Anomaly-Based Manipulation Detection in Satellite Images

János Horváth[1], David Güera[1], Sri Kalyan Yarlagadda[1], Paolo Bestagini[2], Fengqing Maggie Zhu[1]
Stefano Tubaro[2], Edward J. Delp[1]

[1]Video and Image Processing Laboratory (VIPER), Purdue University, West Lafayette, IN, USA
[2]Dipartimento di Informazione, Elettronica e Bioingegneria, Politecnico di Milano, Milan, Italy

Abstract

Satellite overhead imagery can be easily acquired and shared. The integrity of these type of images cannot longer be assumed, due to availability of sophisticated classical and machine learning based image manipulation tools. In this paper we proposed a deep learning based method for detecting and localizing splicing manipulations in overhead images. Our method uses recent advances in anomaly detection and does not require any prior knowledge of the type of manipulations that an adversary could insert in the satellite imagery. We compare our method against robust satellite-based manipulation detection approaches. We show that our proposed technique outperforms all previous methods, especially in detecting small-sized manipulations.

Figure 1: Examples of overhead images containing spliced objects and their corresponding splicing masks.

1. Introduction

The advent of satellites equipped with advanced imaging technology has enabled the creation of companies and military branches that rely entirely on overhead imagery. Commercial vendors and research institutions offer access to Earth observation imagery to the broad public [1, 2, 3]. These images can be forged as easily as any other image. Consumer image and video editing software such as GIMP and Photoshop, are capable of forging imagery which easily fools the human eye. This capability boost comes from improvements in hardware, software and the machine learning field. Deepfake videos [4, 5] are the epitome of this trend. Using machine learning open source software tools, it is now easy for anyone to create believable face swaps in videos that leave few traces of manipulation. Coupled with a continuously increasing amount of digital content being shared online, disregarding the misuse considerations of these image and video editing tools is not an option. Governmental institutions across the globe are already assessing the threat posed by satellite image manipulation, especially worrisome when done by a state-sponsored adversary [6].

Recent examples of satellite images that were manipulated include those of the shot down Malaysia Airlines Flight 17 [7] and nighttime flyovers of India during the Diwali festivity [8]. The forensics community has developed methods for proving image authenticity and assessing integrity [9, 10]. Some of these approaches focus on specific kinds of manipulation like splicing [11] and detect them through image provenance analysis [12]. Other methods have been designed to spot local manipulations generated by a generative adversarial networks (GANs) [13, 14, 15]. However, many state-of-the-art forensic methods are not efficient if blindly applied to overhead image analysis [16, 17, 18]. This is due to the fact that common image forensic methods are often developed for images taken with consumer cameras [19, 20]. These images sig-

nificantly differ from satellites images (e.g., different compression schemes, post-processing, sensors, color channels, ...). It remains an unresolved issue to verify the authenticity of overhead images without prior knowledge of the expected manipulations.

For this reason, it is urgent to develop forensic methods specifically tailored to analyze overhead images. In this paper, we propose a new robust deep anomaly detection and localization method, targeting splicing manipulations inserted in overhead images. Figure 1 shows examples of the kind of manipulations we expect to detect. In our work, we assume that we do not have access to forged images for training. Our method builds upon the findings of Yarlagadda *et al*. [17] and leverages recent advances in the anomaly detection field by Ruff *et al*. [21]. Our contributions for this paper are:

- The proposed approach significantly outperforms all previously presented satellite manipulation detection methods.
- By using anomaly detection techniques, we do not require access to any manipulated data for training.
- We introduce a new splicing detection function that takes into account the statistical properties of satellite data.

2. Related Work

The forensic community has designed several methods for detecting various types of forgeries in images. Most of these methods are designed for images captured using consumer cameras and smartphones [22, 20, 23, 24]. Since overhead images are captured using imaging sensors onboard satellites, their acquisition process is quite different when compared with images from consumer cameras. These include unique post-processing techniques such as orthorectification or multispectral color correction. The used compression schemes also differ from those commonly found in commercial photography. To bridge this gap, several methods to detect manipulations in satellite images have been proposed [25, 26, 17, 18].

In [26], Ho and Woon use watermarks to detect manipulations in satellite images. While methods using watermarks for assessing an image's integrity are certainly effective, they are rendered useless if the watermark is not inserted at image acquisition time by a trusted source. For this reason, several methods have been proposed to detect and localize splicing in satellite images in a blind scenario. Yarlagadda *et al*. [17] use a one-class support vector machine (OCSVM) adversarially trained on patches from pristine images to compress them to a low dimensional representation. For testing, the trained OCSVM acts on a compressed representation of patches coming from input images and classifies them as pristine or forged. Bartusiak *et al*. [18] train a conditional generative adversarial network (cGAN) on both pristine and forged images to detect and localize forgeries. Despite the great results of this last approach, it requires examples of the forgeries for training.

Inspired by previous literature on blind satellite manipulation detection and advances in anomaly detection, we pose the splicing detection and localization problem as an anomaly detection problem. By anomaly detection, we mean the problem of identifying unusual samples in the data. If we consider pristine overhead images to be our data, then spliced objects in them are the anomalies to be detected in the context of verifying their integrity. Many anomaly detection have been proposed, these include supervised [27] and semi-supervised approaches [28], or one-class neural networks [29, 21].

