Manipulation Detection in Satellite Images Using Vision Transformer

János Horváth, Sriram Baireddy, Hanxiang Hao, Daniel Mas Montserrat, Edward J. Delp
Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

Abstract

A growing number of commercial satellite companies provide easily accessible satellite imagery. Overhead imagery is used by numerous industries including agriculture, forestry, natural disaster analysis, and meteorology. Satellite images, just as any other images, can be tampered with image manipulation tools. Manipulation detection methods created for images captured by "consumer cameras" tend to fail when used on satellite images due to the differences in image sensors, image acquisition, and processing. In this paper we propose an unsupervised technique that uses a Vision Transformer to detect spliced areas within satellite images. We introduce a new dataset which includes manipulated satellite images that contain spliced objects. We show that our proposed approach performs better than existing unsupervised splicing detection techniques.

1. Introduction

The exponentially growing number of commercial satellites orbiting the Earth generate an enormous amount of imagery. A large variety of applications makes use of satellite imagery, including agricultural crop classification [19, 46], scene classification [10, 42], wildlife monitoring [14, 20], forest characterization [22, 33], meteorological analysis [31, 39], infrastructure levels assessment, building localization [17, 38], and soil moisture estimation [16, 18].

Popular image editing tools, such as GIMP or Photoshop, can easily alter or manipulate satellite images. Figure 1 shows some examples of manipulated satellite images. Advances in machine learning have simplified the process of manipulating images and even creating highly-realistic "fake" images [26, 56]. Several altered satellite images have been used to spread misinformation on the Internet. Some examples include the Malaysian flight incident over Ukraine [29], the images of fake Chinese bridges [15], the Australian bushfires [43], and the Diwali Festival nighttime flyovers over India [5].

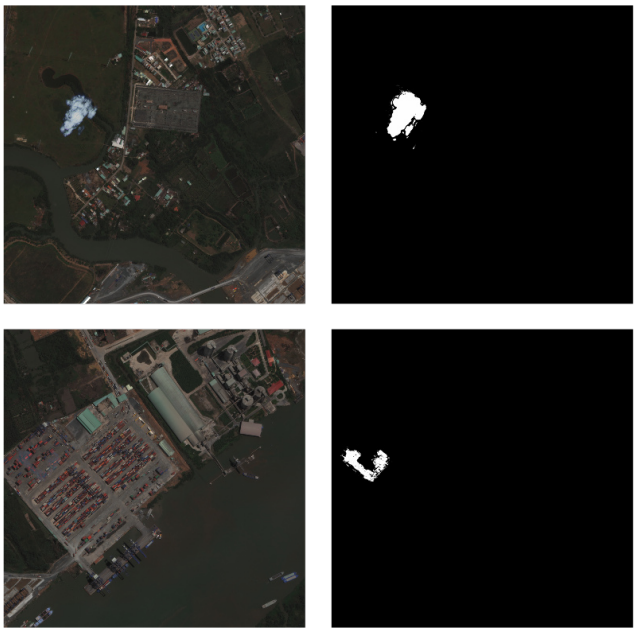

Figure 1. Examples of satellite images: manipulated images (left) and their corresponding ground-truth of the spliced area (right).

Several image manipulation methods have been proposed. Some image manipulation techniques include splicing [11], blending objects created by Generative Adversarial Networks (GANs) [44], and copy-move methods [51]. Various methods have been proposed to detect alterations in images captured by consumer cameras [1, 3, 9, 45]. These techniques tend to fail in detecting alterations in satellite images due to the difference in the image types. These differences include acquisition sensors, compression schemes, color channels, and post-processing operations like orthorectification. Despite the growing number of techniques developed to detect manipulations in satellite imagery, it remains an open problem.

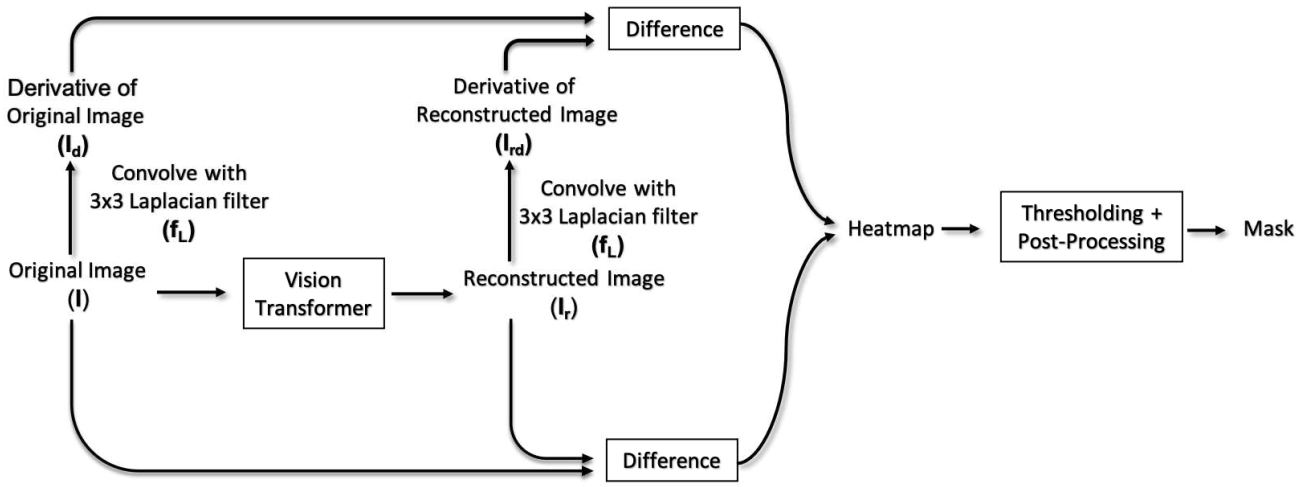

Figure 2. Our proposed method for manipulation detection.

