Feature Pyramid Deep Matching and Localization Network for Image Forensics

Kui Ye, Jing Dong, Wei Wang, Bo Peng, and Tieniu Tan
Center for Research on Intelligent Perception and Computing,
Institute of Automation, Chinese Academy of Sciences
P.O. Box 2728, Beijing, P.R. China, 100190
E-mail: kui.ye@cripac.ia.ac.cn
{jdong, wwang, bo.peng, tnt}@nlpr.ia.ac.cn

***Abstract*—To advance the state of the art of image forensics technologies, a new formulation of splicing localization is proposed, which aims to obtain the masks for both the query and donor images for a pair of query(probe) image and potential donor image if a region of the donor image was spliced into the probe. The former Deep Matching and Validation Network(DMVN) addresses the problem with a novel end-to-end learning based solution. Inheriting the deep dense matching layer, we propose Feature Pyramid Deep Matching and Localization Network(FPLN), whose contributions are three folds. Firstly, instead of using just one feature map as in DMVN, FPLN utilizes a pyramid of feature maps with different resolutions w.r.t. the input image to achieve better localization performance, especially for small objects. Secondly, we add a fusion layer that fuses together all the features after deep dense matching layer, which not only takes full advantage of the correlation information between those features, but is also able to integrate two pathways in DMVN into just one simple pathway, simplifying the subsequent architecture. Lastly, we employ focal loss to address the imbalance problem, as the foreground area is usually much smaller than the background area. The experiments demonstrate the superior performance of our proposed method in detection accuracy and in localizing small tempered regions.**

I. Introduction

With the rapid growth of social networks and the wide use of digital cameras, image and video content have been ubiquitous. At the same time, the easy access to advanced image processing softwares like Photoshop and Gimp has made manipulating and editing digital images become increasingly handy. Therefore seeing is no longer believing. Meanwhile, detecting and localizing image forgeries, at a large scale, is becoming increasingly more difficult for new processionals, forensic experts, and legal prosecutors, which necessitates developing novel and scalable image forensics technologies[1].

Splicing detection ordinarily denotes copying one or more regions of an image and pasting them onto a target image. Image splicing and copy-move are often considered as two close problems, which have been studied a lot in the literature. In the pixel level, forged images can be detected by analysing the artefacts left by cloning[2], re-sizing[3], or non-linear filtering[4]. Inconsistencies in chromatic aberrations[5], color filter array interpolation[6] or sensor noise[7] can also be utilized as evidence of image forgery. On the scene level, inconsistencies in lighting[8], shadows[9] or reflections[10] are used as clues to reveal manipulations. It's worth noting that all methods mentioned above are only applicable to a single image. Furthermore, they share a strong assumption that one or more of these artefacts must be present in a spliced image, which is not always valid.

In the recent Nimble 2018 Challenge from National Institute of Standards and Technology[1], there is a new formulation of splicing detection and localization, that is: for a query(probe) image and a potential donor image, its goal is to estimate the probability that a region of the donor image was spliced into the probe, and if so obtain the masks for both the probe and donor images, i.e. segmenting the spliced regions(s) in both the donor and the query image, which is the main focus of this work. Following [1], we also refer to it as the constrained image splicing detection(CISD) problem. As shown in Fig. 1, Q denotes the query image, P the potential donor image, Q_m the ground truth mask for the query image, and P_m the ground truth mask for the potential donor image.

Such a splicing formulation can be rendered as a copy-move detection by combining both the query and donor images as one. However, traditional copy-move algorithms use hand-crafted features that are vulnerable to image transformations such as noise, compression and geometric transformation. Besides, they are usually performed stage by stage instead of jointly. To address these problems, Wu et al.[1] propose Deep Matching and Validation Network(DMVN) — an end-to-end optimized neural network that is able to extract robust features and perform splicing detection and localization, unlike recent deep learning based forgery detection algorithms that only use a deep learning module to extract features[11,12,13,14].

