IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 December 2022, accepted 11 January 2023, date of publication 2 February 2023, date of current version 7 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3241837

SURVEY

Toward Deep-Learning-Based Methods in Image Forgery Detection: A Survey

NAM THANH PHAM[1] AND CHUN-SU PARK[2]
[1]FPT Software Korea, Seoul 07241, South Korea
[2]Department of Computer Education, Sungkyunkwan University, Seoul 03063, South Korea

Corresponding author: Chun-Su Park (cspk@skku.edu)

This work was supported by the Technology Development Program through the Korean Ministry of Small and Medium Enterprises (SMEs) and Startups under Grant S3147433.

ABSTRACT In the last decades, deep learning (DL) has emerged as a powerful and dominant technique for solving challenging problems in various fields. Likewise, in the field of digital image forensics, a large and growing body of literature investigates DL-based techniques for detecting and classifying tampered regions in images. This article aims to provides a comprehensive survey of state-of-the-art DL-based methods for image-forgery detection. Copy-move images and spliced images, two of the most popular types of forged images, were considered. Recently, owing to advances in DL, DL-based approaches have yielded much better results as compared to traditional non-DL-based ones. The surveyed techniques were proposed by developing or fusing various efficient DL methods, such as CNN, RCNN, or LSTM to adapt to detecting tampered traces.

INDEX TERMS Copy-move image, image forgery, spliced image.

I. INTRODUCTION

Digital content, such as images, videos, and audios, is uploaded to social networks every day, with images being the most popular shared resource. However, owing to the enormous popularity of cutting-edge image-editing software, images can be easily edited without leaving perceptible traces. Therefore, it is difficult for practical users to manually identify manipulated images [1], [2]. The field of multimedia forensics, which aims to validate the integrity of digital content, has received considerable scholarly attention. Image forgery, also known as image tampering, image manipulation, or image forensics, is a research branch in which manipulated images are studied to address challenging tasks such as localizing the regions that have been tampered with [1], [2], [3], validating the integrity of images [4], [5], [6], [7], and identifying the source or provenance of tampered images [8], [9], [10], [11].

The images that have not undergone any editing are referred to as authentic images [1], [2], [3], [4], [5], [6], [7], [8], [12], [13], [14], pristine images [15], [16], [17], [18], [19], genuine images [20], or un-tampered images [21], [22]. Several popular types of image forgery in which images are composed from authentic images are as follows:

- Inpainted images are created by modifying a region of the image through merging a large number of small neighboring components [23], [24], [25]. In particular, the content of the new region is interpolated based on the information from its adjacent pixels/regions.
- Copy-move or copy-paste images are composed from an authentic image by copying one or several regions and moving (pasting) them to other regions [1], [14], [15], [26]. The original copied regions and the pasted regions are referred to as the source and target regions, respectively.
- Spliced images are formed by copying one or several regions from an authentic image and pasting these into another authentic image [7], [13], [27]. The two images are respectively referred to as the source and target images [10], or the donor and host images [9], [28], [29].
- Object removal images are created from an authentic image from which objects are removed by either

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

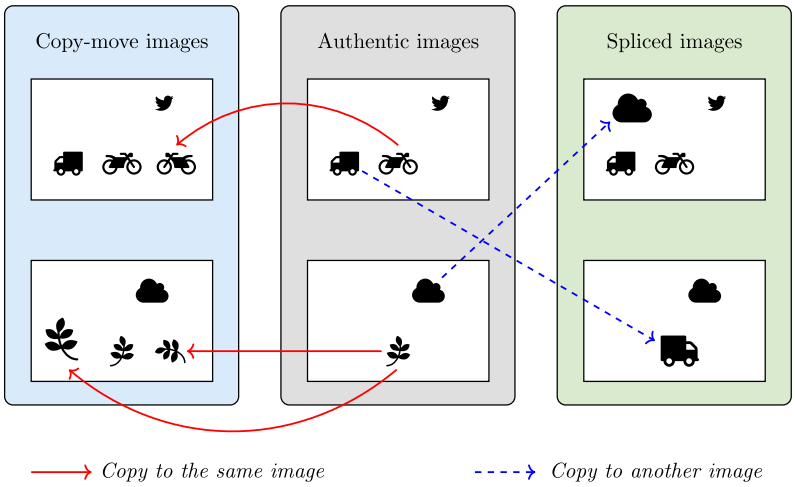

FIGURE 1. Illustration of the composition of two copy-move images (*left*) and two spliced images (*right*) from two authentic images (*middle*). The red solid and blue dashed arrows represent copy-move and splicing operations, respectively. The tampered region can be rotated, scaled, or flipped from the source region or may undergo post-processing operations to thwart forgery detection. Several tampered regions may also be copied from a single source region.

inpainting or copying-moving as defined above [20], [30], [31]. Therefore, copy-move images are occasionally considered as object removal images. The major difference is that in object removal images, the objects to be removed are semantic regions, such as a person or a car in the background [31]. In contrast, the objects to be copied and pasted to other regions in copy-move images (source regions) are not necessarily semantic regions [2]. To be more specific, the copied regions can be any random region in the image, such as a small part in the sky, while the moved (pasted) regions can be a random sky region (non-semantic) or a flying bird (semantic). In addition, an object removal image is a special case of image inpainting.

- Retouched images are generated by performing image processing operations, such as smoothing, sharpening, and contrast enhancement, to improve the visual quality [32], [33].

Meena and Tyagi [34] classified these tampering operations into two main categories: dependent and independent operations. Image splicing and copy-move belong to the dependent category, whereas the remaining forgery operations are in the independent category. Dependent tampering operations are those in which the tampered regions depend on (or are copied from) other regions of the source images. These two dependent types of tampering, image splicing and copy-move, which are also two of the most popular types of forgery, are covered in this survey. Figure 1 presents the detailed composition of spliced and copy-move images.

Examples of copy-move and spliced images together with their corresponding authentic images and ground truth are shown in Figures 2(**a**) and 2(**b**), respectively. We noticed that in a pair of spliced images and their corresponding host (authentic) images, the authentic image could be misconstrued as the object removal image generated from the spliced image. In other words, the relationship between different types of image forgery is complicated.

FIGURE 2. Examples of (a): copy-move images and (b): spliced images. The tampered images appear in the middle row, their corresponding authentic images in the top row, and the ground-truth images of tampered regions (highlighted in white) appear in the bottom row.

In image forgery problems, the term "*detection*" has been used in two different types of problems with different meanings. Image splicing detection (ISD) usually refers to the classification problem between authentic and spliced images [4], [5], [6], [7]. However, it occasionally refers to the spliced region localization problem [21], [28], [35]. Nonetheless, image splicing localization (ISL) has been used more frequently to address the splicing localization problem [3], [36], [37], [38], [39]. By contrast, copy-move forgery detection (CMFD) aims to localize tampered regions, including source (copy) and target (move/paste) regions [2], [26]. In this survey, we used the term image forgery detection (IFD) for the problem of tampered region localization of both copy-move and spliced images.

The remainder of this paper is organized as follows. Section II provides an overview of state-of-the-art surveys on IFD problems. DL backbone networks are discussed in Section III. We then describe DL-based IFD methods in Section IV. The experimental results of the reviewed methods on several benchmark datasets are presented in Section V. Section VI provides possible future directions and concludes the study.

II. REVIEW OF RECENTLY PUBLISHED SURVEYS ON IFD PROBLEMS

In this Section, we analyze IFD survey papers that have been published in the last five years and that focus on the examination of targets with DL-based techniques.

Several recent investigations into image forgery detection have been reported [34], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49] of which a few reviewed deep learning (DL)-based techniques [50], [51], [52]. In this survey, we investigate recent trends in image forgery detection approaches for two of the most popular types of image forgery: copy-move images and spliced images.

Four types of forgery detection, splicing, copy-move, resampling, and retouching, were reviewed [34]. All the IFD methods examined in this survey used handcrafted features to detect tampering. These types of tampering were also analyzed using DL-based methods [50]. Nonetheless, these researchers mainly discussed detection techniques in different categories without providing deep insights into DL architectures. Camacho and Wang [51] surveyed a wide range of problems such as double JPEG compression detection, forgery detection, camera identification, and deep fake detection. The extent to which DL-based methods have been used to solve IFD problems is therefore limited. Abidin et al. [52] briefly reviewed DL-based CMFD papers and compared the differences in the detection pipelines of traditional and DL-based methods.

III. OVERVIEW OF DEEP NEURAL NETWORK ARCHITECTURES

DL has dominated research in various fields over the past decade [53], [54]. In the last five years, DL-based methods have surpassed traditional physics-based methods for solving IFD problems [42], [43], [44], [55], [56]. The general DL-based image forgery detection pipeline is illustrated in Figure 3 where the entire input image or divided patches are fed into a deep neural network (DNN) for feature extraction. These two approaches are known as image-wise and patch-wise approaches. In patch-wise methods, the divided patches can be overlapping or non-overlapping. The final binary output image of this network was the detected forgery image.

