A Deep Learning Approach to Detection of Splicing and Copy-Move Forgeries in Images

Yuan Rao[1], Jiangqun Ni[2]
[#]Guangdong Key Laboratory of Information Security Technology, Sun Yat-Sen University
GuangZhou, GuangDong, P.R. China
[1]faraway3860@163.com, [2]issjqni@mail.sysu.edu.cn

***Abstract*—In this paper, we present a new image forgery detection method based on deep learning technique, which utilizes a convolutional neural network (CNN) to automatically learn hierarchical representations from the input RGB color images. The proposed CNN is specifically designed for image splicing and copy-move detection applications. Rather than a random strategy, the weights at the first layer of our network are initialized with the basic high-pass filter set used in calculation of residual maps in spatial rich model (SRM), which serves as a regularizer to efficiently suppress the effect of image contents and capture the subtle artifacts introduced by the tampering operations. The pre-trained CNN is used as patch descriptor to extract dense features from the test images, and a feature fusion technique is then explored to obtain the final discriminative features for SVM classification. The experimental results on several public datasets show that the proposed CNN based model outperforms some state-of-the-art methods.**

I. Introduction

With the fast development of digital image processing technology and the popularity of digital camera, editing or tampering a digital image becomes much easier even for an inexperienced forger with the aid of some user friendly photo-editing softwares, e.g., Adobe Photoshop. In the past few decades, doctored photographs are appearing with a growing frequency and sophistication and various digital forgery tools seemingly emerge in an endless stream, among which, splicing and copy-move are the most common ones that manipulate the images in a way to be hardly perceived by human perceptual system. Therefore, effective detection of these two kinds of forgeries is of great importance for digital image forensics.

In contrast to the active image forensic approaches [1], e.g., semi-fragile watermarking and image hashing, passive techniques [2] for image forensics are more useful, but more challenging. These techniques work on the assumption that although digital forges may leave no visual clues of what have been tampered with, they may alter the underlying statistics of an image. In recognition of this fact, a variety of image tampering detection techniques have been proposed in recent years. Shi *et al*. [3] proposed a natural image model for image splicing detection. They applied block discrete cosine transform (DCT) to images and combined the features extracted from statistical moments of characteristic functions with the ones from the Markov transition probability matrixes in both spatial and DCT domain, so as to obtain the discriminative feature vectors for support vector machine (SVM) classification. Later, He *et al*. [4] improved the method in [3] by resorting to a causal Markov model applied in both DCT and DWT (discrete wavelet transform) domains, and using the cross-domain features to train a SVM classifier. In [5], Zhao *et al*. further developed a 2-D noncausal Markov model to characterize the underlying image dependency and achieved splicing localization. Different from the model based schemes described above, Lyu *et al*. [6] proposed an alternative approach to detect and localize the splicing images by revealing the inconsistencies of local noises due to camera sensors or post-processing. Inspired by the fact that image tampering may alter the texture micro-patterns in an image, local binary pattern (LBP) and steerable pyramid transform (SPT) were employed in [7] to detect the distortions of textural properties in forged images and achieved so far the state-of-the-art detection performance on CASIA dataset [8].

In recent years, deep neural networks, such as Deep Belief Network [9], Deep Auto Encoder [10] and Convolutional Neural Network (CNN) [11], have shown to be capable of extracting complex statistical dependencies from high-dimensional sensory inputs and efficiently learning their hierarchical representations, allowing it to generalize well across a wide variety of computer vision (CV) tasks, including image classification [12], speech recognition [13], and etc. More recently, the deep learning based approach has also found applications in passive image forensics. In [14], a CNN model was trained for median filtering detection. However, Ying *et al.* [15] showed that the conventional deep learning framework may not be directly applied to image tampering detection, this because, with elaborated designed tools, the forgery images tend to closely resemble the authentic ones not only visually but also statistically. Therefore, they adopted the wavelet features of images as input of their deep autoencoder. Motivated by the similar observation, Bayar *et al.* [16] proposed a new convolutional layer in their CNN model to learn prediction error filters with constraint to discover the traces left by image manipulations.

