



Image Splicing Localization via Semi-global Network and Fully Connected Conditional Random Fields

Xiaodong Cun and Chi-Man Pun^(✉)

University of Macau, Taipa, Macau
{mb55411, cmpun}@umac.mo

Abstract. We address the problem of *image splicing localization*: given an input image, localizing the spliced region which is cut from another image. We formulate this as a classification task but, critically, instead of classifying the spliced region by local patch, we leverage the features from whole image and local patch together to classify patch. We call this structure Semi-Global Network. Our approach exploits the observation that the spliced region should not only highly relate to local features (spliced edges), but also global features (semantic information, illumination, etc.) from the whole image. Furthermore, we first integrate Fully Connected Conditional Random Fields as post-processing technique in image splicing to improve the consistency between the input image and the output of the network. We show that our method outperforms other state-of-the-art methods in three popular datasets.

Keywords: Image splicing localization · Image forgery localization · Multimedia security

1 Introduction

The magic of computer makes digital photos edit possible. Softwares, such as PhotoShop, bring user-friendly interface for tampering image. With the growth of user-uploaded images on the Internet, it is more likely a serious security problem to detect whether an image has been tampered or not and localize the corresponding forgery region. Because artificial tampered images will send wrong message to others. For example, tampered images will make fake news more reliable and throw dust in the eyes of the public; it also convinces people on the impossible natural views and confuses the historian researchers.

In this paper, we focus on image splicing localization, a common forms of photographic manipulation. Image splicing means a particular region of donor image is cut and paste to the host image. Figure 2 is an example procedure to create a spliced image. The detection of image splicing has a long history in the digital image processing community. Many splicing algorithms [3–6] only detect the candidate image has been spliced or not. As for a more challenge task, few techniques [7, 8] attempt to localize the spliced area in the image.

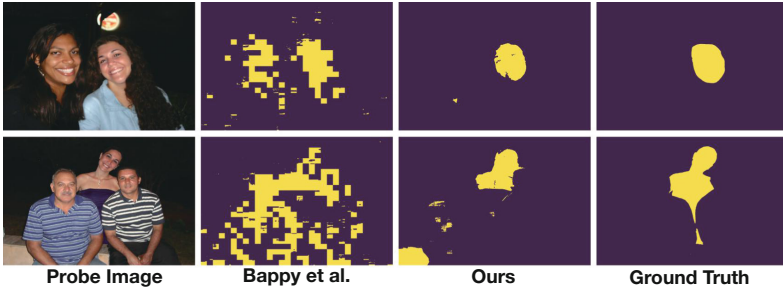


Fig. 1. These images are taken from dataset [1] where spliced regions own different illumination condition. Our network can classify spliced patches by mixing the global feature, while [2] fails because their method only learns from patches.

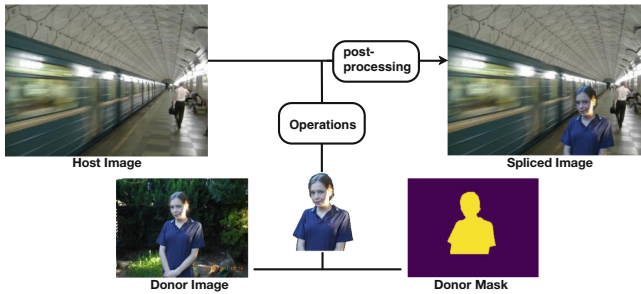


Fig. 2. This figure shows the spliced image is created by two authentic images. By masking the part of donor image, the selected region is pasted to the host image after some operations (translation and rescale the donor region). Sometimes, several post-processing techniques (such as Gaussian filter on the border of selected region) are used to the spliced region for the harmony of the selected region and host image.

The state-of-the-art approaches in image splicing localization analyze the features in frequency domain and/or the properties of statistic [9–11] because the donor image and host image maybe have different feature responses on the edges between splicing region and non-splicing region. Recently, Convolutional Neural Network shows a great success in many Computer Vision tasks, such as image classification [12], object detection [13], and there are also some papers [7, 14, 15] trying to solve image splicing by deep learning. However, current deep learning-based image splicing algorithms often solve image splicing localization from two viewpoints. One type of method often relies on the assumption that some specific features between the spliced region and non-spliced region are different. For example, [14] assume the donor image and host image are taken by different types of cameras, [2, 15] assume that the features in authentic edge and the spliced edge are different. Another type of methods rely on the power of deep learning and the distribution of large dataset. These methods learn splicing region from ground truth label directly, such as [7] propose a splicing localization method based on Fully Convolutional Network [16].

