A Skip Connection Architecture for Localization of Image Manipulations

Ghazal Mazaheri[1], Niluthpol Chowdhury Mithun[1], Jawadul H. Bappy[2], and Amit K. Roy-Chowdhury[1]

[1]University of California, Riverside, CA 92521
[2]JD.Com American Technologies Corporation, Mountain View, CA 94043

gmaza002@ucr.edu, nmithun@ece.ucr.edu, jawadul.bappy@jd.com, amitrc@ece.ucr.edu

Abstract

Detection and localization of image manipulations are becoming of increasing interest to researchers in recent years due to the significant rise of malicious content-changing image tampering on the web. One of the major challenges for an image manipulation detection method is to discriminate between the tampered regions and other regions in an image. We observe that most of the manipulated images leave some traces near boundaries of manipulated regions including blurred edges. In order to exploit these traces in localizing the tampered regions, we propose an encoder-decoder based network where we fuse representations from early layers in the encoder (which are richer in low-level spatial cues, like edges) by skip pooling with representations of the last layer of the decoder and use for manipulation detection. In addition, we utilize resampling features extracted from patches of images by feeding them to LSTM cells to capture the transition between manipulated and non-manipulated blocks in the frequency domain and combine the output of the LSTM with our encoder. The overall framework is capable of detecting different types of image manipulations simultaneously including copy-move, removal, and splicing. Experimental results on two standard benchmark datasets (CASIA 1.0 and NIST'16) demonstrate that the proposed method can achieve a significantly better performance than the state-of-the-art methods and baselines.

1. Introduction

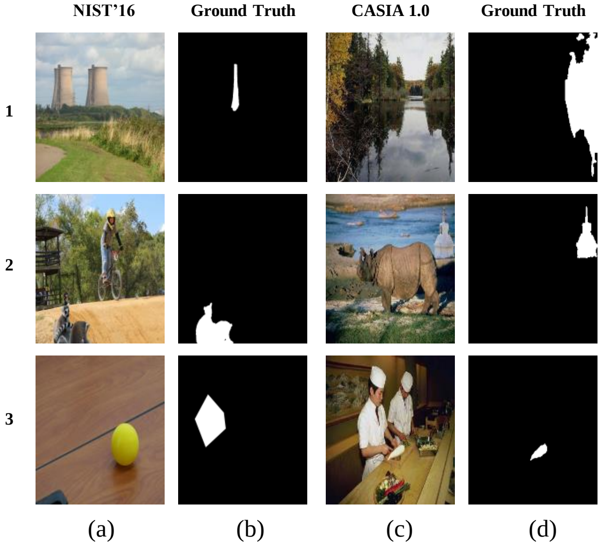


Figure 1. Example of tampered images from (a) NIST'16 [1] and (c) CASIA 1.0 [20]. (a) shows copy-move (first row), splicing (middle row), and removal (third row) manipulations with their corresponding ground truths in column (b). (c) shows copy-move (first row), and splicing (middle and third rows) manipulations with their corresponding ground truths in column (d).

The advent of high technology devices like cameras, smartphones, and tablets have led to significant improvement and availability of image editing programs, e.g., Photoshop, Gimp, Snapseed, and Pixlr. These editors provide users an opportunity for digital altering and tampering of an image without leaving visible traces. Image manipulation is becoming a serious concern as there are increasingly more cases of people trying to hide or add some parts of images for the purpose of misleading. Existing technology offers many tools for skillful manipulators to hide the traces of manipulation in such a way that naked eyes are unlikely to be able to identify image tampering. The types of image manipulation can broadly be classified into two main categories: (1) content-preserving, and (2) content-changing. Content-preserving manipulations (e.g., compression, blur, and contrast enhancement) are considered as less harmful since they do not change the semantic content.

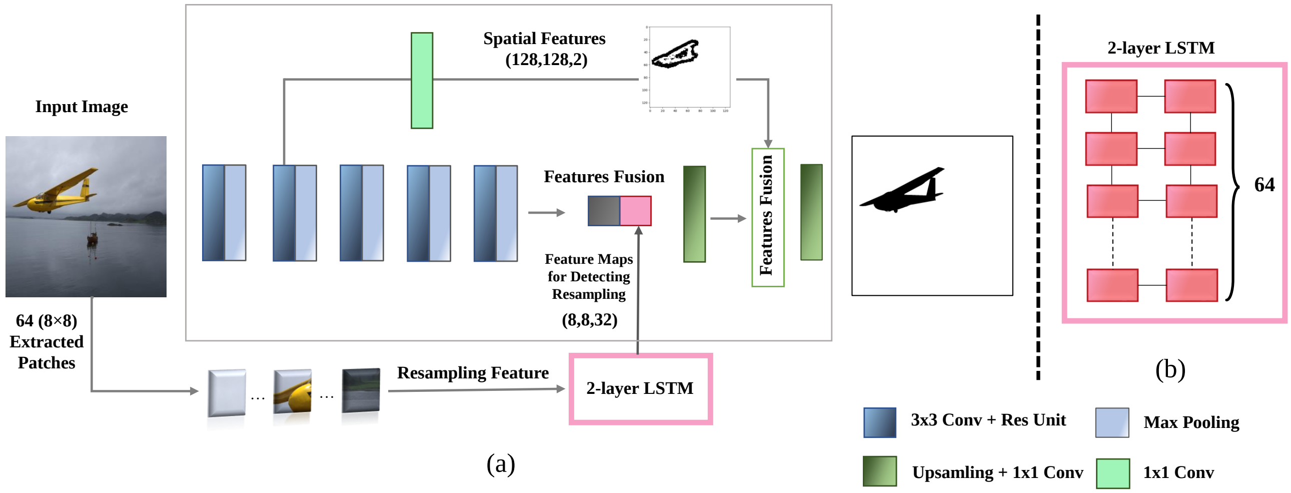


Figure 2. (a) A brief illustration of the proposed framework (LSTM-EnDec-Skip). Encoder-decoder architecture consists of convolutions to learn spatial information. Skip connection is used to take advantage of early layers in CNN which are rich in spatial details. (b) 2-layer LSTM with 64 cells at each layer. Please see Sec. 1.1 for an overview and Sec. 3 for the details.

