Deep Matching and Validation Network

An End-to-End Solution to Constrained Image Splicing Localization and Detection

Yue Wu*
Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292
yue_wu@isi.edu

Wael Abd-Almageed
Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292
wamageed@isi.edu

Prem Natarajan
Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292
pnataraj@isi.edu

ABSTRACT

Image splicing is a very common image manipulation technique that is sometimes used for malicious purposes. A splicing detection and localization algorithm usually takes an input image and produces a binary decision indicating whether the input image has been manipulated, and also a segmentation mask that corresponds to the spliced region. Most existing splicing detection and localization pipelines suffer from two main shortcomings: 1) they use handcrafted features that are not robust against subsequent processing (e.g., compression), and 2) each stage of the pipeline is usually optimized independently. In this paper we extend the formulation of the underlying splicing problem to consider two input images, a query image and a potential donor image. Here the task is to estimate the probability that the donor image has been used to splice the query image, and obtain the splicing masks for both the query and donor images. We introduce a novel deep convolutional neural network architecture, called Deep Matching and Validation Network (DMVN), which simultaneously localizes and detects image splicing. The proposed approach does not depend on handcrafted features and uses raw input images to create deep learned representations. Furthermore, the DMVN is end-to-end optimized to produce the probability estimates and the segmentation masks. Our extensive experiments demonstrate that this approach outperforms state-of-the-art splicing detection methods by a large margin in terms of both AUC score and speed.

KEYWORDS

image forensics; splicing detection and localization; deep learning

1 MOTIVATION

The ubiquity of digital cameras and the rapid growth of social networks have caused a proliferation of image and video content. Image forgery is becoming a rampant problem, as a direct consequence of digital content proliferation. Literally, the common idiom *seeing is believing* does not hold true anymore, especially since in recent years sophisticated image editing tools, such as Adobe PhotoShop™ and GIMP have been pushing the limits of image composition in order to produce more natural and aesthetic images. These tools make it much easier to alter an image maliciously for a non-professional user. Meanwhile, detecting and localizing image forgeries, at a large scale, is becoming increasingly more difficult for new processionals [41], forensic experts, and legal prosecutors. These new challenges necessitate developing novel and scalable image forensics technologies.

Although *image manipulation* is sometimes used to indicate any kind of technique that can be used to modify an image, it often means major manipulations from an image forensics perspective [4, 31], such as *splicing*, *copy-move*, *erasing*, or *retouching*. Fig. 1 illustrates these common manipulations, where *splicing* denotes copying one or more source image regions and pasting them onto a destination image, while the other three types can be done using a single source image. Thus, *splicing* is considered to be more complicated since it involves external images.

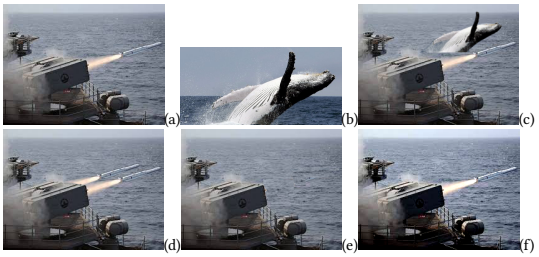


Figure 1: Image manipulation types. (a) and (b) original *missile* and *whale* images; (c) spliced image by compositing *whale* into *missile*; and (d)-(f) resulting images on *missile* after *copy-move*, *erasing*, and *retouching*, respectively.

Traditionally, copy-move and image splicing forgery detection are often thought of as two close problems [1, 4] that can be solved within a general forgery detection framework (GFDF) [10–12], that is: 1) **representation**, in which the characteristics of the underlying image (pixel by pixel or holistically) are extracted as feature vectors; 2) **matching**, in which corresponding regions are determined from the feature representation; and 3) **post-processing**, in which nearest-neighbor detection is linked and filtered to reduce false alarms and improve detection rates. It is worth noting that many copy-move and image splicing detection algorithms rely on strong or specific image hypothesis, e.g., photo-response non-uniformity noise [8, 28], camera characteristics [21], color filter array [35], JPEG compression [2, 6, 9, 26], edge sharpness [18, 30] and local features [22, 23, 45]. Comprehensive reviews of these

*Corresponding author

MM'17, October 23–27, 2017, Mountain View, CA, USA.
 ISBN 978-1-4503-4906-2/17/10…$15.00
DOI: https://doi.org/10.1145/3123266.3123411

approaches can be found in [4, 7, 19, 39]. The main assumption of the approaches, in order to achieve high detection rates, is that one or more of these clues must be present in a spliced image. However, this assumption is not always valid since splicing manipulations are usually followed by transformations (e.g., compression, resampling or geometric transformations) that may hide traces of the manipulation [4, 5].

In the recent Nimble 2017 Challenge from National Institute of Standards and Technology,[1] the image splicing problem has been reformulated as: given a query image Q and a potential donor image P, the goal is to solve not only the detection problem, i.e., whether or not Q contains spliced regions from P, but also the localization problem, i.e., segmenting the spliced region(s) in both the donor and the spliced images. Since this new problem formulation constrains image splicing detection to a pair of images, we refer to it as the constrained image splicing detection (CISD) problem. Fig. 2 shows three input samples along with their ground truth splicing masks and detection labels of CISD. This CISD problem can be viewed as a new formulation of the classic copy-move detection problem in the sense that it also looks for a potential region that is copy-move from image P to image Q. Finally, this new CISD problem also plays an important role in producing an image phylogeny graph [15–17] for a query image given a big dataset, especially in explaining how two images in neighboring nodes are associated.

