Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016
A. Mathur and A. Roychoudhury (Eds.)

doi:10.3233/978-1-61499-617-0-1

Image Region Forgery Detection: A Deep Learning Approach

Ying Zhang [a,1], Jonathan Goh [a], Lei Lei Win [a] and Vrizlynn Thing [a]
[a] *Cyber Security & Intelligence Department, Institute for Infocomm Research, Singapore*

Abstract. In digital forensics, the detection of the presence of tampered images is of significant importance. The problem with the existing literature is that majority of them identify certain features in images tampered by a specific tampering method (such as copy-move, splicing, etc). This means that the method does not work reliably across various tampering methods. In addition, in terms of tampered region localization, most of the work targets only JPEG images due to the exploitation of double compression artifacts left during the re-compression of the manipulated image. However, in reality, digital forensics tools should not be specific to any image format and should also be able to localize the region of the image that was modified.

In this paper, we propose a two stage deep learning approach to learn features in order to detect tampered images in different image formats. For the first stage, we utilize a Stacked Autoencoder model to learn the complex feature for each individual patch. For the second stage, we integrate the contextual information of each patch so that the detection can be conducted more accurately. In our experiments, we were able to obtain an overall tampered region localization accuracy of 91.09% over both JPEG and TIFF images from CASIA dataset, with a fall-out of 4.31% and a precision of 57.67% respectively. The accuracy over the JPEG tampered images is 87.51%, which outperforms the 40.84% and 79.72% obtained from two state of the art tampering detection approaches.

Keywords. Image forgery detection, region localization, deep learning, feature learning

1. Introduction

In the digital era, there are an enormous volume of forged images on social media platforms such as Facebook or Flickr. The distribution of manipulated images can be shared very easily and can be used to mislead viewers from the truth. This may result in very serious consequences so the authenticity of digital images is urgently needed.

While there are a few solutions to automate image tampering detection, some of these methods are specific only to the JPEG file format [5,14,6,4,24,20,23] where they detect the tampered region based on artifacts left by multiple JPEG compressions. Other solutions [1,12,7] also identify features based on a specific tampering method such as

[1]Corresponding Author: Ying Zhang, Cyber Security & Intelligence Department, Institute for Infocomm Research, Singapore; E-mail: zhangy@i2r.a-star.edu.sg

copy-move where objects in the image are copied and pasted to hide or insert object. In these works, the duplicated parts of the image are discovered by invariant features.

There are a few techniques to automate image tampering detection. [10,19,11] determines the authenticity of the image where it is identified either as authentic or tampered [10]. However, these techniques do not identify the tampered region. Hence, a more sophisticated approach to obtain the tampered regions is required as this removes the need to manually identify suspicious regions. Currently, there are a few techniques to identify tampered regions. [5,14,6,4,24,20,23] exploits the artefacts left by multiple JPEG compressions to detect the tampered regions. However, these techniques are applicable only to the JPEG formats. Camera based methods [2,17] have also been explored where the detection is based on demosaicing regularity or sensor pattern noise where the irregularities of the sensor patterns are extracted and compared for anomalies. However, these methods are constricted to specific assumptions. For example, [16] works on the assumption that the image comes from a camera with the presence of Color Filter Array while [9] assumes that there is a presence of sensor patterns pre-obtained from specific camera models. Other methods to detect tampered regions also includes local descriptors. These methods [1,12,7] identify features that identify similar objects in the image which were copied and pasted to hide or insert object. These works are not applicable to tampering techniques such as splicing where objects are copied from one image and inserted into another.

In this paper, we address the above problems by proposing a two stage hierarchy feature learning approach for image tampered region detection. Deep learning provides a novel approach to the identification of features for tampered regions, which inherently represent characteristics of the tampered regions appearing in the dataset [18,21]. Such learning is data-driven and can be applied to images from any category of tampering. It greatly saves us time and energy to find new features from a set of images. In this work, we show that a deep learning Stacked Autoencoder (SAE) model can be used for feature learning for characterizing tampered regions. At the first stage, our experiments have shown that features of tampered regions are not only more effectively represented but it also results in a lower feature dimension. At the second stage, we provide contextual information by including neighboring patches to further improve the detection results. To the best of our knowledge, this work is the first work that utilises deep learning approach for feature learning in the field of tampered region localization.

In the following, Section 2, we will introduce our proposed method. Experimental results are shown in Section 3 and Section 4 concludes the paper.