However, the latter type (e.g., copy-move, splicing, and object removal) is critical as image content is changed arbitrarily and consequently semantic meaning altered. Fig. 1 shows some example images that undergone different tampering techniques from NIST'16 [1] and CASIA 1.0 [20] datasets. These images have been post-processed to deceive the human perceptual system. In recent years, there has been noticeable interest in detecting and localizing content-changing image manipulation. Most prior works on this problem focus on classifying an image as manipulated or non-manipulated [9, 21, 29, 38]. There are also a few works that classify as well as localize manipulated regions in images [6, 5, 65]. For security/surveillance applications, it is critical to not only determine images manipulated by different techniques but also to localize the tampered regions in manipulated images to prevent attackers. In this paper, we present a novel architecture to detect and localize manipulated regions at the pixel level. A significant number of image manipulation approaches focus on a specific type of manipulation (e.g., copy-move, and splicing) [58, 32, 51]. One approach might not do well on other types of tampering. Also, it seems unrealistic to assume that the type of manipulation will be known beforehand. In contrast to most prior works that focus on some specific type of manipulation, we have proposed a method which can localize all three major content-changing manipulation types including copy-move, splicing, and object removal. Our method is inspired by the observation that boundaries of the manipulated regions in an image are often smooth and not as sharp as edges in other regions. We develop a deep convolutional neural network (DCNN) based architecture to exploit this characteristic of tampered artifacts near manipulative edges to improve image forgery detection. Although post-processing procedures may visually hide the traces of tampering in most manipulated images, we believe exploiting features extracted by early layers of convolutional neural networks (which are rich in spatial details) can retrieve sufficiently strong cues for image manipulation detection. While early layers in CNNs mostly encode low-level and generic image cues (e.g., points, and edges), the deeper layer encodes higher-level or specific image cues. As a manipulation detection approach needs to also analyze higher-level image cues along with detecting smooth edges, a combination of spatial resolution cues from different layers would potentially boost detection performance. In this regard, our framework uses a fusion of low-level and high-level representations for manipulation detection (Fig. 2). Besides, we use resampling features to capture artifacts like JPEG quality loss, upsampling, downsampling, rotation, and shearing. These features are extracted by Laplacian filter along with Radon transform and fed to LSTM for learning the transition between manipulated and non-manipulated regions. Experimental results on two standard benchmark datasets demonstrate that the proposed approach outperforms the baselines and existing methods by a large margin with more than 5% absolute improvement in terms of AUC(ROC).

1.1. Framework Overview

Our framework consists of three main parts including 1) LSTM network 2) Encoder-Decoder and 3) Skip connections. Fig. 2 demonstrates our proposed architecture. Resampling features of each patch are extracted and used as input for the LSTM network [25]. The main role of LSTM cells is to learn the transition between manipulated and non-manipulated blocks in the frequency domain. Encoder-Decoder consists of convolutions to learn spatial information and provide a finer representation of the binary mask. Layers in the encoder are a residual block (two convolution layers with short connection), max-pooling, batch normalization, and Rectified Linear Unit (ReLU) as the activation function. A decoder which follows encoder uses a fusion of LSTM output features and low-level features of the encoder as input. The decoder is constructed by upsampling, convolution, batch normalization, and activating feature maps layers. Low-level layers are rich in spatial high-resolution details which can play an important role in detecting blur edges as the key cues for forgery detection. In order to take advantage of blur boundaries in manipulated images, we have concatenated the high-level and the low-level features to find the best layer with rich features suited well for forgery detection. In order to concatenate early layers in encoder and up-sampled intermediate features, we have applied a 1×1 convolution to have the same number of feature map as the intermediate ones. Upsampled features of this concatenation lead to final features for pixel-wise prediction of manipulation.

1.2. Main Contribution

The main contributions of this work are the following. First, we propose a new framework for detecting manipulated image regions by a fusion of spatial features and resampling features extracted in the frequency domain. Second, in order to utilize low-level image cues effectively in detecting boundaries of manipulated regions, we exploit skip connections in a deep CNN based semantic segmentation network to fuse high-level spatial cues with low-level ones. Third, our approach outperforms the baselines and state-of-the-art methods by a large margin, with more than 6% and 5% absolute improvement on NIST'16 [1] and CASIA 1.0 [20] datasets respectively in terms of AUC(ROC). We also analyze the influence of features extracted from different layers of the encoder (as shown in Fig. 4) on the final prediction task by our ablation study.

2. Related Work

There have been variety of works in the field of image forensics for image manipulation detection and localization. Some of them focus on specific types of manipulation including resampling detection [7, 43], detection of copy-move [58, 36, 18, 66], splicing [51, 15, 57, 50, 42, 32, 61, 41], and object removal [60, 54], while a few cover two or more types of manipulation [53], [65], [6]. We briefly discuss some existing works below.

Traditional Physics-based Methods. Before the advances in deep learning-based approaches, traditional image processing-based approaches were very popular to distinguish tampered images. Among the physics-based approaches, frequency domain characteristics and/or statistical properties of images have been explored in several prior works [28, 56, 34, 55]. Analysis of artifacts by multiple JPEG compressions [14, 55] and adding noise to the JPEG compressed image in order to improve the performance of resampling detection [40, 27] are other prominent works in this direction.

