 2019QY2202 and Grant 2019QY(Y)0207.

ABSTRACT Constrained image splicing detection and localization (CISDL) is a newly formulated image forensics task and plays an important role in verifying the generating process of a forged image. CISDL conducts dense matching between two investigated images and detects whether one image has forged regions pasted from the other. In this work, we introduce a novel attention-aware encoder-decoder deep matching network named as AttentionDM for CISDL. An encoder-decoder with atrous convolution is newly designed for hierarchical features dense matching and fine-grained masks generation. A novel attention-aware correlation computation module is built on normalization operations and informative features recalibration with channel attention blocks. Last but not least, VGG and ResNets are respectively formulated as feature extractors for comprehensive comparisons in CISDL. Extensive experiments demonstrate the superior performance of AttentionDM over the state-of-the-art methods.

INDEX TERMS Encoder-decoder, atrous convolution, normalization, channel attention.

I. INTRODUCTION

Malicious image forgery is becoming a global epidemic in recent years, due to the rapidly declining cost of digital cameras and quick development of sophisticated image editing tools [1]. Forgers may use forged images to produce fake news, spread rumors or give false testimony, which result in negative social impacts [2]. Image forensics, which seeks to distinguish forged images and prevent forgers from using forged images for unscrupulous business or political purposes [3], has attracted great attention in research and industrial communities [4].

A variety of image forensics methods investigate an individual image and detect its high-level [5]–[8] or low-level inconsistencies caused by image manipulation [1], [2], [9]. However, it is still a challenging task to accurately distinguish forged images, due to advanced image manipulation techniques and limited information provided by a single image [2], [3]. Moreover, these image forensics methods identify forged images or regions without providing the source of forged regions or specific tampering process, but these auxiliary evidences can provide more clues and make results more convincing in real applications [10].

Considering the afore-mentioned limitations, constrained image splicing detection and localization (CISDL) is newly formulated in the Media Forensics Challenge [10], [11]. Different from "conventional" splicing detection, "constrained" means that the inputs are two images: one is a probe image and the other is a potential donor image. CISDL can be depicted in Figure 1: given a probe image **P** and a potential donor image **D**, CISDL aims to detect if a region of **D** has been spliced into **P**, and consequently provide mask images $\mathbf{P_m}$ and $\mathbf{D_m}$ indicating the region(s) of **P** were spliced from **D**. In [12], Wu *et al.* proposed the pioneering CISDL approach, *i.e.*, Deep Matching and Validation Network (DMVN). DMVN generates correlation maps by comparing high-level low-resolution feature maps of VGG [13], and constructs an inception-based mask deconvolution module [14] to locate suspected regions. However, low-resolution feature maps restrict DMVN's ability to detect accurate boundaries and find small suspected regions. In [10], we proposed a deep matching network based on atrous convolution (DMAC) to generate high-quality candidate masks from high-resolution feature maps. The basic DMAC architecture

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Napoletano.

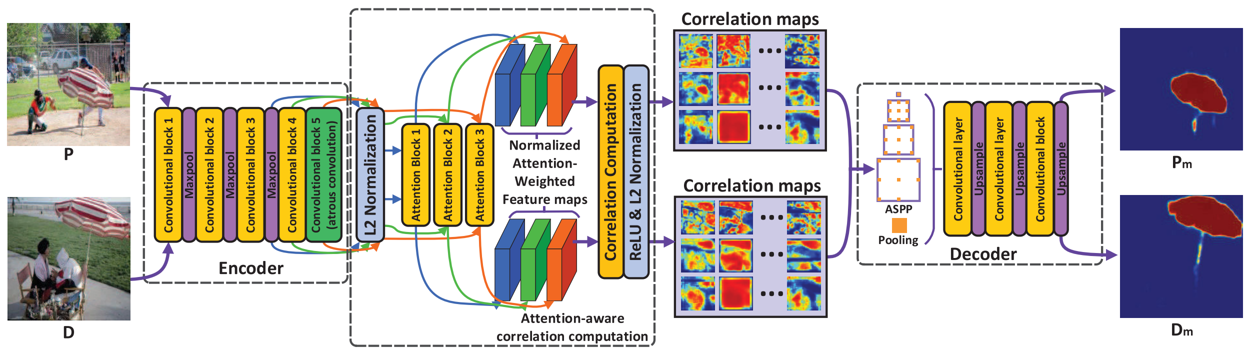

FIGURE 1. Overview of the proposed AttentionDM. In Encoder, three pairs of feature maps with the same size are generated by integrating atrous convolution. Each pair of feature maps are processed by L2 normalization and channel attention. The correlation computation with pre/post-normalization and attention blocks is called attention-aware correlation computation. Then, we use Decoder with ASPP (Atrous Spatial Pyramid Pooling) to generate fine-grained masks.

achieves significant improvements than DMVN, and a hybrid adversarial learning framework was proposed to further optimize the pretrained DMAC. Although massive computations are needed in the adversarial learning procedure, the performance can not be dramatically improved. Besides, the simple scalar product in correlation computation of DMVN and DMAC still limits the discriminative capability of deep matching.

In this work, we propose a novel attention-aware encoder-decoder deep matching network named as AttentionDM, as shown in Figure 1. AttentionDM adopts an encoder-decoder architecture with atrous convolution for fine-grained masks generation. Different from our previous work [10], we propose to construct a decoder with alternative convolutional and upsampling operations to recover the spatial information, instead of a single bilinear upsampling layer at the end in DMAC. During correlation computation, normalization operations are adopted to limit feature values to certain ranges and filter redundant features. A novel channel attention block is proposed to highlight channel-wise informative features in an innovative way, consequently a weighted scalar product can be conducted in the correlation computation procedure. To the best of our knowledge, we first propose to use an attention mechanism [15] in CISDL and adjust it to fit our task, *e.g.*, constructing an embedding network to extract channel-wise high-order features, integrating with our skip architecture [10], [16].

The main contributions of the proposed AttentionDM can be summarized in four folds:

- An encoder-decoder deep matching network with atrous convolution is newly designed for CISDL.
- Attention-aware correlation computation is proposed based on hierarchical feature normalization operations and channel attention blocks.
- ResNets are firstly formulated as the feature extractor in the CISDL task. Abundant comparisons are conducted between VGG and ResNets.
- Extensive experiments on public datasets demonstrate the superior performance of our AttentionDM.

II. RELATED WORK

Image Splicing Detection and Localization have been widely studied in recent years. Because the forger generically tries to satisfy both high-level and low-level consistency constraints during image manipulation, tampering detection and localization can also be conducted from two levels. In the aspect of high-level constraints, different cues can be investigated, *e.g.*, blur type inconsistency [6], shadows and lighting inconsistency [7], traces of perspective and geometry [8]. In certain contexts, forgery detection and localization methods based on high-level traces can achieve excellent performance, however they can not well adapt to complex and practical scenarios [7]. Furthermore, low-level signatures are exploited in a more general way, *e.g.*, photo-response non-uniformity noise [17], color filter array artifacts [18], JPEG coding traces [19], steganalysis features [20]. There are also many researches which are targeted at one specific type of forgery, *e.g.* copy-move [1], [21], seam carving [22]. Despite the tremendous progress so far, much potential and many more discoveries lie ahead because of the breakthrough in deep learning [23], [24], many CNN-based methods are investigated and achieve significant improvements [2], [5], [9], [25]. However, as we describe in Section I, these methods all investigate individual images and can not provide the source of splicing images.

Image Matching using global features extracted by Convolutional Neural Networks (CNNs) has attracted lots of attention [26], [27]. These methods conduct dense comparisons on high-level features extracted by CNNs. A variety of networks have been proposed for estimating inter-frame motion in videos [28] or instance-level homography estimation [29]. These methods attempt to find high-precision correspondences between images, while only need to search surrounding areas with limited appearance variation and background clutter. Otherwise, some deep matching methods were proposed for long-range category-level matching [27], [30]. These methods target at finding objects of the same category with similar appearance, which are quite different from CISDL [10]. Wu *et al.* firstly utilized deep matching techniques to solve the CISDL problem, and proposed DMVN [12]. In [10], we took advantage of the advanced techniques in fully convolutional neural networks [31], and proposed the DMAC network. Both DMVN and DMAC adopt a naive scalar product operation to compute the correlation maps between the two investigated images, and we try to make use of the attention mechanism to enhance this procedure.