2. Related Work

Several methods for detecting alterations in natural images have been described, including finding manipulated pixels in images by using neural networks with domain adaptation [8], using the Radon transform of resampled features and a deep learning classifier [4], and using saturation cues [35]. Other approaches focus on finding double-JPEG compression artifacts [2, 53] or detecting and localizing spliced objects in images using unsupervised approaches [1, 7, 9]. There has been work in detecting spliced regions by using local features in the image and differentiating between the original or splicing images using expectation-maximization [7]. Cozzolino *et al*. [9] developed a technique leveraging the fact that each camera model leaves a unique digital fingerprint, known as a "noiseprint". Instead of finding general model-related artifacts, Bammey *et al*. [1] designed a model focusing specifically on detecting camera demosaicing artifacts. A demosaicing artifact is a repeated pattern in the cycle of 2×2 pixel block. It is caused by the reconstruction of a full color image from the incomplete color samples in digital image process. This repeated pattern will be different between the manipulated region and original image area, which can be detected by the proposed DemosaicingNet [1]. Most of these techniques will fail on satellite images due to the different acquisition process between images captured by consumer cameras and satellites. These differences include different sensor technologies and post-processing steps such as the compression scheme and radiometric corrections.

Several methods for detecting manipulations in satellite images been proposed. These detection techniques are based on more traditional techniques such as watermarking [24] or machine learning approaches that are unsupervised [25, 27, 28] and supervised [26, 41]. In [41] a conditional GAN was used to localize spliced objects in satellite images. In [26] the Nested Attention U-Net for localizing spliced areas in overhead images was described. The authors in [37] propose an authentication protocol for secure satellite image data transfer. While supervised methods tend to detect and localize spliced objects better than unsupervised approaches, they require both manipulated and original data during training.

For developing our method we consider the unsupervised scenario where no manipulated data is available during training. The work introduced in [28] extracts and encodes patches from the input images into a lower dimensional latent space. This encoding is used by a one-class support vector machine (SVM) to determine if a patch contains a manipulation or not. Sat-SVDD, presented in [25], is a modified Support Vector Data Description (SVDD) [48], meaning it is a one-class classifier that detects spliced objects in satellite images. The Sat-SVDD input is patches extracted from the input image which are then encoded into a vector space within a hyper-sphere. During inferencing, the patches whose vector representation are placed outside the hyper-sphere are considered "altered" patches. The authors in [27] use a deep belief network (DBN) [23] constructed by two layers of restricted Boltzmann machines (RBM) [47] following uniform distribution. The Deep Belief Network is trained as an autoencoder which encodes and decodes the input patches. The reconstruction error is then used to detect whether the patch contains alterations or not. The authors in work [36] use an ensemble of auto-regressive networks to detect forgeries in satellite images. The ensemble predicts the probability of whether a pixel is manipulated or not. This latest method performed better than the previously mentioned approaches.

In this paper we describe a splicing detection method using Vision Transformer [13] and morphological filters [57].

In the past Transformers were used mainly for natural language processing [55]. Recently, Transformer methods were developed for images [6, 13] Image-GPT [6] was introduced last year for unsupervised low resolution image generation and image classification. Image-GPT is an autoregressive network which aims to predict pixels without a complete understanding of the 2D input image structure. This GPT model achieved high accuracy on the CIFAR10 dataset. Another model developed for image classification is the Vision Transformer [13]. We will present an overview of Vision Transformer below in Section 3. We will also introduce a morphological filter for binary images and use it in our post-processing step.

3. Proposed Method

In this section we will describe an unsupervised splicing detection method for satellite images that uses Vision Transformer (ViT). A block diagram of our proposed method is shown in Figure 2. In Section 3.1, we will provide some background description of the ViT proposed by Dosovitskiy *et al*. [13]. We extend ViT to an autoencoder-like structure for splicing detection in Section 3.2.

3.1. Vision Transformer

As proposed in [13], ViT uses a Transformer model [50] to replace convolution layers for image classification tasks. As shown in Figure 3 (left), ViT takes image patches as input. In our experiments, the original image size is 128×128 and patch size is 64×64. To reduce the dimensionality of the input image patches, linear projection is used: $T_i = W\hat{I}_i$, where $W \in \mathbb{R}^{D \times N}$ is a learnable linear mapping function, $\hat{I}_i \in \mathbb{R}^N$ is the flattened i-th image patch, and $T_i \in \mathbb{R}^D$ is the i-th image token [13] input to the Transformer. As proposed in [13], we prepend a learnable classification token $T_0 \in \mathbb{R}^R$ to the aforementioned image tokens before inputting them to the Transformer model. Transformers use self-attention modules to add the long range information contained in all of the input tokens [50]. However, the self-attention module is invariant to the input token order. To add positional information about the input patches to Transformer, a set of learnable positional embeddings [13] are used: $P_i \in \mathbb{R}^D$ for $i \in \{0, 1, ...\}$. These positional embeddings contain the unique position information for the different input tokens. After adding the positional embeddings to the input tokens, the position-aware tokens are provided to the Transformer. As proposed in [13], we only take the output from the classification token and pass it to a multi-layer perceptron (MLP) to output the probabilities of object classes.

3.2. Vision Transformer for Splicing Detection

Autoencoders have been successfully used for splicing detection [25, 27, 28]. They are trained to reconstruct images that do not contain manipulations. Then during testing, given a image with manipulated regions, the autoencoder will reconstruct the image using the information it learned from unmanipulated images. We can compare the difference between the input image containing manipulation and the image reconstructed by the autoencoder. Since the autoencoder learned to model the image distribution of the original images, the reconstructed image will be different from the original image in the manipulated areas. Given this, we design a reconstruction approach as shown in the right side of Figure 3 to replace the classification approach (*i.e*. blue dot-line region in Figure 3). We directly reshape the output from the MLP module to construct the output image I_r. To reduce the memory used by the self-attention module in the Transformer, we use the Linformer [54] to reduce the space complexity from the original Transformer used in [13]. For our image reconstruction task, we use smoothed L_1 loss as following:

$$\mathcal{L}_r(I, I_r) = \frac{1}{|I|}\sum_i^{|I|} \begin{cases} \frac{1}{2}(I(i) - I_r(i))^2 & |I(i) - I_r(i)| < 1 \\ |I(i) - I_r(i)| - \frac{1}{2} & \text{Otherwise} \end{cases}$$