Learning-based Methods. In recent years, inspired by the success of deep neural networks in different visual recognition tasks in computer vision, deep learning-based approaches have been popular choices for image forgery detection. Some recent deep learning-based methods such as stacked auto-encoders (SAE) [24] and convolutional neural networks (CNN) [17, 46, 19, 22, 39, 23, 63, 2, 52, 44, 16] have been applied to detect/classify image manipulations. Authors in [10] exploit a convolutional neural network (CNN) to extract characteristic camera model features from image patches and analyze them by use of iterative clustering techniques. To detect JPEG compression, [3] takes advantage of machine learning methods. In [11], authors use the ideas of image querying and retrieval to provide clues to better localize forgeries. Work in [26] exploits an algorithm to reduce the amount of information by having it learn to localize manipulations without ground-truth annotations. Facial retouching is one of the types of image manipulation which has attracted attention in this field. A deep learning approach to identify facial retouching has been applied in [8, 45, 64, 59]. In order to train deep learning-based approaches in a supervised fashion, authors in [49] proposed FaceForensics++ as a database of facial forgeries.

In order to identify the exact position of the manipulated regions in an image, it is necessary to exploit techniques which are able to localize these regions. Works in [62, 5, 12, 32] use machine learning techniques in order to do patch classification. Authors in [65] use object detection method proposed in [47] to identify fake objects. Unlike [65] which utilizes bounding box to coarsely localize manipulated object, we adopt a segmentation approach to segment out manipulated regions by classifying each pixel (manipulated/non-manipulated). Semantic segmentation approaches are suitable for fine-grained localization of tampered regions in an image. A typical semantic segmentation approach focuses on segmenting all meaningful regions (objects). However, a segmentation approach for localization of image manipulation needs to focus only on the possible tampered regions which bring additional challenges to an existing challenging problem. To localize tampered regions, [6] used an LSTM Encode-Decoder architecture and showed good accuracy. We also adopt a similar architecture in this work. The major difference of our approach to [6] is taking advantage of spatial high-resolution details from the early layers of CNN. Our effective use of low-level details in the deep CNN based architecture improves manipulation detection performance significantly.

3. Approach

Image manipulation techniques can be divided into three most popular categories, i.e., copy-move, splicing, and object removal. One of the major challenges in developing a robust manipulation detection method is to simultaneously handle different types of image forgeries. In order to develop an effective general architecture for recognizing manipulation in all three categories mentioned, we used LSTM Encoder-Decoder architecture as the backbone. Our framework simultaneously exploits resampling features in the frequency domain and spatial features extracted from different layers of an encoder-decoder CNN for pixel-wise predictions of manipulation. In order to exploit blurred edges as a primary cue for manipulation localization, we fuse early layers of the encoder which are rich in spatial details with the final layer of the decoder to make the final prediction. We discuss different components of our framework below.

LSTM. In order to localize manipulation in images, besides CNN to extract spatial features, we utilize resampling features to detect manipulations like upsampling, downsampling, and compression. These resampling features in frequency domain can be extracted using Laplacian filter along with Radon transform which has been used in [12]. In this paper, we utilize the LSTM [25] network to learn the correlation between blocks of resampling features as shown in Fig. 2. For an input image with size of $256 \times 256 \times 3$, we first divide it to 64 (8×8) non-overlapping patches. Therefore, each patch has dimension of $32 \times 32 \times 3$. Choosing patch dimension is a challenging issue since, on one hand, resampling is more detectable in larger patch sizes as the resampling signal has more repetitions, on the other hand, small manipulated regions will not be localized that well in small patches. We try to choose a trade-off in selecting the patch size.The first step to produce resampling features is application of Laplacian filter. We use the magnitude square root of 3×3 Laplacian filter which produces an image of the magnitude of linear predictive error as presented in [12]. Next, Radon transform is applied in order to find correlations in the linear predictor error by accumulating errors along various angles of projection. Finally, in the last step to produce resampling features, we take FFT to find the periodic nature of the signal.

Resampling features are the key parts in forgery detection as they capture the characteristics of different artifacts such as JPEG quality above or below a threshold, upsampling, downsampling, rotation clockwise, rotation counterclockwise, and shearing. Therefore, in order to have a network which can detect the most possible manipulation techniques, we need to utilize them. LSTMs are kind of recurrent neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, or numerical times series data. In computer vision, LSTM network has been successfully used to capture the dependency among a series of pixels [13]. In order to detect the correlation among the pixels, we utilize LSTMs with resampling features as inputs. Ordering of patches can effect the performance of LSTMs. Therefore, we use Hilbert curves as they give a mapping between 1D and 2D space that preserves locality fairly well. The Hilbert curve has been shown to outperform many other curves in maintaining the spatial locality, when transforming from a multidimensional space to a one-dimensional space [37]. The basic element of a Hilbert curve is a U-shape. In the first order Hilbert curve, we have a 2×2 square grid. We start with our string in the top left corner, and drape it through the other three squares in the grid to finish in the top right corner. Now imagine that we double the size of the grid to make a 4×4 grid for second order Hilbert curve. The second order Hilbert curve replaces that U-shape by four (smaller) ones, which are linked together. As we have total 64 (8×8) blocks extracted from an image, we need to use third order of Hilbert curve. After determining the order of patches by Hilbert curve, resampling features of these patches are fed to LSTM network.

Encoder-Decoder. Semantic segmentation has wide application in image analysis tasks. Encoder-decoder based networks is a popular choice for pixel-wise segmentation of images. In [4, 48] encoder-decoder architecture has been presented for semantic segmentation. In these cases, deep neural network architectures are presented where convolutional layers are utilized in order to produce spatial heat maps for semantic segmentation. In [6] SegNet [4] has been applied as encoder-decoder network along with LSTM as frequency domain feature extractor. Unlike [6], we follow U-Net [48] architecture for our encoder-decoder network which we empirically found to be more effective for image manipulation detection. Each layer of encoder network consists of convolution, residual unit, pooling, and activation functions. Besides long skip connections, we take advantage of short skip connections in Residual blocks. Convolution layers have increasing number of filters in encoder which are followed by batch normalization and rectified linear unit (ReLU) as an activation function. We apply max-pooling with stride 2 at the end of each layer in the encoder.