Attention Mechanism can be viewed as a strategy to bias the allocation of available processing resources to the most informative components of an input signal [32]. It has been widely applied to recurrent neural networks (RNN) and long short term memory (LSTM) [33] to tackle sequential decision tasks. In [34], a sequence-to-sequence task was formulated as an encoder-decoder network, in which a source sentence was encoded into a fixed-length vector and then fed into a decoder network. To solve the bottleneck problem of the fixed-length vector, Bahdanau *et al.* [35] proposed to utilize the attention mechanism to dynamically generate the vectors. Consequently, a variety of attention-based models were proposed to solve the sequence-to-sequence tasks [15]. The attention mechanism is also applicable to image and video problems in computer vision [36], *e.g.* image classification [37], object detection [38], video classification [32], *etc.*

III. ATTENTIONDM

A. ENCODER-DECODER WITH ATROUS CONVOLUTION

1) BASIC ATTENTIONDM WITH VGG

Encoder-decoder with atrous convolution of AttentionDM is highly motivated by its great success in the semantic segmentation task [31], [39]. In our work, we investigate a decoder architecture with ASPP to capture multi-scale features and gradually generate fine-grained masks from correlation maps. And the correlation maps are computed from hierarchical large feature maps of an encoder with atrous convolution and a skip architecture.

Our detailed encoder-decoder architecture and parameter settings are presented in Figure 2. The encoder is a variant form of VGG [13] with atrous convolution. Let $y(i_s, j_s)$ denote the output of the atrous convolution of a 2-D input signal $x(i_s, j_s)$, and the atrous convolution can be computed as:

$$y(i_s, j_s) = \sum_{k_1, k_2} \phi(k_1, k_2) \times x(i_s + r_s k_1, j_s + r_s k_2) \qquad (1)$$

where $k_1, k_2 \in [-fl(\frac{K}{2}), fl(\frac{K}{2})]$ ($fl(\cdot)$ is a floor function), $\phi(k_1, k_2)$ denotes a $K \times K$ filter, rate r_s denotes the sampling stride of the input signal. Different from the original VGG, we remove the last two maxpooling layers, and adopt atrous convolution with $r_s = 2$ in the last convolutional block to keep their original field-of-views. A skip architecture is used to capture hierarchical information, and three groups of feature maps with the same scale can be generated, *i.e.*, $\mathbf{F}_3$, $\mathbf{F}_4$, $\mathbf{F}_5$, which are all used for correlation computation. In the decoder, atrous rates are set to $\{6, 12, 18\}$, and their feature maps are concatenated and fed into the subsequent layers which are constructed by alternative convolutional and

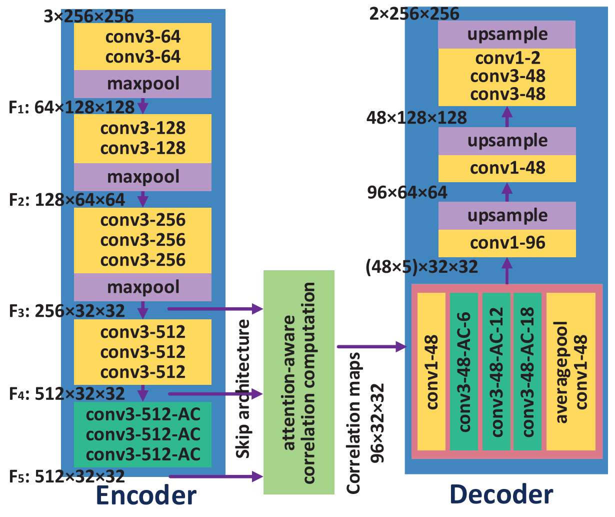

FIGURE 2. Encoder-decoder architecture and parameter settings based on VGG16. "conv3-64" denotes a convolutional layer with 3 × 3 filters and 64 output channels, "AC" denotes the atrous convolutional layer with a default setting $r_s = 2$ or "AC-6" means $r_s = 6$.

upsampling layers to gradually recover high-resolution fine-grained masks.

2) ATTENTIONDM BASED ON ResNet

As we all know, in computer vision tasks, *e.g.*, image classification, object detection and semantic segmentation, deep convolutional neural networks can significantly improve their performance [40]. However, in previous CISDL methods [10], [12], [41], they all use VGG16 as the basic feature extractor, and we do not know whether deep networks can improve the deep matching performance. Thus, we formulate the popular ResNet50 and ResNet101 [42] as the feature extractor, and utilize their hierarchical features by integrating atrous convolution. The detailed architectures with atrous convolution are presented in Table 1. In this formulation, we still can get 3 sets of feature maps with the same size, *i.e.*, $\mathbf{F}_3$, $\mathbf{F}_4$ and $\mathbf{F}_5$. The same as the basic architecture of VGG, these features are fed into correlation layers and mask generation layers.

B. ATTENTION-AWARE CORRELATION COMPUTATION

As shown in Figure 1, the key component of our AttentionDM is the attention-aware correlation computation module. In fact, it consists of three parts, *i.e.*, attention blocks, normalization operations and correlation computation. We first introduce our channel attention block. Then, we present the whole correlation computation procedure with normalization and attention operations.

1) CHANNEL ATTENTION BLOCK

Suppose we have two groups of c-channel $h \times w$ feature maps $\bar{\mathbf{F}}^{(k)} \in \mathbb{R}^{h \times w \times c}$ ($k \in \{1, 2\}$), and we flatten these feature maps to d-dimensional ($d = h \times w$) feature vectors and get $d \times c$ feature matrices $\bar{\mathbf{F}}^{(k)}_{\text{flat}}$. As these feature vectors have high

TABLE 1. Feature extractor formulated based on ResNets.

Output feature map	ResNet50	ResNet101
$\mathbf{F}_1$: 128×128	$7 \times 7, 64$, stride 2	
$\mathbf{F}_2$: 64×64	3×3 max pool, stride 2	
	$\begin{bmatrix} 1 \times 1, 64, \text{stride } 1 \\ 3 \times 3, 64, \text{stride } 1, r_s = 1 \\ 1 \times 1, 256, \text{stride } 1 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64, \text{stride } 1 \\ 3 \times 3, 64, \text{stride } 1, r_s = 1 \\ 1 \times 1, 256, \text{stride } 1 \end{bmatrix} \times 3$
$\mathbf{F}_3$: 32×32	$\begin{bmatrix} 1 \times 1, 128, \text{stride } 2 \\ 3 \times 3, 128, \text{stride } 1, r_s = 1 \\ 1 \times 1, 512, \text{stride } 1 \end{bmatrix} \times 1$	$\begin{bmatrix} 1 \times 1, 128, \text{stride } 2 \\ 3 \times 3, 128, \text{stride } 1, r_s = 1 \\ 1 \times 1, 512, \text{stride } 1 \end{bmatrix} \times 1$
	$\begin{bmatrix} 1 \times 1, 128, \text{stride } 1 \\ 3 \times 3, 128, \text{stride } 1, r_s = 1 \\ 1 \times 1, 512, \text{stride } 1 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128, \text{stride } 1 \\ 3 \times 3, 128, \text{stride } 1, r_s = 1 \\ 1 \times 1, 512, \text{stride } 1 \end{bmatrix} \times 3$
$\mathbf{F}_4$: 32×32	$\begin{bmatrix} 1 \times 1, 256, \text{stride } 1 \\ 3 \times 3, 256, \text{stride } 1, r_s = 2 \\ 1 \times 1, 1024, \text{stride } 1 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256, \text{stride } 1 \\ 3 \times 3, 256, \text{stride } 1, r_s = 2 \\ 1 \times 1, 1024, \text{stride } 1 \end{bmatrix} \times 23$
$\mathbf{F}_5$: 32×32	$\begin{bmatrix} 1 \times 1, 512, \text{stride } 1 \\ 3 \times 3, 512, \text{stride } 1, r_s = 4 \\ 1 \times 1, 2048, \text{stride } 1 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512, \text{stride } 1 \\ 3 \times 3, 512, \text{stride } 1, r_s = 4 \\ 1 \times 1, 2048, \text{stride } 1 \end{bmatrix} \times 3$

dimensions and contain strong spatial information, we propose to use a two-layer embedding network to extract high-order low-dimensional features. Referring to the definition in attention-based sentence embedding of natural language processing [33], our feature extraction network is called the embedding network to extract embedded features:

$$\begin{aligned} \mathbf{E}^{(k)} &= \text{Embed}(\bar{\mathbf{F}}^{(k)}_{\text{flat}}) \\ &= \delta(\mathbf{W}^{\text{E}}_2 \delta(\mathbf{W}^{\text{E}}_1 \bar{\mathbf{F}}^{(k)}_{\text{flat}} + \mathbf{b}^{\text{E}}_1) + \mathbf{b}^{\text{E}}_2) \end{aligned} \tag{2}$$

where $\mathbf{W}^{\text{E}}_1$, $\mathbf{W}^{\text{E}}_2$ denote the parameter matrices of two linear layers with corresponding bias terms $\mathbf{b}^{\text{E}}_1$ and $\mathbf{b}^{\text{E}}_2$, and $\mathbf{W}^{\text{E}}_1 \in \mathbb{R}^{\frac{d}{r} \times d}$, $\mathbf{W}^{\text{E}}_2 \in \mathbb{R}^{\frac{d}{r^2} \times \frac{d}{r}}$, $\mathbf{b}^{\text{E}}_1 \in \mathbb{R}^{\frac{d}{r}}$, $\mathbf{b}^{\text{E}}_2 \in \mathbb{R}^{\frac{d}{r^2}}$. δ refers to a ReLU function. r is a reduction ratio, and is set to 4 in our work. By constructing an embedding network, we get $d_e \times c$ embedded features $\mathbf{E}^{(k)}$, and $d_e = \frac{d}{r^2}$. Channel attention measures channel responses as follows:

$$\begin{aligned} \mathbf{a}^{(k)} &= \text{Atten}(\mathbf{E}^{(k)}) \\ &= \text{softmax}(\mathbf{w}^{\text{A}}_2 \tanh(\mathbf{W}^{\text{A}}_1 \mathbf{E}^{(k)})) \end{aligned} \tag{3}$$

where $\mathbf{W}^{\text{A}}_1 \in \mathbb{R}^{\frac{d_e}{r} \times d_e}$, $\mathbf{w}^{\text{A}}_2$ is a $\frac{d_e}{r}$-dimensional vector. $\mathbf{a}^{(k)}$ is a c-dimensional vector which is designed to indicate channel relations. By multiplying $\mathbf{a}^{(k)}$ and $\bar{\mathbf{F}}^{(k)} \in \mathbb{R}^{h \times w \times c}$, we can recalibrate informative channels to improve the discriminative capability of features. Details are presented as follows.

2) CORRELATION COMPUTATION

Let $\mathbf{F}^{(1)}$, $\mathbf{F}^{(2)}$ denote feature maps extracted by the encoder, and $\vec{f}^{(1)}(i_1, j_1) \in \mathbf{F}^{(1)}$, $\vec{f}^{(2)}(i_2, j_2) \in \mathbf{F}^{(2)}$ denote c-dimensional descriptors at specific coordinates. Before the attention and correlation computation, L2-normalization is conducted:

$$\bar{\vec{f}}^{(k)}(i_k, j_k) = \frac{\vec{f}^{(k)}(i_k, j_k)}{\|\vec{f}^{(k)}(i_k, j_k)\|_2} \tag{4}$$

By adopting L2-normalization, we can restrict the value ranges of descriptors, and adopt two normalized feature maps $\bar{\mathbf{F}}^{(1)}$, $\bar{\mathbf{F}}^{(2)}$. Then, we use $\bar{\mathbf{F}}^{(k)}$ ($k \in \{1, 2\}$) to get channel attention weights $\mathbf{a}^{(k)}$ based on Eq. (2) and Eq. (3). Thus, we can get channel attention weighted feature maps as follows:

$$\ddot{\mathbf{F}}^{(k)} = \mathbf{a}^{(k)} \cdot \bar{\mathbf{F}}^{(k)}, \quad k = 1, 2 \tag{5}$$

By constructing attention blocks, we can recalibrate informative features and improve the descriminative capabilities of features. In fact, channel attention has been adopted for enhancing image classification in [37]. They construct a SE block for each convolutional block, while we only modulate three groups of hierarchical features for the following correlation computation. Thus, we can elaborately construct a more complex architecture as Eq. (2) and Eq. (3), instead of a lightweight gating mechanism in [37].

The correlation maps $\mathbf{C}^{(12)}$ are generated by comparing $\ddot{\vec{f}}^{(1)}(i_1, j_1) \in \ddot{\mathbf{F}}^{(1)}$ and $\vec{f}^{(2)}(i_2, j_2) \in \ddot{\mathbf{F}}^{(2)}$ under strong spatial restrictions:

$$\mathbf{C}^{(12)}(i_{12}, j_{12}, m_{12}) = \ddot{\vec{f}}^{(1)}(i_1, j_1)^T \ddot{\vec{f}}^{(2)}(i_2, j_2) \tag{6}$$

in which

$$\begin{aligned} i_2 &= \text{mod}(i_1 + i_t, h) \\ j_2 &= \text{mod}(j_1 + j_t, w) \\ i_{12} &= i_1, \quad j_{12} = j_1, \; m_{12} = w i_t + j_t \end{aligned} \tag{7}$$

All the compared feature locations in the same channel m_{12} are under the same translation (i_t, j_t), $i_1, i_2, i_t \in [0, h)$ and $j_1, j_2, j_t \in [0, w)$. With the correlation maps $\mathbf{C}^{(12)}$ at our hands, match pooling operations are conducted to suppress uncorrelated information in $\mathbf{C}^{(12)}$. Average, maximum and sorted correlation maps are generated:

$$\mathbf{C}^{(12)}_{\text{avg}}(i_{12}, j_{12}, 0) = \frac{1}{h \times w} \sum_{p_{12}} \mathbf{C}^{(12)}(i_{12}, j_{12}, p_{12}) \tag{8}$$

Algorithm 1 Attention-Aware Correlation Computation

```
Input: Image $\mathbf{I}^{(1)}$ and $\mathbf{I}^{(2)}$
1:  “Hierarchical features extraction:”
2:  $\mathbf{F}_3^{(1)}, \mathbf{F}_4^{(1)}, \mathbf{F}_5^{(1)} = \text{Encoder}(\mathbf{I}^{(1)})$
3:  $\mathbf{F}_3^{(2)}, \mathbf{F}_4^{(2)}, \mathbf{F}_5^{(2)} = \text{Encoder}(\mathbf{I}^{(2)})$
4:  “Attention weighted feature maps generation for hierarchical features:”
5:  for $l = 3$ to $5$ do
        “L2 normalization of Eq. (4)”
6:      $\bar{\mathbf{F}}_l^{(1)} = \text{L2\_norm}(\mathbf{F}_l^{(1)})$
7:      $\bar{\mathbf{F}}_l^{(2)} = \text{L2\_norm}(\mathbf{F}_l^{(2)})$
        “Refer to Eq. (2), Eq. (3) and Eq. (5)”
8:      $\ddot{\mathbf{F}}_l^{(1)} = \text{Atten}_l(\text{Embed}_l(\bar{\mathbf{F}}_l^{(1)})) \cdot \bar{\mathbf{F}}_l^{(1)}$
9:      $\ddot{\mathbf{F}}_l^{(2)} = \text{Atten}_l(\text{Embed}_l(\bar{\mathbf{F}}_l^{(2)})) \cdot \bar{\mathbf{F}}_l^{(2)}$
10: end for
11: “Correlation computation based on hierarchical attention weighted feature maps:”
12: for $l = 3$ to $5$ do
        “Refer to Eq. (11) based on Eq. (6)-(10)”
13:     $\hat{\mathbf{C}}_l^{(12)} = \text{Corr}(\ddot{\mathbf{F}}_l^{(1)}, \ddot{\mathbf{F}}_l^{(2)})$
14:     $\hat{\mathbf{C}}_l^{(11)} = \text{Corr}(\ddot{\mathbf{F}}_l^{(1)}, \ddot{\mathbf{F}}_l^{(1)})$
15:     $\hat{\mathbf{C}}_l^{(21)} = \text{Corr}(\ddot{\mathbf{F}}_l^{(2)}, \ddot{\mathbf{F}}_l^{(1)})$
16:     $\hat{\mathbf{C}}_l^{(22)} = \text{Corr}(\ddot{\mathbf{F}}_l^{(2)}, \ddot{\mathbf{F}}_l^{(2)})$
        “Concatenate correlation maps”
17:     $\mathbf{C}_l^{(1)} = \{\hat{\mathbf{C}}_l^{(12)}, \hat{\mathbf{C}}_l^{(11)}\}$
18:     $\mathbf{C}_l^{(2)} = \{\hat{\mathbf{C}}_l^{(21)}, \hat{\mathbf{C}}_l^{(22)}\}$
19: end for
    “Concatenate hierarchical correlation maps”
20: $\mathbf{C}^{(1)} = \{\mathbf{C}_3^{(1)}, \mathbf{C}_4^{(1)}, \mathbf{C}_5^{(1)}\}$
21: $\mathbf{C}^{(2)} = \{\mathbf{C}_3^{(2)}, \mathbf{C}_4^{(2)}, \mathbf{C}_5^{(2)}\}$
    “ReLU and L2 normalization”
22: $\bar{\mathbf{C}}^{(1)} = \text{L2\_norm}(\max(\mathbf{C}^{(1)}, 0))$
23: $\bar{\mathbf{C}}^{(2)} = \text{L2\_norm}(\max(\mathbf{C}^{(2)}, 0))$
Output: Correlation maps $\bar{\mathbf{C}}^{(1)}$ and $\bar{\mathbf{C}}^{(2)}$ of $\mathbf{I}^{(1)}$ and $\mathbf{I}^{(2)}$
```

