Detection of Image Manipulations Using Siamese Convolutional Neural Networks

Chapter · November 2019

DOI: 10.1007/978-3-030-34869-4_25

4 authors, including:

Aniruddha Mazumdar
Indian Institute of Technology Guwahati
11 PUBLICATIONS **75** CITATIONS
SEE PROFILE

Prabin Bora
Indian Institute of Technology Guwahati
132 PUBLICATIONS **1,342** CITATIONS
SEE PROFILE

Detection of Image Manipulations Using Siamese Convolutional Neural Networks

Aniruddha Mazumdar(✉), Jaya Singh, Yosha Singh Tomar, and P. K. Bora

Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, Assam 781039, India
{m.aniruddha,prabin}@iitg.ac.in

Abstract. The processing history of an image can reveal the application of different types of image editing/manipulation operations applied to images and also can expose forgeries. This paper proposes a novel deep learning-based manipulation detection method using a siamese neural network. The advantage of the proposed method is that it can even detect manipulations not present in the training stage. The network is first trained to differentiate between different types of image editing operations. Once the network learns feature that can discriminate different image editing operations present in the training stage, the unknown manipulations are detected using the *one-shot classification* strategy. We show that the network can also check whether an image is downloaded from a social media platform or not. The experimental results validate the efficacy of the proposed method.

Keywords: Image forensics · CNN · Manipulation detection · Siamese network

1 Introduction

While creating an image forgery, the forger generally applies different types of image editing operations either to make the forged image look realistic or to remove the trace of the manipulations. An image uploaded to a social media platform is also modified using different post-processing and compression techniques. Therefore, it is an important problem to detect the presence of different image editing operations applied on images.

A number of methods are available in the literature for detecting different types of image manipulation operations: JPEG compression [2,13], median filtering [10], resampling [14], contrast enhancement [16] etc. Caldelli *et al.* [3] proposed a method to check whether an image is downloaded from social media platforms, e.g. Facebook, Instagram, or not. The method is based on the idea that when an image is uploaded to a social networking site, it undergoes JPEG compression for reducing its size. Therefore, the authors proposed to extract

B. Deka et al. (Eds.): PReMI 2019, LNCS 11941, pp. 226–233, 2019.
https://doi.org/10.1007/978-3-030-34869-4_25

features based on discrete cosine transform coefficients to detect the source of an image.

Although these methods are good at detecting different types of manipulations, methods from each of the categories work only under their respective assumptions about the traces of manipulations left by the forgery process. For example, the methods designed for detecting re-sampling operations cannot detect JPEG compression related traces. To handle this limitation, a few general purpose manipulation detection methods have been proposed, which aim at detecting many different manipulations in a single framework. The first general purpose forensics method was proposed by Qiu *et al.* [15], where different steganalysis features were used to detect different types of image processing operations. The method is based on the observation that different image editing operations destroy the natural statistics of the image pixels present in an authentic image in the same way steganography methods do while manipulating the pixels for embedding a message. Fan *et al.* [6] proposed another general-purpose forensics method for detecting different types of image editing operations. The authors proposed to create a Gaussian mixture model (GMM) of image patches corresponding to each editing operation. Then, the average log-likelihood of patches under the different GMMs corresponding to different classes are compared to decide the class of the patches.

Bayar and Stamm [1] proposed a deep learning-based general purpose forensics method for detecting different types of image manipulating operations. The image manipulation features are automatically learned from the training data by employing a convolutional neural network (CNN). The authors proposed a new convolutional layer, which suppresses the image content and enhances features important for detecting different editing operations. Although this method performs well, different manipulation operations have to be known exactly before training the network. In real forensics scenarios, there can be any image editing operations applied to an image. In this case, all the general purpose manipulation detection methods will fail as these methods assume that the operations are known before designing the algorithms.

This paper proposes a novel deep learning-based forensics method for detecting image manipulations. Instead of learning features to classify image patches to different manipulation classes as in [1], the proposed method learns the features which can whether they come from the same or different manipulation operations. For this, the proposed method employs a deep siamese CNN, which has twin CNNs accepting two image patches as the input and learns to classifies the patch pair as either identically processed (IP) or differently processed (DP). Once, the network learns to differentiate between different manipulations present in the, it is used to detect unknown manipulations using the *one-shot classification* strategy. Also we show the ability of the network in detecting the social media source of images.