At the inference stage, we input an image $\mathbf{I}$ into the trained Vision Transformer and obtain a reconstructed image $\mathbf{I_r}$ as seen in Figure 2. We also do a convolution for each channel of $\mathbf{I}, \mathbf{I_r}$ with a 3×3 Laplacian filter $\mathbf{f_L}$, obtaining two more images $\mathbf{I_d}, \mathbf{I_{rd}}$, where:

$$\mathbf{I_d} = \mathbf{I} \circledast \mathbf{f_L}$$

and

$$\mathbf{I_{rd}} = \mathbf{I_r} \circledast \mathbf{f_L}$$

The reason for using the Laplacian filter on the input and reconstructed images is that autoencoders have some difficulties when reconstructing the high frequency components of an image and the Laplacian filter acts an edge detector [32] that will highlight these high frequency components. We construct a heatmap from the difference of $\mathbf{I}, \mathbf{I_r}$ and from the difference of $\mathbf{I_d}, \mathbf{I_{rd}}$ by averaging them. Next, we threshold the heatmap to create a binary mask and use a post-processing stage to output a final mask that indicates the region detected as spliced.

3.3. Post-Processing with Morphological Filters

The post-processing consists of several morphological filters [21]. The goal of the post-processing is to decrease the number of false negatives and false positives. This can be achieved by filling in holes and removing small objects in the binary mask. There are efficient techniques which can fill holes in binary masks. We use the function *binary_fill_holes* from the SciPy library [52]. After filling the holes in the mask, we remove the small objects.

For removing small objects, there are filters specifically designed for this task: erosion and opening [21]. While both

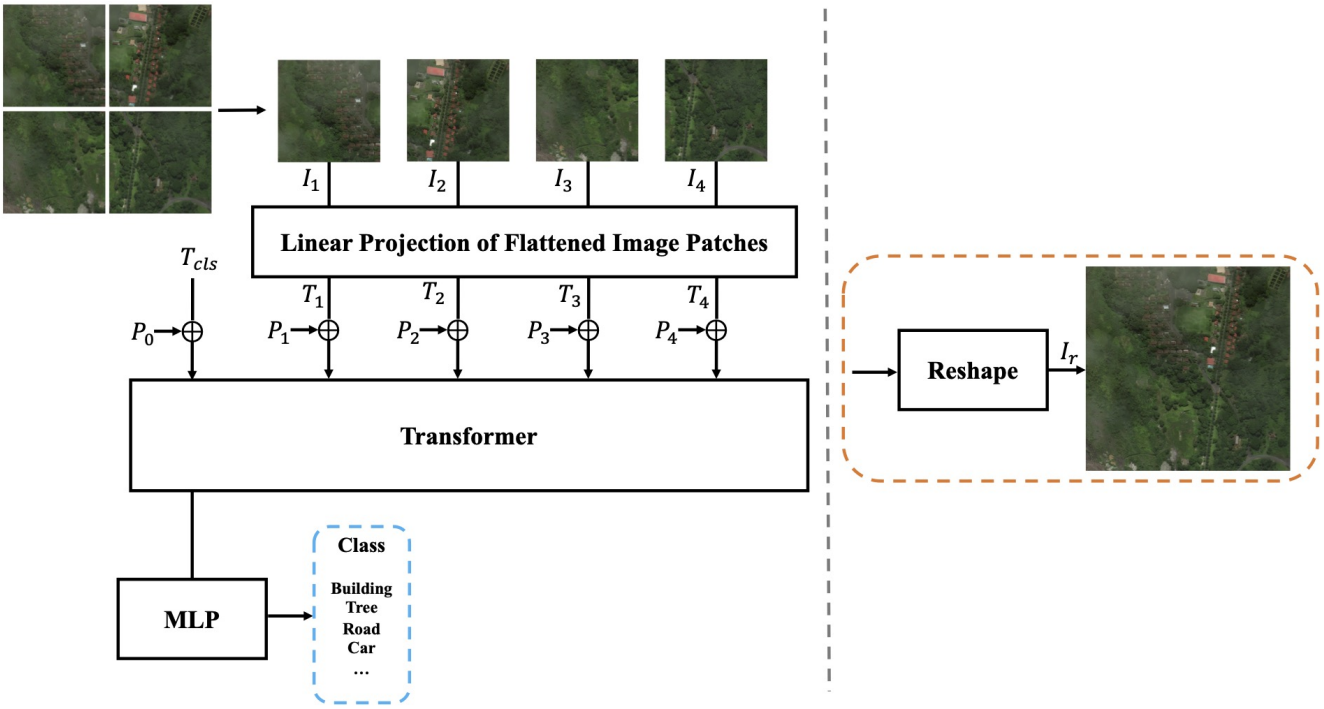

Figure 3. Vision Transformer (ViT) block diagram (left) for image classification. Vision Transformer image reconstruction (right).

of these filters remove small objects, they have the issue of also modifying larger objects. Erosion decreases the area of large objects, while opening destroys the fine detail of the boundary [57]. We propose **ErodeIsolated**, a morphological filter to erode smaller objects while leaving larger objects unchanged. It has two parameters, a and b, that are used to construct the structuring element:

$$\mathbf{f_{EI,a,b}} = \mathbf{ErodeIsolated(a, b)}$$

Consider the first example shown in Figure 4, where $a = 1$ and $b = 2$. The resulting structural element has a square shape with a side length of $2b + 1$ (in this case, 5). Within this structural element, we have an inner square with a side length of $2a + 1$ (in this case, 3). We keep the values in the inner square at 0 and the remaining values of the structural element at 1. By doing so, with **OR** ($\vee$) and **AND** ($\wedge$) operators as shown in Figure 4, we can remove small objects, while keeping the larger objects untouched.