$$\mathbf{C}_{\max}^{(12)}(i_{12}, j_{12}, 0) = \arg\max_{p_{12}}\{\mathbf{C}^{(12)}(i_{12}, j_{12}, p_{12})\} \tag{9}$$

$$\mathbf{C}_{\text{srt}}^{(12)}(i_{12}, j_{12}, p) = \mathbf{C}^{(12)}(i_{12}, j_{12}, p_t), \quad p_t \in \text{Top\_T\_index} \times (\text{sort}_{p_{12}}(\text{sum}(\mathbf{C}^{(12)}(:, :, p_{12})))) \tag{10}$$

where Top_T_index(·) denotes the function which selects indexes of the top-T values. Finally, we can get the output feature maps $\hat{\mathbf{C}}^{(12)} = \{\mathbf{C}_{\text{avg}}^{(12)}, \mathbf{C}_{\max}^{(12)}, \mathbf{C}_{\text{srt}}^{(12)}\}$, and $\hat{\mathbf{C}}^{(12)} \in \mathbb{R}^{h \times w \times (\mathrm{T}+2)}$, in which 2 dimensions are the average and max correlation maps, and the other T dimensions are the sorted maps. The afore-mentioned procedure (Eq. (6)-(10)) is denoted as:

$$\hat{\mathbf{C}}^{(12)} = \text{Corr}(\ddot{\mathbf{F}}^{(1)}, \ddot{\mathbf{F}}^{(2)}) \tag{11}$$

With three pairs of feature maps (in different levels as shown in Figure 2, ResNets are shown in Table 1) as inputs, *i.e.*, $\mathbf{F}_3^{(k)}, \mathbf{F}_4^{(k)}, \mathbf{F}_5^{(k)}$, the attention-aware correlation computation procedure can be summarized as Algorithm 1.

The generated raw correlation maps $\mathbf{C}^{(k)}$ are followed by a ReLU layer to zero out negative values. The consideration is that features of close correlated regions should have the same sign, thus the correlation values should be positive. We zero out negative values to discard weak correlated regions and reduce computational costs. Then these maps are processed by L2-normalization (Eq. (4)) to get the normalized correlation maps $\bar{\mathbf{C}}^{(k)}$. Finally, $\bar{\mathbf{C}}^{(k)}$ are fed into the decoder based on ASPP introduced in Section III-A to generate the final mask.

IV. EXPERIMENTS

A. STEP-BY-STEP ANALYSES OF LOCALIZATION PERFORMANCE

Localization performance is evaluated on our released synthetic testing foreground pairs [43]. According to the ratios r_{pa} of the pasted areas, the image pairs are divided into three sets, namely Difficult ($1\% \le r_{\text{pa}} < 10\%$), Normal ($10\% \le r_{\text{pa}} < 25\%$), and Easy ($25\% \le r_{\text{pa}} < 50\%$). For each set, 3000 image pairs are generated with annotated ground-truth masks. Localization performance is evaluated by the pixel-level IoU (Intersection over Union) [40], MCC (Matthews Correlation Coefficient), NMM (Nimble Mask Metric) [11]. We compute the average IoU, MCC and NMM of all the tested image pairs. **Note that since the state-of-the-art CISDL methods [10], [12], [41] all adopt VGG as the basic feature extractor, the default AttentionDM adopts VGG and is directly denoted as “AttentionDM”. Models using ResNet50/ResNet101 are denoted as “AttentionDM-ResNet50/ResNet101”.**

Compared with our previous work [10], three major improvements are made. In this section, we conduct a detailed step-by-step analysis on the synthetic testing sets:

• Firstly, we test the effectiveness of the proposed encoder-decoder architecture with atrous convolution, *i.e.*, “Encoder-Decoder” in Table 2. “Encoder-Decoder” directly utilizes convolutional features extracted from encoder for correlation computation without L2-normalization and channel attention. “Encoder-Decoder” can already achieve better performance than DMVN and DMAC. Especially, a huge leap is achieved on the Difficult set, which demonstrates its great ability to detect small regions.

• Then, we attempt to normalize input hierarchical features, and process computed correlation maps using ReLU and L2-normalization, *i.e.*, “Encoder-Decoder-Norm”. The localization scores are further improved, and the basic encoder-decoder architecture with normalization builds a solid foundation for our outstanding performance.

• Finally, we evaluate our channel attention block by building it on “Encoder-Decoder-Norm”, *i.e.*, “AttentionDM”. Figure 3 provides IoU scores across training iterations on the Difficult set. Our channel attention block can yield consistent improvements over the basic architecture both in the training procedure (Figure 3) and the final scores (Table 2). We successfully propose an effective channel attention block which can measure channel relations and recalibrate channel-

TABLE 2. Step-by-step analyses on the synthetic testing sets.

Method	Difficult			Normal			Easy		
	IoU	MCC	NMM	IoU	MCC	NMM	IoU	MCC	NMM
DMVN [12]	0.2772	0.3533	-0.4382	0.6818	0.7570	0.4042	0.8198	0.8544	0.6770
DMAC [10]	0.5114	0.6308	0.0335	0.8279	0.8815	0.6840	0.9222	0.9395	0.8685
DMAC-adv [10]	0.5433	0.6584	0.1026	0.8317	0.8833	0.6877	0.9237	0.9411	0.8655
Encoder-Decoder	0.6130	0.7269	0.2434	0.8572	0.9041	0.7391	0.9417	0.9553	0.8997
Encoder-Decoder-Norm	0.6973	0.7904	0.4250	0.8852	0.9189	0.7961	0.9591	0.9685	0.9358
AttentionDM	**0.7228**	**0.8108**	**0.4793**	**0.8980**	**0.9320**	**0.8253**	**0.9602**	**0.9693**	**0.9388**

TABLE 3. Sliding window based matching strategy evaluation on the synthetic testing sets.

Method	Difficult			Normal			Easy		
	IoU	MCC	NMM	IoU	MCC	NMM	IoU	MCC	NMM
DMAC-adv [10]	0.5433	0.6584	0.1026	0.8317	0.8833	0.6877	0.9237	0.9411	0.8655
DMAC-adv-SR256 [10]	0.6426	0.7496	0.3115	0.8521	0.8970	0.7422	0.9368	0.9504	0.9024
DMAC-adv-SR128 [10]	0.6911	0.7930	0.4198	0.8602	0.9047	0.7742	0.9349	0.9484	0.9108
AttentionDM	0.7228	0.8108	0.4793	0.8980	0.9320	0.8253	**0.9602**	**0.9693**	0.9388
AttentionDM-SR256	0.7557	0.8387	0.5608	**0.9005**	**0.9339**	0.8410	0.9597	0.9686	**0.9465**
AttentionDM-SR128	**0.7608**	**0.8477**	**0.5819**	0.8953	0.9313	**0.8416**	0.9540	0.9639	0.9422

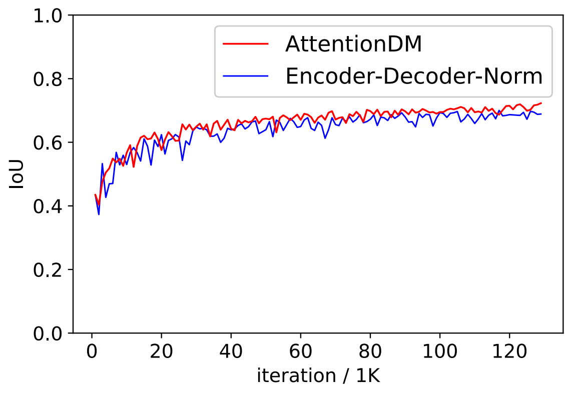

FIGURE 3. IoU scores across training iterations on the difficult set of synthetic testing sets.