The applicability of each of the previously mentioned methods depends on the nature of the problem. In our case, since we assume no knowledge about the nature of the anomaly, both supervised and semi-supervised methods are not applicable. Since our goal is to be able to design a method that detects and localizes splicing using only information from pristine satellite images, a one-class approach seems to be the only viable solution. If we can model pristine images as belonging to one class and all of forged data (irrespective of the type of forgery) to belong to other classes, then our task would be to build a one-class classifier tailored to pristine data.

3. Proposed Method

We propose a deep one-class classifier that we name satellite support vector data description (SatSVDD) and extends the model proposed by Ruff *et al*. [21]. Images are split into patches and used to train an autoencoder, which encodes them into a low dimensional space. Then, a support vector data description (SVDD) classifier is trained jointly with the autoencoder to distinguish these low dimensional pristine encodings from encodings coming from forged patches. Although our overall strategy is similar to the one proposed by in [17], we empirically obtain better results, which we attribute to jointly training the autoencoder and the SVDD classifier as opposed to doing it separately. By doing so, the autoencoder is able to learn better latent representations that are more suitable to the SVDD.

The main idea behind the proposed splicing detection and localization method is to learn a compact representation of pristine data that captures the salient features of the image acquisition process. Splicing pristine data will lead to loss of some or all of the information of the image acquisition process, hence its compact representation will be very different from that of compact representation of pristine data. To obtain this representation, we first train our deep one-class classifier to extract an anomaly score from

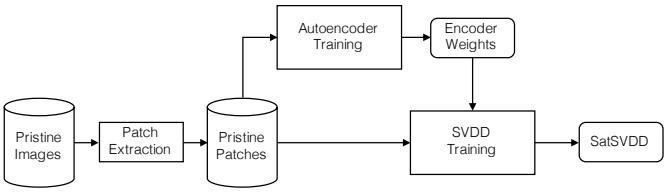


Figure 2: Overview of the training process of SatSVDD

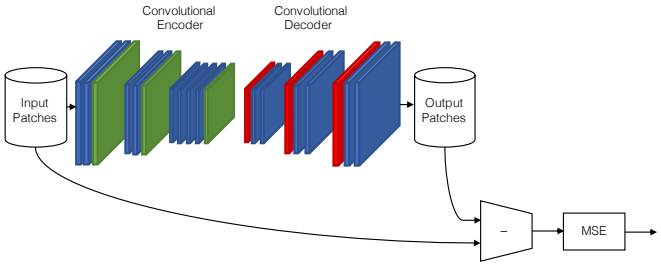


Figure 3: Overview of the autoencoder training step

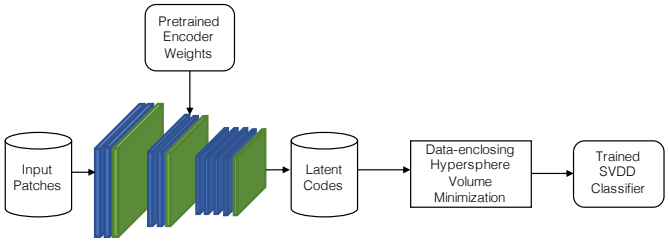


Figure 4: Overview of the SVDD training step

each image patch. When an image is processed, we combine all the estimated anomaly scores associated with the image patches into a splicing mask and splicing score. We provide details about the system training and deployment below.

3.1. System Training

Given a training dataset of pristine overhead images, we extract a set of overlapping patches at fixed resolution from all of them. These patches are used to train our deep one-class classifier following the two-step approach shown in Figure 2:

1. An autoencoder is trained to reconstruct the pristine image patches while simultaneously mapping them to a low dimensional space.

2. An SVDD one-class classifier is jointly trained with the encoder of the pre-trained autoencoder to obtain the final compact representation.

Autoencoder. We use the architecture proposed by Ruff *et al*. [21]. This is a convolutional autoencoder that encodes the extracted patches into a low dimensional feature vector as indicated in Figure 3. The autoencoder can be ideally split into two parts: an encoder that turns patches into feature vectors and a decoder that turns feature vectors into patches. This autoencoder is trained to minimize the mean squared error (MSE) between a patch and the output of the decoder.

Support Vector Data Description Classifier. We train the one-class classifier jointly with the pre-trained encoder as shown in Figure 4. The one-class classifier is chosen to be a support vector data description (SVDD) [30]. SVDD is a classifier related to the one class support vector machine (OC-SVM) where a hypersphere is used to separate the data instead of a hyperplane. SVDD works by minimizing the volume of the hyperspace that encloses all of the compact representations of the pristine patches. Specifically, SVDD minimizes the loss function defined as

$$\begin{aligned} &\min_{R,c,\zeta} \; R^2 + \frac{1}{vn}\sum_i \zeta_i \\ &\|\phi(x_i) - c\| < R^s + \zeta_i \text{ and } \zeta_i >= 0 \end{aligned} \tag{1}$$

Here ϕ refers to the encoder, R refers to the radius of the hypersphere, x_i refers to the i-th patch, ζ_i refers to the i-th slack variable and v is a hyperparameter that controls the trade off between ζ_i and R.