In this paper, we propose a novel image forgery detection approach that can automatically learn feature representations based on deep learning framework. The primary contributions are summarized as follows: (1) We first train a supervised CNN to learn the hierarchical features of tampering operations (splicing and copy-move) with labeled patches ($p \times p$) from the training images. The first convolutional layer of the CNN serves as the pre-processing module to efficiently suppress the

978-1-5090-1138-4/16/$31.00 ©2016 IEEE

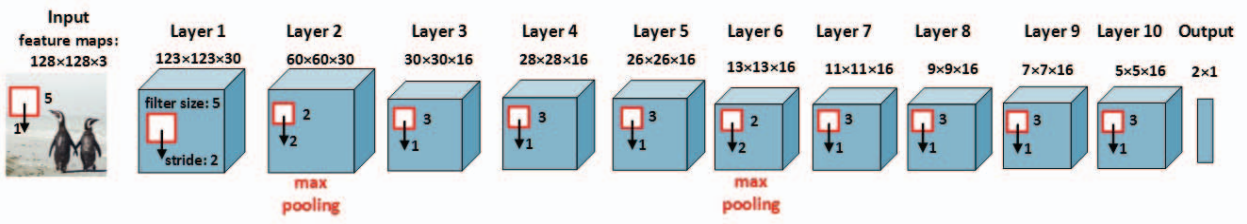

Fig. 1. The architecture of the proposed 10-layer CNN.

effect of image contents. Instead of the random strategy, the kernel weights of the first layer are initialized with the 30 basic high-pass filters used in calculation of residual maps in spatial rich model (SRM) [17], which helps to improve the generalization ability and accelerate the convergence of the network. (2) We then extract the features for an image with the pre-trained CNN on the basis of $p \times p$ patch by applying a patch-sized sliding-window to scan the whole image. The generated image representation is then condensed by a simple feature fusion technique, i.e. regional pooling, to obtain the final discriminative feature. (3) Finally, a SVM classifier is trained based on the resulting feature representation for binary classification (authentic/forged). The experimental results on several public datasets demonstrate that the proposed scheme can outperform some state-of-the-art methods.

The rest of the paper is organized as follows. In Section II, we present the proposed image forgery detection scheme, which includes CNN based feature learning and the process of feature fusion. The experimental results, comparisons and analysis are included in Section III. Finally, the conclusion remarks are drawn in Section IV.

II. The Proposed Method

The proposed method consists of two major steps, i.e., feature learning and feature extraction. In the first step, we pre-train a CNN model based on the labelled patch samples from the training images. The positive patch samples are elaborately drawn along the boundaries of the tampered regions in forged images, i.e., the boundaries of splicing and cloned patches, while the negative ones are randomly sampled from the authentic images. In this way, the CNN could concentrate on the local artifacts due to tampering operations and learn a hierarchical representation for the forged image. In the second step, the pre-trained CNN is used to extract the patch-based features for an image by applying a patch-sized sliding-window to scan the whole image. The patch-based features are aggregated through feature fusion to obtain the discriminate feature for an image, which is then used to train the SVM for image forgery detection.

A. Patch Sampling

Massive and representative data samples are usually the prerequisite to train a discriminate CNN model. In light of this, to prepare the positive samples (tampered), we draw the patch (R, G and B patches if color image samples are available) randomly alone the edges of splicing or copy-move regions in a tampered image. The number of positive patches drawn from a tampered image depends on the size of its tampered area in the image. In other words, the larger the tampered area in a forged image, the more positive patches could be drawn from the image. For negative samples, however, we randomly draw the same number of patches from authentic images in the training image set. The sampled patches then constitute the training data set for the CNN. To avoid the overfitting in CNN training and improve its generalization capability, some label-preserving transformations, e.g., transposing and rotating, are carried out on the training data set, which increases the data set by a factor of 8.