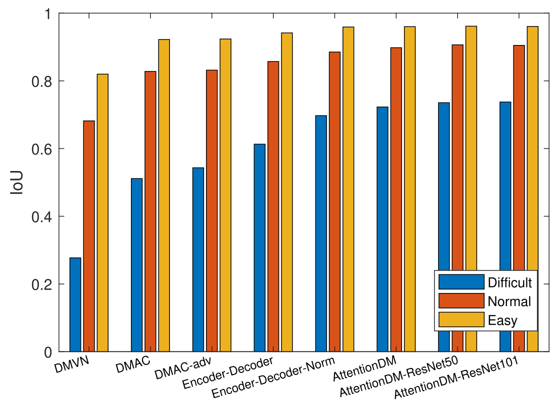

FIGURE 4. IoU scores comparisons on the synthetic testing sets.

wise informative features. Our channel attention block tightly integrates with our correlation computation and skip architectures. AttentionDM only builds three channel attention blocks (Figure 1) with a slight increase of the parameter number, while achieves a steady improvement of localization scores.

1) SLIDING WINDOW BASED MATCHING STRATEGY EVALUATION

In [10], we propose a sliding window based matching strategy to proccess high-resolution images. With the efficiency sacrifice (refer to Table 4 in [10]), they can achieve similar results with our Encoder-Decoder-Norm model. So, we also evaluate the effectiveness of this strategy on our AttentionDM model. Since this strategy contains two stages, *i.e.*, sliding matching and resizing matching, the methods adopted this strategy are annotated with postfix "-SR256/128". "256/128" denotes the sliding stride. Since with the decrease of the sliding stride, the computational complexity increases exponentially, we only test stride 256 and 128 referring to the experiments in [10]. The results are shown in Table 3. With the help of this strategy, the localization scores can be obviously improved on the Difficult set. In Normal and Easy sets, the sliding window based versions can also achieve comparable performance. Because AttentionDM can already achieve superior performance, and has great ability to detect small regions and accurate boundaries, with the help of sliding window based matching, the score improvements are not as large as DMAC, *e.g.*, IoU scores $0.6911 - 0.5433 = 0.1478$ vs. $0.7608 - 0.7228 = 0.0380$ on the Difficult set. Even though, our sliding window based version can achieve the best performance to detect small regions.

2) ResNet FEATURE EXTRACTOR EVALUATION

AttentionDM with different feature extractors and the corresponding sliding window versions are evaluated on the

TABLE 4. Feature extractor evaluation on the synthetic testing sets.

Method	Difficult			Normal			Easy		
	IoU	MCC	NMM	IoU	MCC	NMM	IoU	MCC	NMM
AttentionDM	0.7228	0.8108	0.4793	0.8980	0.9320	0.8253	0.9602	0.9693	0.9388
AttentionDM-ResNet50	0.7355	**0.8215**	0.4982	**0.9065**	**0.9387**	0.8357	**0.9615**	**0.9704**	0.9373
AttentionDM-ResNet101	**0.7375**	0.8192	**0.5113**	0.9049	0.9370	**0.8374**	0.9607	0.9697	**0.9380**
AttentionDM-SR256	0.7557	0.8387	0.5608	0.9005	0.9339	0.8410	0.9597	0.9686	0.9465
AttentionDM-ResNet50-SR256	0.7724	**0.8525**	0.5850	**0.9131**	**0.9432**	**0.8581**	**0.9649**	**0.9727**	**0.9507**
AttentionDM-ResNet101-SR256	**0.7748**	0.8508	**0.5972**	0.9111	0.9412	0.8564	0.9643	0.9723	0.9498
AttentionDM-SR128	0.7608	0.8477	0.5819	0.8953	0.9313	0.8416	0.9540	0.9639	0.9422
AttentionDM-ResNet50-SR128	0.7770	0.8610	0.6010	**0.9124**	**0.9440**	**0.8660**	0.9616	0.9698	0.9494
AttentionDM-ResNet101-SR128	**0.7939**	**0.8711**	**0.6453**	**0.9124**	0.9432	**0.8660**	**0.9631**	**0.9711**	**0.9508**

TABLE 5. Time complexity analyses.

	Time/s	Parameters	Framework
DMVN	0.2968	10,473,788	Keras/Theano
DMAC/DMAC-adv	0.0288	14,920,520	PyTorch
AttentionDM	0.0306	15,758,162	PyTorch
AttentionDM-ResNet50	0.0543	24,660,493	PyTorch
AttentionDM-ResNet101	0.0798	43,652,621	PyTorch

synthetic testing sets, as shown in Table 4. With more complicated feature extractors, the localization performance can be improved slightly. However, the parameter number and computing time obviously increase. Especially, AttentionDM-ResNet50 and AttentionDM-ResNet101 can achieve comparable performance, while AttentionDM-ResNet101 has much more parameters. So we do not recommend the use of ResNet101 for CISDL. In image classification and other corresponding tasks, deep networks can provide richer high-order semantic information. While in the CISDL task, this high-order information is not very useful, and discriminative features with more spatial information are more helpful. For comprehensive comparisons, we compare the IoU scores of CISDL methods and our variants on the synthetic testing sets in Figure 4. It can be clearly seen that AttentionDM achieves a siginificant performance leap, while there is no big gap between the VGG version and ResNets versions.

3) COMPLEXITY ANALYSES

The testing time, parameters numbers and implemented frameworks are reported in Table 5. All the experiments are conducted on a machine with Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz, 64GB RAM and a single GPU (TITAN X). With slightly more parameters, AttentionDM is slightly slower than DMAC. While AttentionDM indeed achieves significant performance improvements. With more complicated feature extrators and more parameters, the ResNets versions are slower than the VGG version, *i.e.*, AttentionDM.

B. DETECTION PERFORMANCE COMPARISONS

We evaluate the detection performance of AttnetionDM on (1) *The paired CASIA dataset*: In [12], they generated $3,642$ positive samples by pairing the $1,821$ spliced images in CASIA TIDEv2.0 dataset with their true donor images, and collected $5,000$ negative samples by randomly pairing $7,491$ color images from the same CASIA-defined content category. For the lack of ground truth masks, this dataset is designed for evaluating detection performance [12]. (2) *The MFC2018 dataset*: There are $1,327$ positive image pairs and $16,673$ negative pairs in the evaluation dataset of Media Forensics Challenge 2018 [11]. Detection performance is quantitatively evaluated, and the localization performance is evaluated by visual comparisons for the large number of negative pairs and some imperfect ground-truths. (3) *The PS-Battles dataset*: it has $11,142$ subsets consisting of the original image and several corresponding derivative fake images for a total of $102,028$ images which are collected from the online Reddit community of Photoshop battles [44]. The detection performance is measured by the precision, recall, F1-score, AUC (Area Under Curve), EER (Equal Error Rate) and detection rate [10].

1) CASIA

In [12], the authors compared DMVN with copy-move forgery detection methods [45]–[48] for the lack of other CISDL methods. In the comparison on CASIA, we directly borrow their scores from [12]. Using our AttentionDM, the forged probabilities are computed as: for each generated mask, we compute the average score $\{s^{(k)}|k = 1, 2\}$ of the detected regions, and the final forged probability is computed as their mean value $(s^{(1)} + s^{(2)})/2$. As shown in Table 6, detection scores on CASIA have been lifted to a new level by AttentionDM. All our scores of AttentionDM, *i.e.*, precision, recall, F1-measure, and AUC, are greater than 0.9. AttentionDM can achieve the highest recall, F1-score and AUC scores. Visual comparisons are provided in Figure 5, in which AttentionDM achieve clearly better performance. It has strong ability to detect small regions and accurate boundaries, and it is robust to deformation and rotation changes. While the ResNet50 version can achieve higher precision with lower recall, its F1-score and AUC are slightly lower than the VGG version. The detection performance of the ResNet101 version is even worse. Since the majority of images in CASIA are smaller than 512×512, we do not test the sliding strategy in this dataset [10].