Decoder network consists of convolution to decrease the number of feature maps and upsampling to produce binary mask with the same size of original image. Different structures for decoder with different feature maps are shown in Fig. 1 and 4. In the last layer of decoder, two heat maps are used for the prediction of manipulated and non-manipulated class. Each decoder follows basic operations - upsample, convolution, and batch normalization. A softmax layer is added at the end of the network for segmentation prediction and classification.

Skip Connections. Both spatial details from early layers and semantic ones from higher ones play an important role in image segmentation. Success of U-Net [48] shows the effect of layer fusion on image segmentation. Skip connections help traverse information in deep neural networks. Corresponding to our task for image forgery localization, we have to find out the most effective features from either high-level or low-level layers. In other words, we are trying to find the layers which are rich in features for forgery detection. Manipulated boundaries are the prominent features to localize fake regions of an image. In most cases, manipulated regions have smooth boundaries. The best option to focus on boundaries of an image is to take advantage of early layers in CNN which are rich in spatial details. In our proposed network, we use this important feature to improve the state-of-art frameworks. Thus, we concatenate the early layer of convolution network in encoder into the last layer in decoder. In spite of U-net architecture, we have just used the spatial information of early layers in encoder. This network is shown in Fig. 2

Training Loss. During training, we use cross entropy loss which is minimized to find the optimal set of parameters of the network. Let θ be the parameter vector corresponding to image tamper localization task. So, the cross entropy loss can be computed as

$$L(\theta) = \frac{-1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N}\mathbb{1}(Y^m = n)\log(Y^m = n|y^m;\theta) \tag{1}$$

Here, M and N denote the total number of pixels and the number of class. y represents the input pixel. $\mathbb{1}(.)$ is an indicator function, which equals to 1 if $m = n$, otherwise it equals 0. We use adaptive moment estimation (Adam) optimization technique in order to minimize the loss of the network, shown in Eqn. 1. After optimizing the loss function over several epochs, we learn the optimal set of parameters of the network. With these optimal parameters, the network is able to predict pixel-wise classification given a test image.

4. Experiments

In this section, we show our results for the proposed model in comparison to state-of-art ones. As current datasets do not have enough data for training process, we pre-train our model on synthesized data created by [6]. In addition, we finetune our model with training set provided by NIST'16 [1] and CASIA 2.0 [20]. We have evaluated our model on two datasets including NIST'16 [1] and CASIA 1.0 [20].

4.1. Datasets

NIST'16 [1] is one of the most challenging datasets for the task as it contains images tampered by all three tampering techniques including splicing, copy-move, and removal. We utilize this dataset in order to show the ability of our approach in detecting all kinds of manipulated images. Also, the images are post-processed in order to hide detectable cues which make them more complicated.

CASIA [20] is one of the common datasets which has been used to evaluate models for forgery detection. In this dataset, two techniques including copy-move and splicing are used to manipulate images. We exploit CASIA 2.0 in training and CASIA 1.0 as the testing set.

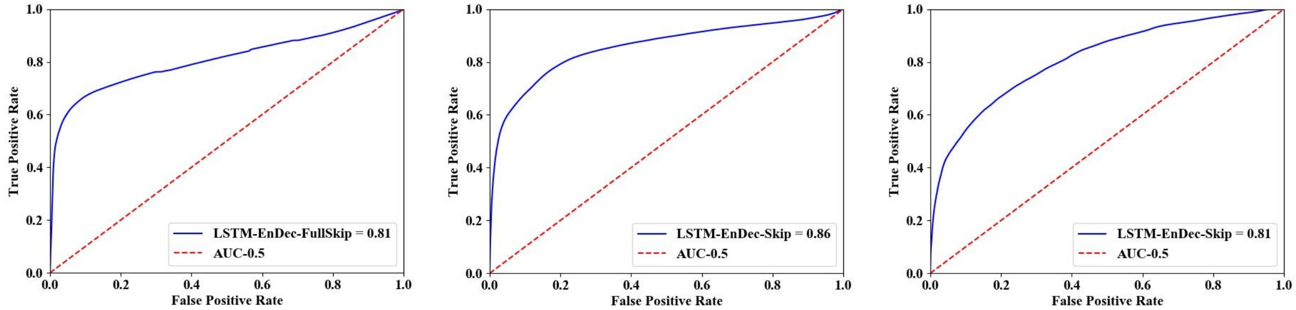
4.2. Experimental Analysis

In this section, we discuss our evaluation results and compare our models with state-of-art approaches. To train the model, we use TensorFlow to define different layers of the network. To run the experiment, we utilize multi-GPU setting. We exploit two NVIDIA Tesla K80 GPUs in different sets of experiments. We use Adam optimizer with a fixed learning rate of 0.0003. We train our model for 200K iterations with a batch size of 16.

Evaluation Metric. We use F1 score and receiver operating characteristic (ROC) curve as evaluation metrics for comparing the performance of models. ROC curve measures the performance of binary classification task by varying the threshold on prediction score. The area under the ROC curve (AUC) is computed from the ROC curve that measures the ability of a system for binary classification and allows comparison between different methods. F1 score is a pixel level evaluation metric for image manipulation detection. We vary different thresholds and use the highest F1 score as the final score for each image.