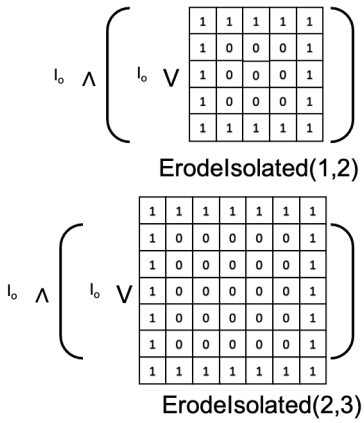

Figure 4. Two example of **ErodeIsolated** morphological filter.

Our post-processing technique is shown in Figure 5. First, we use a closing operation with a very small structuring element, in order to ensure that all pixels in a large object are connected. After that step we enter into a while loop. In this while loop we use a series of ErodeIsolated filters with different structuring elements. The goal is to erode small objects while leaving the larger objects untouched. We exit the while loop when the series of filters no longer improves the binary image. It is interesting to note that the while loop cannot iterate more than the number of ErodeIsolated filters inside the loop; in our case, this is no more than five.

4. Experimental Results

For evaluating the performance our proposed method we use two datasets in our experiments. The first dataset was introduced in [27], while the other dataset is new and will be discussed below. The dataset described in [27] is composed of satellite images of regions of Slovenia. The images have dimensions of 1000×1000 and were captured by the Sentinel-2 satellite [40]. We shall refer to this dataset as *Dataset 1*. We used 98 original images for training and

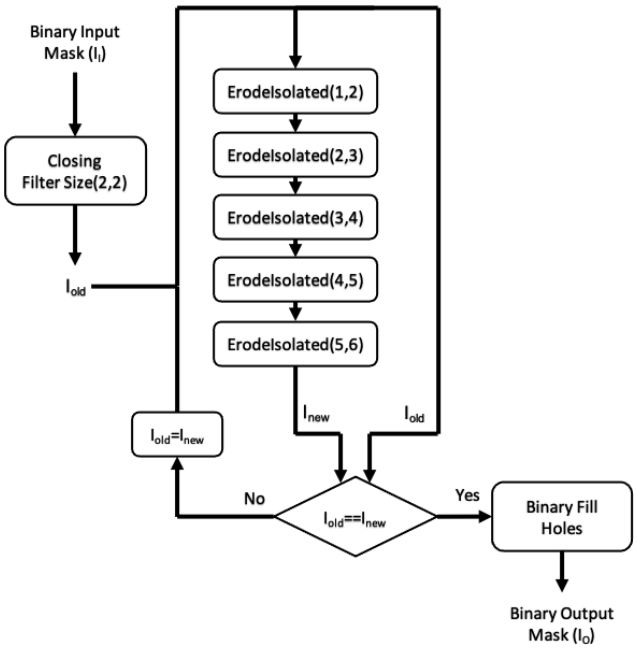

Figure 5. Proposed post-processing method.

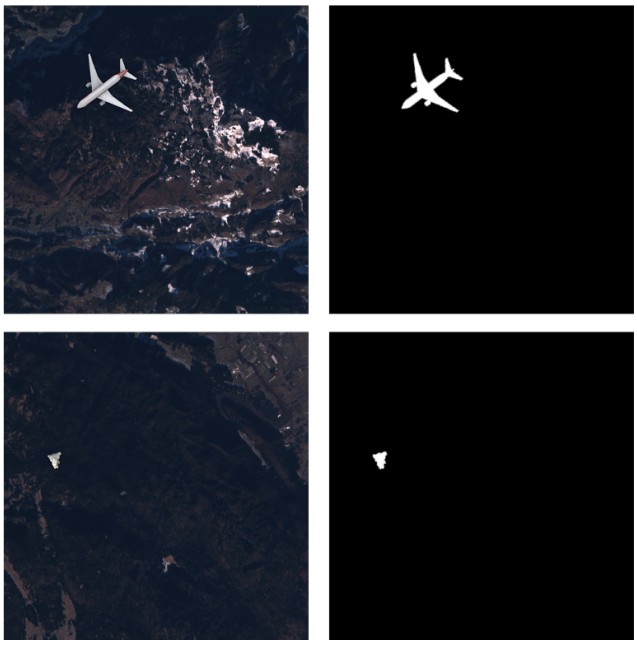

Figure 6. $Dataset1$: On the left are the manipulated images and on the right the corresponding ground-truth.

500 manipulated images with their corresponding ground-truth masks for testing. Each manipulated image contains one spliced object randomly selected from nineteen different objects such as drones, planes, and clouds. The objects are spliced into the images at different locations, rotation angles, and sizes including 16×16, 32×32, 64×64, 128×128, and 256×256 pixels. Some examples from *Dataset 1* are shown in Figure 6.

We also constructed a new dataset, *Dataset 2*, composed of satellite images captured by the WorldView-3 satellite from various locations such as coast, urban, and vegetation areas [49]. The resolution of these image varies from six to eight megapixels. We used 28 images for training and 859 manipulated images with their corresponding ground-truth masks for testing. Each manipulated image contains a spliced object extracted from images captured by a PlanetScope satellite [30]. The objects were spliced into the WorldView-3 images using several steps including multiple blending functions. Each image contains an object with sizes varying from several hundred pixels to several megapixels. Some examples from *Dataset 2* are shown in Figure 7.

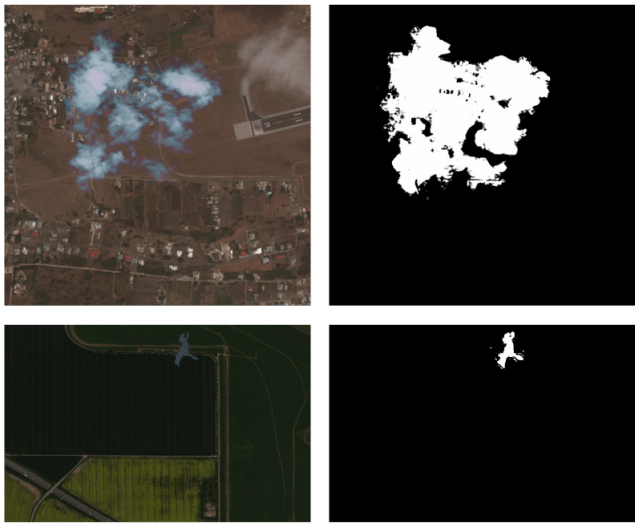

Figure 7. $Dataset2$: On the left are manipulated images and on the right the corresponding ground-truth.