3.2. System Deployment

The testing pipeline is shown in Figure 5. Specifically, we compute a splicing mask from the test image, then we estimate a splicing detection score. Following we describe these two steps.

Splicing Map Computation. We extract overlapping patches from the image under analysis, as done during training. Then we test every extracted patch with the SatSVDD model and get an output value for every patch, from now on referred as anomaly score. The anomaly score indicates, how similar the patch features are to the feature of the patches used in training. The higher the anomaly score, the more likely that the patch contains an anomaly. Using the anomaly scores we construct a splicing mask.

To construct the splicing mask, we merge anomaly scores obtained from all patches. Specifically, the splicing mask is a 2D matrix the same size of the image under analysis. Each pixel of the splicing mask is the average of the anomaly scores computed for all patches overlapped with that pixel position in the analyzed image. Once this mask has been estimated, to localize anomaly we use the same method in [17]. First, we threshold each splicing mask to obtain a binary mask. For each image and used threshold, we compute: the true positive rate as the percentage of forged pixels correctly detected and false positive rate as the percentage of pristine pixels detected as forged. Based

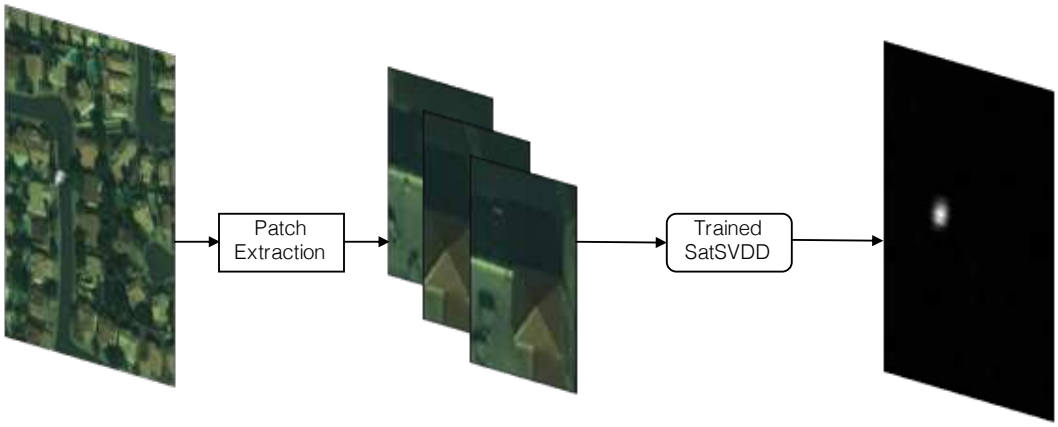


Figure 5: Overview of the testing process for manipulation detection and localization.

on these two values, we compute the ROC curves.

Detection Function. To detect whether an image contains a forgery, we need to turn the splicing mask into a detection value that can be compared to a threshold.

We consider spliced objects as anomalies and propose a detection function to compute the scalar splicing detection value from the splicing masks. Recall that an anomaly is defined as a subset of data whose properties deviate from the rest of the data [31]. In our problem, the splicing mask is our data, and anomalies should be detected as splicing mask values whose properties do not match the rest of the mask. If our classifier is properly trained, it learns how to extract some representative features of pristine patches. Therefore, for patches that do not contain anomalies, it returns a low anomaly score. Conversely, the classifier returns a high anomaly score from patches containing anomalies. From these two premises, we can expect that splicing masks belonging to images containing anomalies exhibit the following properties:

1. The splicing masks have a high maximum value on an absolute scale.
2. The maximum values of the splicing masks are higher compared to the average splicing masks values.
3. Due to the high maximum anomaly score, masks have a low normalized standard deviation.

The first property only takes into account one mask pixel at a time. The second and third properties take into account every pixel of the mask. The detection function proposed in [17] makes use of the first property as splicing detection score. We introduce a detection function for splicing masks that takes into account the second and third properties of the anomalies. We compute the splicing detection value as

$$d(\mathbf{M}) = \frac{\max(\mathbf{M}) - \mu_{\mathbf{M}}}{\sqrt{\frac{\sum_{x \in I}(x - \mu_{\mathbf{M}})^2}{\max(|\mathbf{M}|)}}}, \tag{2}$$

where $\mathbf{M}$ is the 2D mask composed by N pixels, $\mu_{\mathbf{M}} = \sum_{x \in \mathbf{M}} \frac{x}{N}$ is the average mask value, and $\max(\mathbf{M})$ is the maximum value of $\mathbf{M}$.

In using this score, we better capture the presence of an anomaly rather than simply thresholding the maximum $\mathbf{M}$ value.

4. Architectural Improvements

Updated Optimizer The original Deep SVDD implementation uses the Adam optimizer [32] for training. Reddi *et al*. [33] showed that Adam does not always convergence to the optimal solution. They also propose an alternative to Adam called Amsgrad, which is empirically shown to convergence to a better solution than Adam. Specifically, Amsgrad fixes Adams convergence issue by including a long term memory of past gradients.

Several works [34, 35, 36] have increased the popularity of Amsgrad to the extent that extensions to it have also been proposed [37, 38]. For this reason, we also use Amsgrad as the optimizer in our implementation.