Image segmentation and IFD problems are closely related because both are pixel-based classifications and DL networks applied to one problem could also be effective for the other [57], [58], [59]. In image semantic segmentation, recent advanced DL techniques have achieved great success in segmenting images into m regions with n labeled classes, where $n \leq m$ [60], [61], [62]. IFD can be considered a traditional type of image segmentation with $m = n = 2$, where two classes are classified: objects (the foreground) and the background. Similarly, object detection-based methods, such as region-based convolutional neural network (R-CNN), Mask R-CNN, have also found wide application in tampering detection. DL techniques, including long short-term memory (LSTM) and recurrent neural network (RNN), that are usually used in the language modeling or speech recognition fields have been widely employed in IFD.

In this Section, we provide a brief overview of backbone DL networks for IFD problems.

A. CONVOLUTIONAL NEURAL NETWORK

CNNs are among the most popular deep learning architectures with applications in many fields [63]. A typical CNN consists of three main layers: a convolutional layer, a pooling layer, and a fully connected layer [64]. Figure 4 illustrates a simple representation of a CNN, in which each convolutional layer is followed by a pooling layer to reduce the complexity of further layers while maintaining prominent features [64], [65]. Each neuron in the fully condensed layer is connected to every neuron in both the previous and next layers, as shown in Figure 4. The vast number of parameters in the fully connected layers gives rise to high computational complexity. In this survey, the term CNN represents the classical CNN mentioned above, whereas the various successive deep networks are mentioned with specific terminologies.

B. ENCODER-DECODER AND AUTO-ENCODER NETWORKS

The encoder–decoder is an unsupervised neural network that can be simplified into three components: the encoder, latent space representation, and decoder. The input is encoded to generate the latent vector in the hidden state, and this vector is then decoded to obtain the closed reconstructed version of the input or intended output. In the encoder-decoder network, the spatial resolution is encoded in exchange for learning features or finding details in the inputs. The encoder and decoder were trained to minimize the reconstruction error. Figure 5 shows an example of an encoder–decoder network for IFD problems.

An auto-encoder is a special case of an encoder–decoder network with a single hidden layer, and it reconstructs the inputs from the encoded data. Auto-encoders are typically applied to various problems, such as denoising data, classification, clustering, and anomaly detection [67].

C. R-CNN AND ITS DERIVED NETWORKS

R-CNN [68] and its extensions, including Fast R-CNN [69], Faster R-CNN [70], and Mask R-CNN [58] have achieved spectacular success with object detection problems using region-based approaches.

R-CNN is a neural network for object detection that feeds region-of-interest (ROI) bounding boxes into a CNN to classify regions. Fast R-CNN was developed to overcome the

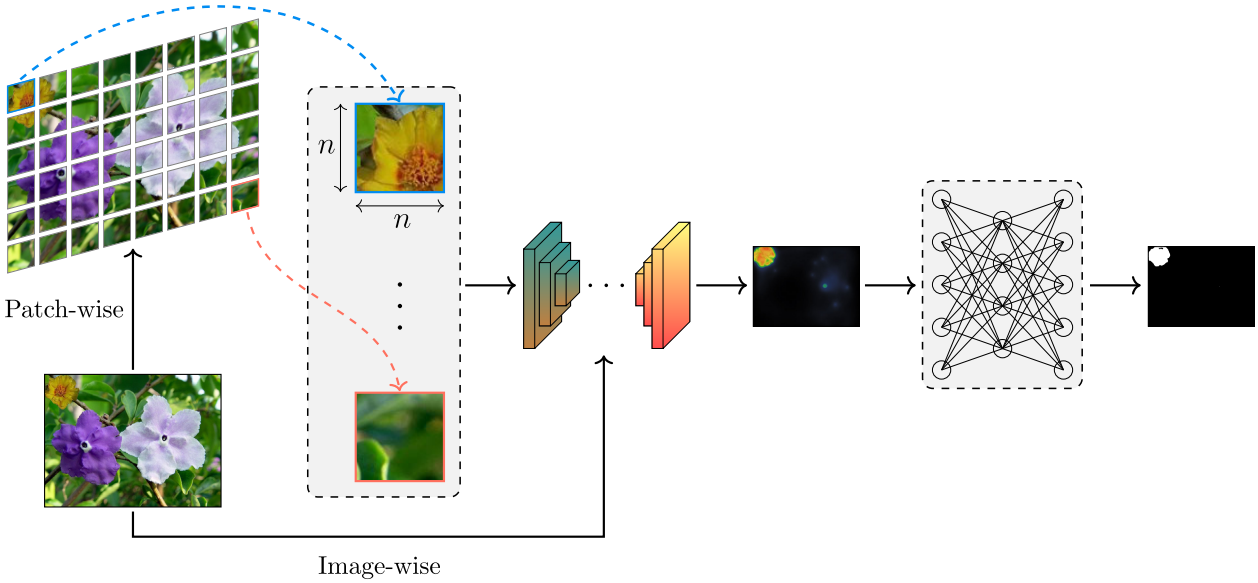

FIGURE 3. General pipeline of deep-learning-based image forgery detection methods, where image-wise or patch-wise methods are usually applied. In the patch-wise approaches, the input images are divided into overlapping and non-overlapping patches. Then single or fused neural networks are used to detect the manipulation, and a segmentation is performed to generate the final localization results.

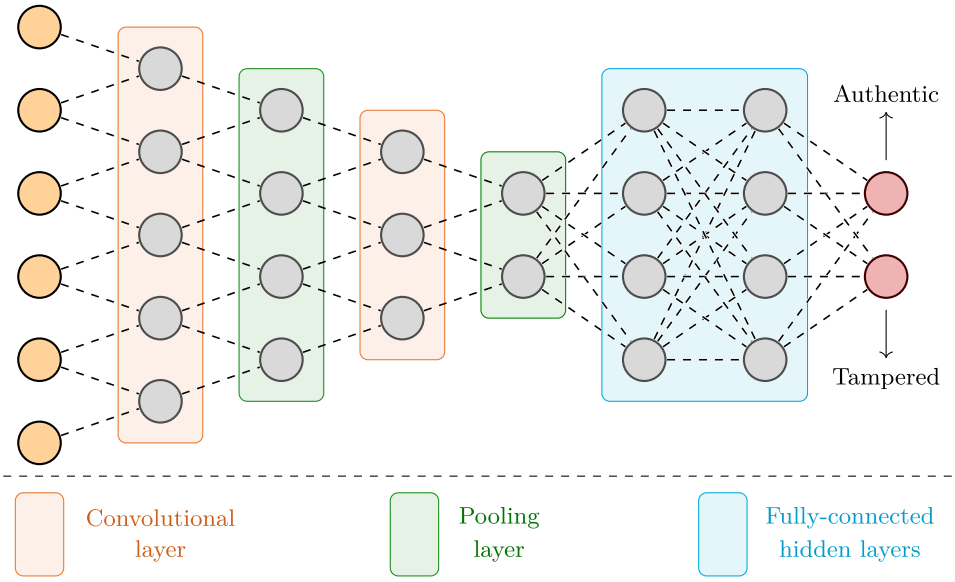

FIGURE 4. Simple illustration of the convolutional neural network used for solving IFD problems. This figure was redrawn from [66].

shortcomings of selective search in R-CNN, which generates thousands of forward passes for each image. In Fast R-CNN, the input image is directly fed into the CNN instead of region proposals to create a convolutional feature map.

Faster R-CNN improves on Fast R-CNN by using a region proposal network (RPN) instead of a selective search for ROI generation. Mask R-CNN was built on top of Faster R-CNN to generate the object masks.

D. U-NET

U-Net, a U-shaped neural network with simple architecture, was originally developed for image segmentation [59], [71], [72]. U-Net consists of convolutional, ReLU, max pooling, and up- convolutional layers designed for down- and up-sampling to capture context and symmetric features via contracting paths. In this network, the up-sampling layers replace pooling operators to localize the features at the pixel level. One of the advantages of this network is that it can achieve highly precise segmentation with very few images used for training, owing to the augmentation of available annotated samples.