We denote by $output\_stride$ the ratio of input image spatial resolution to final output resolution[15]. The feature response used for deep dense matching in DMVN[1] is 16 times smaller than the input image dimension and thus $output\_stride = 16$. Whereas we facilitate a pyramid of feature responses, one having a $output\_stride = 16$ and another having a $output\_stride = 8$. To some extent, the larger the $output\_stride$, the coarser the feature response, so the latter feature response with smaller $output\_stride$ contains finer details of the input image that is helpful for detecting and localizing small objects, as evidenced by later experiments. There are two pathways in DMVN[1] to separately handle two sets of features after deep dense matching layer, in which case the correlation information between the two sets of features are ignored and the architecture is more complex, whereas

[1] https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2018

Fig. 1. Constrained image splicing detection problem, where true spliced pixels are labeled as white. P and Q are donor and probe images respectively; P_m and Q_m are their corresponding masks. The 1st row represents a positive sample, while the 2nd row a negative sample.

we propose to add a fusion layer that fuses together two sets of feature responses of different sizes, thus makes full use of their correlation information, and simplifies the design of subsequent structure. Further more, the problem of predicting masks can be seen as many binary classification problems. Since the spliced object is usually quite small compared with the background, the number of negative samples are much larger than the positive ones if we deem foreground pixel as positive and background pixel as negative. Therefore, besides Binary Cross Entropy Loss, we also explore the effect of focal loss[16] to address the imbalance problem as well.

In summary, our contributions are three folds: 1)we facilitate a pyramid of feature maps to capture fine-grained features so as to better localize small tempered objects. 2)we design a new fusion layer to utilize the correlation information between multi-level feature maps, and redesign the mask decoder that greatly simplifies the architecture. 3)we employ focal loss[16] to address the imbalance problem.

The remainder of this paper is organized as follows. Section 2 describes the proposed FPLN and the training procedure. Section 3 presents experimental results and comparisons against state-of-the-art methods. In section 4, conclusions are drawn.

II. Feature Pyramid Deep Matching and Localization Network

Given a query image Q and a potential donor image P, the goal of the method is essentially to segment the similar regions in Q and P, if existed. Shown in Fig. 2 is the architecture of FPLN whose steps consist of feature pyramid extraction, deep dense matching, feature fusion, feature refinement, and mask decoder. we'll detail each of the steps as following.

A. Feature Pyramid Extraction

In the original DMVN model[1], only one feature map, the output of the fourth convolutional block of VGG16, is used as the block-wise learned representations of the input images. Yet, its $output\_stride = 16$, which might hinder the model's ability to localize small tempered regions as claimed in [1]. In order to tackle the issue, our proposed FPLN, however, extracts a pyramid of feature responses from multi-level convolutional blocks of VGG16 (other CNNs like ResNet, DenseNet should work as well) that contain much more fine-grained information, as shown in Fig. 2.

B. Deep Dense Matching

Following the footsteps of [1], we also adopt the deep dense matching module which is made up of two steps: deep feature correlation and correspondence match pooling. For the sake of completeness, we will describe it here again. The purpose of Deep Dense Matching is to find possible matching regions between representations.

Deep feature correlation. Matching response using cross-correlation are exhaustively computed over all possible translations as in (1)

$$corr(P_f, Q_f)[x, y, i, j] = trans(P_f, x, y)[:, i, j] \cdot Q_f[:, i, j] \tag{1}$$

where P_f and Q_f are extracted feature map, $\cdot$ is the dot product operator, and $trans(Z_f, x, y)$ circularly translates Z_f w.r.t. (x, y) pixels, as defined in Eq. (2)

$$trans(Z_f, x, y)[:, i, j] = Z_f[:, mod(i + x, W), mod(j + y, H)] \tag{2}$$

where W and H are the width and height of the feature map being calculated.

Correspondence match pooling. In this step, meaningful response maps is extracted using three types of pooling: average pooling, max pooling and argsort pooling as defined in Eq.(3), (4), (5) respectively:

$$avgPool(corr(P_f, Q_f))[i, j] = \sum_{x=0}^{W} \sum_{y=0}^{H} corr(P_f, Q_f)[x, y, i, j]/WH \tag{3}$$

$$maxPool(corr(P_f, Q_f))[i, j] = \max_{x,y} \{corr(P_f, Q_f)[x, y, i, j]\} \tag{4}$$

$$argsortPool(corr(P_f, Q_f))[k] = corr(P_f, Q_f)[k_x, k_y, i, j] \tag{5}$$

where (k_x, k_y) in Eq. (5) is determined by the k-th maximum response over all translations. The final dense matching response R_{PQ} between P_f and Q_f is obtained by concatenating one average, one max and the top few(dependent on what level of feature are being used) argsort response along the feature dimension. R_{QQ}, R_{QP} and R_{PP} can, by the same formulation, be obtained as well.