Smooth Activation Function How to determine the most efficient activation function for each task is still an open research problem [39, 40]. Therefore, we also explored whether the original Deep SVDD [21] activation function between layers could be improved. Deep SVDD makes use of the leaky version of the rectified linear unit (Leaky ReLU), which is defined as

$$f(x) = \begin{cases} 0.01\,x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{3}$$

Leaky ReLU is continuous on its domain, and its derivative exists everywhere except for $x = 0$.

The function and its derivative are monotonic, but they are not approximate identities near the origin. Despite Leaky ReLU being broadly used in the machine learning

community [41, 40, 42], the literature has shown that the use of non-differentiable activation functions is not an optimal choice in some situations [43]. The same has been shown also by Ying *et al*. [44], where the authors introduced a novel differentiable graph pooling module which can learn a hierarchical representation of graphs.

Based on these findings, we change the activation function between layers from a non-differentiable one to a differentiable one. Specifically, we consider the exponential linear unit (ELU) defined as

$$f(\alpha, x) = \begin{cases} \alpha\left(e^{x} - 1\right) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{4}$$

where $\alpha = 1$ in our experiments. ELU with $\alpha = 1$ is continuous and differentiable. The function and its derivative are monotonic, and they are approximate identities near the origin. As we will show in Section 5, this choice leads to significantly better performance.

5. Experimental Analysis

In order to validate the proposed SatSVDD method and the different changes made to the Deep SVDD method [21], we carefully design a set of incremental experiments to test the final contribution of each change. We report all details about the used dataset, simulation setup, and the achieved results.

5.1. Dataset

We use the same dataset used by Yarlagadda *et al*. [17] for training and testing our approach. The dataset contains images captured for the Landsat Science program, more specifically images taken by the Landsat 8 satellite. The dataset is composed of 230 images with resolution 650×650. 30 pristine images are used for training, the rest are for testing. 50 testing images are pristine, whereas the remaining 150 contain forgeries. The 150 forged images were created starting from 50 pristine images, and applying forgeries of different sizes. Specifically, 50 images contain a forged region of about 32×32 pixel, another 50 images contain a forged region of about 64×64 pixel, and another 50 images contain a forged region of about 128×128 pixel.

In terms of image-to-patch splitting policy, we tested patches of resolution 32×32, 64×64, and 128×128 pixel. For both training and testing we extracted overlapping patches. During training we used a stride of 32×32, whereas during testing we used a stride of 13×13.

5.2. Experimental Setup

As the proposed method improves over Deep SVDD by adding a series of different or additional operations, we test different solutions by enabling or disabling the proposed modifications. Specifically, we carry out to the following five experiments:

- Deep SVDD model without any changes, as presented by Ruff *et al*. [21].
- Deep SVDD model with the improved optimizer, which we name SatSVDD-v1.
- Deep SVDD model with the improved activation function, which we name SatSVDD-v2.
- Deep SVDD model with the improved optimizer and activation function, which we name SatSVDD-v3.
- Deep SVDD model with the improved optimizer, activation function, and detection function, which we name SatSVDD-v4 in our experiments and is the finally proposed SatSVDD method.

For the sake of comparison, we also implement the baseline solution proposed by Yarlagadda *et al*. [17] for overhead image splicing manipulation detection and localization. For every experiment we analyze the results for different patch sizes, considering both detection and localization scores, in terms of area under the curve (AUC) for the precision and recall (P/R) and the receiver operating characteristic (ROC) curves.

6. Results And Discussion

Figure 6 shows an example of the splicing masks obtained with the different tested solutions on a subset of test images. It can easily be seen that how the visual quality of the results gets better as we progressively add the proposed modifications to our method.

To provide a better insight on the performance obtained by the proposed solution, Table 1 and Table 2 report the metrics obtained considering all tested scenarios in terms of detection and localization. For completeness, we also include in Figure 7 the ROC and P/R curves obtained with the different splicing sizes. It is possible to notice that our final method is significantly better at detecting and localizing small anomalies than the baselines. Moreover, the combined modifications to the Deep SVDD classifier lead to an increased anomaly detection and localization performance. Applying only one modification did not lead to better performance compare to the unchanged implementation of Deep SVDD, but by applying all of them we can achieve a significant improvement in all the reported metrics. Finally, Figure 8 shows the histograms of anomaly scores extracted from pristine and forged patches using SatSVDD-v3, SatSVDD-v4, Deep SVDD[21] with the baseline [17] and Deep SVDD[21] with the proposed detection function.