We trained two Vision Transformers as autoencoders using the two datasets. We assumed that we do not know prior information of the spliced objects; thus, we used only original images during training. After training we tested our proposed technique on spliced images. We compared our method with previously introduced unsupervised splicing detection techniques, such as Splicebuster [7], NoisePrint [9], a Generative Ensemble of Gated PixelCNNs [36], and DemosaicingNet [1]. The output of these techniques is a heatmap. We thresholded the heatmaps in order to produce a binary mask as an output.

In order to evaluate the effectiveness of our proposed post-processing scheme, we compare different post-processing methods. First, as a baseline, we do not have any post-processing of the output mask; we refer to this method as "Vision Transformer" below. The next post-processing approach consists of opening and closing operations [57], both with a structuring element of ones in a 2×2 matrix,

as well as *binary_fill_holes* from the SciPy library [52]. We refer to this as "Vision Transformer with Post-Processing-V1". Finally, we use our proposed post-processing technique (shown in Figure 5), which we refer to as "Vision Transformer with Post-Processing-V2".

We used two evaluation metrics to characterize performance. These two metrics measure the similarity between the generated masks and the corresponding ground-truth masks. The first metric is the Dice Score, which is also known as the F1 score [12]. A high F1 score, indicates that there is no problem with false positives or false negatives. The F1 score is the harmonic mean of the Precision and Recall.

$$F1 = \frac{Precision * Recall}{2 * (Precision + Recall)}$$

or in an another form

$$F1 = \frac{2 * TP}{2 * TP + FP + FN}$$

where TP is the True Positive, FP is the False Positive and FN is the False Negative. Precision is the ratio of True Positive to the sum of True Positive and False Positive.

$$Precision = \frac{TP}{TP + FP}$$

Recall is the proportion of positive scores that have been incorrectly predicted.

$$Recall = \frac{TP}{TP + FN}$$

The second metric is the Jaccard Index, also known as intersection over union [34].

$$Jaccard\ Index\ (JI) = \frac{TP}{TP + FP + FN}$$

5. Results

For *Dataset 1* the spliced images can be grouped by the sizes of the spliced objects (*i.e.*, 16×16, 32×32, 64×64, 128×128, and 256×256 pixels). For *Dataset 2* the spliced objects size varies from image to image, thus we cannot divide it into further groups. Table 1 shows the results for *Dataset 1*. From this table we can see that all of the methods seem to detect larger spliced objects better than smaller objects. We also see that the Vision Transformer generates a better splicing detection mask than NoisePrint, the Gated PixelCNN Ensemble, and DemosaicingNet. Table 1 shows in the last two rows that the Vision Transformer output mask can be improved by post-processing. It also shows that if we use the **ErodeIsolated** filter we can produce a better splicing detection mask. The difference between the two post-processing schemes is significant, as shown in Table 1 and Table 2.

Table 2 shows the results using *Dataset 2*. From this table we can see that Vision Transformer generates a better splicing detection mask than NoisePrint and marginally better splicing detection mask than the Gated PixelCNN Ensemble and DemosaicingNet. Both post-processing schemes output a better splicing detection mask and using ErodeIsolated is beneficial.

Figure 8 presents two spliced images from each dataset with their corresponding ground-truth masks. It also shows the generated mask of our proposed method and the other techniques. By visually inspecting these examples, we can see that NoisePrint is a good splicing detection method but fails to detect some details of the spliced objects and produces false positives. SpliceBuster does well on larger objects, but fails to localize smaller objects correctly. DemosaicingNet fails to detect the manipulated regions in both datasets, especially for the upper two cases as shown in Figure 8. DemosaicingNet, which is designed for consumer cameras, is not able to provide an accurate detection. We conclude that Vision Transformer is better at detecting spliced objects in satellite images than NoisePrint, SpliceBuster, DemosaicingNet and the Gated PixelCNN Ensemble. Using post-processing on the binary splicing detection mask improves the detection performance with respect to the Jaccard Index and Dice Score. The use of the **ErodeIsolated** morphological filter further improves the performance. We show several examples in Figure 8 to compare the different post processing schemes.

6. Conclusion

In this paper, we introduce an unsupervised splicing detection technique for detecting spliced objects in overhead images. This technique uses Vision Transformer trained to localize manipulated areas. We evaluated the performance of our approach on two datasets. From the experiments the proposed method has better performance than previously introduced unsupervised splicing detection techniques. In the future we plan to investigate other Transformer architectures to improve the performance. We also plan to introduce more datasets for evaluating performance.

7. Acknowledgment

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the 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 or AFRL or the U.S.