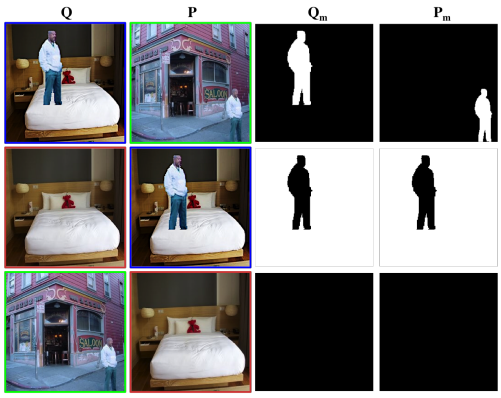


Figure 2: Constrained image splicing detection problem, where true spliced pixels are labeled as white. From top to bottom, sample detection labels are 1, 1, and 0, respectively.

It is worth noting that the two-input nature of the new CISD problem makes many existing image hypothesis used in classic copy-move and splicing detection no longer applicable. For example, [2] proposed an image splicing detection algorithm by differentiating single and double JPEG compression, but it is not useful for the CISD problem because 1) we can neither guarantee that the two inputs will be in JPEG format, nor that the inputs are compressed with the same quantization table at the same level of quality; and 2) even though both are of JPEG format and one region in P is detected as doubly compressed, the CISD question "whether this region is originally from Q" is still unanswered. As a result, the new CISD definition urges the use of more robust assumptions and features.

[1] https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation

Fortunately, visual clues, the visual correspondences between splicing regions [3, 22, 24, 29, 37, 43, 45], are still useful in the new CISD problem because they are probably the weakest assumptions that we can make for that problem. This implies that we need two things: 1) representations for visual clues, and 2) rules to determine which two representations match. As one can see, these are exactly the first two steps ("representation" and "matching") in the GFDF, while the last step ("post-processing") in the GFDF is really to take advantage of the consistency within a set of true matchings to reduce false alarms. Though it seems that classic copy-move and splicing detection algorithms [22, 23, 37, 43, 45] relying on visual features can be easily modified for the new CISD problem, we note that the two major drawbacks of these existing algorithms are: 1) handcrafted features are less robust against image transformations (e.g., compression, noise addition and geometric transformations) and are surely not optimal for the CISD problem; and 2) tuning each of three stages in GFDF on its own only optimizes performance disjointly instead of jointly.

In this paper we conceptually follow the GFDF and propose Deep Matching and Verification Network (DMVN)—a novel deep learning-based splicing detection and localization method that is 1) unlike previous GFDF approaches, end-to-end optimized, 2) does not depend on extracting handcrafted, unrobust feature representations, 3) uses fully learnable parameters to determine matching or not, and 4) mimics the human validation process to see whether the found visual evidence is enough to determine a detection. Furthermore, the proposed method is also distinct from recent deep learning based forgery detection practices [13, 14, 32, 42] in the sense that our approach 1) is a full end-to-end deep learning solution instead of a deep learning module only for feature extraction [13, 14], 2) performs both localization and detection tasks instead of one or the other [14, 32, 42], and 3) invents unique deep learning modules (*Deep Dense Matching* and *Visual Consistency Validator*) for performing visual matching and validation (see Fig.3-(a)). The remainder of this paper is organized as follows. Section 2 describes the proposed DMVN and discusses the training procedure and settings. Experimental results and comparisons against state-of-the-art methods are presented in Section 3. In Section 4, we conclude the paper and provide directions for future research.

2 DEEP MATCHING VALIDATION NETWORK

2.1 Architecture Overview

As previously mentioned, the CISD problem is formulated as follows: given a query image Q and a potential donor image P, we need to determine whether the query image is indeed spliced and then segment the spliced region in the query image and its corresponding region in the donor image.

As shown in Fig. 3, the overall network is designed such that block-wise learned representations of the input images are extracted using a convolutional network network – *CNN Feature Extractor* (e.g., AlexNet, ResNet or VGG) and fed into the proposed *Deep Dense Matching* module, which performs (as the name implies) dense matching between the two input images. In order to segment the splicing masks in the two images, we use an inception-based *Mask Deconvolution* module [36]. Further, the predicted masks are fed into a *Visual Consistency Validator* module that forces the model