B. Architecture of the Proposed CNN

A convolutional neural network (CNN) consists of several cascaded convolutional layer and ends with some fully-connected layer followed by a softmax classifier. The input and output of a convolutional layer are sets of arrays called feature maps, while each convolutional layer usually produces feature maps through a three-step process, i.e., convolution, non-linear activation and pooling. Let us denote by $F^n(X)$ the feature map in layer n of the convolution with the kernel (filter) and bias defined by W^n and B^n, respectively, we have:

$$F^n(X) = pooling(f^n(F^{n-1}(X) * W^n + B^n)) \qquad (1)$$

where $F^0(X) = X$ is the input data, $f^n(\cdot)$ is a non-linear activation function that applies to each element of its input and $pooling(\cdot)$ represents the pooling operation that perform downsampling along spatial dimension using MAX or MEAN operation (max- or mean-pooling). Generally speaking, the non-linear activation and pooling operation are optional in a specific layer.

The architecture of the proposed CNN model is illustrated in Fig.1. It is shown that the proposed CNN consists of 8 convolutional layers, 2 pooling layers and a fully-connected layer with a 2-way softmax classifier. The input volume of the CNN are patches of size 128×128×3 (128×128 patch, 3 color channels). The first and second convolutional layers have 30 kernels with receptive field (a local region of input volume that each neuron connected in the convolutional layer) of 5×5, while other layers all have 16 kernels of size 3×3. For activation function, we apply Rectified Linear Units (ReLU) to neurons to make them selectively respond to useful signals in the input [18]. Note that both the second and forth convolutional layers are followed by a non-overlapping max-pooling

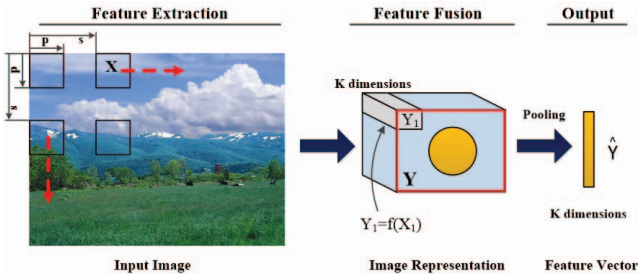

Fig. 2. The process of feature fusion.

with filter of size 2×2, which resizes the input spatially and discards 75% of the activations. This is because the max-pooling operation helps to retain more texture information and improve the convergence performance [19]. In addition, to improve the generalization, local response normalization is also applied to the feature maps before the pooling layer where the central value in each neighborhood is normalized by the surrounding pixel values. Finally, the extracted 400-D features (5×5×16) are passed to the fully-connected layer with 2-way softmax classifier through "dropout" [12] which sets to zero the neurons in fully-connected layer with probability of 0.5. Different from other conventional CNN architectures that employ two or more fully-connected layers, we use only one necessary fully-connected layer at the end of our network. This is because the fully-connected layer usually involves too many parameters to be trained, that could easily lead to overfitting, especially when the training set is not big enough, which is the case for our task.

C. The Proposed Initialization Method

Recall that the first convolutional layer of our CNN model serves as the pre-processing module to efficiently suppress the effect of image content. This coincides with the effort to build rich media model (RM) [17] in image steganalysis. As illustrated in [17], the advantage of modeling residuals instead of the pixel values is that the image content is largely suppressed in residual image, or the SNR (stego signal to image content) is greatly increased, so that the subtle artifacts introduced by steganography can be better detected. Interestingly, the strategy was also adopted in two recently proposed forgery detection schemes [20][21] by incorporating the high-pass filters devised in SRM. Such attempts are reasonable because sharp edges introduced by simple tampering operations, especially splicing, can be exposed to a large extent after high-pass filtering. Inspired by these, we propose to initialize the weights of the first convolutional layer of our CNN model with 30 basic high-pass filters used in calculation of the residual maps in SRM. These basic filters correspond to 7 residual classes in SRM, which include 8 filters in class "1st", 4 in class "2nd", 8 in class "3rd", 1 in class "SQUARE3×3", 4 in class "EDGE3×3", 1 in class "SQUARE5×5" and 4 in class "EDGE5×5". If we denote the index set of filters for i^{th} class by c_i $(i = 1, \cdots, 7)$, we have $c_1 = [1, \cdots, 8]$, $c_2 = [9, \cdots, 12]$, $\cdots$, $c_7 = [27, \cdots, 30]$, and