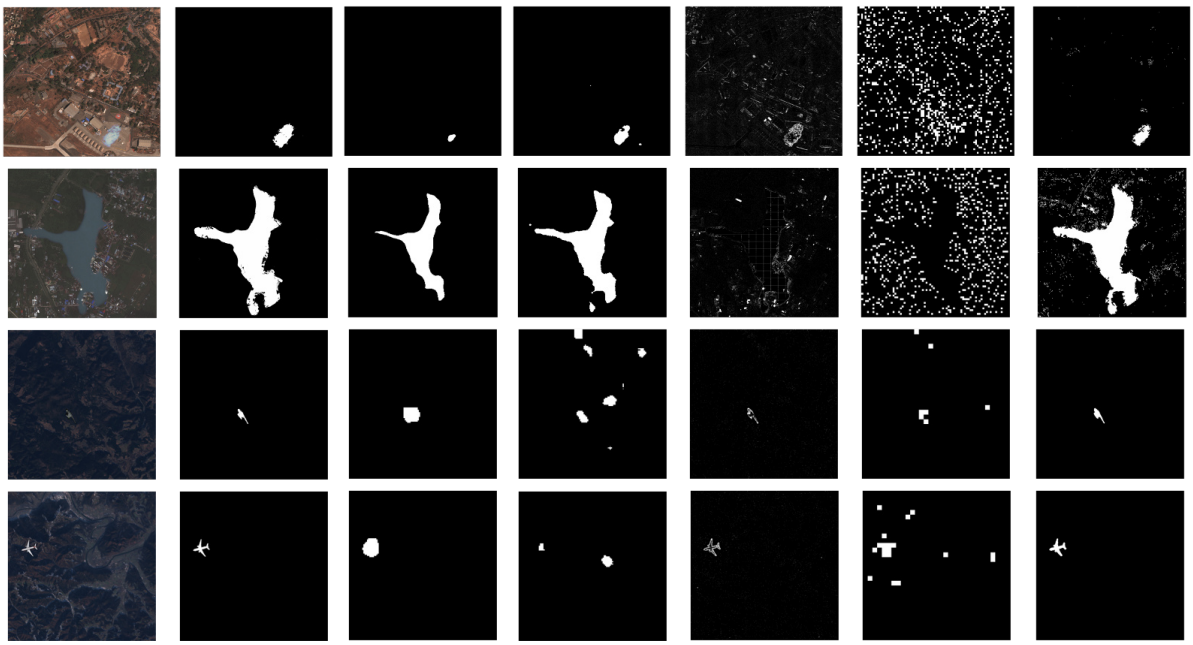

Figure 8. The spliced image, its corresponding ground-truth mask, detection mask generated with Noiseprint, SpliceBuster, Gated PixelCNN Ensemble, DemosaicingNet, Vision Transformer with Post-Processing-v2

Table 1. Results for *Dataset 1*, where "ViT" stands for Vision Transformer and "PP" stands for post-processing

Method	$F1_{16}$	$F1_{32}$	$F1_{64}$	$F1_{128}$	$F1_{256}$	JI_{16}	JI_{32}	JI_{64}	JI_{128}	JI_{256}
NoisePrint	0.000	0.001	0.066	0.148	0.174	0.000	0.001	0.042	0.096	0.111
SpliceBuster	0.001	0.012	0.094	0.360	0.504	0.000	0.006	0.049	0.219	0.337
Gated PixelCNN Ensemble	0.028	0.053	0.095	0.145	0.190	0.014	0.024	0.053	0.085	0.117
DemosaicingNet	0.004	0.013	0.057	0.120	0.181	0.002	0.007	0.029	0.064	0.100
ViT	0.129	0.304	0.413	0.433	0.358	0.077	0.206	0.283	0.296	0.231
ViT PP-v1	0.134	0.321	0.532	0.571	0.489	0.080	0.223	0.408	0.451	0.380
ViT PP-v2	**0.215**	**0.411**	**0.614**	**0.694**	**0.672**	**0.140**	**0.302**	**0.493**	**0.582**	**0.587**

Table 2. Results for *Dataset 2*, where "ViT" stands for Vision Transformer and "PP" stands for post-processing

Method	F1	JI
NoisePrint	0.066	0.037
SpliceBuster	0.337	0.202
Gated PixelCNN Ensemble	0.341	0.249
DemosaicingNet	0.022	0.011
ViT	0.345	0.254
ViT PP-v1	0.354	0.268
ViT PP-v2	**0.364**	**0.275**

Government.

Address all correspondence to Edward J. Delp, ace@ecn.purdue.edu.

References

[1] Q. Bammey, R. G. von Gioi, and J. M. Morel. An Adaptive Neural Network for Unsupervised Mosaic Consistency Analysis in Image Forensics. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 14182–14192, June 2020. Seattle, WA. 1, 2, 5

[2] 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*, pages 1687–1690, May 2010. Paris, France. 2

[3] B. Bayar and M. C. Stamm. A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer. *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, page 5–10, June 2016. Vigo, Galicia, Spain. 1

[4] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson. Detection and Localization