C. Feature Fusion and Refinement block

Different from [1] that uses two pathways to separately learn two sets of features after deep dense matching, ignoring the fact that the two sets of features correlate with each other whose information when used together can probably boost performance. Therefore, we fuse all the features after deep dense matching before later processing. In this way, not only is the correlation information utilized, but the structure thereafter can also be greatly simplified to be just one pathway instead of two as in [1]. It's worth noting that when dealing with multi-level features from different scales with different feature map

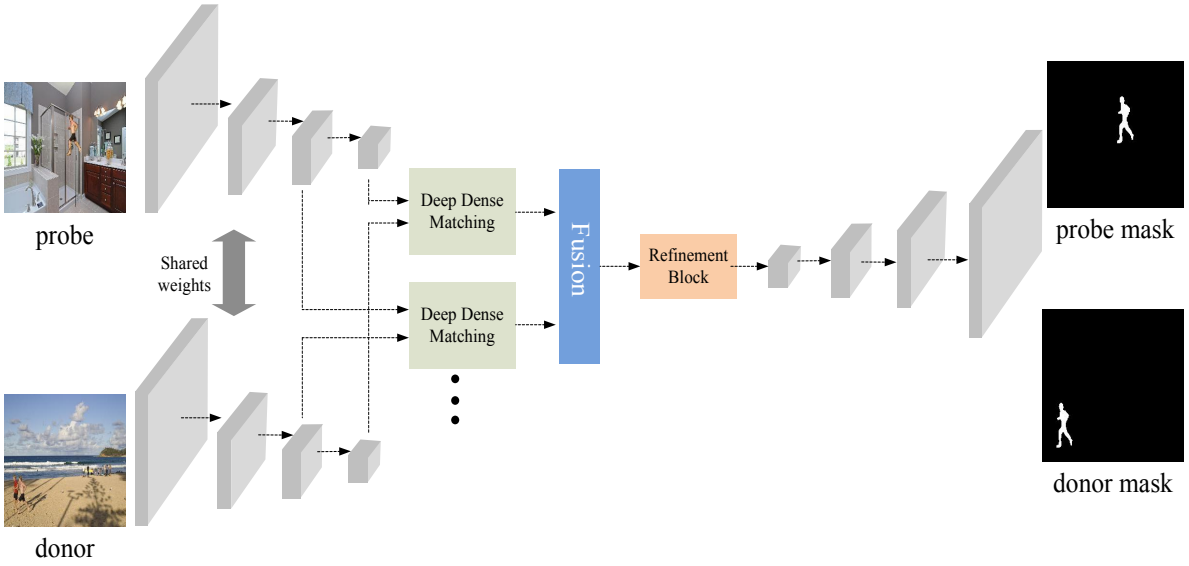

Fig. 2. The architecture of FPLN. It consists of feature pyramid extraction, deep dense matching, feature fusion, feature refinement, and mask decoder.

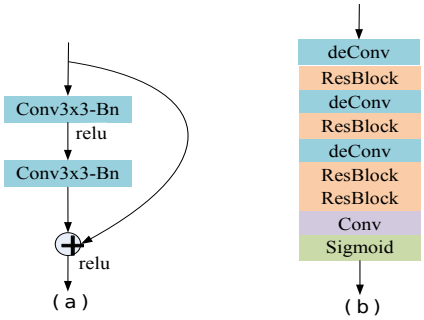

Fig. 3. (a)A ResNet block architecture. (b) The architecture of mask decoder.

sizes, the subsequent dense matching responses vary in size, so in order to align all the features to have the same size before fusion(concatenation), we upsample(by transpose convolution) the features with smaller sizes without losing information by compression as down sampling the features with larger sizes would.

Right after feature fusion, A refinement block is used to further refine the features after fusion. It is composed of several ResNet blocks[17] which is well known for its suitability to build very deep neural networks. Fig. 3(a) illustrates the components of a ResNet block, where $Conv3 \times 3 - BN$ means convolution with kernel size of 3, followed by batch normalization.

D. Mask Decoder

Fig. 3(b) shows the structure of mask decoder responsible for producing two masks for the probe and donor images. In the figure, deConv means a transpose convolution(also known as de-convolution), ResBlock a ResNet block, Conv a convolution operation, and Sigmoid the sigmoid operation. The key difference between our mask decoder and that in [1] lies in the fact that our mask decoder directly decodes two masks whereas there are two mask decoders in[1], each responsible for decoding one mask. Another difference is that rather than using upsampling layers as in [1] to calculate feature maps, we employ learnable transpose convolution with stride of 2, ensuring that every pixel contributes and the adjacent pixels transform in a synergistic manner[15].