Different from previous methods which only consider certain assumptions or learn from the large dataset, we rethink image splicing from the beginning of the human intuition. Human often identifies the splicing region from the candidate image by the clues from many aspects. For example, as the spliced image in Fig. 2, the first observation aspect from human is local edge: the spliced region will have a sharper edge because these borders are manufactured by human/software which is not 100% perfect. Another observation viewpoint is the consistency of light: the sunshine in the face and clothes of the girl is weird when the background is an underground metro station. These evidence means people will not only search the details in the local edges to identify the spliced region but also try to classify the regions from the global level, such as illumination consistent and semantic consistent.

By above observation, we formulate our network as a multi-inputs classification network. To classify each candidate region, the network will preserve the local details features by the input of local patch and calculate the global features by the input of the whole image. From the features of global image and candidate region, the network classifies the candidate region is spliced or not. We call this structure Semi-Global Network. Furthermore, to design a high-performance network structure, we argue that both the relationships of neighbourhood pixels in local patches and the global image features are important. Thus, we use a structure which preserves the local relationship between pixels in [2] as our local feature branch of the network. Furthermore, we borrow the framework from image classification [12] as global feature network. The idea of combining the global and local structure is not only used in the training network, we also add a Fully Connected Conditional Random Fields (CRF) to constraint the output mask should own the similar shape with the original image. As shown in Fig. 1, our method show a significant better result than the method which only consider the local patch.

Our main contributions are as follows:

- By considering the prerequisite of image splicing task is the combination of global features and local features, we propose a Semi-Global network to solve this problem.
- Besides the combination of global features and local features in patch based classification, we firstly add the Fully Connected CRF as post-processing technique in image splicing task.
- We add a new smooth term in loss function for the task harmony in patch-based classification and patch-based segmentation.
- Our method can achieve state-of-the-art performance in several popular datasets.

2 Related Works

Traditional Image Splicing Method. Localizing spliced region in the image has been long studied as part of detecting and localizing manipulated region from images. Some researches [9–11, 17] assume that different images will own

different noise levels because of the combination of camera model or the post-processing techniques when manipulating. A significant direction of image splicing have assumed that different cameras will show different internal patterns. Such as, Color Filter Array (CFA) [18, 19], CFA transforms incoming light to different color channels and reconstructs the color image. Another important pattern is Camera Response Function [20]. Camera Response Function maps the incoming light to linear for making the image more visually appealing. These two internal image features are highly related to the whole image which means the images are taken by different cameras will show different internal patterns. Another important direction in image splicing is JPEG compression features [21, 22]. These techniques squeeze the feature by the observation that different images will have different JPEG compression levels or JPEG features. Such as, Li *et al.* [22] extract the block artifacts from the JPEG image for comparison with other block.

Deep Learning for Image Splicing. Recently, Deep Learning-based techniques have been utilized in many Computer Vision and Digital Image Processing tasks. A lot of interests in learning to localize the image splicing region from a single image has been driven by the ability of Convolution Neural Networks [2, 7, 8, 14, 15, 23]. Liu *et al.* [8] predict the mask of forgery region by a combination of a multi-scale neural network. With the similar idea, Salloum *et al.* [7] propose a multi-task fully convolutional network to localize image splicing region. They not only optimize the splicing region by ground truth mask directly but also constrain the edge in the output of predicted mask. These two methods only rely on the power of deep learning and the structure of network often design for image classification, which will ignore the low-level features. Inspired by traditional camera internal pattern-based method, Bondi *et al.* [14] use a pre-trained camera identification neural network to predict the original camera in input patches levels and analyse the results by the clustering algorithm. This method has strong assumption that splicing region and the original images are taken by the different camera. Following the traditional Camera Response Function based method, a novel feature designed by [15], is proposed for image splicing localization. Chen *et al.* [15] extract the Camera Response Function firstly and then try to classify the splicing regions in the feature domain by Neural Network. However, this method only can classify the patches in the edge of splicing region. Currently, Wu *et al.* [23] propose an algorithm for constrained image splicing problem which focuses on finding the spliced region by two images. Thus it is not design for single image splicing localization. Most recently, Bappy *et al.* [2] propose a hybrid deep learning based method by jointing the training of classification and segmentation for image forgery localization. However, by considering the splicing region often only connect to the local patches, this method is only trying to classify the local patch.