4.2.1 Comparison against Existing Approaches

Some of the tamper localization techniques include DCT Histograms [31], ADJPEG [9], NADJPEG [9], PatchMatch [18], Error level analysis [33], Block Features [30], Noise Inconsistencies [35], J-Conv-LSTM-Conv [5], and LSTM-EnDec [6]. In order to compare our proposed architecture with existing approaches, we measure the performance of our method using area under the ROC curve (AUC) metric tested on NIST'16 [1] dataset. From Table 1, it can be observed that our proposed network outperforms the existing methods (6.40%). The main reason why our method achieves better performance than LSTM-EnDec [6] is that LSTM-EnDec-Skip (ours) exploits spatial information of early layers to seek tampering artifacts including blurred edges. We also compare our network with Error level analysis [33], Noise Inconsistencies [35], CFA



(a) (b) (c)

Figure 3. ROC Curve on (a), (b) NIST'16 [1], and (c) CASIA 1.0 [20] for pixel-wise classification (segmentation).

Table 1. Pixel level AUC comparison on NIST16 [1] dataset. The proposed (LSTM-EnDec-Skip) approach performs significantly better than several state-of-the-art approaches and baselines.

Method	NIST'16
DCT Histograms	0.545
ADJPEG	0.589
NADJPEG	0.656
PatchMatch	0.651
Error level analysis	0.428
Block Features	0.478
Noise Inconsistencies	0.487
LSTM-EnDec	0.793
LSTM-EnDec-Skip	**0.857**

pattern estimation [21], and RGB-N [65] tested on CASIA 1.0 [20] dataset. Table 2 shows AUC and F1 score comparison between our methods and some state-of-art ones. From this table, we can see that our method outperforms existing methods in both AUC and F1 score on CASIA [20] dataset. Our method has better results than conventional methods like Error level analysis [33] and Noise Inconsistencies [35] since they all focus on specific tampering artifacts for localization, which limits their performance while ours exploits both resampling and low-level features.

4.2.2 ROC Curve

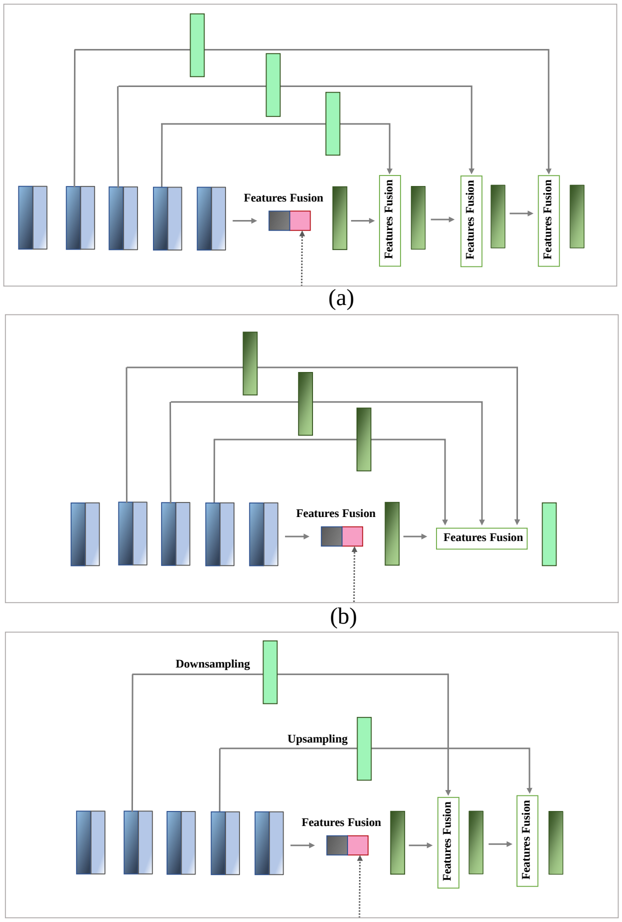
ROC curve illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. Figures 3 (a,b) show the ROC plots for image tamper localization on NIST16 [1] for LSTM-EnDec-FullSkip (section 4.2.3) and LSTM-EnDec-Skip and Fig. 3 (c) demostrates ROC curve of LSTM-EnDec-Skip on CASIA 1.0 [20]. We also measure the area under the ROC curve. Table 1 and Table 2 demonstrate the AUC scores obtained by different approaches. Our model achieves AUC of 0.857, 0.814 on NIST16 and CASIA 1.0 respectively. From the ROC curves as shown in Figs. 3 (b) and (c), we can see that the proposed network classifies tampered pixels with high confidence.

Table 2. Pixel level AUC and F1 score comparison on CASIA [20] dataset. We observe that the proposed LSTM-EnDec-Skip approach outperforms other methods by a large margin in both the evaluation metric.

Method	AUC score	F1 score
Error level analysis	0.613	0.214
Noise Inconsistencies	0.612	0.263
CFA1	0.522	0.207
RGB-N	0.795	0.408
LSTM-EnDec	0.762	0.391
LSTM-EnDec-Skip	**0.814**	**0.432**

4.2.3 Performance with Different Decoder Network

In order to emphasize the importance of early layers of encoder in manipulation localization, we perform an ablation study with three different decoder architectures. In each convolution layer of encoder, we use kernel size of $3 \times 3 \times d$, where d is the depth of a filter. In the first layer of convolution, we use 32 feature maps. Number of feature maps doubles in each convolution layer and size of them reduces by factor 2 after each max-pooling. Thus, we have 32, 64, 128, 256, and 512 feature maps in the first, second, third, fourth, and fifth layer of encoder architecture respectively. After 5 layers of max-pooling, 8×8 size feature maps are produced as the outputs of encoder. In the residual unit, we utilize 2 convolution layers and short skip connections to sum the features. We utilize batch normalization at each convolutional layer. As an activation function, we choose



(a)

(b)

(c)

Figure 4. Different fusion of low-level and high-level features. (a) LSTM-EnDec-FullSkip (b) LSTM-EnDec-FullSkip-2 (c) LSTM-EnDec-CrossoverSkip

rectified linear unit (ReLU) which follows each convolutional layer in encoder. We employ 3 × 3 size kernel for decoder network.