The proposed detection function makes the pristine and forged image scores distributions more distinguishable,

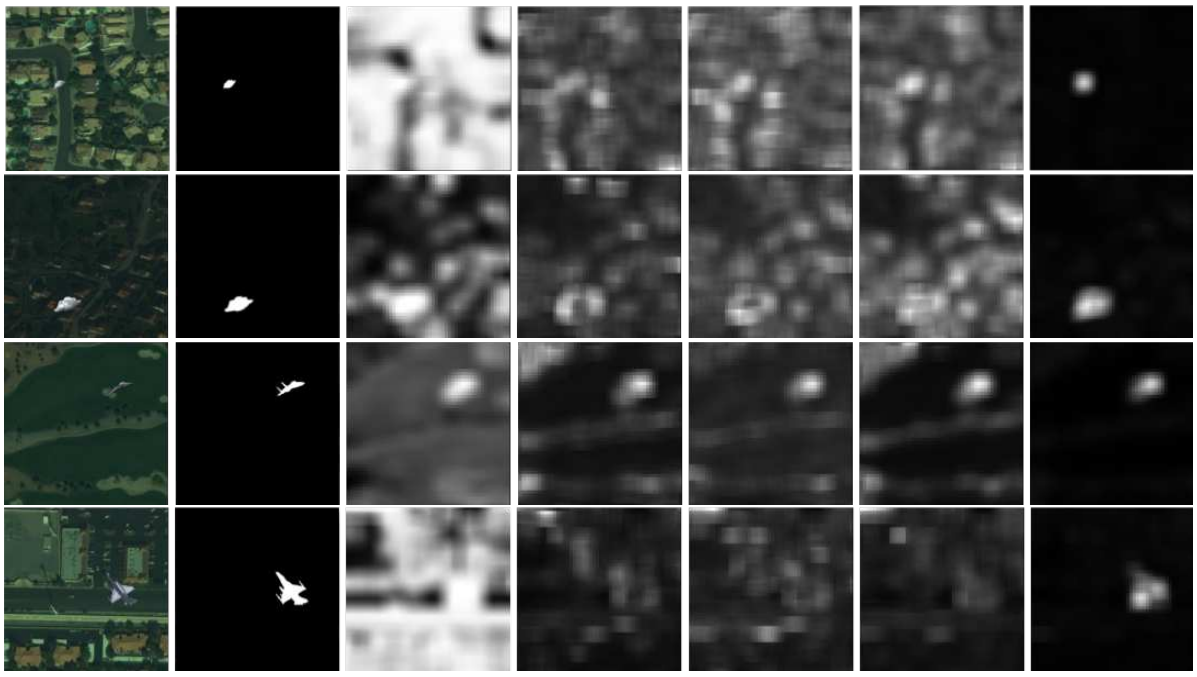


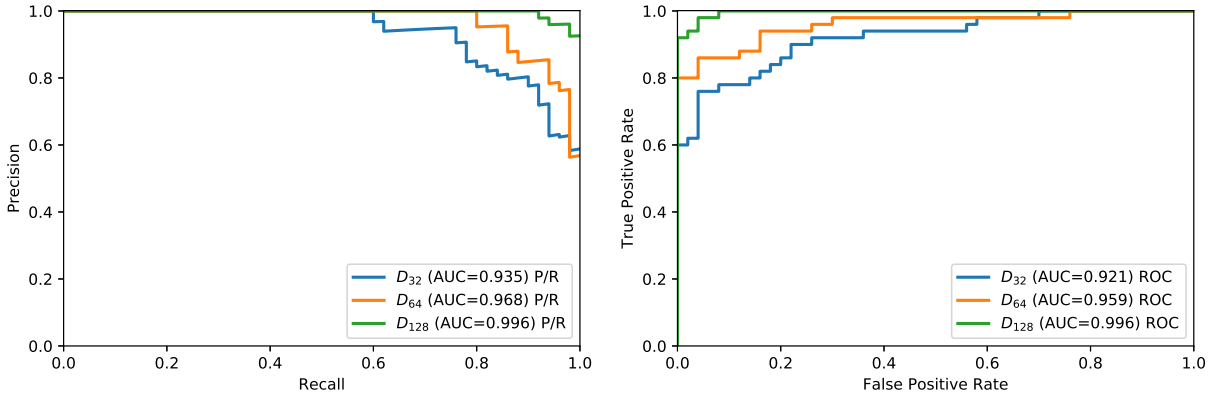
Figure 6: From left to right for each row: input image, splicing mask, baseline method [17], Deep SVDD [21], SatSVDD-v1, SatSVDD-v2, and SatSVDD-v3.

Table 1: AUC scores (%) for the detection task (ROC and P/R metrics). The subscript denotes the manipulation size. Results that surpass all competing methods are **bold**. Our final proposed model, SatSVDD-v4, outperforms all previous approaches.

	Yarlagadda *et al*. [17]	Ruff *et al*. [21]	SatSVDD-v1	SatSVDD-v2	SatSVDD-v3	SatSVDD-v4
ROC_{32}	77.0	64.7	67.7	69.0	88.3	**92.1**
ROC_{64}	89.3	69.2	67.7	72.4	95.1	**95.9**
ROC_{128}	94.2	86.9	86.4	81.0	99.5	**99.6**
P/R_{32}	79.3	61.3	61.8	65.6	90.4	**93.5**
P/R_{64}	92.3	64.6	62.5	66.2	96.1	**96.8**
P/R_{128}	96.0	86.8	82.4	79.2	99.5	**99.6**