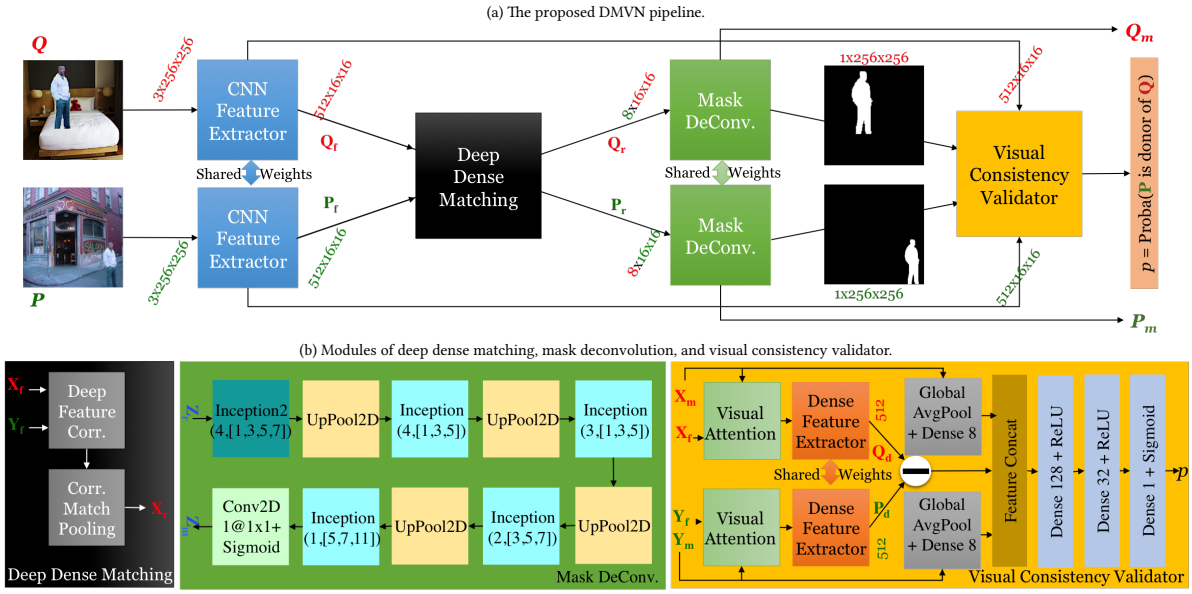


Figure 3: Deep matching and validation network for the constrained image splicing detection and localization.

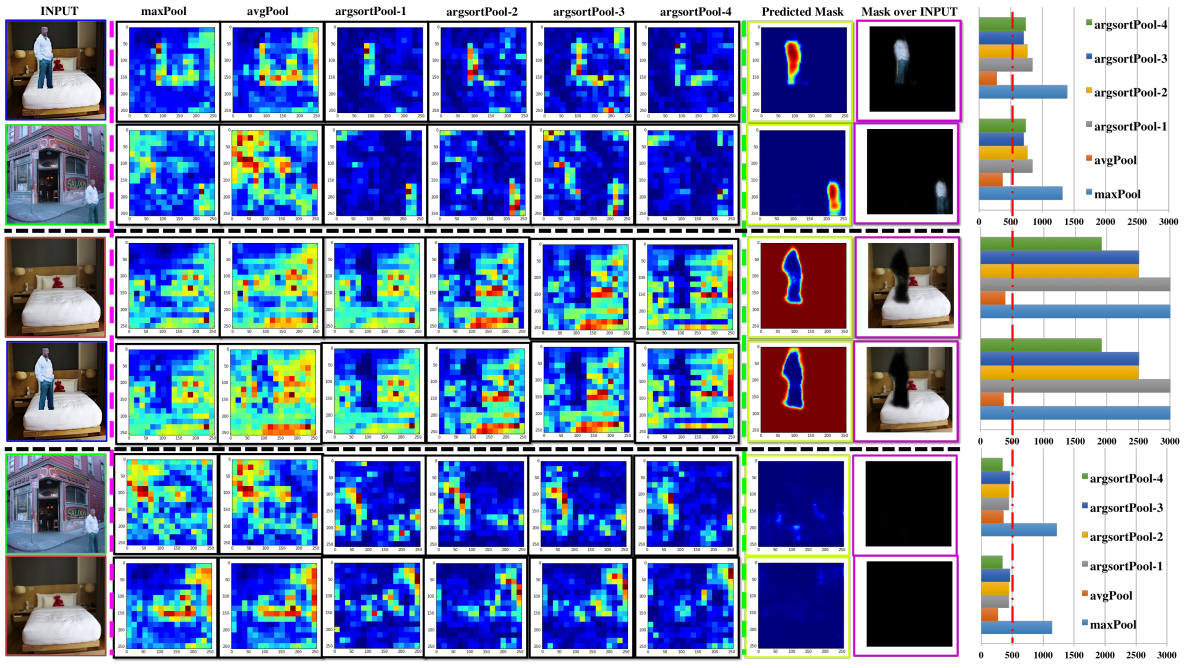


Figure 4: Visualization of selected DMVN layers and the statistics of the 5th maximum responses from correlation matching pooling layers. Note: 1) all layers are shown by linearly rescaling to [0,1], and visualized *w.r.t.* the Jet color map, where more red means higher response; 2) data ranges of pooled matching response maps could be very different, but predicted masks share the same range [0,1]; and 3) rough maximum values of matching responses can be seen in the statistics on the far-right.

to focus on the segmented areas in both images. Finally, a Siamese-like module is used to extract splicing-specific dense representations of the segmented regions in the donor and query images, and it produces a probability value indicating the likelihood that the donor image was used to splice the query image. We describe the details of each of these stages in the following.

2.2 Splicing Localization

Although other CNN models (e.g., ResNet [20]) can be also applied, we use the first four convolutional blocks of the VGG16 model [34] just for the sake of simplicity. Consequently, the two network inputs Q and P (of shape $3 \times 256 \times 256$) are transformed into deep tensor representations Q_f and P_f (of shape $512 \times 16 \times 16$.) It is well known that CNN features like Q_f and P_f have already exhibited certain level of invariance to luminance, scale, and rotation.

The purpose of *Deep Dense Matching* is to find possible matching regions between representations Q_f and P_f. As shown in Fig. 3(b), this is achieved through two steps, namely *Deep Feature Correlation* and *Correspondence Match Pooling*. In *Deep Feature Correlation*, we exhaustively compute matching response using cross-correlation over all possible translations, as shown in Eq. (1)

$$\mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)[x, y, i, j] = \mathrm{trans}(\mathrm{P}_f, x, y)[:, i, j] \cdot \mathrm{Q}_f[:, i, j] \quad (1)$$

where $\cdot$ is the dot product operator, and $\mathrm{trans}(\mathrm{Z}_f, x, y)$ circularly translates Z_f *w.r.t.* (x, y) pixels, as defined in Eq. (2).

$$\mathrm{trans}(\mathrm{Z}_f, x, y)[:, i, j] = \mathrm{Z}_f[:, \mathrm{mod}(i + x, 16), \mathrm{mod}(j + y, 16)] \quad (2)$$