2. Proposed Method

2.1. Basic Features Generation

As we are looking into tampered characteristics, we are unable to use pixels dependency as per traditional deep learning object recognition models [13]. Hence, we have to devise some basic features for the deep learning to learn and transform the initial features. We first converted the image into a YCrCb color space as studies [22] have shown that this color space is known to be more sensitive to tampering artifacts. We then segmented the image into 32 by 32 patches. Finally, we applied a 3 Level 2D Daubechies Wavelet

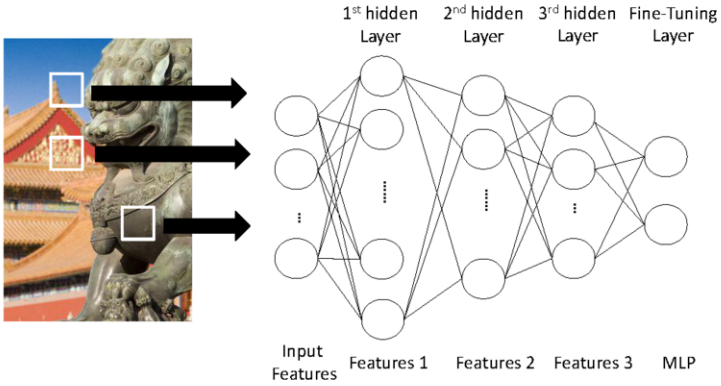


Figure 1. Stacked Autoencoder Architecture. The 1st, 2nd, 3rd hidden layers are for unsupervised learning with the 4th layer being a supervised classification network for feature fine tuning.

decomposition to each YCrCb component of the patches. We then obtained the standard deviation, mean, and the sum for each of the approximation, horizontal, vertical and diagonal coefficients to obtain 90 features. In addition, we applied Daubechies orthogonal wavelets D2-D5 to obtain a total of 450 basic features.

2.2. Two Stage Training Hierarchy for Tampering Detection

Based on the above raw input, we derive a complex feature with better discriminative ability to distinguish the tampered patch from the authentic ones. To achieve this, we propose a two-stage training hierarchy as follows. For the first stage, we utilize a SAE model to learn the complex feature for each individual patch. For the second stage, we integrate the contextual information of each patch so that the detection can be conducted more accurately.

2.2.1. 1st Stage: Stacked Autoencoders for Complex Feature Learning

At the 1st stage, we use a SAE for complex feature learning. A SAE is a neural network that is built by stacking multiple layers of basic autoencoders together. The outputs of each layer are treated as inputs to the successive layer. On top of the SAE, there usually comes with an additional MLP layer to further tune its parameters. The overall structure is shown as in Fig. 1 with three hidden layers plus a MLP layer. A SAE is known to be able to learn features that can represent its input. Typically, at the 1st layer, it learns the first order features. At the 2nd layer, it learns the second order feature which corresponds to the patterns of the first order features and so forth . Consequently, high layers of the SAE tends to learn higher-order features and have very good discriminative power. Consider a SAE with parameters W^l, b^l denoting the parameters for l^{th} autoencoder. Output of the l^{th} layer is $ae^{(l)}$ with its input z^l. Then the encoding of the input feature vectors over stacked autoencoders is performed by the encoding of each layer forwards as follows:

$$ae^{(l)} = f(z^{(l)}) \tag{1}$$

$$z^{(l+1)} = W^{(l)} ae^{(l)} + b^{(l)} \tag{2}$$

where $f(\cdot)$ is an activation function and a common choice is the sigma function:

$$f(x) = \frac{1}{(1+\exp(-x))} \tag{3}$$

The decoding of SAE is performed in a reverse way, i.e., by an decoding of each layer backwards:

$$ae^{(n+l)} = f(z^{(n+l)}) \tag{4}$$

$$z^{(n+l+1)} = W^{(n-l)}ae^{(n+l)} + b^{(n-l)} \tag{5}$$

In this aspect, if we have a SAE with n layers, then the transformed complex feature y is encapsulated as the activation of the last layer:

$$y = ae^{(n)} \tag{6}$$

To obtain the parameters of the SAE, we use the greedy layer-wise training [3]. Specifically, for the first layer, we use the raw input as introduced in Section 2.1 to obtain the parameters $W^{(1)}, b^{(1)}$. Then the activation output is used as the input of the second layer to obtain the parameters $W^{(2)}, b^{(2)}$. To obtain parameters more accurately, at the top of the second layer, we additionally add a layer each two nodes indicating either tampering or authentic and unroll and trained the whole architecture as a MLP [3]. So that for a new incoming image patch, we can represent them using Eq. (6).