Unlike most deep learning based methods in image splicing which only consider the patches [2, 8, 14, 15] or global image based end-to-end training [7, 23], we argue that image splicing is a task not only relate to local feature, such as

the features of edge between splicing region and the host image [15], but also global features, such as light condition [24], camera models [14, 15], etc. Thus, in this paper, we consider from the viewpoint on the combination of global feature and pixel-level local patch classification in the task of image splicing.

3 Methods

We model image splicing localization as a conditional classification task with post-processing. As shown in Fig. 3, the whole overview of our framework can be divided into training network and post-processing.

In training network, giving a candidate image I and its non-overlapped patch sets I_p , the goal of our neural network SGN is to identify each patch is in the spliced region or not (classification of the patch) and which pixels in the patch $P_i (P_i \in I_p)$ belong to spliced region (segmentation of the patch). So our model can be written as:

$$L_i, M_i = SGN(I, P_i) \quad P_i \in I_p$$

where L_i is the label of current patch P_i and M_i is the segmentation results of the spliced region.

Combining the global image and local patch is not only used in network classification, it also performs in post-processing stage in our framework. In post-processing stage, we utilize the Fully Connected CRF to force the connection between colour and position. Notice that we only use the output of segmentation mask as the unary probability of Conditional Random Fields. The final splicing mask M_{pp} of input image I can be formulated as:

$$M_{pp} = CRF(I, M)$$

where $M = \sum_{i \in I_p} M_i$ is the output segmentation probability mask of our network.

3.1 Semi-global Network

As shown in Fig. 3, our Semi-Global Network can be divided into Global Feature Network and Patch Feature Network. These two parts learn different features from the patch and whole image, respectively. The two branches of the network are trained synchronously in end-to-end style by ground truth label and ground truth mask.

Patch Feature Network. We use the network described in [2] as our local feature extraction network. To achieve the goal of feature extraction from local patch, as shown in the *Patch Feature Network* of Fig. 3, in each forward of the neural network, one of the non-overlap patches from the original image is fed to the neural network for classification and segmentation.

In patch-based classification, a patch with 64×64 spatial resolution is fed into two convolutional layers for extracting a 2D low-level feature map firstly, then

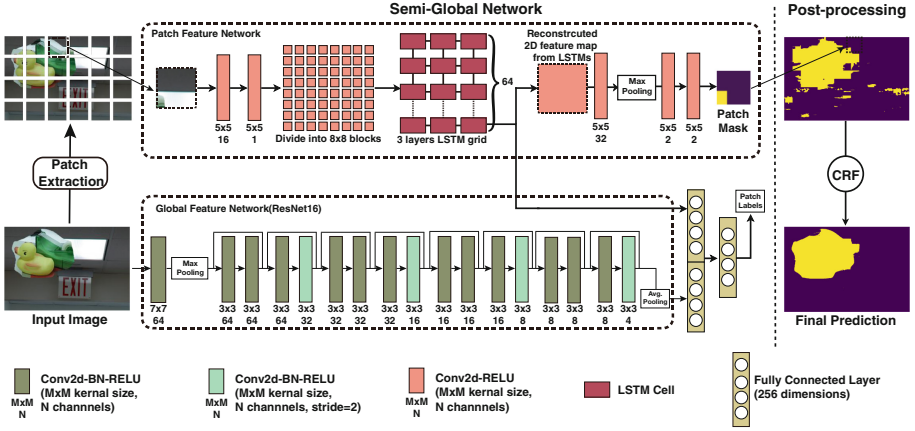


Fig. 3. The overview of our framework.

the feature map is uniformly divided into 8×8 blocks where each block owning 8×8 pixels. For modelling the relationships between the pixels of neighborhood, every block can be viewed as the input of Long-Short Term Memory [25] (LSTM) cell with 256 dimension features. LSTM models the relationship between pixels in the patches without decreasing the size of feature maps. Because low level feature is important for coarse edge detection. Next, the output of LSTM is not only used for image classification but also can be reconstructed into 2D feature map for final segmentation task. As shown in Fig. 3, the output of LSTM is reshaped to the original image according to the blocks we divided. Then two Convolutional layers model the reconstructed feature map for final segmentation results. A Softmax layer is added at the end of network for segmentation prediction and classification, respectively. This model can essentially extract pixel level features from patch while traditional coarse-to-fine network structure will break the relationship between pixels.

Compared with Bappy *et al.* [2], our method utilizes their network structure for local feature extraction in image splicing localization because their network model the local relationship between pixels. However, Bappy *et al.* [2] just rebuild the image from patch output. And we only use the output of patch segmentation as the input of post-processing method we provided. But [2] mixed the results of label and segmentation for final results. More results are discussed in experiments.