2 Proposed Method

The universal forensics method proposed by Bayar and Stamm [1] shows that CNNs can automatically learn features important for detecting different image editing operations. However, there are some limitations of the method. For training the CNN, all the image editing operations should be known *a priori*. But, in a real forensics scenario, we may need to check whether an image is being processed by a particular type of editing operation which may not be present in the training stage. In this case, the exisitng methods [15], [6], [1] will fail. Therefore, there is a need to develop forensics method which can not only detect manipulations present in the training stage but also can generalize to unknown manipulations.

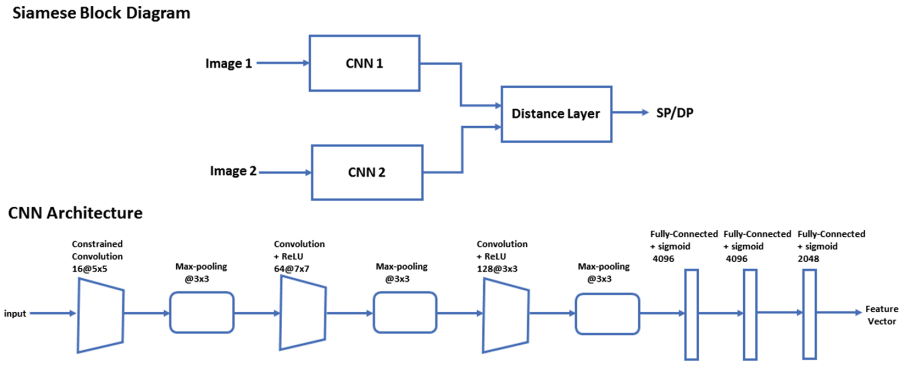

Fig. 1. The CNN architecture used in the proposed siamese network.

To overcome these limitations, we propose a distance metric learning method via a siamese neural network [11], which learns to check whether two image patches have undergone the same operation or two different operations. Once the network learns to discriminate between the manipulations present in the training stage, the network can be employed to check the presence of a manipulation not present in the training stage using the one-shot classification strategy. The reasons for the pair-wise classification image patches through distance metric learning technique is as follows: the distance metric learning technique enables the siamese network to learn more generic image manipulation related features than a simple CNN-based method [11]. This helps the network generalize to manipulations not present in the training stage.

Figure 1 shows the block diagram of the proposed framework. It has twin neural networks CNN1 and CNN2 sharing the same set of weights. It accepts two input images, which are independently processed by CNN1 and CNN2 and then a distance layer [11] computes a distance metric between the outputs of the twin networks. Because of the sharing of weights, CNN1 and CNN2 map two similar input images to very close points in the feature space. The proposed

siamese CNN automatically learns the features that can check whether a pair of images has been similarly or differently manipulated.

2.1 Network Architecture

CNN. The first convolutional layer in each of CNN1 and CNN2 is a constrained convolutional layer [1]. The filters in the constrained convolutional layer are forced to learn a set of prediction error filters, which suppress image contents and produce prediction error. Each of CNN1 and CNN2 has the architecture shown in Fig. 1. It contains 3 convolutional layers, 2 max-pooling layers and 3 fully-connected layers. The first convolutional layer is the constrained convolutional layer [1] with 16 prediction error filters of size 5×5 and stride 1. This layer is followed by an unconstrained convolutional layer with 64 filters of size 7×7 with stride 2. The ReLU nonlinearity is applied element-wise to the output of this layer followed by the max-pooling layer with a kernel size 3×3 and stride 2. The output of this layer is fed to another unconstrained convolutional layer with 128 filters of size 3×3 and stride 1. The ReLU nonlinearity is applied element-wise to the output of this layer. It is followed by a max-pooling layer with a kernel size 3×3 and stride 2. This layer is followed by three fully-connected layers with 4096, 4096 and 2048 neurons respectively. The sigmoid non-linearity is used in each of these layers. The neurons in the fully-connected layers are dropped out [8] with a probability of 0.5 at each iteration of the training process. The output of the final fully-connected layer represents the features learned by the CNN.

Distance Layer. Given a pair of image patches $\mathbf{x}_1$ and $\mathbf{x}_2$ as input, CNN1 and CNN2 compute the feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ respectively. A distance layer computes a distance metric between them, which is then fed to a single sigmoidal output neuron. This neuron computes the prediction of the input image patch pair as $p = \sigma(\sum_j \alpha_j \, |f_1(j) - f_2(j)|)$, where σ is the sigmoid non-linearity function and α_j is a learnable parameter representing the importance of each component of the feature vectors in the classification of the patch-pair.