E. LSTM

The LSTM architecture applied in [20], [73], [74], and [75], illustrated in Figure 6, is the most popular among many variants of the LSTM architecture [76], [77]. The current LSTM

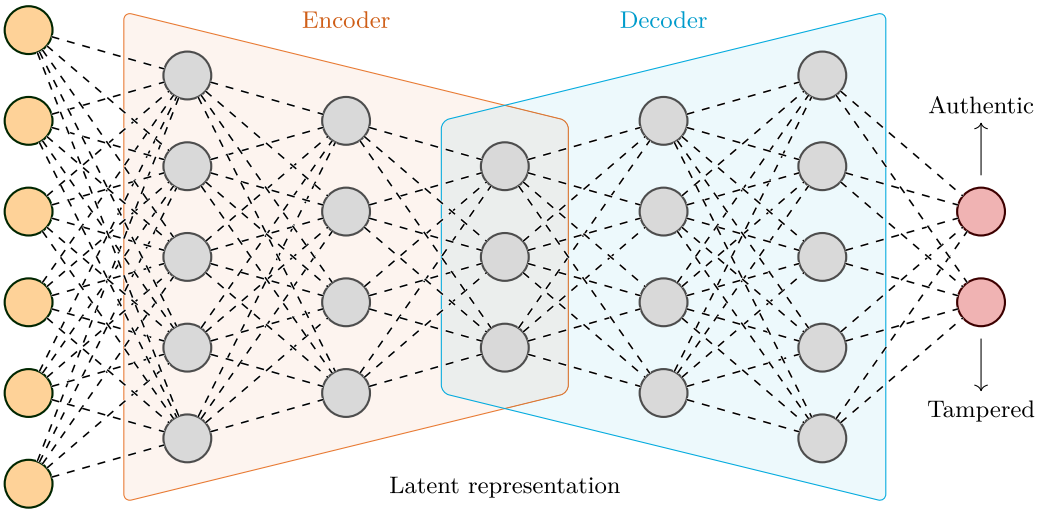

FIGURE 5. Encoder-decoder network used for solving IFD problems. This figure was redrawn from [66].

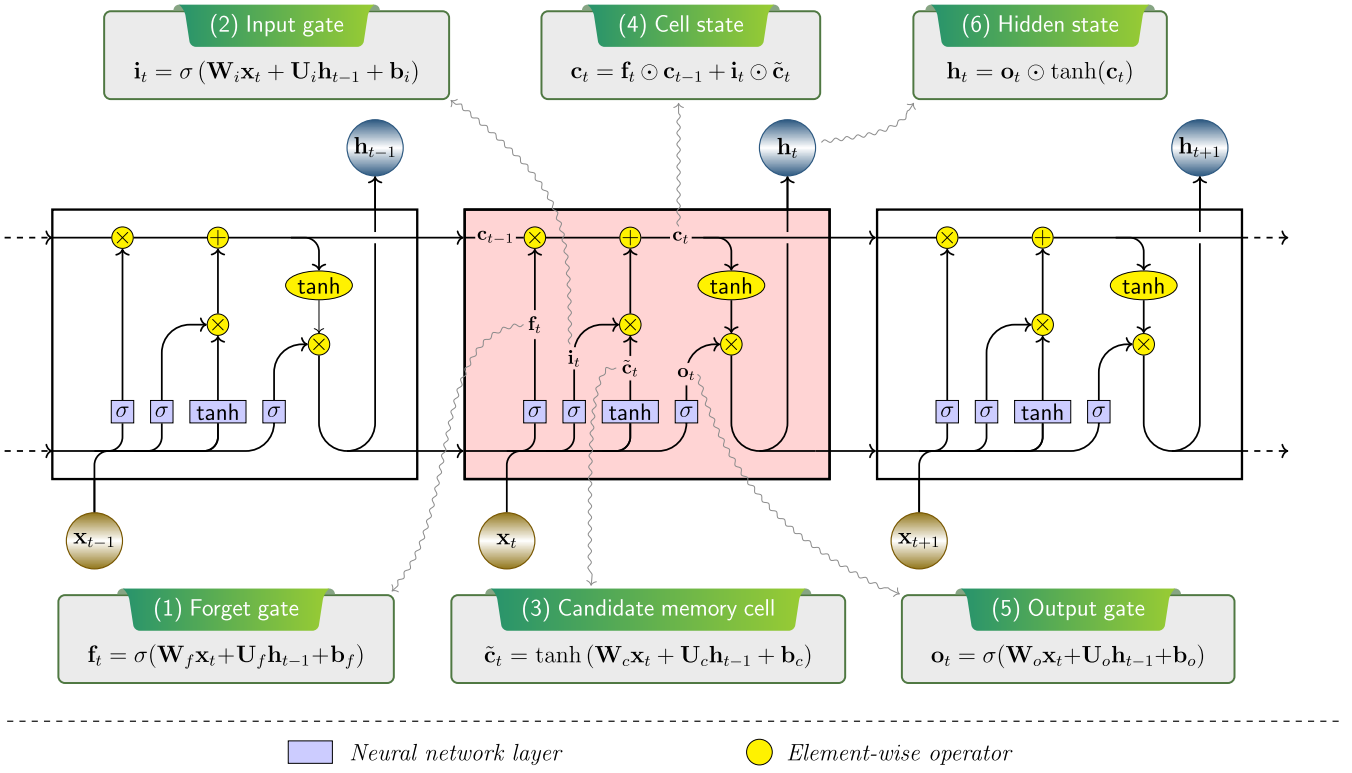

FIGURE 6. Illustration of adjacent cells in LSTM architecture applied in [20], [73], [74], and [75]. For $k \in \{f, i, o, c\}$, $W_k \in \mathbb{R}^{m\times n}$ are input-to-hidden transformations, $U_k \in \mathbb{R}^{m\times m}$ are hidden-to-hidden transformations, and $b_k \in \mathbb{R}^m$ are bias vectors. The current cell is highlighted in light pink background with explanation for cell states using formulas. This figure was drawn on the basis of the idea in [76].

cell at time step t, represented by the rectangle in the middle in Figure 6, is connected to neighboring cells, as depicted in the blurred rectangles. The cell state $\mathbf{c}_t$ of the current cell is controlled by the gating mechanism of the forget gate $\mathbf{f}$, input gate $\mathbf{i}$, and output gate $\mathbf{o}$. Cell t uses $\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}$ as inputs and produces $\mathbf{h}_t, \mathbf{c}_t$ via the intermediate equations in Figure 6. For $k \in \{f, i, o, c\}$, $\mathbf{W}_k \in \mathbb{R}^{m\times n}$ are input-to-hidden transformations, $\mathbf{U}_k \in \mathbb{R}^{m\times m}$ are hidden-to-hidden transformations, and $\mathbf{b}_k \in \mathbb{R}^m$ are bias vectors. The forget gate $\mathbf{f}_t$ is the sigmoid layer, which takes $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$ as inputs and decides to retain or forget the previous memory cell $\mathbf{c}_{t-1}$:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{1}$$

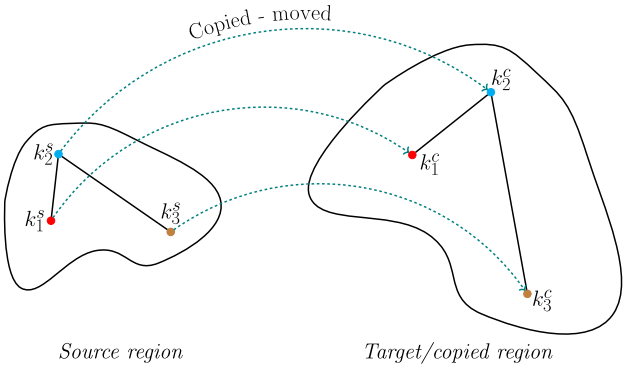

FIGURE 7. Classical feature matching method to detect copy-move tampering. The target region is scaled up 150% and rotated by 45∘. Figure reproduced with permission from [1].

where σ denotes sigmoid function. The input gate determines whether to write the data to the cell state.

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i\right). \tag{2}$$

A candidate vector created by a tanh layer controls the data to be written to the cell state.

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c\right). \tag{3}$$

The current cell state is updated as follows:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{4}$$

where $\odot$ is pointwise multiplication operator. The hidden state in which the output is produced, controlled by the output gate, is defined as follows:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o). \tag{5}$$

The cell state is filtered by a sigmoid layer and then multiplied by the hidden state in which the output is produced to obtain the cell output.

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{6}$$

IV. DEEP-LEARNING-BASED IMAGE FORGERY DETECTION MODELS

In this Section, we presented in detail how the backbone networks reviewed in Section III and their adapted variants are used in IFD problem. Before the widespread use of DL in various fields, traditional physics-based methods were popular for solving IFD problems. During preprocessing, the input images might be converted to the frequency domain using methods such as discrete cosine transform (DCT) [4], [78], [79] or discrete wavelet transform (DWT) [27], or they may be converted to another color space, such as YC_bC_r. Various types of image features, such as scale-invariant feature transform (SIFT) [1], [2], [12], [80], speeded-up robust features (SURF) [81], [82], [83], local binary pattern (LBP) [4], [84], and Zenike moments [2], were extracted in block-based or keypoint-based methods. Usually, the final stages are feature matching and filtering to generate detected heatmaps. Figure 7 illustrates feature matching in copy-move image detection using classical methods, where SIFT keypoints were extracted in the source and target regions. In various computer vision tasks, features designed by DL models are more robust than handcrafted features [63], [85], [86], particularly in problems with large-scale data [87]. Figure 8 presents the timeline of notable DL-based IFD techniques and their backbone networks to show the research trend in this field.