$$c = [c_1, \cdots, c_7] \tag{2}$$

Let W^c_{SRM} and W^c_{CNN} be the filter kernels of size $m \times n$ in SRM and the weight matrix of size 5×5 used in the first convolutional layer, respectively, we have:

$$W^c_{CNN}(x,y) = \begin{cases} W^c_{SRM}(x,y), & if\ x \le m, y \le n \\ 0, & otherwise \end{cases} \tag{3}$$

If the input volume of the first convolutional layer includes 3 color channels (R, G and B), each of the 30 output feature maps correspond to 3 weight matrices of size 5×5, and there are 90 weight matrices in total needed to be initialized with the 30 basic filters. According to our experiment results, to make the CNN as a better descriptor for the local patch, the 3 basic filters used for a specific feature map should be similar but not identical. In another word, we try to use those filters from the same residual class whenever possible. Therefore, we take the following measures to initialize the weight kernels in the first layer. Let $\boldsymbol{W}_j = [W^1_j\ W^2_j\ W^3_j]$ denote the weight kernel for j^{th} output feature map $(j = 1, \cdots, 30)$ and take the notations in (3), $\boldsymbol{W}_j$ is initialized as:

$$\boldsymbol{W}_j = [W^{3k-2}_{CNN} \quad W^{3k-1}_{CNN} \quad W^{3k}_{CNN}] \tag{4}$$

where $k = ((j-1) \bmod 10) + 1$. This initialization strategy acts as a regularization term in machine learning, which dramatically narrows down the feasible parameter space so as to facilitate the network to learn more robust features for the tampering operations rather than the complex image contents. Except for the first convolutional layer, Xavier initialization [22] that automatically determines the scale of initialization based on the number of input and output neurons is adopted for other layers, which often turns out to work much better than random initialization.

D. Feature Fusion

The pre-trained CNN learns a feature mapping f that transforms an input patch $X \in \mathbb{R}^{p \times p}$ (patch of size $p \times p$) to a much condensed representation $Y = f(X) \in \mathbb{R}^K$ (features of K-Dimension) and serves as the local descriptor for the input patch. As shown in Fig.2, for a test image of size $m \times n$, we scan it by applying a sliding-window of size $p \times p$ with a stride of s , compute the K-D local descriptor Y_i for each sampled patch X_i, and concatenate all the Y_i together to obtain the new image representation:

$$\boldsymbol{Y} = [Y_1\ \cdots\ Y_T] \tag{5}$$

where $T = (\lceil (m-w)/s \rceil + 1) \times (\lceil (n-w)/s \rceil + 1)$, and $\lceil x \rceil$ is the ceiling function. In order to obtain a more accurate representation of an image, denser feature is preferred in practice, so that s is usually set to be smaller than or equal to p. For the obtained image representation $\boldsymbol{Y}$ with T local descriptors Y_i, pooling operation (max or mean pooling) is applied on each dimension of Y_i over T sampled patches, i.e.,