2.2 Learning

The proposed siamese network is a binary classifier with label $y(\mathbf{x}_1, \mathbf{x}_2) = 1$ when both input image patches $\mathbf{x}_1$ and $\mathbf{x}_2$ come from the same manipulation class, and $y(\mathbf{x}_1, \mathbf{x}_2) = 0$ when $\mathbf{x}_1$ and $\mathbf{x}_2$ come from two different manipulation classes. The network is trained by minimising the average cross-entropy loss function C over a batch of pairs given by [12]

$$C = \frac{1}{M} \sum_{i=1}^{M} y(\mathbf{x}_1^i, \mathbf{x}_2^i) \log p(\mathbf{x}_1^i, \mathbf{x}_2^i) + (1 - y(\mathbf{x}_1^i, \mathbf{x}_2^i)) \log(1 - p(\mathbf{x}_1^i, \mathbf{x}_2^i)) \qquad (1)$$

where M is the number of images in each batch. The parameters of the network are learnt in the training phase by minimising C using the stochastic gradient descent (SGD)-based backpropagation technique.

Table 1. Different manipulations considered in this paper

Manipulation	Detail
Gaussian blurring	Kernel size = 5×5 and standard deviation (σ) = 1.1
Median filtering	Kernel size = 5×5
Resampling	Scaling factor = 1.5 and bilinear interpolation
Noise addition	AWGN with standard deviation (σ) = 2
Gamma correction	Parameter (γ) = 1.5
JPEG compression	Quality Factor (QF) = 70

2.3 Manipulation Detection

Once the network learns to differentiate between different image editing operations, it is used to detect different manipulations either present or not present in the training stage using one-shot classification technique. In one-shot classification technique, there is at least one reference image from each class and the class of the test image is determined by computing pair-wise prediction of test image paired with the reference image from each of the classes. Let $\mathbf{I}$ be a test image and $\mathbf{I}_c, c = 1, 2, ..., C$ be the reference images, then the class of the test image, c^*, is computing as

$$c^* = \text{argmax}_c p_c \tag{2}$$

where, p_c is the pair-wise prediction for the pair corresponding to the reference image $\mathbf{I}_c$.

3 Experiments and Results

3.1 Dataset and Setup

A dataset was created using the unprocessed raw images taken from the Dresden Image Database [7]. The database contains more than 14,000 images with resolutions of about 2000×3000 captured by 73 different digital cameras. We have saved 1566 images in the JPEG format with 100% quality factor (QF) and converted them into grayscale images by considering only the green channel of the images. We cropped image patches of size 150×150 from these images, resulting in 114,000 unaltered image patches. Six different versions of these unaltered patches are created by editing them with the operations listed in Table 1.

The proposed system was implemented using the Keras [4] deep learning library on a Tesla K20c GPU with 5 GB of RAM. The Nadam optimiser [5] was used with the parameters set as: $learningrate(\eta) = 0.002$, $momentum(\mu) = 0.002$ and $decay = 0.005$ and $regularizationterm(\lambda) = 0.0001$. We have used the learning rate decay technique to converge to the minimum of C by reducing the fluctuations [17]. The training batch size was set to 16 images. We have used the batch normalisation technique [9] as it helps in achieving faster convergence and higher generalisation accuracy.

3.2 Results

In the first experiment, we have trained the network using the image patches coming from four different classes: original, Gaussian blurring, median filtering, and resampling. We randomly selected 40,000 patches from each class to create the training set. We sample 500,000 IP pairs of patches randomly where both image patches of a pair come from the same class (i.e. both patches come either from unaltered class or from the same manipulation class). Similarly, we sample 500,000 DP pairs randomly, where the two images of a pair come from two different classes. The validation set contains 10,000 IP pairs and 10,000 DP pairs and the test set contains 50,000 IP pairs and 50,000 DP pairs. The network training was stopped at 70,000 iterations as it started converging and saved the final parameters of the model for the future use. On this test set, the model achieves an accuracy of 99.26%. This experiment shows ability of the proposed siamese network in discriminate the different types of image editing operations.

In this experiment, we classify each of the four manipulation types individually, i.e. original, Gaussian blurring, median filtering and resampling using the one-shot classification technique described in Sect. 2.3. For this, we have created a test set by randomly sampling 10,000 images from each of the four manipulations. For comparison, we have implemented Bayar and Stamm's method and tested its performance on these test images. It should be noted that the size of image patches used in this experiment is 150×150. Table 2 shows the performance of the proposed siamese network along with the CNN-based method [1].