A. CNN-BASED METHODS

Since the dawn of DL, convolutional neural networks (CNN) have been the most commonly applied artificial neural networks to solve visual imagery problems such as detection, recognition, classification, and segmentation [88].

Bondi et al. [17] proposed a CNN to detect splicing traces based on camera characteristics. In this study, the input images were divided into non-overlapping patches of size 64×64 and fed into a CNN model to discover the camera model that had been used to capture the image patches. To this end, a pretrained CNN model was utilized to extract a feature vector $\mathbf{f}$ of size N_{cams}, where N_{cams} is the number of cameras used for training. The i^{th} element $\mathbf{f}_i^{\mathbf{P}}$ of the feature vector $\mathbf{f}^{\mathbf{P}}$ represents the confidence score of patch $\mathbf{P}$ captured using the i^{th} camera in the list and $\sum_{i=1}^{N_{cams}} \mathbf{f}_i^{\mathbf{P}} = 1$. For camera models not included in the dataset that was used to train the CNN model (referred to as unknown camera models), the feature vectors $\mathbf{f}$ of the corresponding patches captured by the same camera behave similarly because of the coherence of the image patches [89]. The splicing localization of this method was determined at the patch level, where a binary mask $\hat{\mathbf{M}}$ was computed based on the k-means clustering algorithm using all feature vectors $\mathbf{f}$ and a confidence score matrix. This method achieved detection accuracies of 90.8% and 81% for known and unknown camera models, respectively. Similarly, Cozzolino et al. [18] utilized camera noiseprint features in their CNN-based method.

The limitations of CNN-based methods using camera features [17], [18], [89] are as follows: (*i*) these methods localized splicing regions at a patch size of 64×64; (*ii*) these methods may fail to detect spliced images if the tampered and pristine regions are composed from images captured by the same camera model; (*iii*) the spliced images with small tampered regions might be detected as authentic images; (*iv*) these methods assumed that the majority of the patches belong to the pristine region, whereas the minority of the patches belong to the spliced region; therefore, they are not robust to images with large spliced regions and spliced regions could be detected as pristine regions, and vice versa.

In [90], Rao et al. proposed a network of 10 convolutional layers, where mean or max pooling was performed in the second and sixth layers. The first layer was responsible for pre-processing, and the weights were computed using 30 high-pass filters used in the estimation of residual maps in a spatially rich model [91]. $\mathbf{F}^n(\mathbf{X})$, the feature map in layer n of input $\mathbf{X}$, was computed from kernel $\mathbf{W}^n$ and bias $\mathbf{B}^n$ as

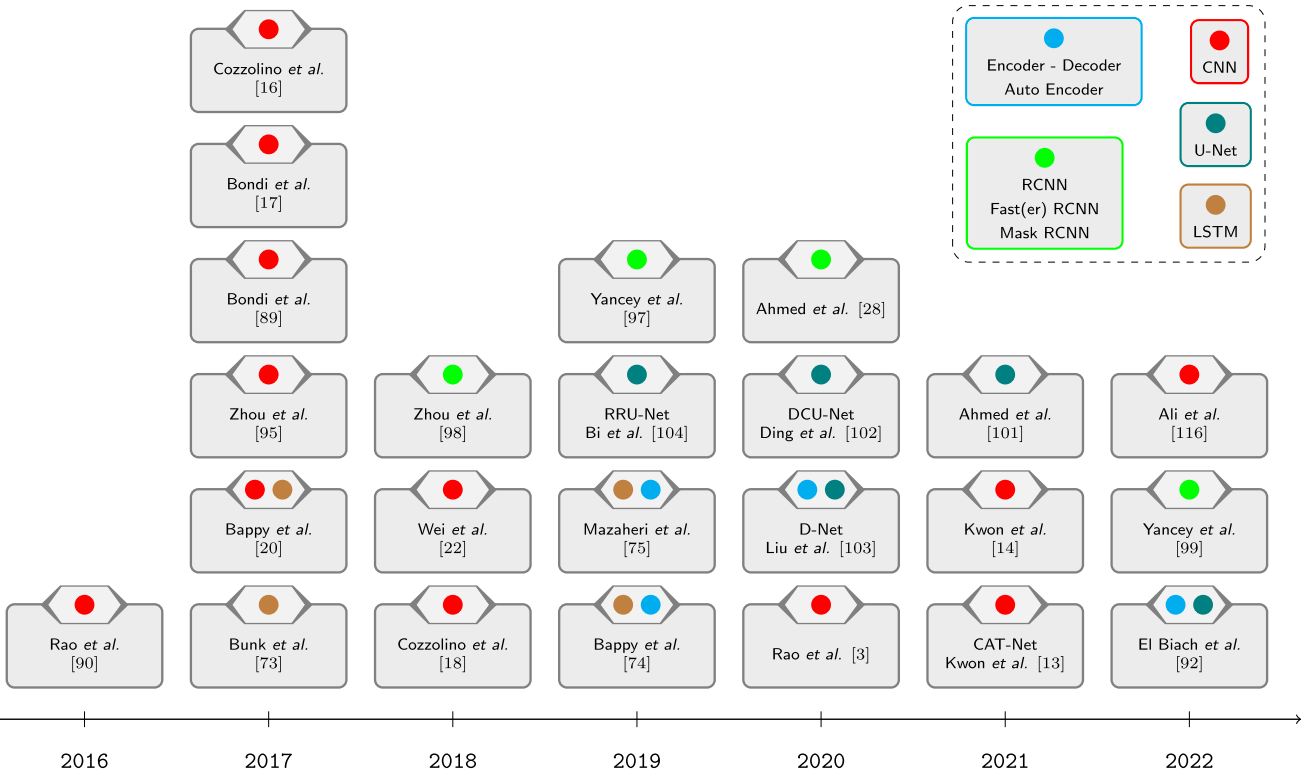

FIGURE 8. DL-based IFD reports in the literature and their corresponding DL backbone networks denoted by colored circles.

follows:

$$\mathbf{F}^n(\mathbf{X}) = \text{pooling}(f^n(\mathbf{F}^{n-1}(\mathbf{X}) \odot \mathbf{W}^n + \mathbf{B}^n)), \tag{7}$$

where $f^n(.)$ was the activation function. In this network, only the last layer was fully connected to reduce the parameter training complexity and avoid possible overfitting.

A back-to-back CNN named C2RNet, including Coarse-CNN, followed by Refined-CNN was proposed for splicing detection in [21]. The input image was fed into the Coarse-CNN with 13 convolutional layers, five max pooling layers, and two fully connected layers to detect suspicious coarse spliced regions by extracting the differences between the authentic and spliced regions. The output feature map of the Coarse-CNN was then fed into the Refined-CNN with 16 convolutional layers, five max pooling layers, and three fully connected layers to refine the detected results. Furthermore, post-processing operations, including morphological operations, adaptive filtering, and convex-full filling, were performed to refine the detected splicing results using C2RNet. However, the adaptive filter was not guaranteed to precisely remove inaccurately detected edges of authentic regions, and the convex-full filling algorithm might fail to detect non-simply connected spliced regions.

B. ENCODER-DECODER-BASED MODELS

The encoder–decoder network in [74] is based on SegNet [57]. Unlike other traditional encoder–decoder architectures, the latent representation in this work was a combination of the outputs of its encoder and the LSTM network adopted from [20]. The LSTM branch of this hybrid network is discussed in Section IV-D.

The image-splicing localization network proposed by Mazaheri et al. [75] has three main components: LSTM, encoder–decoder, and skip connections. The LSTM part of this method is similar to that of [20] and [74]. The encoder–decoder of this method is different from that in [74] because it was inspired by U-Net [59]. The encoder part included convolutional layers, max-pooling, batch normalization, and a rectified linear unit (ReLU) was used as the activation function. In the residual blocks of the encoder, both long and short skip connections were used to obtain Because it was a hybrid LSTM–encoder–decoder method, the combination of LSTM and encoder outputs was combined to feed into the decoder.

Another encoder–decoder network was proposed in [92], in which the encoder was adopted from ResNet [93]. A full-size image with zero padding was used as input. In each stage of the encoder, the image resolution was halved, while the depth was doubled.

C. R-CNN-BASED METHODS

The IFD problem, which attempts to localize regions that have been tampered with, benefited from these object detection studies because a number of studies on IFD involved a R-CNN [21], [28], [94], [95], [96], [97].