Global Feature Network. Whether the goal of our global feature network is to extract the global features (such as light, semantic information) from the input image, networks, we interpolate the pre-trained image classification network on large available dataset for global feature extraction. For global feature extraction, a ResNet18 [12] network structure, which is pre-trained on ImageNet [26], is added for global feature extraction. In our task, we remove the fully connected classification layer by replacing it with a new fully-connected layer in 256

dimensions. This new layer can learn the global features we need automatically from ResNet18 by the back-propagation of training data. We also freeze all the weights in Convolutional Layers and Batch Normalization layers in ResNet18, because comparing with ImageNet, our dataset is too small for the global features extraction. Thus, by leveraging the weights learning from ImageNet, our network has the ability to learn from small dataset. Notice that the global feature is only connected to the features of patch classification because the feature of patch segmentation is highly related to the position of pixels. So we can not add the global feature to segmentation branch as classification branch. However, feature concatenation in patch classification can also benefit the results of segmentation task because we train the network synchronously.

3.2 Loss Function

By considering the spliced region and host image are two categories, our network is a hybrid system of binary classification task $\Phi_{classification}$ and binary segmentation task $\Phi_{segmentation}$. We use Weighted Cross Entropy to model this two losses. So the loss function of classification is:

$$\Phi_{classification}(L, L_{gt}) = \frac{1}{N} \sum_{i \in I_p} W_n(1 - L_{gt})\log(1 - L_i) + W_s L_{gt}\log(L_i)$$

where N is the number of patches totally, L_i is the probability of the patch i in the spliced region, L_{gt} is the ground truth label of current patch, and W_s , W_n are the weight of spliced region and non-spliced region, respectively.

The segmentation loss is almost the same as classification loss except the input mask M_i and the ground truth mask M_{gt} are 2D probability maps on each pixels:

$$\Phi_{segmentation}(M, M_{gt}) = \frac{1}{N} \sum_{i \in I_p} \sum_{j \in M_i} W_n(1 - M_{gt})\log(1 - M) + W_s M_{gt}\log(M_i)$$

Because the splicing dataset is totally unbalanced, we set the weight between spliced region W_s and weight of non-spliced region W_n according to the statistics percentage on the ground truth mask of the training set. The weighted strategy makes our model more sensitive to the spliced region.

Furthermore, for making classification results and segmentation results unity, we add an extra smooth loss Φ_{smooth} for classification results and segmentation results. This smooth loss is added by the observation that patches label probability and the mean of patch segmentation will be minimum when the network convergence. If we think the classification results as the output of mask, or if we think the patches results as the output of label, these two parts will show the same probabilities. So we force the mean of mask probability equals to the patch label, our smooth criterion can be written as:

$$\Phi_{smooth}(M, L) = \left| \frac{\sum_{i \in I_p} M_i}{numel(M_i)} - L \right|$$

where $numel$ is a function to get the size of patch masks M_i . So the final loss function Φ can be written as the sum of classification criterion, segmentations criterion and smooth criterion:

$$\Phi = \Phi_{classification} + \beta\Phi_{segmentation} + \lambda\Phi_{smooth}$$

We also add two hyper-parameter β and λ for better results. In the experiment we found that classification is a relative easier task than segmentation, so we set $\beta = 10$. As for the smooth hyper-parameter λ , we set this parameter to $\lambda = 0.01$ by thinking the classification as the main task.

3.3 Conditional Random Fields as Post Processing

Because the output of our network still fails in some patches of the image, and the patch segmentation task is more complex than patch classification task. We exploit the Fully Connected Conditional Random Fields in [27] for further exploit the global information to our network and get better results. Although CRF has been utilized in Semantic Segmentation widely [28, 29], it has never been used in image splicing task.

The fully connected CRF can be written as an energy function:

$$E(\mathbf{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j)$$

where \mathbf{x} is label assignment for pixels. The unary potential $\theta_i(x_i) = -\log(M_i)$ where i is each pixels in the probability mask M . The probability mask M is created by the output of patch segmentation. Then, a fully-connected graph is used for efficient influence the pairwise potential. So the pairwise potential in [27] can be expressed as:

$$\theta_{i,j} = \mu(x_i, x_j) \left[\omega_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + \omega_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right) \right]$$

where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$ and zero otherwise. Then, two Gaussian Kernels are applied in different feature spaces. The first is related to positions and RGB colors, and the second only measure the connection between pixels. These two kernels are used for feature constraint. While the first kernel restraint the pixels which have similar color and position as the same label, the second kernel penalizes the smoothness in position. As illustrated in Fig. 4, the results of our network benefit from fully connected CRF.