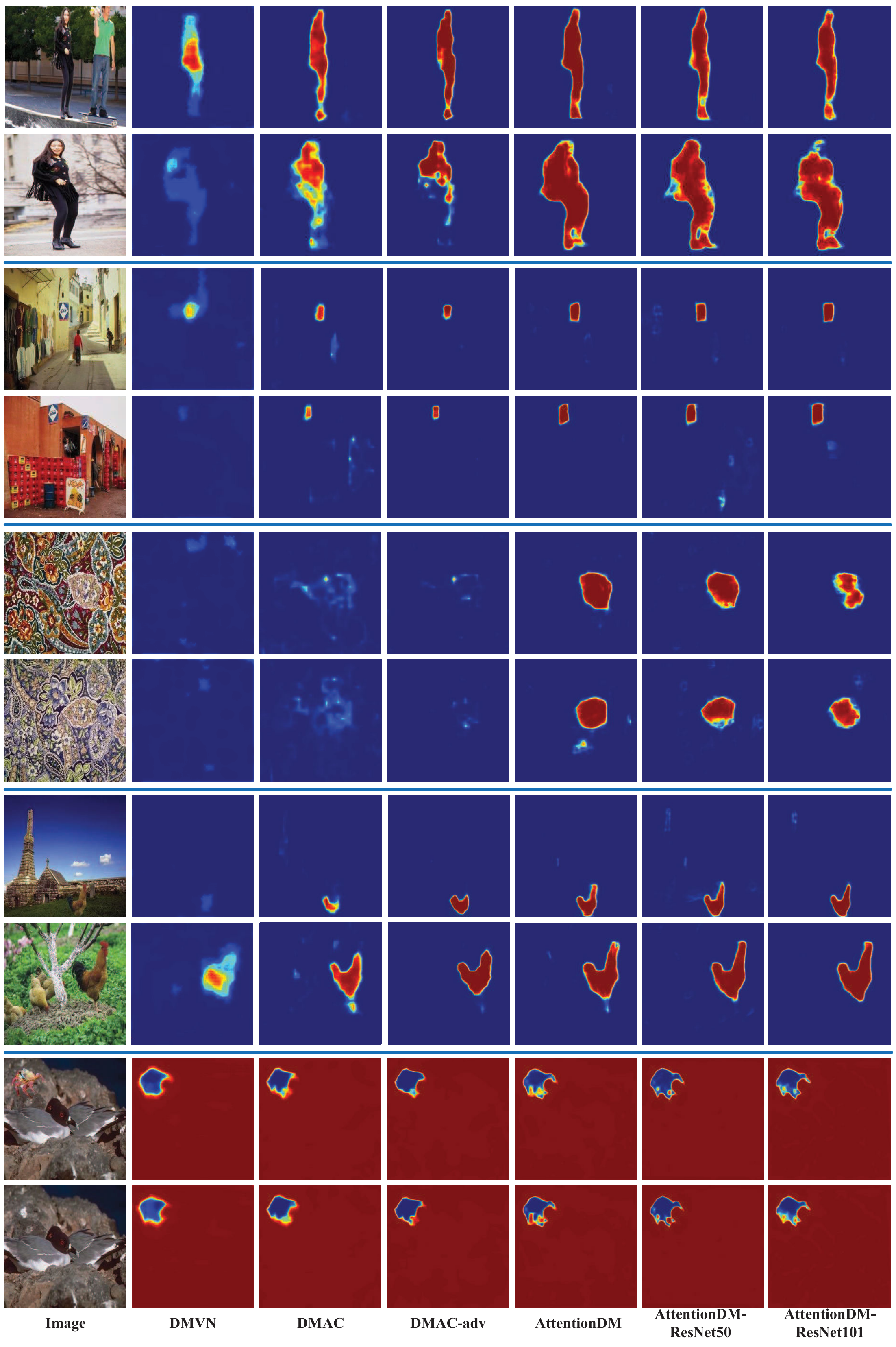

FIGURE 5. Visual comparisons on CASIA.

TABLE 6. Comparisons on CASIA.

Method	Precision	Recall	F1-score	AUC
Christlein et al. [45]	0.5164	0.8292	0.6364	0.8097
Luo et al. [46]	**0.9969**	0.5353	0.6966	0.7677
Ryu et al. [47]	0.9614	0.5895	0.7309	0.7909
Cozzolino et al. [48]	0.9897	0.6334	0.7725	0.8157
DMVN-loc [12]	0.9152	0.7918	0.8491	0.9052
DMVN-det [12]	0.9415	0.7908	0.8596	0.9244
DMAC [10]	0.9255	0.8668	0.8952	0.9468
DMAC-adv [10]	0.9657	0.8576	0.9085	0.9472
AttentionDM	0.9288	**0.9204**	**0.9246**	**0.9676**
AttentionDM-ResNet50	0.9373	0.9071	0.9220	0.9609
AttentionDM-ResNet101	0.9575	0.8841	0.9193	0.9530

TABLE 7. Comparisons on MFC2018.

Method	AUC	EER
DMVN-loc [12]	0.6584	0.4000
DMVN-det [12]	0.6970	0.3665
DMAC [10]	0.7542	0.3123
DMAC-adv [10]	0.7511	0.3093
AttentionDM	**0.7922**	0.2756
AttentionDM-ResNet50	0.7820	**0.2568**
AttentionDM-ResNet101	0.7555	0.2648
DMAC-adv-SR256 [10]	0.7723	0.2946
DMAC-adv-SR128 [10]	0.7739	0.3110
AttentionDM-SR256	**0.8164**	0.2691
AttentionDM-SR128	0.8156	**0.2608**
AttentionDM-ResNet50-SR256	0.8021	0.2723
AttentionDM-ResNet50-SR128	0.8115	0.2784
AttentionDM-ResNet101-SR256	0.8006	0.2634
AttentionDM-ResNet101-SR128	0.8050	0.2750

2) MFC2018

Since experiments on the CASIA dataset have demonstrated the superiority over conventional copy-move forgery detection methods which have high computational complexity [49], we compare AttentionDM with DMVN and DMAC in the experiments on MFC2018. AUC and EER scores on MFC2018 are shown in Table 7. AttentionDM can achieve the highest AUC score, and AttentionDM-ResNet50 can achieve the lowest EER score. Visual comparisons are provided in Figure 6. Apparently, AttentionDM can generate more accurate boundaries and detect small regions. Besides, the majority of images in MFC2018 have high resolutions, we also compare the sliding window matching versions. It shows that with the help of sliding window matching, the performance of AttentionDM is further improved.

3) PS-BATTLES

Images in the PS-Battles dataset are elaborately designed and edited by amateur or professional digital artists, and these images have been uploaded to the online Reddit community. There are many challenging image pairs which can evaluate the effectiveness of CISDL methods. Since all the image pairs are correlated, in other words, one is a fake image, and the other is the source image in one image pair. So we use the detection rate to evaluate the compared methods, as shown in Table 8. And visual comparisons are provided in Figure 7. It can be seen that AttentionDM and its corresponding sliding-window version can achieve the highest detection rates, while the ResNets versions have lower detection rates. Although the detection rate is slightly lower, the ResNet50 version has better localization performance according to Figure 7. While AttentionDM-ResNet101 has a lower detection rate, higher computation complexity and unremarkable localization performance. According to our experiments on four datasets, we can conclude that AttentionDM and AttentionDM-ResNet50 can achieve comparable detection performance, while AttentionDM-ResNet50 can achieve better localization performance. And we do not recommend the use of ResNet101 in the CISDL task.

TABLE 8. Detection rates on the PS-Battles dataset.