Zhou et al. [98] proposed a two-stream Faster R-CNN network using the input RGB image and its noise stream as the inputs for the convolutional layers. In this method, both streams utilize region proposals obtained from the RGB convolutional layers. The ROI features were generated for both streams from the ROI pooling layer. Both feature maps were passed through FCN and softmax layers to predict the tampering map, whereas the bounding box was predicted from the RGB ROI features.

Similar to [98], Yancey et al. [99] also adopted a Faster R-CNN network with two input streams, where the JPEG compression stream, which was generated from the Block Artifact Grid and Error Level Analysis, replaced the noise stream. The spatial features were selected by the ROI pooling layer from each stream to generate a fixed-length feature vector, which was then used for localization.

In [28], the features of the input image were extracted from one of two ResNet architectures: ResNet-50 and ResNet-101 [93]. The ResNet backbone network was followed by a CNN with four convolutional and pooling layers, where the feature map of the convolutional layer n, $\mathbf{F}^n(\mathbf{X})$, is represented by Equation 7; therefore, it was named ResNet-conv. This method simplifies ResNet by utilizing convolutional layers instead of the feature pyramid network (FPN) [100] to slightly increase the convergence speed. This idea was based on the observation that the tampering features can be learned in the first several layers. The feature map was then trained using Mask R-CNN to detect the tampering regions. Specifically, the first stage of the Mask R-CNN, the region proposal network (RPN), which processes the CNN feature output to generate the ROI, was adopted from Faster R-CNN [70] whereas the second stage was entirely from Mask R-CNN [58]. Another major contribution of this method is the construction of a forgery image dataset for training.

D. LSTM-BASED METHODS

Resampling features were exploited to feed into LSTM in [20], [73], [74], and [75] based on the assumption that negative correlations were usually observed at the boundaries of the tampered regions. LSTM was effectively utilized to model chronological sequences such as language or speech [105], [106]. LSTM [107] was designed to alleviate the problem of an exploding or vanishing gradient of classical RNN [108].

When solving IFD problems, the chain-like structure of LSTM has the advantage of learning the correlations of adjacent blocks; therefore, it is capable of capturing the discrepancy at the boundaries of manipulated regions [20]. An image-splicing detection method using resampling features was proposed [73]. In this study, the resampling features were extracted from overlapping patches with a size of 64×64 and stride of 8 to generate a multichannel characteristic heatmap, where each channel represented one type of resampling in image patches. Image patches were trained using two separate deep learning architectures to detect six resampling features: JPEG compression, upsampling, downsampling, clockwise rotation, counterclockwise rotation, and shearing. The resampling features, represented as periodic correspondences, were effectively extracted in the densely overlapping patches using the Radon transform and fast Fourier transform (FFT) in two fully connected neural networks. These two networks are designed to handle different types of resampling. The resampling feature map was divided into blocks sized 8×8, and then these blocks were sequentially fed into an LSTM with three stacked layers, each of which had 64 cells. The LSTM cells were connected to their neighbors by an input gate, a forget gate, and an output gate. The cell state and output state of the current cell t were denoted as $\mathbf{c}_t$ and $\mathbf{h}_t$, respectively. $\tilde{\mathbf{c}}_t$ denotes the new cell state candidate produced by $\mathbf{c}_t$. The output state of the current cell t is defined as

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{8}$$

where $\odot$, $\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$ denote pointwise multiplication, input, forget, and output, respectively. In the last layer of the LSTM, each cell generated a 256- dimensional feature vector, which was fed into a softmax classifier to predict the label (with or without tampering) for each patch. Finally, two segmentation methods, Otsu's thresholding and Random Walk, were employed to extract the localization results from the heatmap extracted by the softmax classifier.

A hybrid CNN-LSTM network was proposed [20], with LSTM adopted from Bunk et al. [73]. Two convolutional layers were used in the preprocessing step to extract low-level image features, such as edges and textures. The output feature map of this step was then divided into patches sized 8×8 to be fed into the LSTM with three stacked layers, as described in [73]. The LSTM generates patch labels and a 2D feature map, which are then fed into the last three convolutional layers of the network to segment the manipulation result at the pixel level. The final result, the detection of a tampered image, was augmented using patch-wise tampering classification (patch labels generated by LSTM) and pixel-wise manipulation segmentation. The method proposed in this study localizes three types of forgery traces: splicing, copy-move, and removal. However, only the target regions were localized in the copy-move images, while the copy regions were not specified. The presence of noise is an important indicator of forgery in IFD [13], [14], [95]. Therefore, in this method, max pooling is used only in the third layer to avoid information loss.

Bappy et al. developed fused architecture comprising an LSTM and encoder-decoder [74] based on their previous studies [20], [73], where a novel localization framework employed features in both the frequency and spatial domains for tampering localization. Specifically, the resampling features in the frequency domain obtained from non-overlapping patches sized 8×8 were fed into the LSTM, whereas the full image was encoded by the four-layer encoder. The detected tampering mask was finally generated by decoding the fusion of the feature vectors from the LSTM and encoder. This

study utilized the image and divided patches as the inputs for two different DL networks; therefore, it can detect traces of tampering in local and global contexts.

The method proposed by Mazaheri et al. [75] adopted a two-layer LSTM network with resampling features from 8×8 non-overlapping patches in the frequency domain [74] for their hybrid architecture. The contribution of this research to the design of a novel encoder-decoder network is presented in section IV-B.

In these LSTM-based IFD methods [20], [73], [74], [75], the cross-entropy loss was used as the training loss function, as follows:

$$L(\theta) = \frac{-1}{M} \sum_{m=1}^{M} \sum_{n=0}^{1} \delta\left(Y_m, n\right) \log\left(Y_m = n \mid y_m; \theta\right) \quad (9)$$

where δ denotes the Kronecker delta function, M denotes the number of pixels, and y_m and Y_m represent the m^{th} pixel values of the input image and the detection mask, respectively.

The detailed architectures and a few remarks relating to the reviewed papers, categorized according to their backbone DL networks, are listed in Table 1.

V. EXPERIMENTAL COMPARISONS

A. IMAGE FORGERY DATASETS

This section provides a brief overview of popular datasets used for IFD problem solving. Usually, researchers only conducted their experiments using several among many of the public datasets. Detailed information on the image forgery datasets reviewed in this survey is provided in Table 2.

1) COLUMBIA DATASETS

The Columbia Image Splicing Detection Evaluation Dataset [117] was the first dataset designed to contain spliced images and was published in 2004. All 933 authentic and 912 spliced images in this dataset were black-and-white with a resolution of 128×128 and included simple splicing operators. In the second version, the Columbia Uncompressed dataset consisted of only 183 authentic and 180 spliced images in color. In this version, the image size is approximately five to six times larger. The manipulation operations used to tamper with the images in this dataset were also simple, such that human eyes could easily identify forgery.

2) CASIA DATASETS

CASIA forensic datasets are arguably the most popular datasets used for IFD problems. Two versions of the CASIA image forgery datasets were constructed: CASIA v1 and CASIA v2 [111], where the latter is an extension of the former. Because these datasets were provided without official ground-truth images, the third-party ground-truth dataset created by Pham et al. [10] has been widely used for benchmarking. The first version includes 800 authentic images and 921 tampered images, whereas the extended version includes 7200 authentic images and 5123 forged images. Both CASIA datasets contain multiple categories of images, such as architecture, nature, indoors, and animals. Tampered regions of various sizes, ranging from small to very large, were created. The preprocessing operations include rotation, resizing, distortion, and the fusion of two of these three operations. In the post-processing step, the dataset was manually edited using Photoshop to make it more realistic. However, the images in these datasets were low resolution, with less than 1000 pixels in each dimension.

3) FORENSICS DATASET

The forensic dataset [118] was published in the first IEEE Forensics challenge by the Information Forensics and Security Technical Committee (IFSTC) in 2013. This dataset comprises 450 training and 700 testing images sized 2018×1536.

4) CoMoFoD DATASET

CoMoFoD [112], a dataset comprising copy-move images, is an acronym for Copy-Move Forgery Detection. This dataset consists of 200 sets of small images, 512×512, and 60 sets of large images, 3000×2000. In this dataset, the tampered regions account for 0.11% to 17.34% of the images. A small number of tampered images have multiple copied regions. Five types of copy-move manipulations were conducted to create this dataset: translation, rotation, scaling, combination, and distortion. After the manipulation, post-processing, such as JPEG compression, blurring, noise addition, and color reduction, was also performed to make the dataset more challenging. Each post-processing operation was performed with multiple parameters, for example, nine JPEG compression quality factors, three sigma values for image blurring, and three noise-averaging filters. In total, 10,400 small and 3120 large copy-move images were produced.