E. Training Data and Strategy

Since no existing dataset is available for directly training the proposed FPLN, we use the SUN397 dataset[18] and the MS COCO dataset[19] to generate training samples according to the unsupervised generation process described in [20]. Briefly, a random image X with polygon-based object annotations and an random object in X are selected, then randomly transform this object and paste it to another randomly selected image Y to generate a resulting composite image Z. Different from [1], we only harvest no more than two(one positive and one negative) training samples for each unsupervised data generation, because we mainly care about foreground instances rather than the background appearing in both the probe and donor images. For instance, Fig. 1 gives a set of two training samples of this type.

More specifically, when generating data samples, we randomly pick an image and an object, and apply a random scale in $U(0.5, 2)$, a random rotation in $U(-15, 15)$, a random luminance change in $U(0.8, 1.2)$, and a random shift and translation as long as it doesn't transcend the image. In this way, we are able to generate as many samples as needed to train the end-to-end network. We finally generate 100k(thousand), 10k, and 10k synthesized samples for training, validation, and testing respectively. Our dataset is smaller than that in [1] due to hardware and speed concerns. It's also worth noting that we exclude those samples whose mask regions are spread sparsely and that we also apply erosion and dilation operation to wipe

out very tiny regions, because we find it might undermine training the network. In addition, we also observe that the performance for probe and donor images are quite different, and that the model is essentially trying to locate similar regions in probe and donor images, so we randomly exchange the position of probe image and donor image as data augmentation to address this problem.

The proposed FPLN is implemented using PyTorch deep learning framework. Our model is trained with Adam optimizer w.r.t. log loss(binary cross entropy loss or focal loss). we set beta1 to be 0.5, beta2 0.99, and initial learning 1e-3. The learning rate strategy is to decay itself by a factor of 0.2 if the validation loss stops decreasing for several(three in our experiments) epochs, and the minimum learning rate is 1e-5. It's worth noting that when training FPLN, we first set the learning rate for the pre-trained vgg network to be zero, and gradually increase its learning rate to be 1e-6, with the intention to smooth the transition when vgg is suddenly being finetuned with a relatively large learning rate.

III. Experiments

A. Baseline Methods and Test Settings

Since DMVN model[1] outperforms the classic block matching-based approach[21], the classic Zernike moments-based block matching[22] with nearest-neighbor search, the SURF feature-base keypoint matching[23] and the dense field matching[24], it's a quite state-of-the-art approach by the CISD formulation which is completely new and thus we consider it as the baseline method, as will denoted by dmvn-loc thereafter. Our FPLN model is run on Nividia TitanX GPU.

B. Dataset

We conduct evaluation experiments on two large dataset:1)the generated test set, 2)the NIST-provided Nimble 2018 image splicing detection dataset, as will be denoted by NIST set.

The generated test set. As described in II-E, it contains 10k image pairs, with 5k positive pairs and 5k negative pairs. Keep in mind that for generating each positive pair, random scale, random shift, random translation, and random luminance are applied.

The NIST set. It is provided by NIST and designed for the CISD task. It is very large containing more than half a million samples. The manipulations applied are sophisticated including image impainting and seam-carving etc. And the ratio of negative samples to positive samples is extremely huge, mimicking the real application scenario.

It is worth emphasizing that 1) for better comparison, we, following [1], also directly test the FPLN model trained by our synthetic data without any finetuning, and that 2)ground truth splicing masks are available for the generated test set but not always accurate for the NIST dataset, so we didn't perform pixel level evaluation on it.

C. Evaluation Metrics

Image level evaluation. We consider the metrics of precision, recall, f-score, ROC curve and the area under the ROC curve(AUC). We denote by TP $true\ positive$ i.e. correctly detected as spliced, FN $false\ negative$ i.e. incorrectly detected as not-spliced, FP $false\ positive$ i.e. incorrectly detected as spliced and TN $true\ negative$ i.e. correctly detected as not-spliced. Precision, recall and f-score are defined as follows:

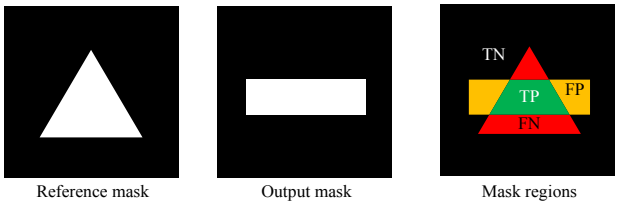

Fig. 4. Definitions of TP, TN, FP and FN for pixel level evaluation.