Method	PS-Battles dataset
DMVN-loc [12]	0.6792
DMVN-det [12]	0.7102
DMAC [10]	0.8028
DMAC-adv [10]	0.7793
AttentionDM	**0.8346**
AttentionDM-ResNet50	0.8102
AttentionDM-ResNet101	0.8048
DMAC-adv-SR256 [10]	0.8477
DMAC-adv-SR128 [10]	0.8870
AttentionDM-SR256	0.8935
AttentionDM-SR128	0.9228
AttentionDM-ResNet50-SR256	0.8877
AttentionDM-ResNet50-SR128	**0.9322**
AttentionDM-ResNet101-SR256	0.8580
AttentionDM-ResNet101-SR128	0.9077

C. IMPLEMENTATION DETAILS

AttentionDM is implemented on PyTorch, and is trained using a single spatial cross entropy loss. The parameters in the encoder are initialized using VGG16/ResNet50/ResNet101 which are trained for image classification. Three epochs of training are conducted, the batch size is set to 24, and more than 129, 000 iterations are conducted. The Adadelta optimizer is adopted with PyTorch default settings. AttentionDM is trained on synthetic training image pairs.

We automatically generate over one million synthetic training image pairs from the MS COCO dataset [50]. We randomly select one annotated region in one image under different transformations, and past it into another randomly selected image. Five types of transformations are adopted, *i.e.*, shift $\mathbb{U}(-256, 256)$, rotation $\mathbb{U}(-30, 30)$, scale $\mathbb{U}(0.5, 4)$, luminance $\mathbb{U}(-32, 32)$, deformation $\mathbb{U}(0.5, 2)$ changes. Specifically, all pasted regions suffer from the

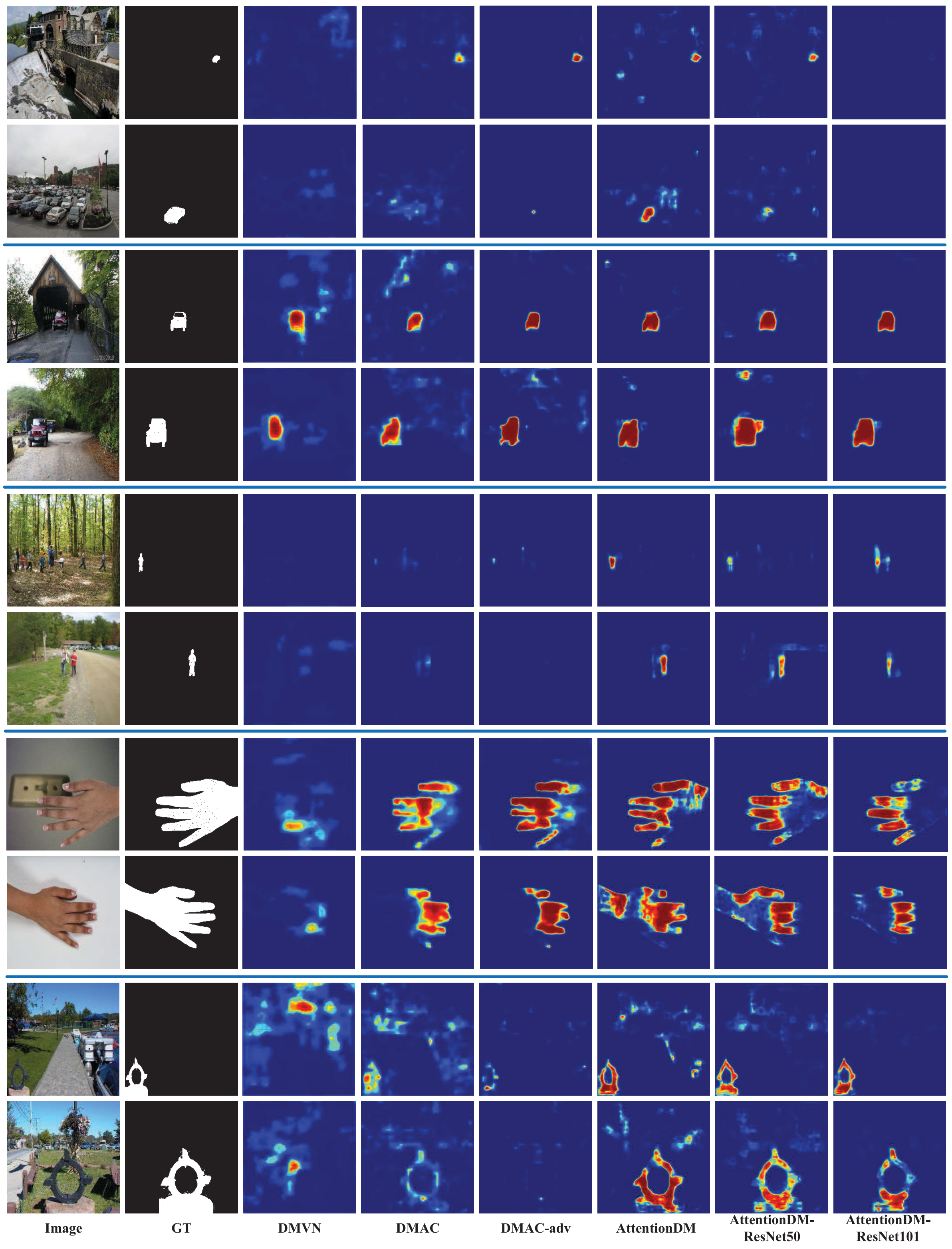

FIGURE 6. **Visual comparisons on MFC2018.**

shift change. For other types of transformations, it has a 50% probability of suffering each transformation. The forged regions in synthetic images may suffer from several types of transformations. The selected regions all satisfy that their areas should be larger than 1% of the images and smaller than 50%, for that extremely small regions are too difficult to