5) GRIP DATASET

The GRIP dataset [15] consisted of 80 copy-move images and 80 corresponding authentic images of XGA size. All copy-move images in this dataset have a single tampering region ranging from small to medium in size. The images in GRIP were manipulated using only copy-move translation, without any rotation or scaling.

6) COVERAGE DATASET

COVERAGE [113] is a copy-move image dataset with similar genuine objects in the copied regions. Because of this unique property, COVERAGE has presented a challenge for traditional keypoint-based matching detection methods. Six types of manipulation operations were employed to create tampered images: translation, scaling, rotation, free-form transformation, illumination change, and fusion. Among these six types of manipulation, the former three operations were considered simple tampering, and the latter three were complex tampering.

TABLE 1. Details of DL-based methods for solving IFD problems. The abbreviations used in this table are as follows: IW, image-wise; PW, patch-wise; O, overlapping; N-O -, non-overlapping; CM, copy-move; Sp, spliced.

	Methods	Input			Main DL architecture	Processing		Forgery type		Remarks
		IW	PW			Pre-	Post-	CM	Sp	
			O	N-O						
CNN-based	Rao *et al.* 2016 [90]		✓		8 conv. layers, 2 pooling layers	✓		✓	✓	Training R, G, B patches around the boundary of tampered regions
	Xiao *et al.* 2020 [21]	✓	✓		Back-to-back CNN, including 13 and 16 layers		✓		✓	Morphological operations, followed by adaptive filtering and convex-full filling
	Bondi *et al.* 2017 [17]			✓	4 conv. layers, 3 max pooling layers, 2 fully connected layers, and ReLU, softmax layers		✓		✓	Morphological operations were used to refine detected results
	Bondi *et al.* 2017 [89]			✓	4 conv. layers, 3 max pooling layers, 2 inner product layers, softmax layer		✓		✓	Only 128 features were used
	Cozzolino *et al.* 2018 [18]			✓	Adopted denoising CNN [109] Conv. layer + ReLU with or without batch normalization		✓		✓	Noiseprint for camera model was used
	Ali *et al.* 2022 [116]	✓			3 conv. layers followed by dense fully connected layer	✓	✓	✓	✓	Fast, light-weight model by learning double compression
Encoder-Decoder-based	Bappy *et al.* 2019 [74]			✓	Encoder used for spatial feature map detection Decoder used for forgery detection from fused features by LSTM and encoder	✓	✓	✓	✓	The decoder detected pixel-wise forgery from low-resolution feature maps
	Mazaheri *et al.* 2019 [75]			✓	Encoder-decoder used for spatial learning			✓	✓	Similar to [74]
	El Biach *et al.* 2022 [92]	✓			Encoder adopted from ResNet [93] Each decoder block consists of 3×3 convolutional kernel, followed by batch normalization and ReLU			✓	✓	After each stage, the depth size is doubled, and the input size is halved
R-CNN-based	Ahmed *et al.* 2020 [28]	✓			ResNet-convolution architecture obtained by replacing feature pyramid network in ResNet-FPN with convolutional layers		✓		✓	Mask-RCNN model was used with ResNet model [94] to extract the initial feature map
	Zhou *et al.* 2018 [98]	✓			Adopted Faster R-CNN [70] Two-stream network for RGB and noise RPN uses the RGB stream to detect the tampered regions		✓	✓	✓	Combine the spatial co-occurrence features from the RGB and noise streams
	Yancey *et al.* 2019 [97]	✓			Two-stream Faster R-CNN for RGB and JPEG compression	✓		✓	✓	Non-JPEG images were converted to JPEG prior to being input to the model for ELA
	Yancey *et al.* 2022 [99]	✓			Two-stream Faster R-CNN for RGB and JPEG compression			✓	✓	JPEG compression input was generated from Block Artifact Grid and Error Level Analysis
LSTM-based	Bunk *et al.* 2017 [73]		✓		Resampling feature detection by DNN, patch classification by LSTM		✓		✓	Random Walker [110] used for heatmap segmentation
	Bappy *et al.* 2017 [20]		✓		Pre- and post-processing by CNN, patch classification by LSTM	✓	✓	✓	✓	Only target region was detected in CM images
	Bappy *et al.* 2019 [74]			✓	Resampling feature is detected by LSTM, then combined with spatial feature map detected by encoder	✓	✓	✓	✓	Space-filling curve (Hilbert curve) was used to preserve the spatial locality of the patches in LSTM
	Mazaheri *et al.* 2019 [75]			✓	LSTM used for resampling feature detection			✓	✓	

TABLE 2. Details of forgery image datasets. In the column listing the sizes of the tampered regions, S, M, and L denote small, medium, and large, respectively. The manipulation types are denoted as follows: *a*) rotation, *b*) scaling, *c*) JPEG compression, *d*) noise addition, *e*) blurring, *f*) illumination changing, *g*) contrast adjustment, and *h*) distortion.

Dataset	# of authentic images	# of spliced images	# of copy-move images	Multiple tampered regions	Tampered regions' size	Types of manipulation
CASIA v1 [111]	800	470	451		S - M - L	*a-b-e-h*
CASIA v2 [111]	7.491	1.849	3.274		S - M - L	*a-b-e-h*
CoMoFoD [112]	260	0	13.520	✓	S	*a-c-d-e-f-g*
GRIP [15]	80	0	80		S	*d-f-g*
COVERAGE [113]	100	0	100		S - M	*a-b-f-h*
FAU [114]	48	0	48	✓	S - M	*a-b-c-d*
MICC-F600 [115]	440	0	160	✓	S - M - L	*a-b-h*

7) MICC-F600 DATASET

The MICC-F600 dataset [115] consists of 440 authentic images and 160 copy-move images with the corresponding ground truth images. However, the copy-move images in this dataset were not created skillfully; hence, the manipulation could be easily recognized by the human eye. Other

TABLE 3. Percentage of pixel-wise splicing detection of some representative methods.

Methods	CASIA 1			CASIA 2 [111]			Columbia [117]			Forensics [118]			NIST [119]			COVERAGE [113]		
	M_P	M_R	M_F	M_P	M_R	M_F	M_P	M_R	M_F	M_P	M_R	M_F	M_P	M_R	M_F	M_P	M_R	M_F
Xiao et al. 2020 [21]	-	-	-	58.1	80.8	67.58	80.4	61.2	69.5	36.7	74.7	49.2						
Bappy et al. 2017 [20]	-	-	-	-	-	-	-	-	-	-	-	72.38	-	-	76.41	-	-	61.37
Bappy et al. 2019 [74]	-	-	-	-	-	-	-	-	-	-	-	91.19	-	-	94.8	-	-	88.76
Bi et al. 2019 [104]	-	-	-	84.8	83.4	84.1	91.8	82.2	86.7	-	-	-	78.3	78.2	78.3	-	-	-
Bi et al. 2020 [103]	-	-	-	86.6	85.2	85.9	96	90.1	93	-	-	-	86.3	84.2	85.2	-	-	-
Wei et al. 2019 [96]	-	-	-	51.58	66.08	57.94	76.44	73.89	75.14	-	-	-	95.03	95.63	95.33	-	-	-
Zhou et al. 2018 [98]	-	-	40.8	-	-	-	-	-	69.7	-	-	-	-	-	72.2	-	-	43.7
Yancey et al. 2019 [99]	-	-	-	-	-	69	-	-	-	-	-	-	-	-	-	-	-	82

MICC datasets, including MICC-F8multi, MICC-F220, and MICC-F2000, were not considered here because they did not provide ground truth images.

8) FAU DATASET

The FAU dataset [114] contains 48 authentic medium- to high-resolution images that were adopted from the MICC-600 dataset. A handcrafted copy-move image was created from each authentic image. The tampered regions accounted for approximately 10% of the average image size. In addition, JPEG compression, noise addition, rotation, and scaling were used to tamper with the images. Because the manipulations were performed skillfully, all the copy-move images of this dataset appeared realistic.

B. EVALUATION METRICS

Copy-move detection studies differentiate between the source and target regions [12], [29]. However, almost all the approaches

To evaluate the performance of manipulation localization quantitatively, localization methods use pixel-oriented metrics, precision P and recall R, which are defined as follows:

$$M_P = \frac{\#\text{ correctly detected pixels}}{\#\text{ all detected pixels}}, \tag{10}$$

and

$$M_R = \frac{\#\text{ correctly detected pixels}}{\#\text{ all spliced pixels}}. \tag{11}$$

A trade-off exists between precision and recall; consequently, to consider both of these measures, their harmonic mean M_F, the F_1score, is computed as follows:

$$M_F = \frac{2M_P M_R}{M_P + M_R}. \tag{12}$$

C. BENCHMARKING THE METHODS

To the best of our knowledge, although several image forgery datasets have been published in this field, a standard dataset that contains images that meet all of the following experimental criteria: ground-truth images, data for training, testing, and validation, has not yet been constructed. Therefore, each method was used to conduct experiments on its own setup with several specific datasets, and it is difficult to compare the performance of IFD methods in the field.