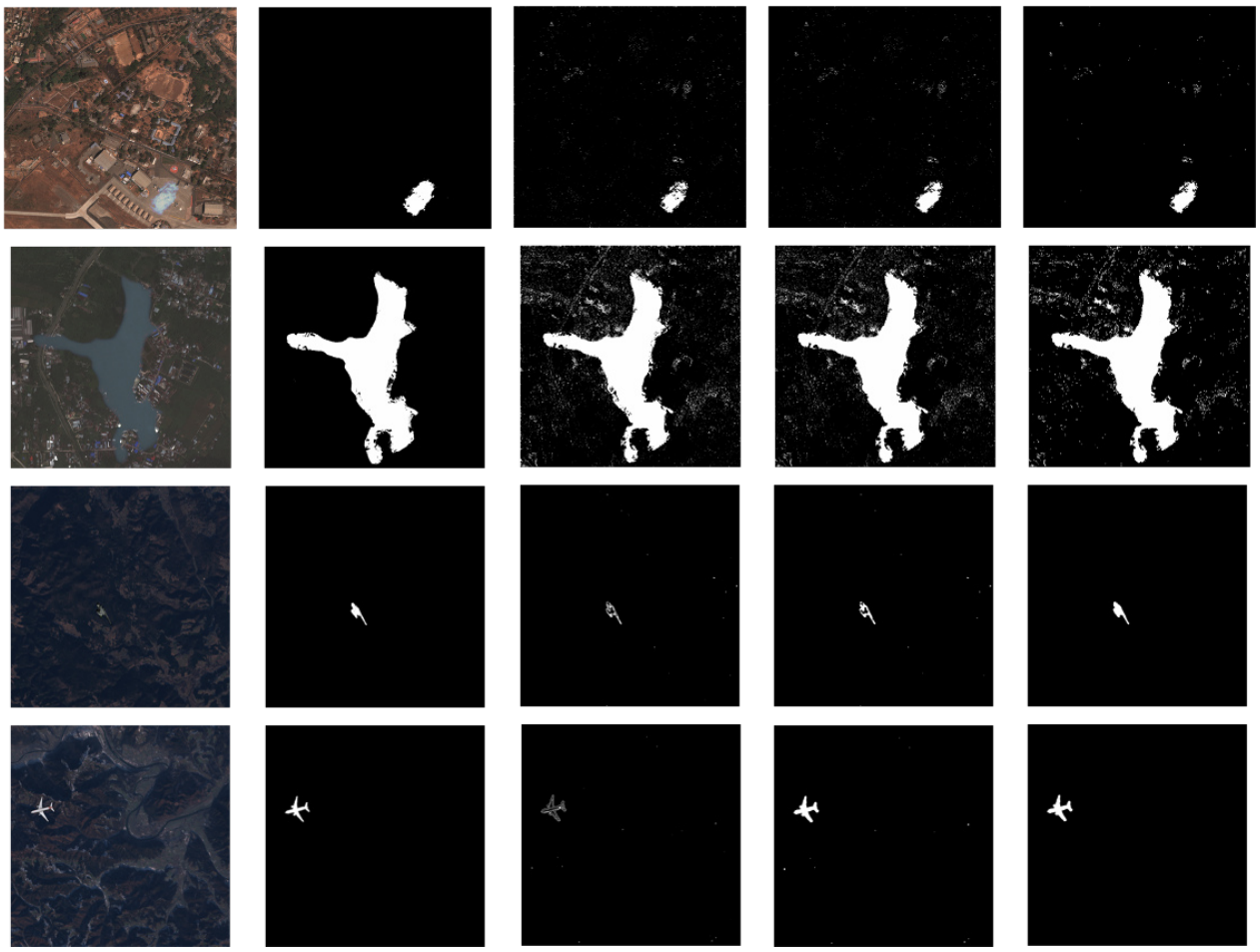

Figure 9. The spliced image, its corresponding ground-truth mask, detection mask generated with Vision Transformer, Vision Transformer with Post-Processing-v1, and Vision Transformer with Post-Processing-v2

of Image Forgeries Using Resampling Features and Deep Learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1881–1889, July 2017. Honolulu, HI. 2

[5] Deborah Byrd. Fake Image of Diwali Still Circulating. https://earthsky.org/earth/fake-image-of-india-during-diwali-versus-the-real-thing. 1

[6] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative Pretraining From Pixels. *Proceedings of the International Conference on Machine Learning*, pages 1691–1703, July 2020. Virtual. 3

[7] 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*, pages 1–6, November 2015. Rome, Italy. 2, 5

[8] D. Cozzolino, J. Thies, A. Rossler, C. Riess, M. Niessner, and L. Verdoliva. ForensicTransfer: Weakly-Supervised Domain Adaptation for Forgery Detection . *arXiv preprint arXiv:1812.02510*, December 2018. 2

[9] D. Cozzolino and L. Verdoliva. Noiseprint: A CNN-Based Camera Model Fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2020. 1, 2, 5

[10] A. Davari, V. Christlein, S. Vesal, A. Maier, and C. Riess. GMM Supervectors for Limited Training Data in Hyperspectral Remote Sensing Image Classification. *Proceedings of the International Conference on Computer Analysis of Images and Patterns*, pages 296–306, July 2017. Ystad, Sweden. 1

[11] T. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7):1182–1194, July 2013. 1

[12] L. R. Dice. Measures of the Amount of Ecologic Association Between Species . *Ecology*, 26(3):297–302, July 1945. 6

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, Matthias M., G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[14] I. Duporge, O Isupova, S. Reece, D. Macdonald, and T. Wang. Using Very High-Resolution Satellite Imagery and Deep Learning to Detect and Count African Elephants in

Heterogeneous Landscapes. *Remote Sensing in Ecology and Conservation*, December 2020. 1

[15] J. Edwards. China Uses GAN Technique to Tamper With Earth Images. https://earthsky.org/earth/fake-image-of-india-during-diwali-versus-the-real-thing. 1

[16] N. Efremova, D. Zausaev, and G. Antipov. Prediction of Soil Moisture Content Based on Satellite Data and Sequence-to-Sequence Networks. *arXiv preprint arXiv:1907.03697*, June 2019. 1

[17] A. Femin and K. S. Biju. Accurate Detection of Buildings from Satellite Images Using CNN. *Proceedings of the International Conference on Electrical, Communication, and Computer Engineering*, pages 1–5, June 2020. Istanbul, Turkey. 1

[18] M. Foucras, M. Zribi, and A. Kallel. Soil Moisture Estimation at 500m using Sentinel-1: Application to African Sites. *Proceedings of the International Conference on Advanced Technologies for Signal and Image Processing*, pages 1–5, September 2020. Sousse, Tunisia. 1

[19] H. Gao, C. Wang, G. Wang, Q. Li, and J. Zhu. A New Crop Classification Method Based on the Time-Varying Feature Curves of Time Series Dual-Polarization Sentinel-1 Data Sets. *IEEE Geoscience and Remote Sensing Letters*, 17(7):1183–1187, October 2020. 1

[20] E. Guirado, S. Tabik, M. L. Rivas, D. Alcaraz-Segura, and F. Herrera. Whale Counting in Satellite and Aerial Images with Deep Learning. *Scientific Reports*, 9:14259–14259, October 2019. 1

[21] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image Analysis Using Mathematical Morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):532–550, April 1987. 3

[22] E. Helmer, N. R. Goodwin, V. Gond, C. M. Souza, Jr., and G. P. Asner. Characterizing Tropical Forests with Multispectral Imagery. In Prasad S. Thenkaibail, editor, *Land Resources: Monitoring, Modeling and Mapping*, volume 2, pages 367–396. CRC Press, Boca Raton, Florida, 2015. 1

[23] G. E. Hinton, S. Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets . *Neural Computer*, 18(7), July 2006. 2

[24] 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*, pages 1–4, July 2005. Seoul, South Korea. 2

[25] J. Horvath, D. Guera, S. K. Yarlagadda, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp. Anomaly-Based Manipulation Detection in Satellite Images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 62–71, June 2019. Long Beach, CA. 2, 3