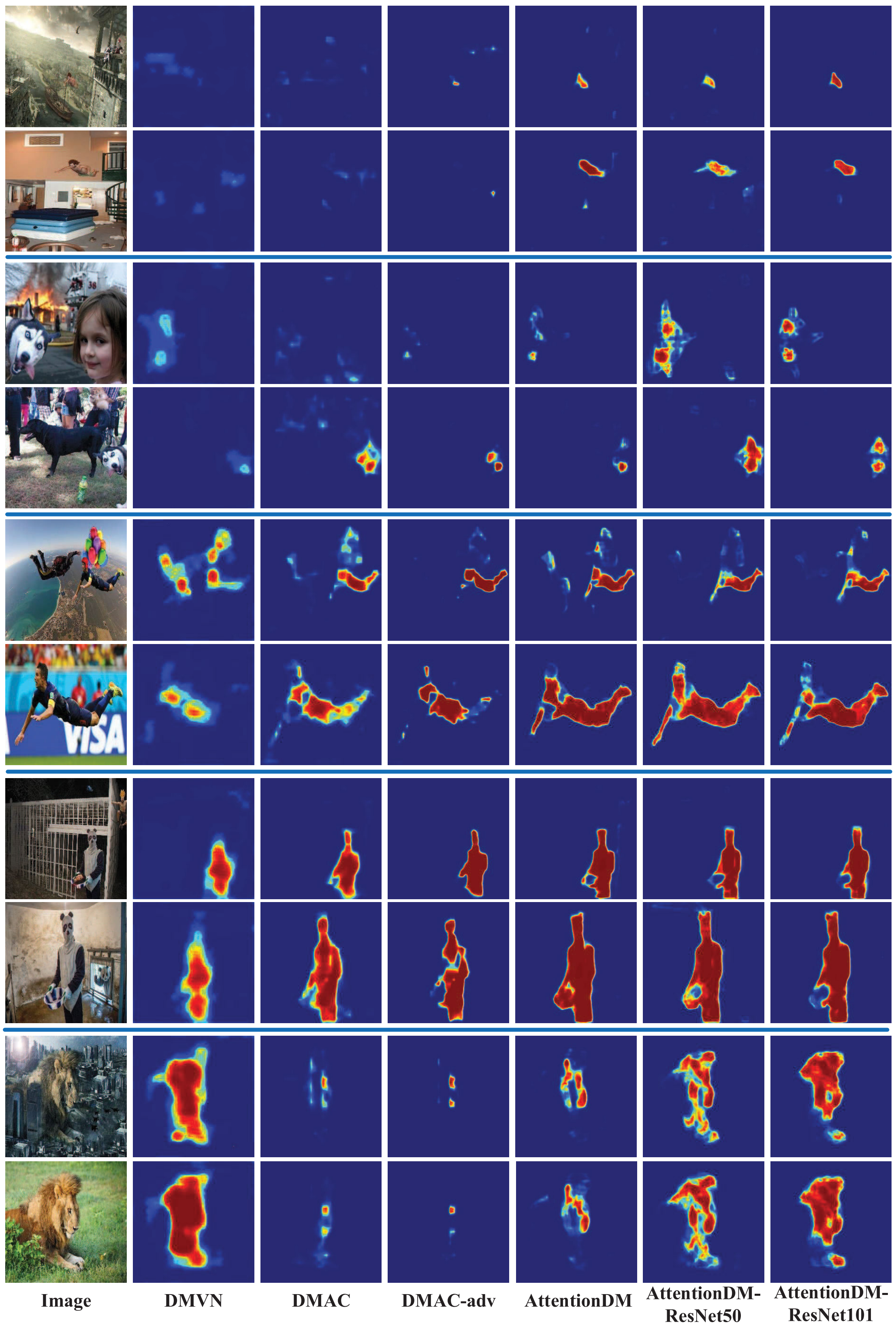

FIGURE 7. **Visual comparisons on the PS-Battles dataset.**

detect and excessively large regions are meaningless. Finally, over one million training pairs are generated with $1/3$ foreground pairs, $1/3$ background pairs and $1/3$ negative pairs.

V. CONCLUSION

In this work, an attention-aware encoder-decoder deep matching network named as AttentionDM is proposed for CISDL. An encoder-decoder architecture with atrous convolution is constructed for hierarchical features dense matching and fine-grained masks generation. Normalization operations are designed to re-enforce the convolutional features and correlation features. A channel attention block is proposed for channel-wise features recalibration to enhance correlation computation. Extensive experiments verify that AttentionDM achieves a significant performance leap. To the best of our knowledge, AttentionDM can achieve the best performance among all the published CISDL methods.

Although AttentionDM can achieve excellent performance, it still has some limitations. For example, similar as the forerunners [10], [12], [41], it can only process fixed-size images. The remedial measure of our sliding window based matching strategy results in high computational complexity. Besides, AttentionDM can achieve the lowest EER score on MFC2018, however there are still many false-alarm images, especially the majority of investigated images are uncorrelated or unforged in real applications [10]. The objects which have similar appearance can mislead our detection. There is still a long way to go for real application.

REFERENCES

[1] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy-move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 170–186.

[2] Y. Liu, Q. Guan, X. Zhao, and Y. Cao, "Image forgery localization based on multi–scale convolutional neural networks," in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, 2018, pp. 85–90.

[3] H. Li, W. Luo, X. Qiu, and J. Huang, "Image forgery localization via integrating tampering possibility maps," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1240–1252, May 2017.

[4] W. Wang, J. Dong, and T. Tan, "A survey of passive image tampering detection," in *Proc. 8th Int. Workshop Digit. Watermarking*, Oct. 2009, pp. 308–322.

[5] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN–based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020.

[6] K. Bahrami, A. C. Kot, L. Li, and H. Li, "Blurred image splicing localization by exposing blur type inconsistency," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 999–1009, May 2015.

[7] B. Peng, W. Wang, J. Dong, and T. Tan, "Optimized 3D lighting environment estimation for image forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 479–494, Feb. 2017.

[8] W. Zhang, X. Cao, Y. Qu, Y. Hou, H. Zhao, and C. Zhang, "Detecting and extracting the photo composites using planar homography and graph cut," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 544–555, Sep. 2010.

[9] 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.

[10] 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. 14, no. 10, pp. 2551–2566, Oct. 2019.

[11] The National Institute of Standards and Technology (NIST). (2018). *Media Forensics Challenge 2018*. [Online]. Available: https://www.nist.gov/itl/iad/mig/media-forensicschallenge-2018

[12] 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. 1480–1488.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, Dec. 2017, pp. 6000–6010.

[16] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[17] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian–MRF approach for PRNU–based image forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 554–567, Apr. 2014.

[18] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine–grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[19] T. Bianchi and A. Piva, "Image forgery localization via block–grained analysis of JPEG artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1003–1017, Jun. 2012.

[20] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[21] Y. Liu, Q. Guan, and X. Zhao, "Copy-move forgery detection based on convolutional kernel network," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18269–18293, Jul. 2018.

[22] K. Wattanachote, T. K. Shih, W.-L. Chang, and H.-H. Chang, "Tamper detection of JPEG image due to seam modifications," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2477–2491, Dec. 2015.

[23] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly detection in nanofibrous materials by CNN–based self–similarity," *Sensors*, vol. 18, no. 2, p. 209, Jan. 2018.

[24] P. Napoletano, "Visual descriptors for content-based retrieval of remote-sensing images," *Int. J. Remote Sens.*, vol. 39, no. 5, pp. 1343–1376, Mar. 2018.

[25] Z. Shi, X. Shen, H. Kang, and Y. Lv, "Image manipulation detection and localization based on the dual–domain convolutional neural networks," *IEEE Access*, vol. 6, pp. 76437–76453, 2018.

[26] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self–supervised learning of geometrically stable features through probabilistic introspection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3637–3645.

[27] I. Rocco, R. Arandjelovic, and J. Sivic, "End-to-end weakly–supervised semantic alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6917–6925.

[28] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2758–2766.

[29] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," 2016, *arXiv:1606.03798*. [Online]. Available: https://arxiv.org/abs/1606.03798

[30] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3475–3484.

[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

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

[33] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017. [Online]. Available: https://openreview.net/forum?id=BJC_jUqxe

[34] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Neural Inf. Process. Syst.*, Dec. 2014, pp. 3104–3112.

[35] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[36] P. Napoletano, G. Boccignone, and F. Tisato, "Attentive monitoring of multiple video streams driven by a Bayesian foraging strategy," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3266–3281, Nov. 2015.

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

[38] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[39] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 833–851.

[40] Y. Liu, X. Zhang, X. Zhu, Q. Guan, and X. Zhao, "ListNet-based object proposals ranking," *Neurocomputing*, vol. 267, pp. 182–194, Dec. 2017.

[41] K. Ye, J. Dong, W. Wang, B. Peng, and T. Tan, "Feature pyramid deep matching and localization network for image forensics," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2018, pp. 1796–1802.

[42] 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.

[43] Y. Liu. (2018). *DMAC Codes*. [Online]. Available: https://github.com/yaqiliu-cs/CISDL-DMAC

[44] S. Heller, L. Rossetto, and H. Schuldt, "The PS-battles dataset—An image collection for image manipulation detection," 2018, *arXiv:1804.04866*. [Online]. Available: http://arxiv.org/abs/1804.04866

[45] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy–move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[46] W. Luo, J. Huang, and G. Qiu, "Robust detection of region–duplication forgery in digital image," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 746–749.

[47] S. Ryu, M. Lee, and H. Lee, "Detection of copy-rotate-move forgery using Zernike moments," in *Proc. ACM Workshop Inf. Hiding*, Jun. 2010, pp. 51–65.

[48] 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, Nov. 2015.

[49] L. Verdoliva, D. Cozzolino, and G. Poggi, "A reliable order-statistics-based approximate nearest neighbor search algorithm," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 237–250, Jan. 2017.

[50] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 740–755.

YAQI LIU received the B.E. and M.E. degrees from Beijing Technology and Business University, Beijing, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in cyber security with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing. His research interests include multimedia forensics, computer vision, pattern recognition, and image processing.

XIANFENG ZHAO received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2003. From 2003 to 2005, he was a Postdoctoral Fellow with the Data Assurance and Communication Security Center, Chinese Academy of Sciences (CAS), Beijing. From 2006 to 2011, he was an Associate Professor with the State Key Laboratory of Information Security (SKLOIS), Institute of Software, CAS. Since 2012, he has been a Professor with SKLOIS, which was moved to the Institute of Information Engineering, CAS, in 2012. Since 2013, he has been a Professor with the University of Chinese Academy of Sciences. His research interests are information hiding and multimedia security, including watermarking, steganography, steganalysis, and multimedia forensics. He has published more than 100 articles. He holds 15 patents. He has coauthored four books in these fields. He has served as a PC Member or the Co-Chair for many conferences, such as IWDW, AVSS, and InsCrypt. Since 2014, he has been an Associate Editor of the *International Journal of Digital Crime and Forensics*.