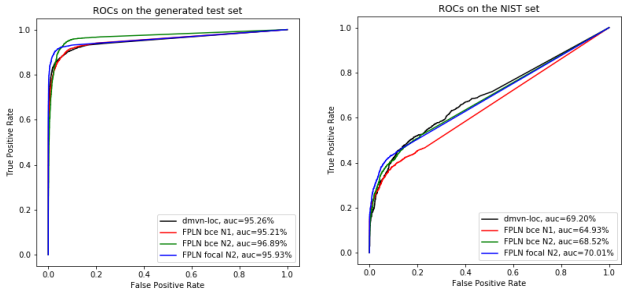

Fig. 5. ROCs of different methods on the generated test set and NIST set.

$$precision = TP\ /\ (TP + FP) \tag{6}$$

$$recall = TP\ /\ (TP + FN) \tag{7}$$

$$f - score = 2TP\ /\ (2TP + FN + FP) \tag{8}$$

ROC curve is determined as the function of $true\ positive\ rate$(TPR) in terms of $false\ positive\ rate$(FPR), where TPR and FPR are defined as Eqs.(9) and (10). AUC quantifies the overall ability of the system to discriminate between two class, which is also the only official metric used by NIST.

$$TPR = TP\ /\ (TP + FN) \tag{9}$$

$$FPR = FP\ /\ (TN + FP) \tag{10}$$

Pixel level evaluation. Metrics of Intersection over Union(IoU), NMM, and The Matthews Correlation Coefficient (MCC) are considered. IoU, NMM and MCC are all used to measure the accuracy of a system output mask. The NMM is invariant to translation, rotation, resizing, and cropping (under certain conditions). If MCC = 1, there is perfect correlation between the target and system output masks. If MCC = 0, there is no correlation between the target and system output masks. If MCC = -1, there is perfect anticorrelation between the target and system output masks. IoU, NMM and MCC are defined as following:

$$IoU = TP\ /\ (TP + FP + FN) \tag{11}$$

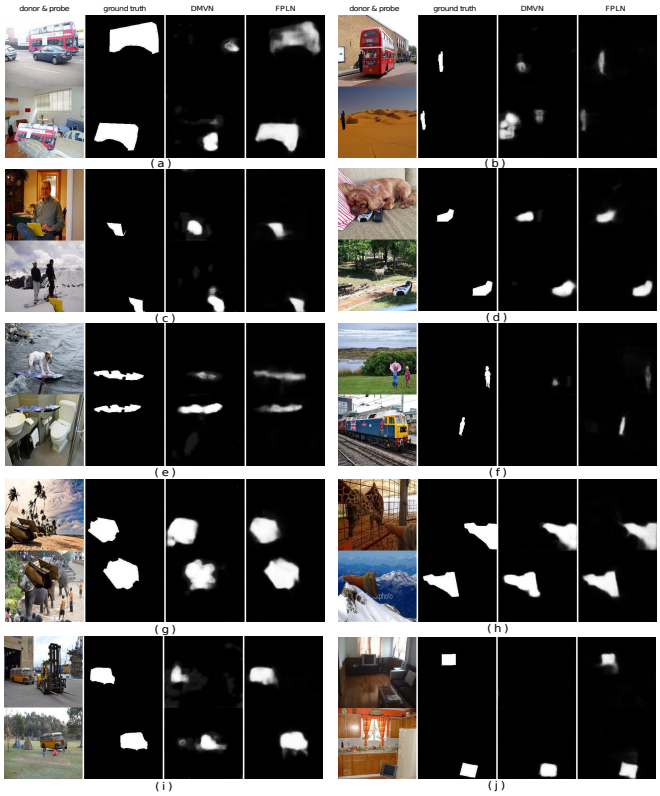

Fig. 6. Qualitative evaluation on positive samples in the generated test set.

Fig. 7. Qualitative evaluation on negative samples in the generated test set.

TABLE I. EXPERIMENTS ON THE GENERATED TEST SET.

Method	Precision	Recall	F-score	IoU	NMM	MCC
dmvn-loc	0.790	0.940	0.859	0.516	0.035	0.635
FPLN bce N1	0.858	0.922	0.889	0.505	0.016	0.638
FPLN bce N2	0.817	**0.968**	0.886	**0.592**	**0.193**	**0.719**
FPLN focal N2	**0.894**	0.932	**0.913**	0.535	0.076	0.664

$$NMM = max\{\frac{TP - FN - FP}{TP + FN}, -1\} \tag{12}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}$$

where TP, TN, FP, FN are defined as Fig. 4 shows.