- **LSTM-EnDec-FullSkip.** This architecture uses the idea of U-nets [48] for image segmentation. In order to see the effect of both low-level and high-level features, we hierarchically fuse each layer of encoder to the corresponding layer in decoder. Therefore, we take advantage of both semantic and spatial information of layers in encoder. This architecture is shown in Fig. 4(a).
- **LSTM-EnDec-FullSkip-2.** In this architecture, we use all upsampled layers of encoder. we apply convolution with kernel size of $3 \times 3 \times 32$ before concatenating them all together in order to have same number of layers for all skip connections. Also a $1 \times 1 \times 2$ convolution is applied in the last layer to produce final masks. This architecture exploits fusion of all low-level and high-level features, shown in Fig. 4(b).
- **LSTM-EnDec-CrossoverSkip.** In order to use semantic details in the last layer of decoder, instead of concatenating deep layers in encoder to early layers in decoder (similar to U-net), we fuse deep layers to the last layer of decoder. Also, spatial information from early layers of encoder is fused to early layers in decoder. This architecture is shown in Fig. 4(c).

Table 3. Pixel level AUC comparison with different decoder networks on NIST16 [1] dataset to analyze the proposed network.

Method	NIST'16
LSTM-EnDec-FullSkip	0.812
LSTM-EnDec-FullSkip-2	0.765
LSTM-EnDec-CrossoverSkip	0.781
LSTM-EnDec-Skip	0.857

In order to compare the performance of models discussed in this section, we test our finetuned models on NIST16 [1]. Table 4.2.3 shows the area under the ROC curve (AUC) for different architectures. From this table, it is clear that our proposed architecture, LSTM-EnDec-Skip, outperforms others. This is due to usage of low-level layer which is rich in high-resolution features to detect manipulation. As [5] has mentioned, blurred edges as tampering artifacts play important role in forgery detection. Therefore, we combine the high-level semantic details with the low-level ones to keep spatial resolution and make a better estimation of manipulated regions.

4.2.4 Qualitative Analysis of Segmentation

We present some qualitative results in Fig. 5 and 6 for a comparison between LSTM-EnDec [6] and LSTM-EnDec-Skip (Fig. 2) networks in two-class image manipulation localization. The images are selected from the NIST'16 [1] and CASIA 1.0 [20]. It is evident from the figures that our proposed method outperforms LSTM-EnDec [6] in localizing manipulated regions in most cases. More accurate output masks can ease the problem of recognizing exact tampering region. NIST'16 [1] is one of the challenging datasets as it contains three types of manipulation. Among different types of manipulations, object removal is considered more complicated to identify as it leaves less distinguishable traces. We observe that our approach performs better than LSTM-EnDec in case of object removal manipulation detection. From Fig. 5, we observe that our approach performs significantly better in localizing the removal region. LSTM-EnDec [6] fails to detect the manipulated region in the third image and predicts some non-manipulated

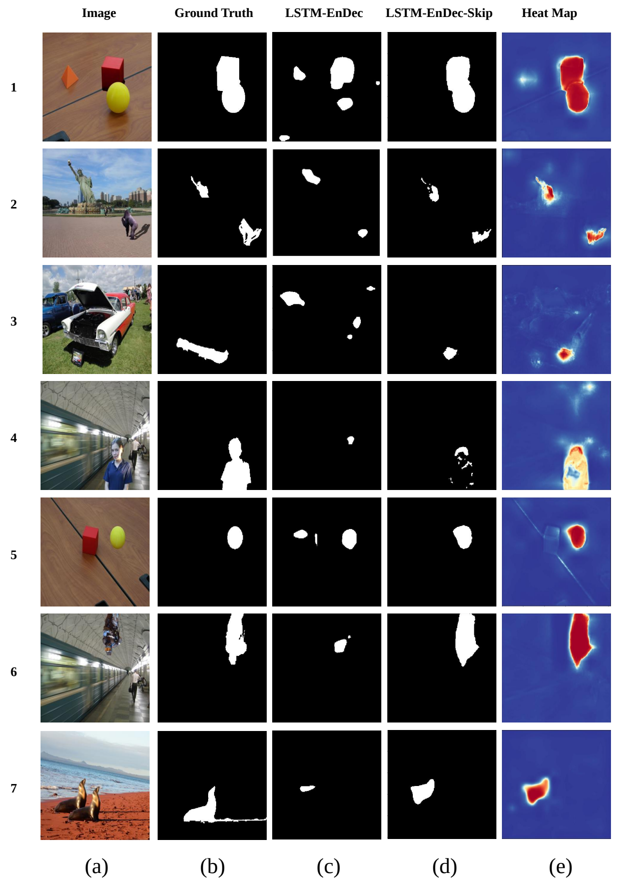


Figure 5. Qualitative results for pixel-wise image manipulation detection on NIST'16 [1] dataset. Columns(a) and (b) are input images and ground-truth masks for manipulated regions. (c) shows predicted binary mask for LSTM-EnDec [6]. Columns (d) and (e) demonstrate predicted binary mask and the probability heat map for our proposed model

regions as tampered. On the contrary, the proposed method shows reasonable success. We also observe that, LSTM-EnDec localizes both non-manipulated and manipulated regions in first and fifth images of Fig. 5. However, our approach shows high recognition performance in these cases. The seventh example of Fig. 5 shows one of our failure cases. In this case, the manipulated object has been copied from original image. Although our method was able to detect partial edges of the manipulated region, it failed to distinguish between the pristine and fake object.

Fig. 6 shows some qualitative results on CASIA 1.0 [20]. Similar to Fig. 5, we observe that our proposed method performs better than state-of-the-art method LSTM-EnDec [6]. LSTM-EnDec predicts several non-manipulated regions as manipulated (e.g., case 2,3,4,5). On the other hand, the proposed model localizes manipulated regions with high accuracy and only generate false positive predictions for the fifth image. The Fifth image is a very challenging image due to its complex texture and both LSTM-EnDec [6] and LSTM-EnDec-Skip (ours) failed to localize the manipulated region accurately.