2.2.2. 2nd Stage: Context Learning for Tampered Regions

Since tampered regions usually span across a few patches and would consist of different shapes and sizes, so the contextual information can additionally indicate if a patch is tampered or authentic. Therefore, for each patch, we introduced another layer to integrate the contextual information. To be specific, we firstly divide the image into non-overlapping 32 by 32 small patches. For each patch p, we determine its contextual neighborhood, say $\mathbf{N}(p)$, and we assume that there should exist a consistent feature pattern among patches within this neighborhood.

$$\mathbf{N}(p) = [y_p^0, y_p^1, y_p^2, ..., y_p^k] \tag{7}$$

where y_p^0 is the complex feature learnt by the introduced SAE from the patch p. And $y_p^i, i \geq 1$ is the feature of its i^{th} neighboring patch. Through this representation, the relation among spatially-close patches is preserved. The final prediction label is determined by an average value over its neighboring patches.

$$Prob(p) = \begin{cases} 1 & \frac{1}{k+1}\sum_{y_p^i \in \mathbf{N}(p)} MLP(y_p^i) \geq \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where $MLP(\cdot)$ is the probability value predicted by the MLP trained in the first stage and α is an threshold to binarize the map. In this paper, α is taken as 0.5, a midpoint of the maximal probability. For the neighborhood selection, we choose $k = 3$. That is to say, for each patch, we include its right, bottom and bottom-right three patches, position of each of which overlaps with the patch itself by half patch size horizontally, vertically and both, respectively.

3. Experimental Results

3.1. Data Setup

One of the major obstacles in image region localisation is the lack of an open source database available for benchmarking. The Columbia Image Splicing Database [15] and the CASIA database [8] are the only Image Forgery database available to date. However, these databases only denote whether the images were tampered or not and do not provide ground truths of the tampered regions. In order to overcome this obstacle, we manually labelled 1000 images randomly selected from the CASIA 1 and 2 databases. For the task of labelling the ground truth, we referred to their published instructions [8] on how the images were tampered and labelled the ground truths accordingly. Fig. 2 shows an example of the tampered image and the manually labelled ground truth respectively.

(a) Tampered image (b) Labelled ground truth

Figure 2. Example of tampered region and ground truth.

For training using the SAE strategy, we randomly selected 770 images for patch level training. There are totally 36439 tampered patches and 63561 authentic patches. The data is mildly imbalanced as the tampered regions are typically smaller than the rest of the authentic regions. However, we show in our experimental results that the imbalanced training data does not affect the final overall accuracy.

The remaining 230 unseen images (where there are 135 authentic images and 95 tampered images) were then used for validating our proposed method.

3.2. Training

Using the training architecture discussed in Section 2, in the first layer, we learnt our features using a SAE with 3 hidden layers. Hence, the network's input is 450 dimensions and the number of neurons in the SAE's remaining layers are 500-256-128-2. At the second layer which integrates contextual features, the average of the MLP values of the 3 half-overlapped neighbouring patches are further used to calculate the final prediction value.

The SAE training was performed using a Core i7 PC with 24GB RAM in a Matlab environment. The total training time was 13 hours based on the amount of training data.

3.3. Evaluation

Utilising the proposed structure, we generated the detected tampered regions for each of the 230 unseen images. As our model uses patches as inputs, we evaluated our proposed method based on patches as well. Since the ground truth is labelled at pixel-level, so

the corresponding grids will be labelled with the same labels, based on which all the afterwards evaluation are conducted. To be specific, the images are segmented into 32 by 32 grids, overlapped with the ground truth and labeled accordingly. Fig. 3 shows an example, the left figure is the original pixel-based ground truth while the right one is the patch-based ground truth. Based on this, there are 4 possible scenarios among the results:

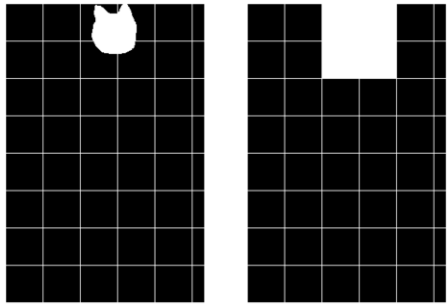


Figure 3. An illustration of patch-wise ground truth.