Table 2. Classification accuracies on different manipulation classes

Manipulation	Proposed method	Bayar and Stamm [1]
Original	**99.35**	99.01
Gaussian blurring	**99.51**	99.22
Median filtering	**99.64**	99.34
Resampling	**99.26**	98.88

Table 3. Classification accuracies on manipulations not present in the training stage

Manipulation	Accuracy (%)
AWGN ($\sigma = 2$)	95.09
Gamma correction ($\gamma = 1.5$)	92.17
JPEG compression ($QF = 70$)	93.6

The next experiment is carried out to see the ability of the proposed method in detecting manipulations not present at the training phase. The unknown

Table 4. Classification accuracies on images coming from different social media platforms

Platform	Accuracy (%)
Facebook	77.45
Instagram	77.85
Flickr	69.95

manipulations are detected using the one-shot classification technique. For this, we assume that we have at least one image from each of the manipulations that we want to detect. The test set contains 10,000 images coming from the following three manipulation: corrupting the images with additive white Gaussian noise (AWGN) and applying gamma correction, and JPEG compression with quality factor (QF) 70. As can be seen in Table 3, the network achieved classification accuracies of 95.09% and 92.17% and 93.6% on AWGN, gamma correction and JPEG classes respectively. From these results, it is evident that the network can generalize to images come from manipulation classes not used in the training stage. This is a huge advantage of the proposed method over the state-of-the-art method [15], [6], [1] as they can not be applied in this case.

An experiment was carried out to test the performance of the proposed method in checking if an image is downloaded from any of the following social media platforms: Facebook, Instagram, Flickr or not. We collected a set of 100 images downloaded from each platform. The images are divided into patches of size 150×150 to create a test set, and each patch is considered as an individual test image. The test set contains 3648 images in Facebook class, 3000 images in Instagram class, 4214 classes in Flickr class and 4000 images which are not downloaded from any platform. The images from each category are detected in the one-shot classification strategy. The detection accuracy are shown in Table 4. These results show the ability of the network in detecting the social media source of an image, without training specifically for the task. The above detection accuracies may be further improved by training the network with JPEG compressed images with different QF.

4 Conclusions and Future Work

In this paper, a novel image forensics method was proposed which can detect different types of image manipulations carried out on images. The proposed method employs a siamese CNN-based metric learning technique which takes a pair of image patches as input and learns to check whether they are identically or differently processed. Once, the network is trained the known/unknown manipulations are detected using the one-shot classification strategy. The experimental results show the ability of the method in detecting known/unknown manipulations. Also, the ability of the proposed method in detecting social media source of images are also established experimentally. The future work will involve further exploration of the universal nature of the proposed method.

References

1. Bayar, B., Stamm, M.C.: Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. IEEE Trans. Inf. Forensics Secur. **13**(11), 2691–2706 (2018)
2. Bianchi, T., Piva, A.: Image forgery localization via block-grained analysis of jpeg artifacts. IEEE Trans. Inf. Forensics Secur. **7**(3), 1003–1017 (2012)
3. Caldelli, R., Becarelli, R., Amerini, I.: Image origin classification based on social network provenance. IEEE Trans. Inf. Forensics Secur. **12**(6), 1299–1308 (2017)
4. Chollet, F., et al.: Keras (2015)
5. Dozat, T.: Incorporating Nesterov momentum into Adam (2016)
6. Fan, W., Wang, K., Cayre, F.: General-purpose image forensics using patch likelihood under image statistical models. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6. IEEE (2015)
7. Gloe, T., Böhme, R.: The 'Dresden image database' for benchmarking digital image forensics. In: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1584–1590. ACM (2010)
8. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
10. Kang, X., Stamm, M.C., Peng, A., Liu, K.R.: Robust median filtering forensics using an autoregressive model. IEEE Trans. Inf. Forensics Secur. **8**(9), 1456–1468 (2013)
11. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop, vol. 2 (2015)
12. LeCun, Y., Huang, F.J.: Loss functions for discriminative training of energy-based models. In: AIStats, vol. 6, p. 34 (2005)
13. Liu, Q.: An approach to detecting JPEG down-recompression and seam carving forgery under recompression anti-forensics. Pattern Recognit. **65**, 35–46 (2017)
14. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Trans. Signal Process. **53**(2), 758–767 (2005)
15. Qiu, X., Li, H., Luo, W., Huang, J.: A universal image forensic strategy based on steganalytic model. In: Proceedings of the 2nd ACM workshop on Information hiding and multimedia security, pp. 165–170. ACM (2014)
16. Stamm, M., Liu, K.R.: Blind forensics of contrast enhancement in digital images. In: 2008 15th IEEE International Conference on Image Processing, pp. 3112–3115. IEEE (2008)
17. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 681–688 (2011)