[26] J. Horvath, D. Mas Montserrat, and E. J. Delp. Nested Attention U-Net: A Splicing Detection Method for Satellite Images. *Proceedings of the International Conference on Pattern Recognition*, pages 516–529, January 2021. Virtual. 1, 2

[27] J. Horváth, D. Mas Montserrat, H. Hao, and E. J. Delp. Manipulation Detection in Satellite Images Using Deep Belief Networks . *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2832–2840, June 2020. Seattle, WA. 2, 3, 4

[28] S. K. Yarlagadda, D. Güera, P. Bestagini, F. Zhu, S. Tubaro, and E. J. Delp. Satellite Image Forgery Detection and Localization Using GAN and One-Class Classifier . *Proceedings of the IS&T International Symposium on Electronic Imaging*, pages 214–1–214–9, February 2018. Burlingame, CA. 2, 3

[29] 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. 1

[30] Planet Labs. Planet Scope. https://sentinel.esa.int/web/sentinel/missions. 5

[31] V. Lebedev, V. Ivashkin, I. Rudenko, A. Ganshin, A. Molchanov, S. Ovcharenko, R. Grokhovetskiy, I. Bushmarinov, and D. Solomentsev. Precipitation Nowcasting with Satellite Imagery. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2680–2688, August 2019. Anchorage, AK. 1

[32] J.S.J. Lee, L.G. Shapiro, and R. M. Haralick. Morphologic Edge Detection. *Proceedings of the International Conference on Pattern Recognition*, pages 369–373, October 1986. Paris, France. 3

[33] J. H. Lee, J. T. S. Sumantyo, M. M. Waqar, and J. H. Kim. Analysis of Forest Loss by Sentinel-1 SAR Time Series. *Proceedings of the International Conference on Information and Communication Technology Convergence*, pages 182–184, October 2020. Jeju, South Korea. 1

[34] M. Levandowsky and D. Winter. Distance Between Sets . *Nature*, 234:34–35, November 1971. 6

[35] S. McCloskey and M. Albright. Detecting GAN-Generated Imagery Using Saturation Cues . *Proceedings of the IEEE International Conference on Image Processing*, pages 4584–4588, September 2019. Taipei, Taiwan. 2

[36] D. Mas Montserrat, J. Horvath, S. K. Yarlagadda, F. Zhu, and E. J. Delp. Generative Autoregressive Ensembles for Satellite Imagery Manipulation Detection . *Proceedings of the IEEE International Workshop on Information Forensics and Security*, pages 1–6, December 2020. Virtual. 2, 5

[37] A. Murtaza and L. Jianwei. A Simple, Secure and Efficient Authentication Protocol for Real-Time Earth Observation Through Satellite. *Proceedings of the International Bhurban Conference on Applied Sciences and Technology*, pages 822–830, January 2018. Islamabad, Pakistan. 2

[38] B. Oshri, A. Hu, P. Adelson, X. Chen, P. Dupas, J. Weinstein, M. Burke, D. Lobell, and S. Ermon. Infrastructure Quality Assessment in Africa Using Satellite Imagery and Deep Learning. *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, pages 616–625, August 2018. London, United Kingdom. 1

[39] B. L. Pavuluri, R. S. Vejendla, P. Jithendra, T. Deepika, and S. Bano. Forecasting Meteorological Analysis Using Machine Learning Algorithms. *Proceedings of the International Conference on Smart Electronics and Communication*, pages 456–461, September 2020. Tiruchirappalli, India. 1

[40] Sentinel Program. Sentinel Missions. https://sentinel.esa.int/web/sentinel/missions. 4

[41] E. R. Bartusiak, S. K. 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*, pages 91–96, March 2019. San Jose, CA. 2

[42] K. Raiyani, T. Goncalves, L. Rato, P. Salgueiro, and J. Silva. Sentinel-2 Image Scene Classification: A Comparison between Sen2Cor and a Machine Learning Approach. *Remote Sensing*, 13(2):300, January 2021. 1

[43] G. Rannard. Australia fires: Misleading maps and pictures go viral. https://www.bbc.com/news/blogs-trending-51020564. 1

[44] C. X. Ren, A. Ziemann, J. Theiler, and A. M.S. Durieux. Deep Snow: Synthesizing Remote Sensing Imagery with Generative Adversarial Nets. *arXiv preprint arXiv:2005.08892*, May 2020. 1

[45] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein. Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics. *ACM Computing Surveys*, 43(4):26:1–26:42, October 2011. 1

[46] M. Rußwurm, S. Lefèvre, and M. Körner. BreizhCrops: A Satellite Time Series Dataset for Crop Type Identification. *arXiv preprint arXiv:1905.11893*, May 2019. 1

[47] P. Smolensky. *Information Processing in Dynamical Systems: Foundations of Harmony Theory* . MIT Press, Cambridge, MA, 1986. 2

[48] D. M. J. Tax and R. P. W. Duin. Support Vector Data Description. *Machine Learning*, 54(1):45–66, January 2004. 2

[49] Defense Innovation Unit. xView2. https://www.xview2.org. 5

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All You Need. *Proceedings of the International Conference on Neural Information Processing Systems*, pages 6000–6010, December 2017. Long Beach, CA. 3

[51] L. Verdoliva. Media Forensics and DeepFakes: an overview. *arXiv preprint arXiv:2001.06564*, January 2020. 1

[52] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C J Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, February 2020. 3, 6

[53] Q. Wang and R. Zhang. Double JPEG Compression Forensics Based on a Convolutional Neural Network. *EURASIP Journal on Information Security*, 2016(23), October 2016. 2

[54] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-Attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*, June 2020. 3

[55] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. HuggingFace's Transformers: State-of-the-Art Natural Language Processing. *arXiv preprint arXiv:1910.03771*, June 2020. 3

[56] X. Zhou, S. Huang, B. Li, Y. Li, J. Li, and Z. Zhang. Text Guided Person Image Synthesis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3658–3667, June 2019. Long Beach, CA. 1

[57] X. Zhuang and R. M. Haralick. Morphological Structuring Element Decomposition . *Computer Vision, Graphics, and Image Processing*, 35:370–382, April 1986. 2, 4, 5