1) if both the output and the ground truth are labelled as tampered (in white color), we take the result as a True Positive (TP). 2) if the output grid is predicted as tampered (in white color) but the ground truth grid is labeled as authentic (in black color), the result is a False Positive (FP), 3) if the ground truth grid is authentic, but the output grid is predicted as authentic, the result is a False Negative (FN), 4) if both the output and ground truth label of a grid are authentic, the result is a True Negative (TN). Based on the above, we evaluate the results using the following criteria:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{9}$$

$$Fallout = \frac{FP}{FP+TN} \tag{10}$$

$$Precison = \frac{TP}{TP+FP} \tag{11}$$

For accuracy and Precison, the higher the values are, the better the classifier performs. For the Fall-out, the lower the value is, the better the classifier is. The statistics of the above criteria are summarized in Table 1. While our Fall-out is kept to a minimal of 4.31%, our precision is at 57.67%, which indicates that our proposed method can well predict the label of individual patch. As our method is dependent on patches, the detected tampered regions may not necessarily cover the exact parts of the ground truth grids and this results in more false positives which may effect the value of precision. However, this is can be solved by some image based post processing techniques. One possible solution is to leverage image segmentation in which way the labels of the segments are propagated from the corresponding patch labels. Therefore, we can effectively recover the tampered object since the localized region have already been identified.

Since the CASIA dataset contains both JPEG and TIFF images, and our method does not depend on specific image format, we further investigate the performance for each of them. As there is no authentic image with TIFF format, we reported the results for

Table 1. Performance Matrix for all test images.

Fall-out	4.31%
Precision	57.67%
Overall accuracy	91.09%

(a) Input TIFF image 1. (b) Ground truth of (a). (c) Detected results.

(d) Input TIFF image 2. (e) Ground truth of (d). (f) Detected results.

Figure 4. Detection results for TIFF image examples from CASIA 2 database.

tampered images only. Among the 95 tampered images, there are 51 JPEG and 44 TIFF, with results shown in Table 2. From the data, we can see that the performance for JPEG and TIFF are mostly similar, which indicates that our proposed method has the capability to be applied to various format images. As illustrated in table 2, the overall accuracies are consistent for each image format at 87.51% for JPEG and 81.91% for TIFF respectively. In addition, the fall-outs are kept to a minimal of 7.09% and 4.39% for JPEG and TIFF, separately.

Table 2. Performance Matrix between Tampered JPEG and TIFF images

	JPEG	TIFF
Fall-out	7.09%	4.39%
Precision	59.43%	80.65%
Overall accuracy	87.51%	81.91%

To visualize the results, we provided some examples in Fig. 4 and Fig. 5. Fig. 4 includes two examples of TIFF images while Fig. 5 illustrates two JPEG examples. Within each figure, the first column is the tampered images, the second column is the ground truth and the last column is our detection results. Note that for the detection results, we plot them in a grayscale version instead of binary values just for better visualization purpose. But all the statistics reported in the above table are based on binary label (i.e., either tampered or authentic). As shown in the these examples, our proposed method is capable of identifying the tampered regions accurately.

(a) Input JPEG image 1. (b) Ground truth of (a). (c) Detected results.

(d) Input JPEG image 2. (e) Ground truth of (d). (f) Detected results.

Figure 5. JPEG images from CASIA 2 database

3.4. Comparison with existing work

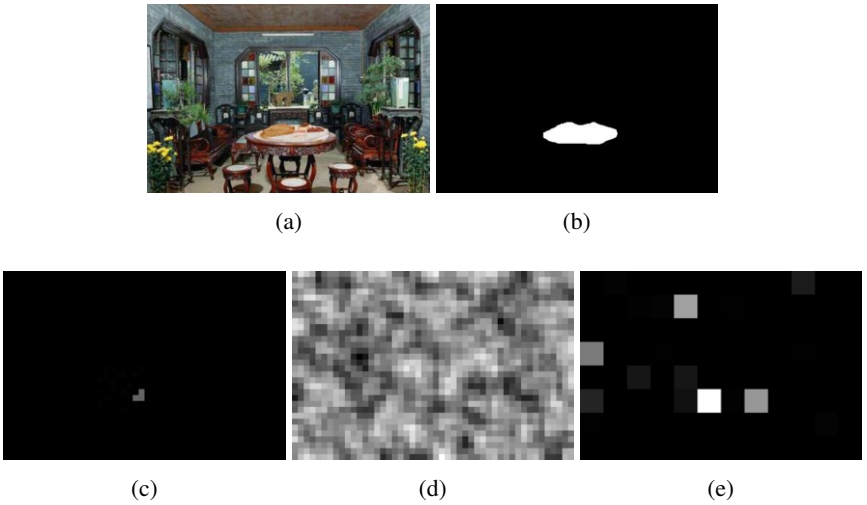
The above provides a results overview of our proposed method. In this section, we evaluate our method against existing works. Since JPEG image is the most popular image format today and majority of the work which localize tampered region such as [20,14,6,4,24] are only applicable to JPEG images, we compare our work with two existing studies in JPEG domain. Both of these works exploit the double compression artefacts left after tampering. The first method is proposed by Thing et al. [20] which leverages the double quantization effect among JPEG images for tampering detection. We choose it as it claims to be robust in practical applications even with a relatively limited or small training data set available. Additionally, this work is reported to perform well for CASIA dataset, so we believe it to be a good baseline. The second one is proposed by Bianchi et al. [4] based on an improved and unified statistical model characterizing the artifacts that appear in the presence of both A-DJPG or NA-DJPG.