In *Correspondence Match Pooling*, we extract out meaningful response maps using three types of pooling—average pooling as defined in Eq. (3)

$$\mathrm{avgPool}\big(\mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)\big)[i, j] = \sum_{x=0}^{15} \sum_{y=0}^{15} \mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)[x, y, i, j]/256 \quad (3)$$

max pooling as defined in Eq. (4)

$$\mathrm{maxPool}\big(\mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)\big)[i, j] = \max_{x,y} \{\mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)[x, y, i, j]\} \quad (4)$$

and argsort pooling as defined in Eq. (5)

$$\mathrm{argsortPool}\big(\mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)\big)[k] = \mathrm{corr}(\mathrm{P}_f, \mathrm{Q}_f)[k_x, k_y, i, j] \quad (5)$$

where (k_x, k_y) in Eq. (5) is determined by the kth maximum response over all translations. Finally, we concatenate one average, one max, and the top six argsort response maps along the feature dimension and obtain the dense matching response Q_r of shape $8 \times 16 \times 16$ for Q. By interchanging the roles of P_f and Q_f in $\mathrm{corr}(\cdot, \cdot)$, one can therefore obtain P_r.

Fig. 4 visualizes intermediate results of the proposed *Deep Dense Matching* layer for the three testing samples from Fig. 2. As one can see, the proposed deep dense matching module 1) successfully localizes potential splicing regions, and 2) produces substantially higher responses to the two positive samples above than the negative sample (see the red dash line on the right half stats figure in Fig. 4), implying that the previous *CNN Feature Extractor* and *Deep Feature Correlation* provides meaningful representation with high discernibility.

In order to produce a splicing mask from the dense response map, we use a *Mask Deconvolution* module, as shown in Fig 3(b), where we gradually deconvolve a response map by a factor of 2 until its size reaches the size of input, i.e., 256×256. During each deconvolution stage, we apply an inception module [36] with a larger filter size and a smaller number of filters, where the two types of inception modules can be seen in Fig. 5. This enables us to obtain splicing masks for both image P and Q, i.e., outputs P_m and Q_m (see examples in the "Predicted Mask" column in Fig. 4.)

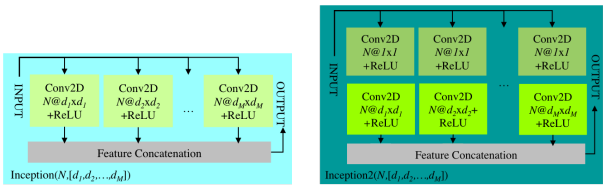


Figure 5: The internal architecture of the two types of used inception modules.

2.3 Splicing Detection

Intuitively, given predicted splicing masks Q_m and P_m, one can easily determine splicing or not through visual inspection, namely verifying whether or notimage contents covered by two masks match. We therefore follow this intuition to design a *Visual Consistency Validation* module to fulfill the splicing detection task as shown in Fig. 3(b). Specifically, we first use the *Visual Attention* module to zero-out all non-spliced regions in the CNN feature, as shown in Eq. (6),

$$\mathrm{visAtt}(\mathrm{Z}_f, \mathrm{Z}'_m)[c, i, j] = \begin{cases} \mathrm{Z}_f[c, i, j], & \text{if } \mathrm{Z}'_m[i, j] > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where Z'_m is the result of Z_m after classic *MaxPooling2D* for a size (16,16). This process is analogous to forcing the network to pay attention only to splicing regions. Furthermore, we follow a Siamese-like network to compare these two attention features—namely, extract a new round of features from the two attention features using the *Deep Feature Extractor* (see Fig. 6 for detailed architecture), and then compute the difference between the two resulting features. We then concatenate this feature with the feature obtained from the average mask responses. Finally, we use three stacked dense layers to infer the probability that $p = \mathrm{Proba}(\mathrm{P}$ is a donor of $\mathrm{Q})$, and this fulfills the detection task as shown in Fig. 3(b).

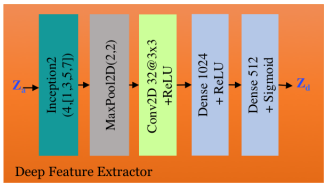


Figure 6: Internal architecture of the deep feature extractor.

2.4 Training Data and Strategy

To the best of our knowledge, no dataset exists that is large enough to be directly used for training the proposed DMVN model. To overcome this limitation, we use the SUN2012 object detection dataset [40] and the MS COCO dataset [25] to create training samples according to the unsupervised generation process described in [44]. Briefly, we begin with a random image X with polygon-based object annotations, randomly select an object in X, then randomly transform this object and paste it to another randomly selected image Y to obtain a resulting composite image Z. We could harvest at most three (two positive and one negative) training

$\{inputs, outputs\}$ samples for each unsupervised data generation. For instance, Fig. 2 gives a set of three training samples of this type.

In terms of the parameters controlling the data generation process, we equally likely pick an image and an object, random affine transformation involving a scale change in $\mathbb{U}(.5, 4)$, rotation in $\mathbb{U}(-10, +10)$, shift in $\mathbb{U}(-127, +127)$, translation $\mathbb{U}(-127, +127)$ and random luminance change in $\mathbb{U}(-32, +32)$. This enables us to create as many samples as needed to train the network end-to-end. Effectively, we create 1.5 M(illion), 0.3M, and 0.3M synthesized samples for training, validation, and testing, respectively.