Fig. 4. The effect of post-processing

4 Experiments

4.1 Preparation

Implementation Details. All experimental benchmarks are obtained by PyTorch [30] framework. ADAM [31] solver with $\beta_1 = 0.9$, $\beta_2 = 0.999$ is used as optimization function for all the experiments. We train the network in 120 epochs and choose the best accuracy model on the validation set as the final model. The initial learning rate is 0.001, we decay the learning rate in 60, 90 epochs to 0.0001 and 0.00001, respectively. The network is trained on two NVIDIA 1080 GPUs.

Datasets Setup. We compare our method with other states-of-the-art methods on NC2016 dataset [32], Carvalho dataset [1] and Columbia dataset [33]. There are 280 spliced samples in NC2016, 100 spliced samples in Carvalho dataset and 180 spliced samples in Columbia dataset. For each dataset, we randomly split the whole image dataset into three categories with training (65%), validation (10%) and testing (25%) as Bappy *et al.* [2]. Then, we extract the patch-global image pairs in training set. By considering the balance of space and time for network training, we resize the original image to 224×224 for the input of global feature network. In patches extraction, we split original image to the non-overlapped 64×64 image blocks. Thus, we have more than 10k training patches on each dataset which is enough for training classification network and segmentation network. Similarly, we obtain validation and test set. As for the ground truth label of patch, following [2], we label the patches which contain more than 87.5% ($7/8$) of the spliced pixels as the positive spliced patches.

Evaluation Metrics. We compare our method with other state-of-the-art methods on F_1 score and Matthews Correlation Coefficient (MCC) for binary classification tasks as [7]. We also exploit the ROC curve and AUC score on three datasets as [2] in Table 1 and Fig. 6.

Baselines. As for deep learning based method, we compare our method with two most relevant methods: Bappy *et al.* [2] use the local patch to classify the manufacture region; MFCN [7] learn to predict the spliced mask and spliced edge from Fully Convolutional Network [16] directly.

Because there are few image splicing localization methods using deep learning, we also compare our results with some state-of-the-art traditional methods. We select four representative methods from different viewpoints: CFA2 [19] utilize Color Filet Array for forgery detection. NOI1 [11] assume that the splicing region will have different local image noise variance. BLK [22] classify the spliced region by detecting the periodic artifacts in JPEG compression. DCT [17] detect inconsistent of JPEG Discrete Cosine Transform coefficients histogram. These four methods are tested and evaluated on the same test datasets as deep learning based methods. We run traditional methods by a public available image splicing toolkits [34].

4.2 Comparisons

Experiments on Columbia Dataset. Columbia dataset is a relatively easier dataset for classification. There are 180 images which certain objects are spliced to host image in different localization and the edge of the spliced region is easy to recognize. In this dataset, the content of spliced region and the background are often totally different. By analyzing the ground truth mask in training dataset, we weight the spliced region and non-spliced region to 1:5 in loss function. As shown in Fig. 6 and Table 1, our method get better results than others. As the similar spliced objects/shapes are shown in both training set and test set, patch-based method can also be detected spliced regions without the help of global features. (First column of Fig. 5 on Bappy *et al.* method). However, comparing with the images which spliced region rarely shown in training set (Second and third columns of Fig. 5), our method gain better results. As for other traditional methods, our network also gain better results, because these methods only detect/analysis the spliced region by certain assumptions.