Table 3. Detection accuracy comparison with existing works among all JPEG images.

Author	Accuracy
Bianchi et al. [4]	40.84%
Thing et al. [20]	79.72%
Proposed Method	**87.51%**

The experiments are conducted among the all JPEG images. For [4], we obtained their codes from the authors website. As for [20], we contacted the authors to obtain their codes for evaluation on our database. However, these existing methods are pixel based as compared to our proposed method which is patch base. Hence, in order to have a fair comparison, we took two average score from the corresponding patch in the existing method. Using the same 32 by 32 patch wise evaluation, we obtain the accuracy as shown in Table 3.

From the results, we can see that our proposed method achieves the highest accuracy at 87.51%, which outperform the second best by Thing et al. [20] by 7.79% and outperform the work by Bianchi et al.[4] by around 46.67%. In [4], the output of their algorithm is a probability map corresponding to the probability of the region being double compressed. Our results illustrate two main drawbacks on this method; firstly, the tampered regions need to be manually identified since the authors did not provide a threshold for detecting the regions. Secondly, the tampered regions are not accurately detected on our test database.



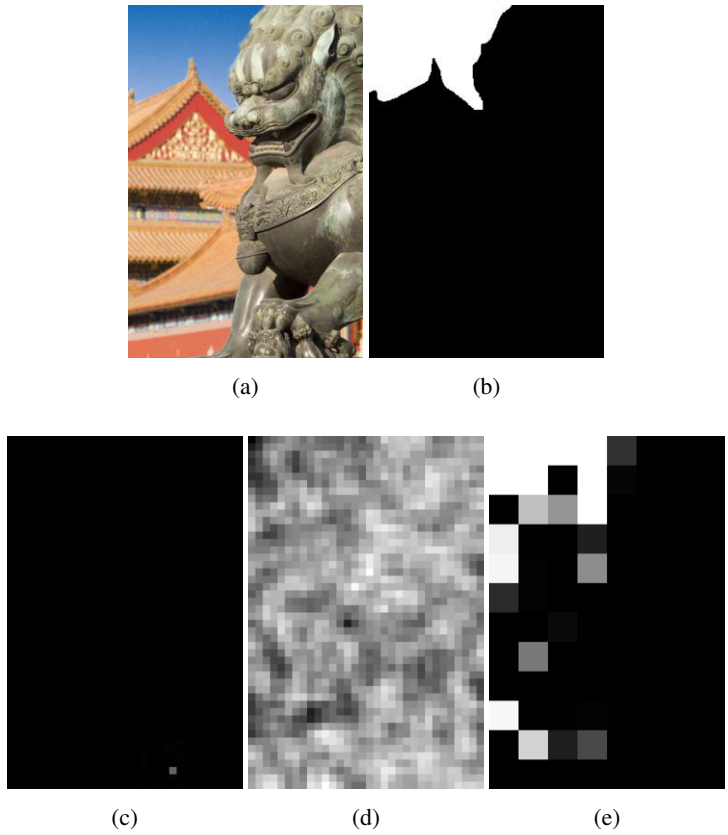
(a) (b) (c) (d) (e)

Figure 6. Comparison of detected tampering region, Example 1. (a) Input JPEG image, (b) Ground truth (white region is tampered), (c) results by Thing et al. (d) results by Bianchi et al. (e) results of our method.