The proposed DMVN was implemented using the *Keras* deep learning framework with the *Theano* backend, including all custom correlation and pooling layers. Our model was trained with the *Adadelta* optimizer *w.r.t.* the *log loss* for both localization and detection branches. Since we design the splicing detection branch as a *Visual Consistency Validator* of image contents on predicted splicing masks, this branch output may not produce meaningful gradients unless the localization branch produced meaningful splicing masks. Thus, we first focus on the localization branch of DMVN model only. Once this localization branch converges, we freeze its weights, add on the detection branch, and train the detection branch until it converges. We finally unfreeze all DMVN weights and train the entire model end-to-end using the stochastic gradient optimizer with a learning rate $1e-5$ and momentum of 0.9. In summary, we achieve 98.52%, 98.67%, and 97.88% prediction accuracy on the localization branch, and 97.75%, 97.53% and 97.69% prediction accuracy on the detection branch on our synthesized training, validation, and testing datasets, respectively. Our pretrained model can be downloaded from https://gitlab.com/rex-yue-wu/Deep-Matching-Validation-Network.git.

3 EXPERIMENTAL RESULTS

3.1 Baseline Methods and Test Settings

Since the CISD formulation is completely new, we compare against baseline algorithms from the state-of-the-art copy-move detection algorithms. We rely on visual clues, because when we concatenate the two inputs from a CISD sample into a single combined image, the resulting image contains copy-move forgery if the CISD sample is positive. Specifically, we choose the classic block matching-based approach [27], the classic Zernike moments-based block matching [33] with nearest-neighbor search, the SURF feature-based keypoint matching [10] and the dense field matching [11]. All used baselines are implemented by either a third-party or by the authors of the corresponding papers.[2]

With regard to preprocessing, we resize an image to 256×256 to fit the input size of the proposed DMVN, and thus a 256×512 image for those baseline algorithms. With regard to postprocessing, we do not apply any to DMVN, i.e., using the outputs from DMVN localization and detection branches directly, while keeping default postprocessing settings of baselines unchanged. Since some baseline methods only output a splicing mask but not a binary decision on detection, we follow the tradition in the classic ISD and copy-move community to determine that a sample is positive if no pixel is labeled as spliced in a mask. Finally, all baseline methods are run on `Intel Xeon` CPU E5-2695@2.40GHz, and the proposed DMVN is run on `Nividia TitanX` GPU.

3.2 Dataset

We conduct evaluation experiments on two large datasets: 1) the paired CASIA dataset, and 2) the NIST-provided Nimble 2017 image splicing detection dataset. The paired CASIA dataset is a modified version of the original CASIA TIDEv2.0 dataset [38][3] which contains 7200 authentic color images and 5123 color images tampered with by realistic manual manipulations (e.g., resize, deform, and blurring) through *Adobe Photoshop CS3*. It was originally collected for both the image copy-move problem and the classic image splicing detection problem. Since the CISD problem requires a pair of inputs, we select pairs of images from the original CASIA dataset to create the new paired CASIA dataset. Among the 5123 CASIA tampered images, we that found 3302 are of the copy-move problem and 1821 are of the classic ISD task. We therefore generate 3642 positive samples by pairing these 1821 spliced images with their true donor images, and collect 5000 negative samples by randomly pairing 7491 color images from the same CASIA-defined content category. Our paired CASIA dataset can be found at https://gitlab.com/rex-yue-wu/Deep-Matching-Validation-Network.git.

With regard to the Nimble 2017 dataset, it is provided by NIST with 98 positive samples and 529,836 negative samples. This challenging dataset is particularly designed for the CISD task with considerations to 1) a very large scale (more than a half million samples), 2) more realistic and artistic manipulations like image impainting, seam-carving etc., 3) difficult negative samples with visually similar foreground and background, and 4) mimicking the real application scenario where the ratio of negative samples to positive samples is extremely huge.

It is worth emphasizing that 1) we directly test the DMVN models trained by our synthetic data without any finetuning, and 2) ground truth splicing masks are not available for both dataset.

3.3 Evaluation Metrics

Since both datasets do not provide ground truth splicing masks, we focus on assessing the splicing detection performance. We follow the tradition of the classic ISD and copy-move community, namely, using precision, recall, and f-score: TP stands for *true positive*, i.e., correctly detected as spliced; FN stands for *false negative*, i.e., incorrectly detected as not-spliced; FP stands for *false positive*, i.e., incorrectly detected as spliced; and TN stands for *true negative*, i.e., correctly detected as not-spliced.

$$\text{precision} = TP/(TP + FP) \tag{7}$$

$$\text{recall} = TP/(TP + FN) \tag{8}$$

$$\text{f-score} = 2TP/(2TP + FN + FP) \tag{9}$$

Furthermore, we also use *area under the ROC curve (AUC)* to evaluate overall system performance at different operation points, where the receiver operating characteristic (ROC) curve is determined as the function of *true positive rate (TPR)* in terms of *false positive rate (FPR)*. TPR and FPR are defined as shown in Eqs. (10) and (11).

[2] Available at https://github.com/rahmatnazali/image-copy-move-detection.git, https://www5.cs.fau.de/research/software/copy-move-forgery-detection/, http://www.grip.unina.it/research/83-image-forensics/90-copy-move-forgery.html as the date of April 10, 2017.