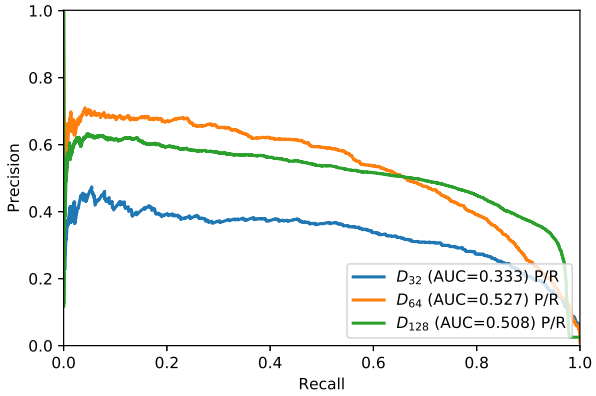
Table 2: AUC scores (%) for the localization task (ROC and P/R metrics). The subscript denotes the manipulation size. Results that surpass all competing methods are **bold**. Our final proposed model, SatSVDD-v4, outperforms almost all previous approaches.

	Yarlagadda *et al*. [17]	Ruff *et al*. [21]	SatSVDD-v1	SatSVDD-v2	SatSVDD-v3	SatSVDD-v4
ROC_{32}	91.2	94.2	97.4	97.5	**99.7**	**99.7**
ROC_{64}	95.1	86.9	95.5	90.9	**99.3**	**99.3**
ROC_{128}	96.1	94.2	93.6	92.7	**99.6**	**99.6**
P/R_{32}	11.4	3.7	11.0	8.2	**33.3**	**33.3**
P/R_{64}	45.1	4.8	18.9	6.7	**52.7**	**52.7**
P/R_{128}	**67.0**	30.8	33.3	25.4	50.8	50.8

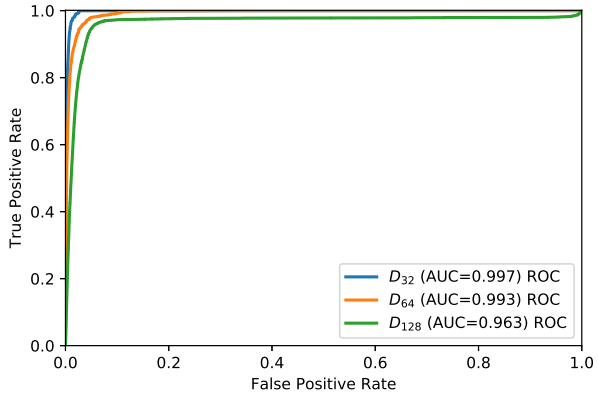


(a) Detection P/R curve

(b) Detection ROC curve



(c) Localization P/R curve



(d) Localization ROC curve

Figure 7: P/R and ROC curves for the anomaly detection and localization tasks

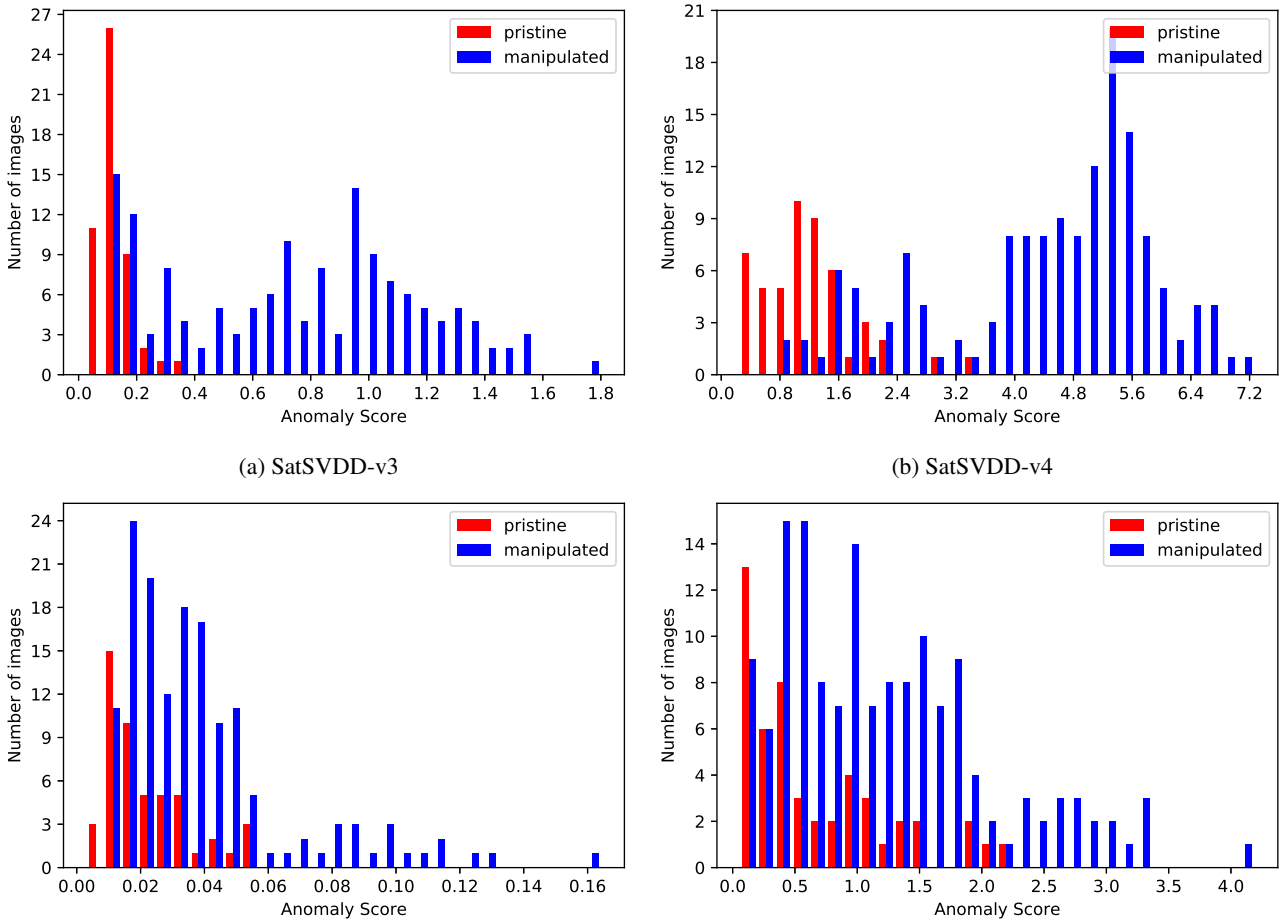
with a 17% decrease in the overlapping area (from 42 to 35 overlapping images). It can also be observed that in the overlapping area, the average anomaly scores of manipulated images became more distinguishable compared to the average anomaly score of non-manipulated images. The two bottom histograms in Figure 8 show that the proposed detection function does not improve the splicing detection if we simply use Deep SVDD. This due to the fact that Deep SVDD does not properly extract representative features of pristine patches. Therefore, it is not guaranteed to return a low anomaly score for patches that do not contain anomalies.