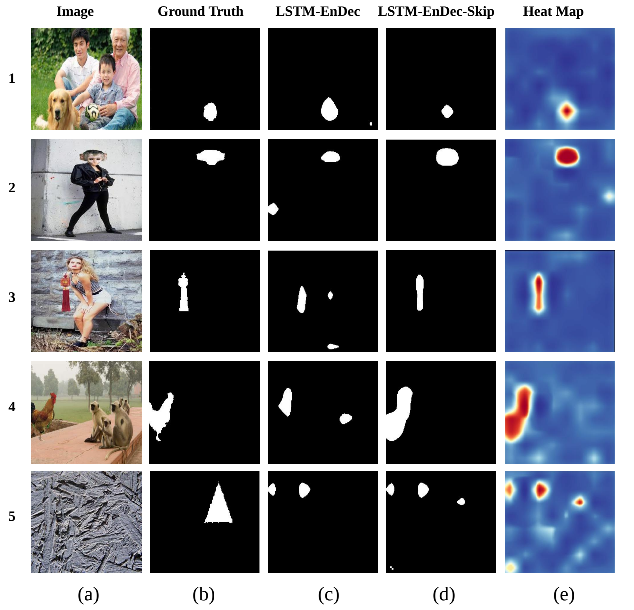


Figure 6. Qualitative results for pixel-wise image manipulation detection on CASIA 1.0 [20] dataset. Columns (a) and (b) are input images and ground-truth masks for manipulated regions. Columns (c) and (d) demonstrate predicted binary mask and the probability heat map for our proposed model

5. Conclusion

In this paper, we propose a new approach to further exploit layers rich in spatial information for localizing manipulation. Detection of blur edges in a manipulated image helps segment tampered region more accurately and prevents false pristine object segmentation. The fusion of low-level layers in our framework leads to detection of smooth edges which is one of the primary traces for tampering detection in an image. Besides, we use resampling features extracted by Laplacian filter along with Radon transform and fed to LSTM for learning the transition between manipulated and non-manipulated regions. Consequently, experiments on standard datasets show that our method not only detects tampering artifacts but also localize different tampered regions with noticeably improved performance compared to previous approaches.

6. Acknowledgement

The project described was supported by Grant Number ORIIR180037 from the Department of Health and Human Services.

References

[1] Nist nimble 2016 datasets. https://www.nist.gov/sites/default/files/documents/2016/11/30/should_i_believe_or_not.pdf. 1, 2, 3, 5, 6, 7, 8

[2] Y. Annadani and C. V. Jawahar. Augment and adapt: A simple approach to image tampering detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2983–2988, Aug 2018. 3

[3] M. Attarifar and M. Baniasadi. Jpeg image security by block size estimation and quality factor classification. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 118–121, Oct 2017. 3

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017. 4

[5] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 3, 5, 7

[6] 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 Transactions on Image Processing*, pages 1–1, 2019. 2, 3, 4, 5, 7, 8

[7] B. Bayar and M. C. Stamm. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2152–2156, March 2017. 3

[8] A. Bharati, M. Vatsa, R. Singh, K. W. Bowyer, and X. Tong. Demography-based facial retouching detection using subclass supervised sparse autoencoder. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 474–482, Oct 2017. 3

[9] T. Bianchi and A. Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, June 2012. 2, 5

[10] L. Bondi, S. Lameri, D. Gera, P. Bestagini, E. J. Delp, and S. Tubaro. Tampering detection and localization through clustering of camera-based cnn features. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1855–1864, July 2017. 3

[11] J. Brogan, P. Bestagini, A. Bharati, A. Pinto, D. Moreira, K. Bowyer, P. Flynn, A. Rocha, and W. Scheirer. Spotting the difference: Context retrieval and analysis for improved forgery detection and localization. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4078–4082, Sep. 2017. 3

[12] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1881–1889, July 2017. 3, 4

[13] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3547–3555, June 2015. 4

[14] I.-C. Chang, J. C. Yu, and C.-C. Chang. A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. *Image and Vision Computing*, 31(1):57 – 71, 2013. 3

[15] C. Chen, S. McCloskey, and J. Yu. Image splicing detection via camera response function analysis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1876–1885, July 2017. 3

[16] C. Chen, X. Zhao, and M. C. Stamm. Mislgan: An anti-forensic camera model falsification framework using a generative adversarial network. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 535–539, Oct 2018. 3

[17] Y. Chen, X. Kang, Y. Q. Shi, and Z. J. Wang. A multi-purpose image forensic method using densely connected convolutional neural networks. *Journal of Real-Time Image Processing*, Mar 2019. 3

[18] D. Cozzolino, G. Poggi, and L. Verdoliva. Efficient dense-field copymove forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, Nov 2015. 3, 5

[19] D. Cozzolino and L. Verdoliva. Camera-based image forgery localization using convolutional neural networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1372–1376, Sep. 2018. 3

[20] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database 2010. http://forensics.idealtest.org. 1, 2, 3, 5, 6, 7, 8

[21] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, Oct 2012. 2, 6

[22] A. Flenner, L. Peterson, J. Bunk, T. Mohammed, L. Nataraj, and B. Manjunath. Resampling forgery detection using deep learning and a-contrario analysis. *Electronic Imaging*, 2018(7):212–1–212–7, 2018. 3

[23] M. Goebel, A. Flenner, L. Nataraj, and B. S. Manjunath. Deep learning methods for event verification and image repurposing detection. *CoRR*, abs/1902.04038, 2019. 3

[24] J. Goh, L. L. Win, and V. L. L. Thing. Image region forgery detection: A deep learning approach. In *SG-CRC*, 2016. 3