[3] http://forensics.idealtest.org/casiav2/

The area under an ROC curve then quantifies the over-ability of the system to discriminate between two classes. It is worth noting that a system which is no better at identifying true positives than random guessing has an area of 0.5, and a perfect system without false positives and false negatives has an area of 1.0; and that AUC is the only official metric used by the Nimble 2017 challenge .

$$TPR = TP/(TP + FN) \tag{10}$$

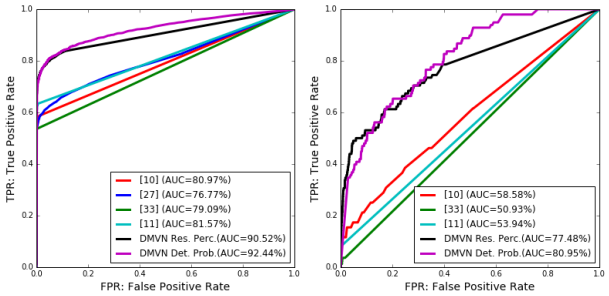
$$FPR = FP/(TN + FP) \tag{11}$$

3.4 Results

The most challenging task in the CISD is to 1) find spliced regions under various transformations like translation, rotation, scale, crop, etc., while dealing with complicated cases like multiple instances (a donor image contains multiple instances that are similar to a true spliced region), and multiple spliced regions (a donor image contributes more than one region); and 2) reduce false alarms on those visually similar but non-spliced regions.

Table 1: CISD performance comparison on CASIA

Method	Precision	Recall	F-score	Time (sec/sample)
[10]	51.64%	82.92%	63.64%	1.85
[27]	99.69%	53.53%	69.66%	$6.27\times10^{+2}$
[33]	96.14%	58.95%	73.09%	8.61
[11]	98.97%	63.34%	77.25%	3.23
DMVN loc.	91.52%	79.18%	84.91%	7.16×10^{-2}
DMVN det	94.15%	79.08%	85.96%	8.29×10^{-2}



(a) ROC on the paired CASIA dataset (b) ROC on the Nimble 2017 dataset

Figure 7: CISD performance comparison using ROC.

Table 1 shows the splicing detection performance of baseline approaches and the proposed DMVN methods on the paired CASIA dataset, where *DMVN loc.* means that we determine whether a sample (a pair of images) is positive or not by checking whether any pixel in a predicted splicing mask (from the DMVN localization branch) is positive, and *DMVN det.* means that we directly use the relation probability (from the DMVN detection branch) to determine a sample's label. As one can see, the proposed DMVN methods outperform peer algorithms by a large margin in terms of f-score (~7% higher) on the paired CASIA dataset. Note also that the proposed DMVN is significantly faster than baseline approaches by 20+ times, and that the DMVN detection branch improves our precision score from 91.52% to 94.15% while only reducing recall score for 0.1%, thus indicating the effectiveness of the proposed validation idea which relies on visual attention and Siamese architecture. Fig. 7 compares ROC and AUC scores for different methods, where *DMVN Res. Perc.* and *DMVN Det. Prob.* means that the threshold used to obtain TPR and FPR is based on the positive pixel percentage in a resulting mask, and the detection branch's output probability, respectively. Again, the proposed DMVN methods are noticeably better than others on AUC scores (10%+ on CASIA, and 20%+ on Nimble 2017).

With regard to splicing localization performance, Fig. 8 shows how the proposed DMVN method conquered this challenge on the paired CASIA dataset, where X_m indicates a splicing mask binarized with threshold 0.5, and $X_m * X$ indicates an overlaid image by using the splicing mask as the alpha channel with 40% transparency. To see this, one shall go to each row in Fig. 8, where the left and right sides show true positive and negative samples, respectively. Note that two samples on each row are intentionally picked from a similar CASIA category and/or a similar object class. As one can see, the proposed DMVN method not only predicts meaningful splicing masks on those positive samples, but also correctly suppresses splicing masks on those negative samples. Fig. 9 shows the manually annotated ground truth masks along with our predicted masks for the Nimble 2017 dataset.

3.5 Discussions

Fig. 10 qualitatively compares the splicing localization performance for all supported baselines. As one can see: 1) classic exhaustive block matching method [27] is sensitive to transformation, but good at capturing nearly duplicate regions; 2) block matching algorithm relying on Zernike moments [33] handles a certain level of transformations, but fails to maintain the completeness of a splicing region (see those holes in "[33]'s Masks" in row 5 of Fig. 10); 3) a keypoint-based detector [10] may fail due to no effective keypoints or noisy keypoints, which can commonly be seen in a homogeneous region or regions with similar texture, and one has to further convert potential matching points to a mask (see the last row; finding correspondence does not mean find a mask); and 4) the proposed DMVN method does not suffer the drawbacks of the previous three methods and gives satisfactory localization results on homogeneous and non-homogeneous regions even under severe transformations.

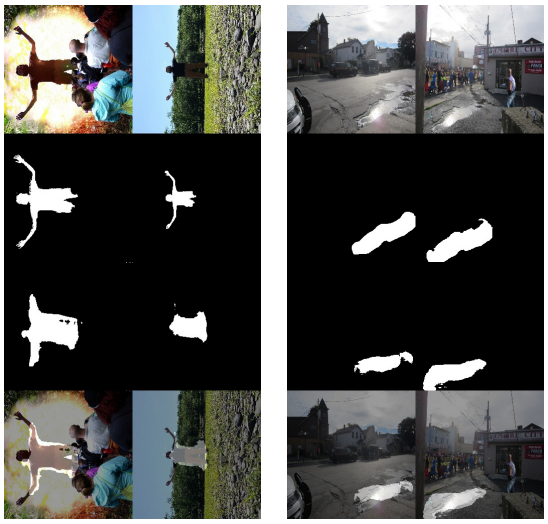
With regard to drawbacks, the proposed DMVN has some difficulty detecting splicing objects smaller than 8×8; this is due to the down-sampling in *CNN Feature Extractor*. As one may notice, our AUC scores on the Nimble 2017 dataset are much lower than those on the paired CASIA dataset. Besides the fact of extremely unbalanced positive (98) and negative samples (529836), we notice that the Nimble 2017 dataset contains much more challenging samples. For example, the proposed DMVN approach produces false alarms when two images P and Q are from consecutive video frames, because it mistakenly predicts those similar but genuine objects in both P and Q as spliced objects.