We visually compare the detected regions with the existing works [20,4] as illustrated in Figs. 6 and 7. From the figures we can see that our proposed methods can detect the tampered region accurately. Furthermore, the method by [4] requires a threshold to be set in order to automatically identify the tampered region. Based on the table, we can observe that our proposed method obtained higher accuracy over existing works on the same test set. Furthermore, the advantage of our proposed method is that it is also applicable to both JPEG and TIFF image formats.

4. Conclusions

In this paper, we propose a deep learning approach for feature learning used to characterize tampered regions across multi format images. Our experiment results demonstrate that the proposed method detects tampered regions well with an overall accuracy of 91.09%. As future work, we will include other image transforms such as DCT as the base feature input. We will also investigate if other deep learning architectures such as Deep Belief Networks will improve the performance of feature learning. In addition, we will continue our efforts to manually label more ground truths from other datasets such as the Columbia Image Splicing dataset [15] as it also includes BMP file formats. This



(a) (b)

(c) (d) (e)

Figure 7. Comparison of detected tampering region, Example 2. (a) Input JPEG image, (b) Ground truth (white region is tampered), (c) results by Thing et al. (d) results by Bianchi et al. (e) results of our method.

will allow the deep learner to learn more characteristics of tampered regions and ensure better accuracy for tampered region localization across different image file formats.

Acknowledgement

This material is based on research work supported by the Singapore National Research Foundation under NCR Award No. NRF2014NCR-NCR001-034.

References

[1] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra. A sift-based forensic method for copy and move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, March 2011.

[2] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva. Estimate of prnu noise based on different noise models for source camera identification. *Crime Prevention Technologies and Applications for Advancing Criminal Investigation*, page 9, 2012.

[3] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[4] T. Bianchi and A. Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, June 2012.

[5] I. C. Chang, J. C. Yu, and C.-C. Chang. A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. *Image and Vision Computing*, 31(1):57–71, January 2013.

[6] Y. L. Chen and C. T. Hsu. Detecting recompression of jpeg images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security*, 6(2):396–406, June 2011.

[7] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on Information Forensics and Security*, 7(6):1841–1854, December 2012.

[8] J. Dong and W. Wang. Casia tampering detection dataset, 2011.

[9] J. Fridrich. Digital image forensics. *Signal Processing Magazine, IEEE*, 26(2):26–37, 2009.

[10] J. Goh and V. L. L. Thing. A hybrid evolutionary algorithm for feature and ensemble selection in image tampering detection. *International Journal of Electronic Security and Digital Forensics*, 7(1):76–104, March 2015.

[11] Z. He, W. Lu, W. Sun, and J. Huang. Digital image splicing detection based on markov features in dct and dwt domain. *Pattern Recognition*, 45(12):4292–4299, 2012.

[12] P. Kakar and N. Sudha. Exposing postprocessed copy-paste forgeries through transform-invariant feature. *IEEE Transactions on Information Forensics and Security*, 7(3):1018–1028, June 2012.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[14] Z. Lin, J. He, X. Tang, and C.-K. Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, January 2009.

[15] T. T. Ng, J. Hsu, and S. F. Chang. Columbia image splicing detection evaluation dataset, 2004.

[16] A. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *Signal Processing, IEEE Transactions on*, 53(10):3948–3959, Oct 2005.

[17] K. Rosenfeld and H. T. Sencar. A study of the robustness of prnu-based camera identification. In *IS&T/SPIE Electronic Imaging*, pages 72540M–72540M. International Society for Optics and Photonics, 2009.

[18] H. C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. *IEEE Pattern Analysis and Machine Intelligence*, 35(8):1930–1943, August 2012.

[19] P. Sutthiwan, Y. Shi, H. Zhao, T.-T. Ng, and W. Su. Markovian rake transform for digital image tampering detection. In *Transactions on Data Hiding and Multimedia Security VI*, volume 6730 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2011.

[20] V. L. L. Thing, Y. Chen, and C. Cheh. An improved double compression detection method for jpeg image forensics. In *IEEE International Symposium on Multimedia*, pages 290–297, December 2012.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, Dec. 2010.

[22] W. Wang, J. Dong, and T. Tan. Effective image splicing detection based on image chroma. In *IEEE International Conference on Image Processing*, pages 1257–1260, November 2009.

[23] W. Wang, J. Dong, and T. Tan. Exploring dct coefficient quantization effects for local tampering detection. *Information Forensics and Security, IEEE Transactions on*, 9(10):1653–1666, Oct 2014.

[24] F. Zach, C. Riess, and E. Angelopoulou. Automated image forgery detection through classification of jpeg ghosts. *Pattern Recognition*, 7476:185–194, January 2012.