D. Results

Image level and pixel level evaluation on the generated test set. Following [1], we deem a sample (a pair of images) as positive if any pixel in a predicted splicing mask is positive, as shown in Table.I. We consider three settings for FPLN model. FPLN bce N1(denoted by FP-B1) is trained with binary cross entropy(BCE) loss using one level of feature map. FPLN bce N2(denoted by FP-B2) is trained with BCE loss using two levels of feature maps(feature pyramid). FPLN focal N2(denoted by FP-F2) is trained with focal loss using two levels of feature maps.

TABLE II. EXPERIMENTS ON NIST DATASET.

Method	Precision	Recall	F-score
dmvn loc	0.414	**0.715**	0.525
FPLN bce N1	**0.6436**	0.419	0.507
FPLN bce N2	0.5252	0.541	0.533
FPLN focal N2	0.5301	0.564	**0.546**

For the image level metrics, all three settings of FPLN outperform dmvn-loc[1] by a moderate margin in terms of f-score(3% or higher). Specifically, the baseline dmvn-loc has relatively low precision but a high recall, yet FP-B1 and FP-F2 outperform dmvn-loc in terms of F-score, the reason of which, as can be seen in Table.I, comes from large precision gains. Whereas FP-B2 surpasses dmvn-loc by precision, recall and F-score.

The left diagram in Fig. 5 compares ROC and AUC scores for different methods on the generated test set, where the threshold used to obtain TPR and FPR is based on the positive pixel percentage in a resulting mask. More specifically, FP-B1 has comparable performance than the baseline dmvn-loc, which is outperformed by FP-B2 and FP-F2 regardless of the loss used, demonstrating the effectiveness of feature pyramid mechanism.

For the pixel level quantitative evaluation, the baseline dmvn-loc achieves 0.516, 0.035 and 0.635 for IoU, NMM and MCC respectively. FP-b1 achieves comparable scores on these metrics, while both FP-B2 and FP-F2 outperform the baseline method, indicating their stronger ability to localize the tempered regions. It's worth noting that FP-F2 behaves a little worse than FP-B2, but as will be shown later, FP-F2 generalizes better on the NIST set.

Image level evaluation on NIST dataset. Similar to the performance on the generated test set, the baseline dmvn-loc achieves a F-score of 0.525, with a relatively low precision

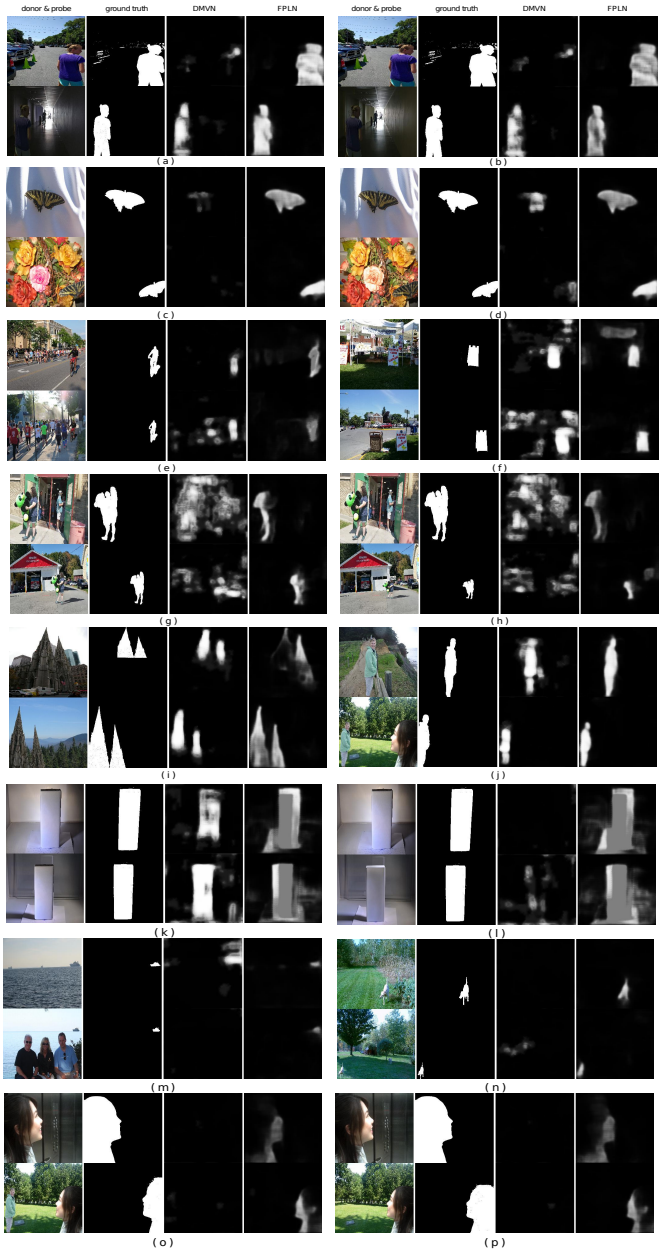

Fig. 8. Qualitative evaluation on positive samples in the NIST set.