4 CONCLUSION

In this paper we propose a new deep neural network based solution for the image splicing detection and localization problems. We show that these two problems can be jointly solved using a multitask network in an end-to-end manner, as shown in Fig. 3. In

(a) True positive samples (b) True negative samples

Figure 8: DMVN localized splicing masks on the paired CASIA TIDEv2.0 dataset. (PID, QID) indicate the original CASIA filename of P and Q. Color blocks indicate different factors which splicing localization need to be robust against, namely ■: translation, ■: scale, ■: rotation, ■: perspective, ■: crop, ■: multiple instances, ■: multiple splicing objects, ■:similar foreground, ■: similar background.



(a) (PID,QID)= (19e32f2b29a2b345dfd28a8ad857b3fc, 797a0b5dd9c47a4b391f4cee60ba9354)

(b) (PID,QID)= (c0cfa4a6a95b3762f9d966e09a0340b9, 030d521fca99981c442e60e4b2155699)

(c) (PID,QID)= (9883d12156ed82344d65fe19575e1d84, 8b570c4bd216f5144774b9c0bc61bbd4)

(d) (PID,QID)= (4bf6a60d74ecd725e3e4b0eedc7d7cc0, 96492c250780ebababfcebc585ef1db6)

Figure 9: Predicted splicing masks on NC2017 dataset. From top to bottom, input pair, manually annotated ground truth masks, predicted splicing masks, and overlaid masks.

particular, we invent the *Deep Dense Matching* layer to find potential splicing regions for two given image features, and we design a *Visual Consistency Validator* module that determines a detection by cross-verifying image content on potential splicing regions. Compared to classic solutions, the proposed approach does not rely on any handcrafted features, heuristic rules and parameters, or extra post-processing, but could fulfill both splicing localization and detection. Our experiments on two very large datasets show that this new approach is much faster and achieves a much higher AUC score than classic approaches, and that it also provides meaningful splicing masks that can help a human conduct further forensics analysis (see Fig. 8). Last but not least, though we train our DMVN

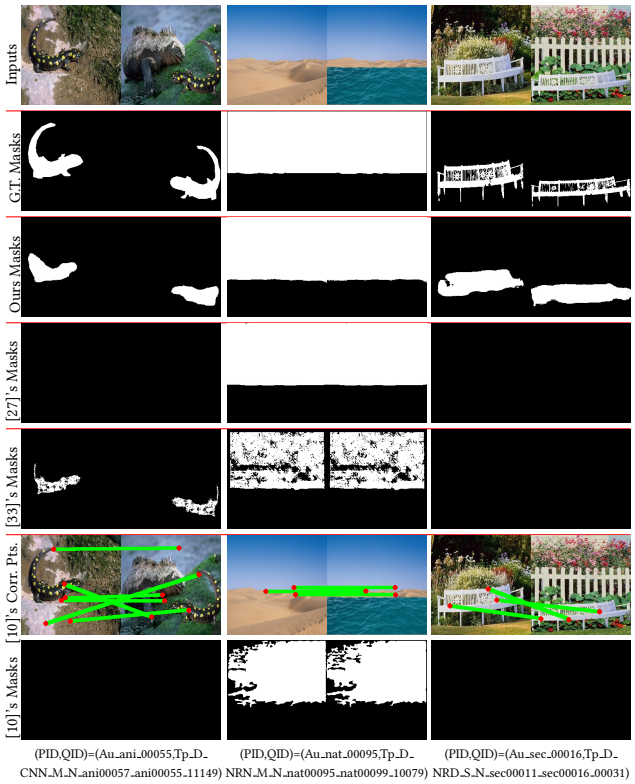


Figure 10: Visual comparison of detected masks. From top to bottom, input pair, manually annotated ground truth masks, predicted masks of using the proposed DMVN method, predicted masks of Alg. [27], [33]; the last two rows are [10]'s matched keypoints and predicted masks.

model *w.r.t.* both localization and detection branches, the proposed DMVN could be trained *w.r.t.* only the detection branch while still attaining the capacity to localize splicing masks due to the feed-forward nature of DMVN. This fact means that the proposed DMVN model can be easily finetuned to a new CISD dataset with only label annotations, and that one can save tremendous time and cost for splicing mask annotation in CISD training data collection.

ACKNOWLEDGEMENT

This work is based on research sponsored by the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Project Agency or the U.S. Government.

REFERENCES

[1] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. 2011. A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security* 6, 3 (2011), 1099–1110.

[2] Irene Amerini, Rudy Becarelli, Roberto Caldelli, and Andrea Del Mastio. 2014. Splicing forgeries localization through the use of first digit features. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*. IEEE, 143–148.

[3] Edoardo Ardizzone, Alessandro Bruno, and Giuseppe Mazzola. 2015. Copy–move forgery detection by matching triangles of keypoints. *IEEE Transactions on Information Forensics and Security* 10, 10 (2015), 2084–2094.

[4] Khurshid Asghar, Zulfiqar Habib, and Muhammad Hussain. 2016. Copy-move and splicing image forgery detection and localization techniques: a review. *Australian Journal of Forensic Sciences* (2016), 1–27.

[5] Mauro Barni, Marco Fontani, and Benedetta Tondi. 2012. A universal technique to hide traces of histogram-based image manipulations. In *Proceedings of the on Multimedia and security*. ACM, 97–104.

[6] Tiziano Bianchi and Alessandro Piva. 2012. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security* 7, 3 (2012), 1003–1017.