$$\hat{Y}[k] = \text{Mean or Max}\{Y_1[k]\ \cdots\ Y_T[k]\} \tag{6}$$

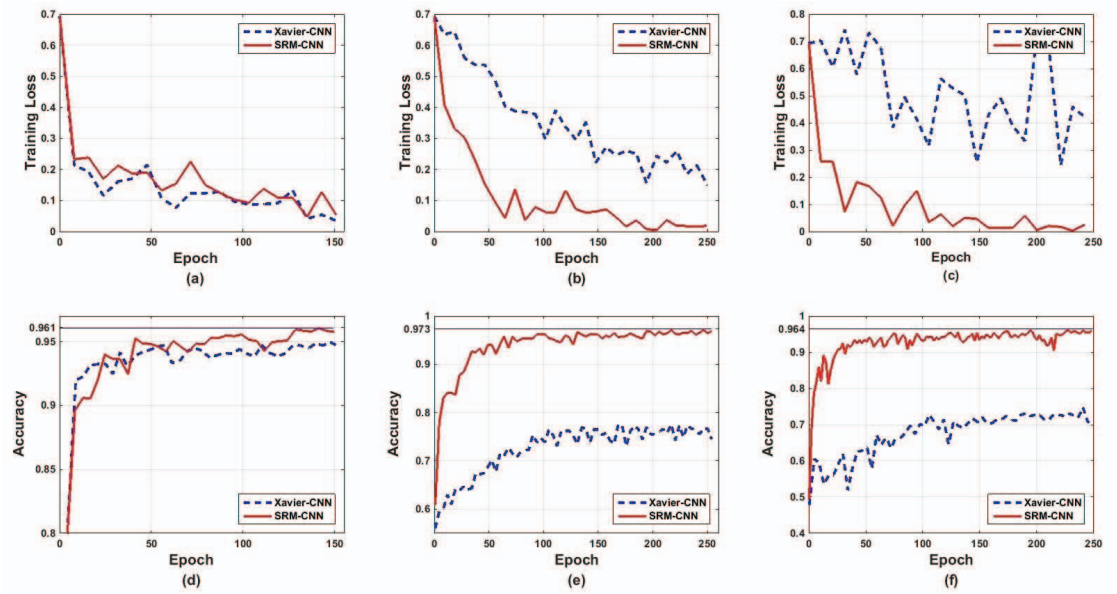

Fig. 3. The convergence and detection performance comparison between SRM- and Xavier-CNN on 3 public datasets. The evolution of training loss (top) and accuracy (bottom) on validation set versus number of epochs on (a) and (d) CASIA v2.0 dataset, (b) and (e) CASIA v1.0 dataset, (c) and (f) DVMM dataset, respectively. The validation set and test set for DVMM dataset are the same.

where $k \in [1, K]$, leading to the K-D feature vector for SVM classification.

III. Experiment and Analysis

In this Section, experiments are carried out to demonstrate the effectiveness of our proposed deep learning approach for image forgery detection. We also compare our scheme with several state-of-the-art image forgery detection methods on public datasets.

A. Image Dataset

All of our experiments are performed on 3 public benchmark datasets for forgery detection, i.e., CASIA v1.0 [8], CASIA v2.0 [8] and Columbia gray DVMM [23]. The CASIA v1.0 dataset contains 1,725 color images of size 384×256 pixels in JPEG format, among which, 925 are forged images generated by pasting the cropped image regions processed by resizing, rotation or deformation. The CASIA v2.0 database is more challenging for introducing post-processing on boundary area of tampered regions. It contains 7,491 authentic and 5,123 forged color images with size ranging from 240×160 to 900×600 pixels in JPEG, BMP and TIFF formats. Both CASIA v1.0 and v2.0 contain splicing and copy-move forged images. The DVMM dataset consists of 933 authentic images and 912 spliced images of size 128×128 pixels without any post-processing.

B. Implementation Details

To evaluate the detection performance of the proposed and other involved schemes, for each data set, we randomly divide it into six equal-sized groups, where each group consists of one sixth of non-repetitive authentic and forged images randomly drawn from the data set. And the performance is based on the average of 6 independent experiments. In each experiment, one group of authentic and forged images is selected for testing while the remaining 5 groups are used to train a SVM classifier.