of 0.414 and a relatively high recall of 0.715, as shown in Table.II. All three setting of FPLN surpass the baseline dmvn-loc in terms of precision by a large margin, but are inferior to the baseline dmvn-loc by recall. However, as for the more comprehensive metric of F-score, both FP-B2 and FP-F2 outperform the baseline method. It's worth noting that FP-F2 outperforms FP-B2 in terms of all three metrics, indicating a better generalization ability of FP-F2 than FP-B2.

The right diagram in Fig. 5 compares ROC and AUC scores for different methods on the NIST set. The baseline dmvn-loc performs better than FP-B1, comparable with FP-B2, and inferior to FP-F2. It's worth noting that there's a large discrepancy between the training data size, where there are 1.5 million training samples for the baseline model, which is 15 times larger than our training dataset containing only 100 thousand training samples. Thus there is still room for improvement for FP-F2 if trained with more samples, given that it already outperforms the baseline method based on a much smaller training set. The qualitative evaluations in the following are compared between dmvn-loc(later denoted as DMVN) and FP-F2(later denoted as FPLN).

Qualitative evaluation on the generated testset. Fig. 6 shows the qualitative results on positive samples, where the 1st column is the (donor, probe) pair, the 2nd column the ground truth masks, the 3rd column the predicted masks of DMVN, and the 4th column the predicted masks of FPLN. (c), (d), (g), and (h) show comparable localization performance between the baseline DMVN and FPLN. But FPLN covers more ground truth regions than DMVN in terms of (a), (e), (i), indicating better performance. In addition, for (b) and (f), FPLN achieving better performance in localizes the small regions, which we contribute to the pyramid features utilized. We see that for (j), DMVN predicts no tempered mask for the donor image at all, and it might due to the bias between probe and donor images, because DMVN is always trained with (probe, donor) pair, whereas our FPLN is trained with (probe, donor) with probability of 0.5, and with (donor, probe) pair of probability of 0.5 as well, which helps mitigate the bias.

Fig. 7 shows the qualitative results on negative samples. Except (a), (c), (f), (h) and (j) where DMVN incorrectly predicts masks for negative samples, the DMVN and FPLN perform comparably well on other negative samples.

Qualitative evaluation on the NIST set. Fig. 8 shows the qualitative results on positive samples. For (a) and (b), FPLN predicts the masks quite well whereas DMVN miss the correct mask for the donor image. If you watch closely, the probe image in (b) chromatically differs from that in (a), which suggests that FPLN is robust to chromatic alteration. Similarly goes (c) and (d) where FPLN is robust to the color change of the central flower while the predicted masks output by DMVN changes dramatically which also fails to localize the butterfly. For (e) and (f), the predicted masks by FPLN shows much less extraneous artefacts than DMVN's, indicating a better performance of FPLN. For (g) and (h), FPLN scales quite well as the ground truth mask for the probe image scales, though the predicted masks are not very accurate. For (i) and (j), it's obvious that the output masks by FPLN preserves better shape information than that by DMVN. For (k) and (l), DMVN is sensitive to the slight change of the probe image, while FPLN is much more robust to it. For (m) and (n), FPLN successfully localize the small tempered regions, while DMVN fails. We contribute it to the feature pyramid architecture of FPLN. For (o) and (p), FPLN is able to handle symmetry, while DMVN can not.

IV. Conclusions

In this paper, we propose the FPLN, a end-to-end network, for the image splicing localization and detection problems. We show that the FPLN that facilitates a pyramid of feature responses is superior in detecting and localizing small spliced regions than DMVN[1]. We fuse together all the feature responses after deep dense matching layer to take full advantage of their correlation information. Based on that, we design

our network that integrates two pathways of producing two separate masks into one pathway, which greatly simplifies the structure. In addition, we also propose to employ focal loss to tackle the imbalance problem between the foreground area and background area that achieves desirable effects. The experiments also demonstrate the robustness against DMVN.

V. Acknowledgements

This work was partly supported by NSFC (No. 61772529, 61502496, U1636201 and U1736119) and the National Key Research and Development Program of China (No. 2016YFB1001003). It was also partly supported by the Key Lab of Information Network Security and the Ministry of Public Security of China.

References

[1] Yue Wu, Wael AbdAlmageed and Prem Natarajan, "Deep Matching and Validation Network ", arXiv:1705.09765.