[7] Gajanan K Birajdar and Vijay H Mankar. 2013. Digital image forgery detection using passive techniques: A survey. *Digital Investigation* 10, 3 (2013), 226–245.

[8] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukás. 2008. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security* 3, 1 (2008), 74–90.

[9] Yi-Lei Chen and Chiou-Ting Hsu. 2011. Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security* 6, 2 (2011), 396–406.

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

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

[12] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. 2015. Splicebuster: A new blind image splicing detector. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. IEEE, 1–6.

[13] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. 2017. Recasting Residual-based Local Descriptors as Convolutional Neural Networks: an Application to Image Forgery Detection. *arXiv preprint arXiv:1703.04615* (2017).

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

[15] Alberto A de Oliveira, Pasquale Ferrara, Alessia De Rosa, Alessandro Piva, Mauro Barni, Siome Goldenstein, Zanoni Dias, and Anderson Rocha. 2016. Multiple parenting phylogeny relationships in digital images. *IEEE Transactions on Information Forensics and Security* 11, 2 (2016), 328–343.

[16] Zanoni Dias, Siome Goldenstein, and Anderson Rocha. 2013. Large-scale image phylogeny: Tracing image ancestral relationships. *Ieee Multimedia* 20, 3 (2013), 58–70.

[17] Zanoni Dias, Anderson Rocha, and Siome Goldenstein. 2012. Image phylogeny by minimal spanning trees. *IEEE Transactions on Information Forensics and Security* 7, 2 (2012), 774–788.

[18] Zhen Fang, Shuozhong Wang, and Xinpeng Zhang. 2010. Image splicing detection using color edge inconsistency. In *Multimedia Information Networking and Security (MINES), 2010 International Conference on*. IEEE, 923–926.

[19] Hany Farid. 2009. Image forgery detection. *IEEE Signal processing magazine* 26, 2 (2009), 16–25.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).

[21] Yu-Feng Hsu and Shih-Fu Chang. 2006. Detecting image splicing using geometry invariants and camera characteristics consistency. IEEE, 549–552.

[22] Ce Li, Qiang Ma, Limei Xiao, Ming Li, and Aihua Zhang. 2017. Image splicing detection based on Markov features in QDCT domain. *Neurocomputing* 228 (2017), 29–36.

[23] Haodong Li, Weiqi Luo, Xiaoqing Qiu, and Jiwu Huang. 2017. Image Forgery Localization via Integrating Tampering Possibility Maps. *IEEE Transactions on Information Forensics and Security* 12, 5 (2017), 1240–1252.

[24] Jian Li, Xiaolong Li, Bin Yang, and Xingming Sun. 2015. Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security* 10, 3 (2015), 507–518.

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

[26] Qingzhong Liu and Zhongxue Chen. 2014. Improved Approaches with Calibrated Neighboring Joint Density to Steganalysis and Seam-Carved Forgery Detection in JPEG Images. *ACM Trans. Intell. Syst. Technol.* 5, 4, Article 63 (Dec. 2014), 30 pages. DOI: http://dx.doi.org/10.1145/2560365

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

[28] Xunyu Pan, Xing Zhang, and Siwei Lyu. 2011. Exposing image forgery with blind noise estimation. In *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*. ACM, 15–20.

[29] Chi-Man Pun, Xiao-Chen Yuan, and Xiu-Li Bi. 2015. Image forgery detection using adaptive oversegmentation and feature point matching. *IEEE Transactions on Information Forensics and Security* 10, 8 (2015), 1705–1716.

[30] Zhenhua Qu, Guoping Qiu, and Jiwu Huang. 2009. Detect digital image splicing with visual cues. In *International workshop on information hiding*. Springer, 247–261.

[31] Muhammad Ali Qureshi and Mohamed Deriche. 2015. A bibliography of pixel-based blind image forgery detection techniques. *Signal Processing: Image Communication* 39 (2015), 46–74.

[32] Yuan Rao and Jiangqun Ni. 2016. 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.

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

[34] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[35] Ashwin Swaminathan, Min Wu, and KJ Ray Liu. 2008. Digital image forensics via intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security* 3, 1 (2008), 101–117.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.

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

[38] Wei Wang, Jing Dong, and Tieniu Tan. 2010. Image tampering detection based on stationary distribution of Markov chain. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2101–2104.

[39] Nor Bakiah Abd Warif, Ainuddin Wahid Abdul Wahab, Mohd Yamani Idna Idris, Roziana Ramli, Rosli Salleh, Shahaboddin Shamshirband, and Kim-Kwang Raymond Choo. 2016. Copy-move forgery detection: Survey, challenges and future directions. *Journal of Network and Computer Applications* 75 (2016), 259–278.

[40] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 3485–3492.

[41] Markos Zampoglou, Symeon Papadopoulos, Yiannis Kompatsiaris, Ruben Bouwmeester, and Jochen Spangenberg. 2016. Web and Social Media Image Forensics for News Professionals. In *Tenth International AAAI Conference on Web and Social Media*.

[42] Ying Zhang, Lei Lei Win, Jonathan Goh, and Vrizlynn LL Thing. 2016. 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.

[43] Xudong Zhao, Shenghong Li, Shilin Wang, Jianhua Li, and Kongjin Yang. 2012. Optimal chroma-like channel design for passive color image splicing detection. *EURASIP Journal on Advances in Signal Processing* 2012, 1 (2012), 240.

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

[45] Ye Zhu, Xuanjing Shen, and Haipeng Chen. 2016. Copy-move forgery detection based on scaled ORB. *Multimedia Tools and Applications* 75, 6 (2016), 3221–3233.