Table 3 lists the image splicing detection results for popular forgery datasets. The best detection results for each dataset are highlighted in bold font.

VI. CONCLUSION

In this study, we surveyed DL-based methods published in the last five years for IFD problems. We categorized papers in which methods based on the well-known DL backbone architectures, such as CNN, LSTM, encoder-decoder, U-Net, and R-CNN, were reported. A large number of state-of-the-art methods and the most popular datasets were included in this survey. The methods were discussed according to the categories of backbone DL architectures and feature vectors. Although DL-based IFD methods have achieved promising results compared with traditional methods using handcrafted features, there is room for different DL-based methods with modifications to be considered in future research.

REFERENCES

[1] C.-S. Park and J. Y. Choeh, "Fast and robust copy-move forgery detection based on scale-space representation," *Multimedia Tools Appl.*, vol. 77, no. 13, pp. 16795–16811, Jul. 2018.
[2] C.-S. Park, C. Kim, J. Lee, and G.-R. Kwon, "Rotation and scale invariant upsampled log-polar Fourier descriptor for copy-move forgery detection," *Multimedia Tools Appl.*, vol. 75, no. 23, pp. 16577–16595, Dec. 2016.
[3] Y. Rao, J. Ni, and H. Zhao, "Deep learning local descriptor for image splicing detection and localization," *IEEE Access*, vol. 8, pp. 25611–25625, 2020.
[4] K. Asghar, X. Sun, P. L. Rosin, M. Saddique, M. Hussain, and Z. Habib, "Edge–texture feature-based image forgery detection with cross-dataset evaluation," *Mach. Vis. Appl.*, vol. 30, nos. 7–8, pp. 1243–1262, Oct. 2019.
[5] C. Li, Q. Ma, L. Xiao, M. Li, and A. Zhang, "Image splicing detection based on Markov features in QDCT domain," *Neurocomputing*, vol. 228, pp. 29–36, Mar. 2017.
[6] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proc. 9th Workshop Multimedia*, 2007, pp. 51–62.
[7] N. T. Pham, J.-W. Lee, G.-R. Kwon, and C.-S. Park, "Efficient image splicing detection algorithm based on Markov features," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12405–12419, Oct. 2018.
[8] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1299–1308, Jun. 2017.
[9] D. Moreira, A. Bharati, J. Brogan, A. Pinto, M. Parowski, K. W. Bowyer, P. J. Flynn, A. Rocha, and W. J. Scheirer, "Image provenance analysis at scale," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6109–6123, Dec. 2018.
[10] N. Pham, J.-W. Lee, G.-R. Kwon, and C.-S. Park, "Hybrid image-retrieval method for image-splicing validation," *Symmetry*, vol. 11, no. 1, p. 83, Jan. 2019.

[11] A. Bharati, D. Moreira, J. Brogan, P. Hale, K. Bowyer, P. Flynn, A. Rocha, and W. Scheirer, "Beyond pixels: Image provenance analysis leveraging metadata," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1692–1702.

[12] N. T. Pham, J.-W. Lee, and C.-S. Park, "Structural correlation based method for image forgery classification and localization," *Appl. Sci.*, vol. 10, no. 13, p. 4458, Jun. 2020.

[13] M.-J. Kwon, I.-J. Yu, S.-H. Nam, and H.-K. Lee, "CAT-Net: Compression artifact tracing network for detection and localization of image splicing," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 375–384.

[14] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim, "Learning JPEG compression artifacts for image manipulation detection and localization," *Int. J. Comput. Vis.*, vol. 130, no. 8, pp. 1875–1895, Aug. 2022.

[15] 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.

[16] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2017, pp. 159–164.

[17] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based CNN features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1855–1864.

[18] D. Cozzolino and L. Verdoliva, "Camera-based image forgery localization using convolutional neural networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1372–1376.

[19] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5302–5306.

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

[21] B. Xiao, Y. Wei, X. Bi, W. Li, and J. Ma, "Image splicing forgery detection combining coarse to refined convolutional neural network and adaptive clustering," *Inf. Sci.*, vol. 511, pp. 172–191, Feb. 2020.

[22] Y. Wei, X. Bi, and B. Xiao, "C2R Net: The coarse to refined network for image forgery detection," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1656–1659.

[23] 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.

[24] J. Jam, C. Kendrick, K. Walker, V. Drouard, J. G.-S. Hsu, and M. H. Yap, "A comprehensive review of past and present image inpainting methods," *Comput. Vis. Image Understand.*, vol. 203, Feb. 2021, Art. no. 103147.

[25] 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.

[26] C. S. Park, "Hybrid copy-move-forgery detection algorithm fusing keypoint-based and block-based approaches," *J. Internet Comput. Services*, vol. 19, no. 4, pp. 7–13, Aug. 2018.

[27] Z. He and W. Lu, "Digital image splicing detection based on Markov features in DCT and DWT domain, *Pattern Recognition*, vol. 45, no. 12, pp. 4292–4299, Dec. 2012.

[28] B. Ahmed, T. A. Gulliver, and S. Al Zahir, "Image splicing detection using mask-RCNN," *Signal, Image Video Process.*, vol. 14, no. 5, pp. 1035–1042, Jul. 2020.

[29] 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.

[30] R. Shetty, M. Fritz, and B. Schiele, "Adversarial scene editing: Automatic object removal from weak supervision," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1–11.

[31] J. Zhang, T. Fukuda, and N. Yabuki, "Automatic object removal with obstructed Façades completion using semantic segmentation and generative adversarial inpainting," *IEEE Access*, vol. 9, pp. 117486–117495, 2021.

[32] Q. Gao and X. Wu, "Real-time deep image retouching based on learnt semantics dependent global transforms," *IEEE Trans. Image Process.*, vol. 30, pp. 7378–7390, 2021.

[33] J. He, Y. Liu, Y. Qiao, and C. Dong, "Conditional sequential modulation for efficient global image retouching," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2020, pp. 679–695.

[34] K. B. Meena and V. Tyagi, "Image forgery detection: Survey and future directions," *Data, Eng. Appl.*, vol. 2, pp. 163–194, Apr. 2019.

[35] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, "Image splicing detection through illumination inconsistencies and deep learning," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3788–3792.

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

[37] 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.

[38] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1480–1502.

[39] 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.

[40] S. Sadeghi, S. Dadkhah, H. A. Jalab, G. Mazzola, and D. Uliyan, "State of the art in passive digital image forgery detection: Copy-move image forgery," *Pattern Anal. Appl.*, vol. 21, no. 2, pp. 291–306, May 2018.

[41] S. Walia and K. Kumar, "Digital image forgery detection: A systematic scrutiny," *Austral. J. Forensic Sci.*, vol. 51, no. 5, pp. 488–526, Sep. 2018.

[42] N. B. A. Warif, M. Y. I. Idris, A. W. A. Wahab, N.-S.-N. Ismail, and R. Salleh, "A comprehensive evaluation procedure for copy-move forgery detection methods: Results from a systematic review," *Multimedia Tools Appl.*, vol. 81, no. 11, pp. 15171–15203, May 2022.

[43] W. D. Ferreira, C. B. R. Ferreira, G. C. Júnior, and F. Soares, "A review of digital image forensics," *Comput. Electr. Eng.*, vol. 85, Jul. 2020, Art. no. 106685.

[44] G. Kaur, N. Singh, and M. Kumar, "Image forgery techniques: A review," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1577–1625, Jun. 2022.

[45] L. Zheng, Y. Zhang, and L. Vrizlynn, "A survey on image tampering and its detection in real-world photos," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 380–399, Jan. 2019.

[46] L. Niu, W. Cong, L. Liu, Y. Hong, B. Zhang, J. Liang, and L. Zhang, "Making images real again: A comprehensive survey on deep image composition," 2021, *arXiv:2106.14490*.

[47] Z. Zhang, C. Wang, and X. Zhou, "A survey on passive image copy-move forgery detection," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 6–31, 2018.

[48] A. Roy, R. Dixit, R. Naskar, and R. S. Chakraborty, "Copy-move forgery detection in digital images survey and accuracy estimation metrics," in *Digital Image Forensics*. Singapore: Springer, 2020, pp. 27–56.

[49] P. Korus, "Digital image integrity—A survey of protection and verification techniques," *Digit. Signal Process.*, vol. 71, pp. 1–26, Dec. 2017.