[25] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. 2, 4

[26] M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[27] B. S. M. Lakshmanan Nataraj, Anindya Sarkar. Improving re-sampling detection by adding noise, 2010. 3

[28] G. Li, Q. Wu, D. Tu, and S. Sun. A sorted neighborhood approach for detecting duplicated regions in image forgeries

based on dwt and svd. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1750–1753, July 2007. 3

[29] H. Li, W. Luo, X. Qiu, and J. Huang. Image forgery localization via integrating tampering possibility maps. *IEEE Transactions on Information Forensics and Security*, 12(5):1240–1252, May 2017. 2

[30] W. Li, Y. Yuan, and N. Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Process.*, 89(9):1821–1829, Sept. 2009. 5

[31] Z. Lin, J. He, X. Tang, and C.-K. Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492 – 2501, 2009. 5

[32] B. Liu and C.-M. Pun. Deep fusion network for splicing forgery localization. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 237–251, Cham, 2019. Springer International Publishing. 2, 3

[33] W. Luo, J. Huang, and G. Qiu. Jpeg error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5:480–491, 2010. 5, 6

[34] B. Mahdian and S. Saic. Detection of copymove forgery using a method based on blur moment invariants. *Forensic Science International*, 171(2):180 – 189, 2007. 3

[35] B. Mahdian and S. Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497 – 1503, 2009. Special Section: Computer Vision Methods for Ambient Intelligence. 5, 6

[36] T. M. Mohammed, J. Bunk, L. Nataraj, J. H. Bappy, A. Flenner, B. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. A. Peterson. Boosting image forgery detection using resampling features and copy-move analysis. *Electronic Imaging*, 2018(7):118–1–118–7, 2018. 3

[37] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Transactions on Knowledge and Data Engineering*, 13(1):124–141, Jan 2001. 4

[38] G. Muhammad, M. Hussain, and G. Bebis. Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digital Investigation*, 9(1):49–57, 2012. 2

[39] L. Nataraj, T. M. Mohammed, B. S. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury. Detecting GAN generated fake images using co-occurrence matrices. *CoRR*, abs/1903.06836, 2019. 3

[40] L. Nataraj, A. Sarkar, and B. S. Manjunath. Adding gaussian noise to denoise jpeg for detecting image resizing. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 1493–1496, Nov 2009. 3

[41] N. Pham, J.-W. Lee, G.-R. Kwon, and C.-S. Park. Efficient image splicing detection algorithm based on markov features. *Multimedia Tools and Applications*, 10 2018. 3

[42] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho. Image splicing detection through illumination inconsistencies and deep learning. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3788–3792, Oct 2018. 3

[43] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, Feb 2005. 3

[44] W. Quan, D. Yan, K. Wang, X. Zhang, and D. Pellerin. Detecting colorized images via convolutional neural networks: Toward high accuracy and good generalization. *CoRR*, abs/1902.06222, 2019. 3

[45] R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch. Transferable deep-cnn features for detecting digital and print-scanned morphed face images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 3

[46] Y. Rao and J. Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Dec 2016. 3

[47] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(06):1137–1149, jun 2017. 3

[48] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 4, 5, 7

[49] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. *CoRR*, abs/1901.08971, 2019. 3

[50] R. Salloum and C. J. Kuo. Efficient image splicing localization via contrastive feature extraction. *CoRR*, abs/1901.07172, 2019. 3

[51] R. Salloum, Y. Ren, and C.-C. J. Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *J. Visual Communication and Image Representation*, 51:201–209, 2018. 2, 3

[52] Z. Shi, X. Shen, H. Kang, and Y. Lv. Image manipulation detection and localization based on the dual-domain convolutional neural networks. *IEEE Access*, 6:76437–76453, 2018. 3

[53] A. Tambo, M. Albright, and S. Mccloskey. Low-and semantic-level cues for forensic splice detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1664–1672, Jan 2019. 3

[54] J. Wang, X. Li, and J. Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[55] W. Wang, J. Dong, and T. Tan. Exploring dct coefficient quantization effects for local tampering detection. *IEEE Transactions on Information Forensics and Security*, 9(10):1653–1666, Oct 2014. 3

[56] L. Wei. Robust detection of region-duplication forgery in digital image. *Chinese Journal of Computers*, 2007. 3

[57] 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 *Pro-*

ceedings of the 25th ACM International Conference on Multimedia, MM '17, pages 1480–1502, New York, NY, USA, 2017. ACM. 3

[58] Y. Wu, W. Abd-Almageed, and P. Natarajan. Busternet: Detecting copy-move image forgery with source/target localization. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 3

[59] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265, May 2019. 3

[60] S. K. Yarlagadda, D. Gera, D. M. Montserrat, F. M. Zhu, E. J. Delp, P. Bestagini, and S. Tubaro. Shadow removal detection and localization for forensics analysis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2677–2681, May 2019. 3

[61] K. Ye, J. Dong, W. Wang, B. Peng, and T. Tan. Feature pyramid deep matching and localization network for image forensics. pages 1796–1802, 11 2018. 3

[62] Z. Zhang, Y. Zhang, Z. Zhou, and J. Luo. Boundary-based image forgery detection by fast shallow cnn. 01 2018. 3

[63] J. Zhou, J. Ni, and Y. Rao. Block-based convolutional neural network for image forgery detection. In C. Kraetzer, Y.-Q. Shi, J. Dittmann, and H. J. Kim, editors, *Digital Forensics and Watermarking*, pages 65–76, Cham, 2017. Springer International Publishing. 3

[64] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-stream neural networks for tampered face detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 3

[65] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 6

[66] Y. Zhu, T. Ng, B. Wen, X. Shen, and B. Li. Copy-move forgery detection in the presence of similar but genuine objects. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pages 25–29, Aug 2017. 3