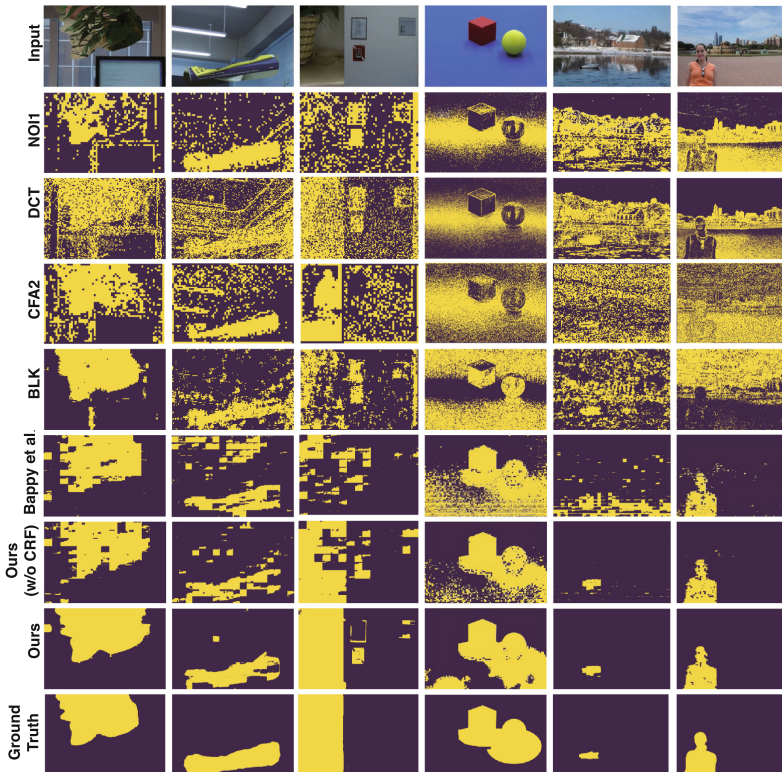


Fig. 5. Results on Columbia [33] dataset (left three) and NC2016 dataset (right three) [32]. (NOI1, CFA2, BLK, DCT are displayed by thresholding the mean probability of whole feature image.)

Experiments on NC2016 Dataset. In NC2016 dataset, some tampered images show very similar “appearance” from human viewpoint but tamper with different operations or post-processing techniques such as, the border of the temper region is utilized Gaussian smooth or not will be considered as two samples in the dataset. These attack methods may huge influence the traditional methods which detect/localize the splicing region from hand-craft features. However, in deep learning-based method, it is a relatively easier task when similar images are shown in both training set and test set. Because neural network need to inference from global high level features and low level features. We list the results on NC2016 dataset in Table 1 and Figs. 5 and 6, our method is significantly better than other states-of-the-art methods on several evaluation metrics.

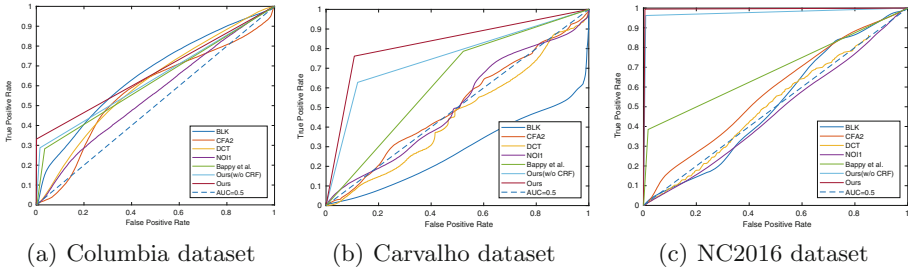


Fig. 6. ROC curves on three different datasets.

Experiments on Carvalho Dataset. We set the weight of the spliced region and non-spliced region to 1:7 on Carvalho Dataset. From Table 1 and Fig. 6, our method is significantly better than other methods in several numeric metrics. Carvalho [1] manufacture input images by splicing the face/body from another image with the inconsistent of illumination color. It is hard to recognize when the network only classifies the local patches [2]. As shown in Figs. 1 and 7, our method can detect spliced region because of the integrate of the global feature while [2] only classify the skin from the image because of their method just classify the local patches from the whole image.

4.3 Evaluation

Evaluation of Patch Classification and Segmentation. We list the accuracy of patch-based classification and patch-based segmentation results on all three datasets for model evaluation. As shown in Table 2, our method gain significantly better results than baseline method [2] because our method integrate of global feature and task harmony loss.

The Effect of Global Feature. Our method needs to connect the feature from the local patch and global image together for final prediction. Do more global

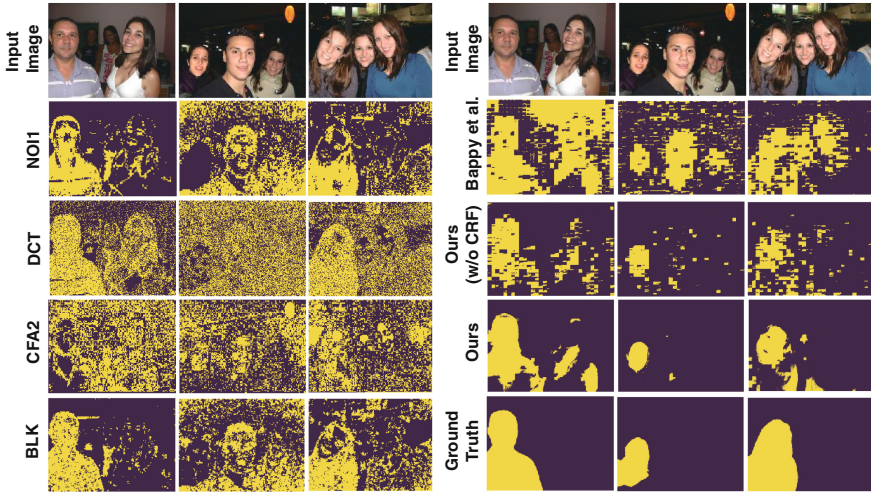


Fig. 7. Results on Carvalho [1] dataset. (NOI1, CFA2, BLK, DCT are displayed by thresholding the mean probability of whole image.)