[50] K. B. Meena and V. Tyagi, "Image splicing forgery detection techniques: A review," in *Proc. Int. Conf. Adv. Comput. Data Sci.*, 2021, pp. 364–388.

[51] I. C. Camacho and K. Wang, "A comprehensive review of deep-learning-based methods for image forensics," *J. Imag.*, vol. 7, no. 4, p. 69, Apr. 2021.

[52] A. B. Z. Abidin, H. B. A. Majid, A. B. A. Samah, and H. B. Hashim, "Copy-move image forgery detection using deep learning methods: A review," in *Proc. 6th Int. Conf. Res. Innov. Inf. Syst. (ICRIIS)*, Dec. 2019, pp. 1–6.

[53] M. T. Vo, A. H. Vo, T. Nguyen, R. Sharma, and T. Le, "Dealing with the class imbalance problem in the detection of fake job descriptions," *Comput., Mater. Continua*, vol. 68, no. 1, pp. 521–535, 2021.

[54] M. T. Vo, T. Nguyen, H. A. Vo, and T. Le, "Noise-adaptive synthetic oversampling technique," *Int. J. Speech Technol.*, vol. 51, no. 11, pp. 7827–7836, Nov. 2021.

[55] S. T. Nabi, M. Kumar, P. Singh, N. Aggarwal, and K. Kumar, "A comprehensive survey of image and video forgery techniques: Variants, challenges, and future directions," *Multimedia Syst.*, vol. 28, no. 3, pp. 939–992, Jun. 2022.

[56] S. Tyagi and D. Yadav, "A detailed analysis of image and video forgery detection techniques," *Vis. Comput.*, pp. 1–21, Jan. 2022.
[57] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[60] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, Sep. 2018.
[61] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, Sep. 2020.
[62] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, Oct. 2020, pp. 1–7.
[63] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Sep. 2015.
[64] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.
[65] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.
[66] I. Neutelings. (2021). *Neural Networks*. Accessed: Sep. 22, 2022. [Online]. Available: https://tikz.net/neural_networks/
[67] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," Mar. 2020, *arXiv:2003.05991*.
[68] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[69] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[70] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[71] Z. Zhou, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Dec. 2020.
[72] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Deep Learning Med. Image Anal. Multimodal Learn. Clin. Decis. Support, 4th Int. Workshop, (DLMIA) 8th Int. Workshop, ML-CDS Held Conjunct. (MICCAI)*, 2018, pp. 3–11.
[73] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson, "Detection and localization of image forgeries using resampling features and deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1881–1889.
[74] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3286–3300, Jul. 2019.
[75] G. Mazaheri, N. C. Mithun, J. H. Bappy, and A. K. Roy-Chowdhury, "A skip connection architecture for localization of image manipulations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 119–129.
[76] C. Olah. (Aug. 2015). *Understanding LSTM Networks, Colah's Blog*. Accessed: Aug. 12, 2021. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[77] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[78] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
[79] T. Nikoukhah, J. Anger, T. Ehret, M. Colom, J. M. Morel, and R. G. V. Gioi, "JPEG grid detection based on the number of DCT zeros and its application to automatic and localized forgery detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9.
[80] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[81] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jan. 2008.
[82] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image copy-move forgery detection based on SURF," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, 2010, pp. 889–892.
[83] C. Wang, Z. Zhang, Q. Li, and X. Zhou, "An image copy-move forgery detection method based on SURF and PCET," *IEEE Access*, vol. 7, pp. 170032–170047, 2019.
[84] H. P. Truong, T. P. Nguyen, and Y.-G. Kim, "Weighted statistical binary patterns for facial feature representation," *Int. J. Speech Technol.*, vol. 52, no. 2, pp. 1893–1912, May 2021.
[85] P. Thodoroff, J. Pineau, and A. Lim, "Learning robust features using deep learning for automatic seizure detection," 2016, *arXiv:1608.00220*.
[86] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time-series," in *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.
[87] W. Lin, K. Hasenstab, G. M. Cunha, and A. Schwartzman, "Comparison of handcrafted features and convolutional neural networks for liver MR image adequacy assessment," *Sci. Rep.*, vol. 10, no. 1, p. 20336, Nov. 2020.
[88] M. V. Valueva, N. N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Math. Comput. Simul.*, vol. 177, pp. 232–243, Nov. 2020.
[89] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 259–263, Mar. 2017.
[90] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.
[91] J. Fridrich and J. Kodovskỳ, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
[92] F. Z. El Biach, I. Iala, H. Laanaya, and K. Minaoui, "Encoder–decoder based convolutional neural networks for image forgery detection," *Multimedia Tools Appl.*, vol. 81, no. 16, pp. 22611–22628, Jul. 2022.
[93] 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.
[94] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, "Constrained R-CNN: A general image manipulation detection model," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.
[95] J. Zhou, J. Ni, and Y. Rao, "Block-based convolutional neural network for image forgery detection," in *Digital Forensics and Watermarking* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2017, pp. 65–76.
[96] X. Wei, Y. Wu, F. Dong, J. Zhang, and S. Sun, "Developing an image manipulation detection algorithm based on edge detection and faster R-CNN," *Symmetry*, vol. 11, no. 10, p. 1223, Oct. 2019.
[97] R. E. Yancey, N. Matloff, and P. Thompson, "Multi-linear faster RCNN with ELA for image tampering detection," *CoRR*, Jun. 2019.
[98] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1053–1061.
[99] R. Elizabeth Yancey, "Deep localization of mixed image tampering techniques," 2019, *arXiv:1904.08484*.
[100] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
[101] B. Ahmed, T. A. Gulliver, and S. Al Zahir, "Localization and detection of copy-move forgeries in post-processed images using U-Net," *Social Netw. Comput. Sci.*, vol. 2, no. 6, p. 476, Sep. 2021.
[102] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong, and X. Cui, "DCU-Net: A dual-channel U-shaped network for image splicing forgery detection," *Neural Comput. Appl.*, pp. 1–17, Aug. 2021.
[103] B. Liu, R. Wu, X. Bi, B. Xiao, W. Li, G. Wang, and X. Gao, "D-UNet: A dual-encoder U-Net for image splicing forgery detection and localization," 2020, *arXiv:2012.01821*.

[104] X. Bi, Y. Wei, B. Xiao, and W. Li, "RRU-Net: The ringed residual U-Net for image splicing forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 30–39.
[105] Y. Goldberg, "A primer on neural network models for natural language processing," *J. Artif. Intell. Res.*, vol. 57, pp. 345–420, Nov. 2016.
[106] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. AAAI Conf. Artif. Intell.*, Dec. 2015, pp. 1–9.
[107] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[108] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, Jun. 2013, pp. 1310–1318.
[109] 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.
[110] L. Grady, "Multilabel random Walker image segmentation using prior models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 763–770.
[111] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, Jul. 2013, pp. 422–426.
[112] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic, "CoMoFoD—New database for copy-move forgery detection," in *Proc. Int. Symp. Electron. Mar. (ELMAR)*, Sep. 2013, pp. 49–54.
[113] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "COVERAGE—A novel database for copy-move forgery detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 161–165.
[114] 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.
[115] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1099–1110, Sep. 2011.
[116] S. S. Ali, I. I. Ganapathi, N.-S. Vu, S. D. Ali, N. Saxena, and N. Werghi, "Image forgery detection using deep learning by recompressing images," *Electronics*, vol. 11, no. 3, p. 403, Jan. 2022.
[117] T.-T. Ng and S.-F. Shih, "A data set of authentic and spliced image blocks," Columbia Univ., USA, Tech. Rep. #203-2004-3, 2004.
[118] "IEEE IFS-TC image forensics challenge, website up for new submissions," *IEEE Signal Process. Soc.*, Dec. 2013.
[119] NIST. (Aug. 2016). *Open Media Forensics Challenge*. [Online]. Available: https://www.nist.gov/itl/iad/mig/open-media-forensics-challenge

NAM THANH PHAM received the B.S. and M.S. degrees in computer science from Vietnam National University, Hanoi, in 2012 and 2015, respectively, and the Ph.D. degree from Sejong University, in 2020. From 2015 to 2016, he was an Internship Student with the National Institute of Informatics, Tokyo, Japan. His research interests include image processing, computer vision, and deep learning.

CHUN-SU PARK received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, in 2003 and 2009, respectively. From 2009 to 2010, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, USA. From 2010 to 2012, he was a Senior Research Engineer with Samsung Electronics. From 2012 to 2014, he was an Assistant Professor with the Department of Information and Telecommunication Engineering, Sangmyung University. From 2014 to 2016, he was an Associate Professor with the Department of Digital Contents, Sejong University. He joined the Department of Computer Education, Sungkyunkwan University, in 2017, where he is currently an Associate Professor. His research interests include video signal processing, parallel computing, and multimedia communications.