For CASIA v1.0 and CASIA v2.0 databases, we draw the image patches of size 128×128 from training groups to train the CNN-based descriptor. To prepare the tampered patches, we find a pair of sample images (tampered image and its origin) each time according to the image label, subtract them and denoise the resulting difference image so as to reveal the edge of tampering operation. As a result, the positive training patches are extracted randomly along the forged boundaries in tampered images while the negative ones are obtained randomly in authentic images. And 5/6 of the extracted patches are randomly picked out to train the CNN descriptor and the rests are used to validate its performance. When training with the color patches in CASIA datasets, we initialize the weights in the first convolutional layer of our CNN model based on formula (4), while (3) is only adopted for training with the gray scale patches in DVMM dataset. The proposed CNN is implemented using Caffe [24] and conducted on a NVIDIA Tesla K40 GPU. The time of feature extraction is typically less than one hour on our PC with Intel(X) E5-2630 CPU and 64GB RAM on CASIA datasets. Stochastic gradient descent is employed to optimize the network with a momentum value of 0.99 and a weight decay of 5×10^{-4}. Initial learning rate is set to 0.01 and declines 10% every 10 epochs. In addition, early stopping is introduced to avoid overfitting.

The pre-trained CNN is treated as a patch descriptor of 400-D (5×5×16, see the input of fully-connected layer in Fig.1) and used to represent a test image in a "multiple patches"

TABLE I
DETECTION PERFORMANCE OF THE PROPOSED SCHEME WITH AND WITHOUT SRM INITIALIZATION WHEN DIFFERENT STRIDE AND POOLING ARE APPLIED.

Dataset	Pooling	Stride	Acc(%)	
			SRM-CNN	Xavier-CNN
CASIA v1.0	Max	64	98.04	88.24
		128	97.39	86.93
	Mean	64	98.04	87.91
		128	97.71	88.24
CASIA v2.0	Max	64	97.42	97.19
		128	97.83	97.30
	Mean	64	97.77	97.42
		128	97.48	97.30
DVMM	–	–	96.38	74.67

fashion. Feature fusion is then applied by pooling on each dimension of the extracted patch descriptors, leading to a 400-D discriminative feature vector. Note that for DVMM dataset with both train and test images of 128×128, we apply the pre-trained CNN to classify the images directly, and bypass the processes of feature fusion and SVM classification. For other two datasets, the resulting discriminative feature is sent to support vector machine LIBSVM [25] for classification. To train the SVM, C-support classification with the non-linear RBF kernel is performed and the parameter set (C, g) is determined with an exhaustive grid search strategy.

C. Performance of the Proposed Scheme

We first compare the convergence performance of the CNN when its first convolutional layer is initialized with the high-pass filters in SRM (denoted as SRM-CNN) or "xavier" (denoted as Xavier-CNN). For fair comparison, except for the first convolutional layer, other layer are all initialized with "xavier", and the involved two CNNs are trained through equal epochs. Fig.3 shows the evolutions of training loss and accuracy on validation set versus number of epochs on CASIA v1.0, CASIA v2.0 and DVMM datasets. Note that the validation set of DVMM dataset is also the test set due to the patch size used (the same as the one of test image). It is observed that the SRM-CNN converges much faster than Xavier-CNN and achieves much higher detection accuracy on both CASIA v1.0 and DVMM datasets. In fact, the Xavier-CNN could not even be converged on DVMM dataset. On the contrary, the convergence performances for both CNNs are almost the same on CASIA v2.0 dataset due to the much more samples in it. The size of CASIA v2.0 dataset is nearly 7 times of the other two, and more than 150,000 patches can be drawn after data augmentation to train the CNN, which contributes to the good convergence performance of the network.

We then proceed to compare the detection performance with and without SRM initialization when different stride setting and pooling operations are applied. In our experiments, the stride of patch is set to be 64 or 128 pixels when we represent the test image with the pre-trained CNN descriptor.