Table 1. Comparison on three datasets. F_1 , MCC score are calculated for each image firstly and then calculate the average value while AUC is calculated amount all the pixels.

Metrics	Methods	NC2016 [32]	Carvalho [1]	Columbia [33]
F_1	MFCN* [7]	0.5707	0.4795	0.6117
	Bappy et al. [2]	0.6242	0.3102	0.5270
	Ours (w/o CRF)	0.7174	0.4236	0.5956
	Ours	0.7900	0.5006	0.6482
MCC	MFCN* [7]	0.5703	0.4074	0.4792
	Bappy et al. [2]	0.6257	0.1882	0.5074
	Ours (w/o CRF)	0.7101	0.3309	0.5557
	Ours	0.7847	0.4379	0.6403
AUC	CFA2 [19]	0.57	0.51	0.54
	NOI1 [11]	0.47	0.55	0.51
	BLK [22]	0.51	0.29	0.64
	DCT [17]	0.51	0.37	0.62
	Bappy et al. [2]	0.68	0.65	0.62
	Ours (w/o CRF)	0.98	0.75	0.64
	Ours	0.99	0.83	0.67

*MFCN is trained on larger dataset and test on these datasets while our method takes parts of datasets as the training set. The size of train data will hugely influence the results. So the experiment results of MFCN is taken by original paper and just for reference.

Table 2. Comparison of classification/segmentation accuracy.

	NC2016 dataset [32]	Carvalho dataset [1]	Columbia dataset [33]
Bappy et al. [2]	95.89%/89.53%	68.57%/53.80%	85.02%/77.95%
Ours	97.81%/89.60%	83.69%/75.10%	89.72%/83.90%

features get better results? To verify this question, we train the network with different percentage between global features and patch features to 0:1 (baseline network [2]), 0.25:1, 0.5:1, 1:1, 2:1. Then we observe the results in the final splicing task. As shown in Table 3, the MCC and F_1 score show the best results when the global features equal to the features from local. And the results get worse slightly when the global feature grows. This conforms to our intuitive sense of the world: although the hybrid of global feature and the local feature can gain better results in image splicing task, it is better to consider the local patch and global patch by suitable percentage.

The Influence of Task Harmony in Loss Function and Post-processing.

In loss function, we add a new smooth term to force the relationship between the loss of classification loss and segmentation loss. As shown in Table 3, the smooth term benefits for our task. We also list the output of our network w/o CRF. Mask segmentation is obviously better than Label classification results because label classification only classifies the uniform patches.

Table 3. Evaluation on Columbia dataset

	F_1 Score	MCC
Ours w/o CRF (Label)	0.5467	0.5305
Ours w/o CRF (Mask)	0.5956	0.5557
Ours w/o smooth term	0.6416	0.5863
Ours (0:1)	0.5270	0.5074
Ours (0.25:1)	0.6287	0.6224
Ours (0.5:1)	0.6281	0.5255
Ours (2:1)	0.6258	0.6204
Ours (1:1)	0.6482	0.6403

5 Conclusion

In this paper, we propose Semi-Global network with fully connected CRFs as post-processing for image splicing localization. Our Semi-Global network interpolates global features to patch classification/segmentation network. In addition, we use CRF-based post processing techniques to refine the output of the network. Extensive experiments on three benchmarks demonstrate that our method

significantly improves the baseline and outperform other state-of-the-art algorithms. We also evaluate our method by removing the necessary parts in the experiments.

We hope that our proposed splicing localization pipeline might potentially help other applications which need to constraint the relationship between local and global when the low-level information (the relationship between pixels) is as important as global features. Such as video splicing detection and scene labeling. We believe our framework is a promise direction for further researches.

Acknowledgements. This work was supported in part by the Research Committee of the University of Macau under Grant MYRG2018-00035-FST, and the Science and Technology Development Fund of Macau SAR under Grant 041/2017/A1.