7. Conclusion

In this paper we propose a method for overhead image forgery detection and localization based on a deep one-class classifier that works in an anomaly detection framework. Specifically, we inherit the autoencoder structure proposed in [17], and adapt it to the Deep SVDD framework, also proposing a series of additional enhancement steps. As a matter of fact, applying Amsgrad as an optimizer and using a smooth activation function between layers significantly improved forgery detection performance. Moreover, the method proposed in the paper outperforms all previous work by a large margin in small to medium sized forgeries. Future work will be devoted to study the generalization capability of the proposed approach to different kinds of satellite imagery, as well as to different overhead image datasets.

Acknowledgement

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. 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 DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.



(a) SatSVDD-v3

(b) SatSVDD-v4

(c) Deep SVDD with the detection function used in [17].

(d) Deep SVDD with the proposed detection function.

Figure 8: Anomaly score histograms with different detection function. Note that SatSVDD-v3 uses the detection function from [17], whereas SatSVDD-v4 uses our proposed detection function.

References

[1] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing*, pp. 3226–3229, July 2017. Fort Worth, TX.

[2] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," *Proceedings of IEEE International Conference on Computer Vision*, pp. 3961–3969, Dec. 2015. Santiago, Chile.

[3] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. Salt Lake City, UT.

[4] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv:1812.08685v1*, Mar. 2018.

[5] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, Nov. 2018. Auckland, New Zealand.

[6] P. Tucker, "The Newest AI-Enabled Weapon: 'Deep-Faking' Photos of the Earth." `https://www.defenseone.com/technology/2019/03/next-phase-ai-deep-faking-whole-world-and-china-ahead/155944/`.

[7] A. E. Kramer, "Russian Images of Malaysia Airlines Flight 17 Were Altered, Report Finds." `https://www.nytimes.com/2016/07/16/world/europe/malaysia-airlines-flight-17-russia.html`.

[8] D. Byrd, "Fake image of Diwali still circulating." `https://earthsky.org/earth/fake-image-of-india-during-diwali-versus-the-real-thing`.

[9] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital im-

age and video forensics," *ACM Computing Surveys*, vol. 43, pp. 26:1–26:42, Oct. 2011.

[10] V. Schetinger, M. Iuliani, A. Piva, and M. M. Oliveira, "Digital image forensics vs. image composition: An indirect arms race," *arXiv:1601.03239*, Jan. 2016.

[11] S. K. Yarlagadda, D. Güera, D. M. Montserrat, F. Zhu, E. J. Delp, P. Bestagini, and S. Tubaro, "Shadow removal detection and localization for forensics analysis," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2677–2681, May 2019. Brighton, United Kingdom.

[12] D. Moreira, A. Bharati, J. Brogan, A. Pinto, M. Parowski, K. W. Bowyer, P. J. Flynn, A. Rocha, and W. J. Scheirer, "Image provenance analysis at scale," *IEEE Transactions on Image Processing*, vol. 27, pp. 6109–6123, Dec. 2018.

[13] H. Liu, C. Li, S. Ge, S. Zhao, and X. Jin, "Generative image inpainting with neural features," *Proceedings of the International Conference on Internet Multimedia Computing and Service*, pp. 23:1–23:5, Aug. 2018. Nanjing, China.

[14] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," *Proceedings of thee IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1316–1324, June 2018. Salt Lake City, UT.

[15] P. Zhou, B. Chen, X. Han, M. Najibi, and L. S. Davis, "Generate, segment and replace: Towards generic manipulation segmentation," *arXiv:1811.09729v2*, Mar. 2019.