TABLE II
DETECTION PERFORMANCE COMPARISON OF THE PROPOSED SCHEME WITH OTHER METHODS.

Methods	CASIA v1.0 Acc(%)	CASIA v2.0 Acc(%)	DVMM Acc(%)
Proposed	**98.04**	**97.83**	**96.38**
Muhammad [7]	94.89	97.33	-
He [4]	-	89.76	93.55
Zhao [5]	-	-	93.36

Either max- or mean-pooling is employed in feature fusion. The detection performances in terms of test accuracy (Acc) on the 3 benchmark datasets are summarized in Table I. It is ready to see that the proposed initialization method improves the detection performances on all datasets, especially in CASIA v1.0 and DVMM datasets. Meanwhile, the effect of the involved stride setting and pooling operations on detection performance is negligible, indicating the robustness of the extracted features.

D. Comparisons With the Other Methods

We also compare our scheme with several state-of-the-art image forgery detection methods. Table II shows the detection performance comparison of the proposed scheme with other 3 methods, i.e., Muhammad in [7], He in [4] and Zhao in [5] on 3 public datasets. In the interest of fair comparison, for competing methods, only the best results are included in Table II, and those results on some datasets, which were not reported, are ignored. It is observed that the proposed scheme outperforms other methods for all the datasets tested. To the best of our knowledge, the method in [7] is the state-of-the-art method on CASIA v1.0 and CASIA v2.0 datasets. Our scheme outperforms the one in [7] by at least 3% and 0.5% on these two datasets, respectively. Note that the method in [7] is based on the Cb or Cr channels and cannot be applied to grayscale images. In this sense, our scheme is more preferable due to its extensibility to grayscale images. For DVMM dataset with grayscale images, our scheme also achieves a performance gain of at least 2.8% over the methods in [4] and [5]. In addition, the average ROC curves of our scheme and other competing schemes over six independent experiments are also given in Fig.4. Note that the AUCs (area under curve) of Muhammad' method [7] in Fig.4 (a) and (b) are larger than the ones reported in [7], this is because in [7] it utilized 9/10 of the images in the dataset for SVM training and others for testing, which is different from our current experimemtal setting. Once more, the AUC performance of our scheme also outperforms other methods as shown in Fig.4.

IV. CONCLUSION

In this paper, we propose a novel image forgery detection scheme based on deep convolutional neural network (CNN). The proposed CNN bears some customized designs for image tampering detection applications. Instead of a random strategy, the weights at the first layer of our network are initialized with the 30 basic high-pass filters used in spatial rich model

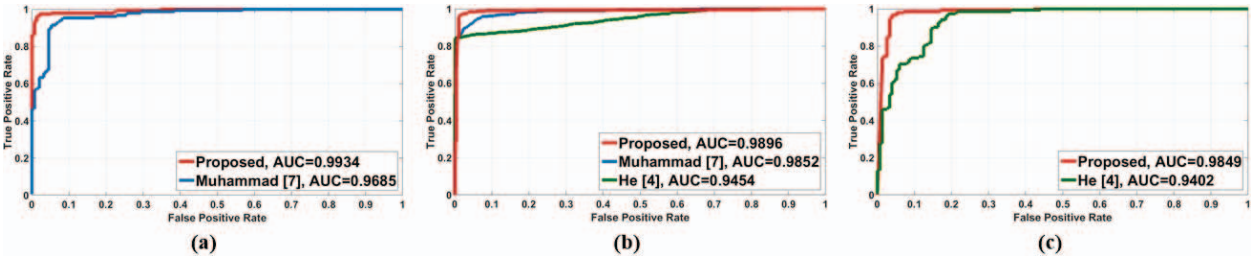

Fig. 4. ROC curve and AUC comparison of the proposed scheme with other methods on (a) CASIA v1.0, (b) CASIA v2.0 and (c) DVMM dataset, respectively.