References

1. de Carvalho, T.J., Riess, C., Angelopoulou, E., Pedrini, H., de Rezende Rocha, A.: Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics Secur.* **8**, 1182–1194 (2013)
2. Bappy, J.H., Roy-Chowdhury, A.K., Bunk, J., Nataraj, L., Manjunath, B.: Exploiting spatial structure for localizing manipulated image regions. In: *International Conference on Computer Vision (ICCV)* (2017)
3. Hsu, Y.-F., Chang, S.-F.: Image splicing detection using camera response function consistency and automatic segmentation. In: *ICME*, pp. 28–31 (2007)
4. Chen, W., Shi, Y.Q., Su, W.: Image splicing detection using 2-D phase congruency and statistical moments of characteristic function. In: *Security, Steganography, and Watermarking of Multimedia Contents*, vol. 6505, p. 65050R (2007)
5. Hsu, Y.-F., Chang, S.-F.: Detecting image splicing using geometry invariants and camera characteristics consistency. In: *ICME*, pp. 549–552 (2006)
6. He, Z., Lu, W., Sun, W., Huang, J.: Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recogn.* **45**(12), 4292–4299 (2012)
7. Salloum, R., Ren, Y., Kuo, C.C.J.: Image splicing localization using a multi-task fully convolutional network (MFCN). *arXiv preprint [arXiv:1709.02016](https://arxiv.org/abs/1709.02016)* (2017)
8. Liu, Y., Guan, Q., Zhao, X., Cao, Y.: Image Forgery Localization Based on Multi-Scale Convolutional Neural Networks. *CoRR cs.CV* (2017)
9. Pun, C.M., Liu, B., Yuan, X.C.: Multi-scale noise estimation for image splicing forgery detection. *J. Vis. Commun. Image Represent.* **38**, 195–206 (2016)
10. Lyu, S., Pan, X., Zhang, X.: Exposing region splicing forgeries with blind local noise estimation. *Int. J. Comput. Vis.* **110**, 202–221 (2014)
11. Mahdian, B., Saic, S.: Using noise inconsistencies for blind image forensics. *Image Vis. Comput.* **27**, 1497–1503 (2009)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR* (2016)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: *ICCV* (2017)
14. Bondi, L., Lameri, S., Guera, D., Bestagini, P., Delp, E.J., Tubaro, S.: Tampering detection and localization through clustering of camera-based CNN features. In: *CVPR Workshops*, pp. 1–10, November 2017
15. Chen, C., McCloskey, S., Yu, J.: Image splicing detection via camera response function analysis. In: *CVPR*, pp. 1876–1885 (2017)

16. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. CoRR cs.CV (2014)
17. Ye, S., Sun, Q., Chang, E.C.: Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In: ICME (2007)
18. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. *IEEE Trans. Sig. Process.* **53**(10), 3948–3959 (2005)
19. Dirik, A.E., Memon, N.D.: Image tamper detection based on demosaicing artifacts. In: ICIP (2009)
20. Hsu, Y.F., Chang, S.F.: Camera response functions for image forensics - an automatic algorithm for splicing detection. *IEEE Trans. Inf. Forensics Secur.* **5**, 816–825 (2010)
21. Farid, H.: Exposing digital forgeries from JPEG ghosts. *IEEE Trans. Inf. Forensics Secur.* **4**, 154–160 (2009)
22. Li, W., Yuan, Y., Yu, N.: Passive detection of doctored JPEG image via block artifact grid extraction. *Sig. Process.* **89**, 1821–1829 (2009)
23. Wu, Y., AbdAlmageed, W., Natarajan, P.: Deep Matching and Validation Network - An End-to-End Solution to Constrained Image Splicing Localization and Detection. arXiv.org (2017)
24. Johnson, M.K., Farid, H.: Exposing digital forgeries in complex lighting environments. *IEEE Trans. Inf. Forensics Secur.* **2**, 450–461 (2007)
25. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
26. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: ImageNet - a large-scale hierarchical image database. In: CVPR (2009)
27. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: NIPS (2011)
28. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab - Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. CoRR (2016)
29. Zheng, S., et al.: Conditional random fields as recurrent neural networks. In: ICCV (2015)
30. Paszke, A., et al.: Automatic differentiation in pytorch (2017)
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
32. NIST: Nimble Media Forensics Challenge Datasets (2016). <https://www.nist.gov/itl/iad/mig/media-forensics-challenge>
33. Ng, T.T.: Columbia image splicing detection evaluation dataset (2004)
34. Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y.: A large-scale evaluation of splicing localization algorithms for web images. *Multimedia Tools Appl.* **76**, 4801–4834 (2017)