[16] A. V. Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," *arXiv:1805.09512*, May 2018.

[17] S. Kalyan Yarlagadda, D. Güera, P. Bestagini, F. Zhu, S. Tubaro, and E. Delp, "Satellite image forgery detection and localization using gan and one-class classifier," *Proceedings of the IS&T International Symposium on Electronic Imaging*, vol. 2018, pp. 214–1–214–9, Feb. 2018. Burlingame, CA.

[18] E. R. Bartusiak, S. Kalyan Yarlagadda, D. Güera, F. M. Zhu, P. Bestagini, S. Tubaro, and E. J. Delp, "Splicing detection and localization in satellite imagery using conditional gans," *Proceedings of the IEEE International Conference on Multimedia Information Processing and Retrieval*, pp. 91–96, Mar. 2019. San Jose, CA.

[19] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, pp. 259–263, Mar. 2017.

[20] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based cnn features," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1855–1864, July 2017. Honolulu, HI.

[21] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," *Proceedings of the International Conference on Machine Learning*, vol. 80, pp. 4393–4402, July 2018. Stockholm, Sweden.

[22] M. Barni, A. Costanzo, and L. Sabatini, "Identification of cut&paste tampering by means of double-JPEG detection and image segmentation," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1687–1690, May 2010. Paris, France.

[23] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," *Proceedings of the IEEE International Workshop on Information Forensics and Security*, Nov. 2015. Rome, Italy.

[24] M. Kirchner and T. Gloe, "Forensic camera model identification," in *Handbook of Digital Forensics of Multimedia Data and Devices*, John Wiley & Sons, Ltd, 2015.

[25] L. Ali, T. Kasetkasem, F. G. Khan, T. Chanwimaluang, and H. Nakahara, "Identification of inpainted satellite images using evalutionary artificial neural network (EANN) and k-nearest neighbor(KNN) algorithm," *Proceedings of the IEEE International Conference of Information and Communication Technology for Embedded Systems*, pp. 1–6, May 2017. Chonburi, Thailand.

[26] A. T. S. Ho and W. M. Woon, "A semi-fragile pinned sine transform watermarking system for content authentication of satellite images," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, pp. 1–4, July 2005.

[27] R. Chalapathy, E. Zare Borzeshi, and M. Piccardi, "An investigation of recurrent neural architectures for drug name recognition," *Proceedings of the International Workshop on Health Text Mining and Information Analysis*, pp. 1–5, Nov. 2016. Honolulu, HI.

[28] M. Nadeem, O. Marshall, S. Singh, X. Fang, and X. Yuan, "Semi-supervised deep neural network for network intrusion detection," *Proceedings of the Kennesaw State University Conference on Cybersecurity, Research and Practice*, Oct. 2016. Kennesaw, GA.

[29] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv:1802.06360*, February 2018.

[30] D. M. Tax and R. P. Duin, "Support vector data description," *Machine Learning*, vol. 54, pp. 45–66, January 2004.

[31] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, pp. 15:1–15:58, July 2009.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the International Conference on Learning Representations*, May 2015. San Diego, CA.

[33] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *Proceedings of the International Conference on Learning Representations*, 2018. Vancouver, Canada.

[34] D. Holden, "Robust solving of optical motion capture data by denoising," *ACM Transactions on Graphics*, vol. 37, pp. 165:1–165:12, July 2018.

[35] J. Lucas, S. Sun, R. Zemel, and R. Grosse, "Aggregated momentum: Stability through passive damping," *Proceedings of the International Conference on Learning Representations*, May 2019. New Orleans, LA.

[36] G. Sterpu, C. Saam, and N. Harte, "Attention-based audio-visual fusion for robust automatic speech recognition," *Proceedings of the International Conference on Multimodal Interaction*, pp. 111–115, Oct. 2018. Boulder, CO.

[37] G. Becigneul and O.-E. Ganea, "Riemannian adaptive optimization methods," *Proceedings of the International Conference on Learning Representations*, May 2019. New Orleans, LA.

[38] J. Chen and Q. Gu, "Closing the generalization gap of adaptive gradient methods in training deep neural networks," *arXiv:1806.06763*, June 2018.

[39] J. Li, C. Fang, and Z. Lin, "Lifted proximal operator machines," *arXiv:1811.01501*, Nov. 2018.

[40] D. Güera, F. Zhu, S. K. Yarlagadda, S. Tubaro, P. Bestagini, and E. J. Delp, "Reliability map estimation for cnn-based camera model attribution," pp. 964–973, Mar. 2018. Lake Tahoe, NV.

[41] S. S. Du, W. Hu, and J. D. Lee, "Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced," *Advances in Neural Information Processing Systems*, pp. 384–395, Dec. 2018. Montréal, Canada.

[42] T. Le and D. Q. Phung, "When can neural networks learn connected decision regions?," *arXiv:1901.08710*, Jan. 2019.

[43] Y. S. Soh and V. Chandrasekaran, "Fitting tractable convex sets to support function evaluations," *arXiv:1901.05331*, Mar. 2019.

[44] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in Neural Information Processing Systems*, pp. 4800–4810, Dec. 2018. Montréal, Canada.