(SRM) for image steganalysis, which helps to efficiently suppress the effect of complex image contents and accelerate the convergence of the network. In our method, the CNN model serves as a local patch descriptor, which is pre-trained based on the labeled patch samples elaborately drawn alone the forged boundaries in tampered images. The pre-trained CNN is then used to extract dense features from the test images, and a feature fusion technique is incorporated to obtain the final discriminative features for SVM classification. Extensive expriments on several public datasets have been carried out, which demonstrates the superior performance of the proposed CNN based scheme over other state-of-the-art image forgery detection methods.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (61379156 and 60970145), the National Research Foundation for the Doctoral Program of Higher Education of China (20120171110037), and the key Program of Natural Science Foundation of Guangdong (S2012020011114).

REFERENCES

[1] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, Madison, Wisconsin, USA, 2003, pp. 94-94.

[2] K. C. Chan, Y. S. Moon, and P. S. Cheng, "Fast fingerprint verification using subregions of fingerprint images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 95–101, Jan. 2004.

[3] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proceedings of the 9th workshop on Multimedia & security*. (MM & Sec), Dallas, TX, USA, 2007, pp. 51–62.

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

[5] X. Zhao, S. Wang, S. Li and J. Li, "Passive Image-Splicing Detection by a 2-D Noncausal Markov Model," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 185–199, Feb. 2015.

[6] S. Lyu, X. Pan, and X. Zhang, "Exposing Region Splicing Forgeries with Blind Local Noise Estimation," *International Journal of Computer Vision*, vol. 110, no. 2, pp. 202–221, 2014.

[7] G. Muhammad, M. Al–Hammadi, M. Hussain, G. Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications*, pp. 1–11, 2013.

[8] Dong, J., Wang, W.: CASIA tampered image detection evaluation (TIDE) database, v1.0 and v2.0 (2011). http://forensics.idealtest.org/

[9] H. Lee, C. Ekanadham, and A.Y. Ng., "Sparse deep belief net model for visual area V2", in *Advances in Neural Information Processing Systems* (NIPS) 20, 2008.

[10] Larochelle, Hugo, et al. "Exploring Strategies for Training Deep Neural Networks." *Journal of Machine Learning Research* 10.10(2009):1-40.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (NIPS), 2012, pp. 1097–1105.

[13] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," in *IEEE Signal Processing Letters*. ,vol. 21, no. 9, pp. 1120–1124, Sept. 2014.

[14] J. Chen, X. Kang, Y. Liu and Z. J. Wang, "Median Filtering Forensics Based on Convolutional Neural Networks," in *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849-1853, Nov. 2015.

[15] Z. Ying, J. Goha, L. Wina and V. Thinga, "Image Region Forgery Detection: A Deep Learning Approach," in *Proceedings of the Singapore Cyber-Security Conference (SG-CRC)*, 2016, vol. 14, p. 1–11.

[16] B. Bayar, and M. C. Stamm. "A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 5–10.

[17] J. Fridrich, and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882, June 2012.

[18] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines," In *Proceedings of the 27th International Conference on Machine Learning*, pp. 807-814, June 21-24, 2010.

[19] Boureau, Ylan, et al. "Learning mid-level features for recognition." in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, San Francisco, CA, 2010, pp. 2559-2566.

[20] L. Verdoliva, D. Cozzolino, and G. Poggi, "A feature-based approach for image tampering detection and localization," in *IEEE Workshop on Information Forensics and Security*, 2014, pp. 149–154.

[21] D. Cozzolino, G. Poggi and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2015, Rome, 2015, pp. 1–6.

[22] Glorot, Xavier, and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks." *Proc. JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, 2010, pp. 249-256.

[23] T.-T. Ng, J. Hsu, and S.-F. Chang. Columbia Image Splicing Detection Evaluation Dataset. [Online]. Available: http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol*., vol. 2, no. 3, pp. 1–27, 2011.