[2] Fridrich, A.J., Soukal, B.D., Luk, A.J., "Detection of copy-move forgery in digital images," in Proceedings of Digital Forensic Research Workshop (2003).

[3] Popescu, A.C., Farid, H., "Statistical tools for digital forensics," in Fridrich, J. (ed.)IH 2004. LNCS, vol. 3200, pp. 128–147. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30114-1_10

[4] Lin, Z., Wang, R., Tang, X., Shum, H.-V., "Detecting doctored images using camera response normality and consistency," in Proceedings of Computer Vision and Pattern Recognition (2005).

[5] Johnson, M.K., Farid, H., "Exposing digital forgeries through chromatic aberration," in Proceedings of ACM Multimedia and Security, Workshop, pp. 48–55 (2006).

[6] Popescu, A.C., Farid, H., "Exposing digital forgeries in color filter array interpolated images," IEEE Trans. Sig. Process. 53(10), 3948–3959 (2005).

[7] Luk, J., Fridrich, J., Goljan, M., "Detecting digital image forgeries using sensor pattern noise," in Proceedings of SPIE Electronic Imaging Security Steganography Watermarking of Multimedia Contents VIII, vol. 6072, pp. 0Y1–0Y11 (2006).

[8] Kee, E., Farid, H., "Exposing digital forgeries from 3-D lighting environments," in 2010 IEEE International Workshop on Information Forensics and Security, pp. 1–6. IEEE (2010).

[9] Kee, E., O'Brien, J.F., Farid, H., "Exposing photo manipulation with inconsistent shadows," ACM Trans. Graph. (ToG) 32(3), 28 (2013).

[10] O'Brien, J., Farid, H., "Exposing photo manipulation with inconsistent reflections," ACM Trans. Graph. 31(1), 1–11 (2012).

[11] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva, "Recasting Residualbased Local Descriptors as Convolutional Neural Networks: an Application toImage Forgery Detection" arXiv preprint arXiv:1703.04615 (2017).

[12] Davide Cozzolino and Luisa Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in Information Forensics and Security (WIFS), 2016 IEEE International Workshop on. IEEE, 1–6.

[13] Yuan Rao and Jiangqun Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in Information Forensics and Security (WIFS), 2016 IEEE International Workshop on. IEEE, 1–6.

[14] Ying Zhang, Lei Lei Win, Jonathan Goh, and Vrizlynn LL Thing, "Image Region Forgery Detection: A Deep Learning Approach," in Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016: Cyber-Security by Design, Vol. 14. IOS Press, 1.

[15] Chen L C, Papandreou G, Schroff F, et al., "Rethinking atrous convolution for semantic image segmentation[J]," arXiv preprint arXiv:1706.05587, 2017.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," arXiv preprint arXiv:1708.02002, 2017.

[17] He K, Zhang X, Ren S, et al., "Deep residual learning for image recognition[C]" Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[18] Xiao J, Hays J, Ehinger K A, et al., "Sun database: Large-scale scene recognition from abbey to zoo[C]" Computer vision and pattern recognition (CVPR), 2010 IEEE conference on. IEEE, 2010: 3485-3492.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick, "Microsoft coco: Common objects in context" in European Conference on Computer Vision, 2014. Springer, 740–755.

[20] Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros, "Learning a discriminative model for the perception of realism in composite images," in Proceedings of the IEEE International Conference on Computer Vision, 2015. 3943–3951.

[21] Weiqi Luo, Jiwu Huang, and Guoping Qiu, "Robust detection of regionduplication forgery in digital image." In Pattern Recognition, ICPR 2006. 18th International Conference on, Vol. 4. IEEE, 746–749.

[22] Seung-Jin Ryu, Min-Jeong Lee, and Heung-Kyu Lee. "Detection of copy-rotate-move forgery using zernike moments." In International Workshop on Information Hiding, 2010. Springer, 51–65.

[23] Vincent Christlein, Christian Riess, Johannes Jordan, Corinna Riess, and Elli Angelopoulou. "An evaluation of popular copy-move forgery detection approaches." IEEE Transactions on information forensics and security 7, 6 (2012), 1841–1854.

[24] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva," Efficient dense-field copy-move forgery detection." IEEE Transactions on Information Forensics and Security 10, 11 (2015), 2284–2297.

[25] Wei Wang, Jing Dong, and Tieniu Tan, "Effective image splicing detection based on image chroma." In Image Processing (ICIP), 2009 16th IEEE International Conference on. IEEE, 1257–1260.