APSIPA Transactions on Signal and Information Processing, 2024, 13, e200

Original Paper

Geo-DefakeHop: High-Performance Geographic Fake Image Detection

Hong-Shuo Chen[1]*, Kaitai Zhang[1], Shuowen Hu[2], Suya You[2] and C.-C. Jay Kuo[1]

[1] *University of Southern California, Los Angeles, California, USA*
[2] *DEVCOM Army Research Laboratory, Adelphi, Maryland, USA*

ABSTRACT

A robust fake satellite image detection method, called Geo-DefakeHop, is proposed in this work. Geo-DefakeHop is developed based on the parallel subspace learning (PSL) methodology. PSL maps the input image space into several feature subspaces using multiple filter banks. By exploring response differences of different channels between real and fake images for filter banks, Geo-DefakeHop learns the most discriminant channels based on the validation dataset, uses their soft decision scores as features, and ensemble them to get the final binary decision. Geo-DefakeHop offers a light-weight high-performance solution to fake satellite images detection. The model size of Geo-DefakeHop ranges from 0.8K to 62K parameters depending on different hyper-parameter settings. Experimental results show that Geo-DefakeHop achieves F1-scores higher than 95% under various common image manipulations such as resizing, compression and noise corruption.

Keywords: Geo Artificial Intelligence (GeoAI), Parallel Subspace Learning (PSL), Generative Adversarial Network (GAN).

*Corresponding author: Hong-Shuo Chen, hongshuo@usc.edu.

Received 26 October 2022; Revised 05 April 2023
ISSN 2048-7703; DOI 10.1561/116.00000072

1 Introduction

Artificial intelligence (AI) and deep learning (DL) techniques have made significant advances in recent years by leveraging more powerful computing resources and larger collected and labeled datasets. Geospatial science [14] and remote sensing [27] benefit from this development, involving increased application of AI to process data arising from cartography and geographic information science (GIS) more effectively. Despite countless advantages brought by AI, misinformation over the Internet, ranging from fake news [30] to fake images and videos [8, 15, 24, 31, 38, 40], poses a serious threat to our society.

Satellite images are utilized in various applications such as weather prediction [37], agriculture crops prediction [22], flood and fire control [23]. If one cannot determine whether a satellite image is real or fake, it would be risky to use it for decision making. Fake satellite images may have impacts on national security. For example, adversaries can create fake satellite images to hide military infrastructure and/or create fake ones to deceive others. Though government analysts could verify the authenticity of geospatial imagery leveraging other satellites or data sources, this would be prohibitively time intensive. It would be extremely difficult for the public to verify the authenticity of satellite images.

It becomes easier to generate realistically looking images due to the rapid growth of generative adversarial networks (GANs). There are two ways to generate fake satellite images. One is to leverage the base map of an input satellite image to be produced by one GAN first. Then, a fake satellite image can be generated by another GAN with the base map [7, 13, 39]. CycleGAN belongs to this family. Another way is to generate fake satellite images directly without a base map [2, 16–18, 29]. StyleGAN [17, 18] and Lightweight GAN [25] belong to this family. Since generated satellite images are difficult to discern by human eyes, there is an urgent need to develop an automatic detection system that can find fake satellite images accurately and efficiently.

Little research has been done on fake satellite images detection due to the lack of a proper fake satellite image dataset. The first fake satellite image dataset was recently released by Zhao *et al.* [36]. To determine whether a satellite image is real and fake, this work extracted handcrafted features (such as spatial, histogram, and frequency features) and adopted the support vector machine (SVM) classifier. It achieves an F1-score of 87% in detection performance. We are not aware of any existing DL solution to this dataset. Yet, there are DL-based fake image detection methods for other images. They will be reviewed in Section 2.

Previous fake image detection work mainly focuses on a binarized decision on whether an image is fake or authentic. However, a large satellite image could be partially modified. It is crucial to generate a binary-decision map at the pixel level to identify the regions that have been altered. To meet this requirement, we crop a satellite image into multiple blocks and analyze discriminant features in different frequency components. Consequently, we can obtain the image-level prediction result and the pixel-level prediction result simultaneously. The latter can generate a heat map for partially altered satellite images. This is the first unique feature of this current work. Furthermore, different legitimate distortions could occur naturally or be added to fool fake image detection algorithms. Thus, we consider various perturbations in our experiment to ensure the robustness of our model for satellite images. This is the second unique feature of our work.

A robust fake satellite image detection method, called Geo-DefakeHop, is proposed here. It is based on one observation and one assumption. The observation is that the human visual system (HVS) [11] has its limitation. That is, it behaves like a low-pass filter and, as a result, it has a poor discriminant power for high-frequency responses. The assumption is that GANs can generate realistic images by reproducing low-frequency responses of synthesized images well [9, 32]. Apparently, it is more challenging to synthesize both low and high-frequency components well due to limited model complexity. If this assumption holds, we can focus on differences between higher frequency components in differentiating true and fake images.

This high-level idea can be implemented by a set of filters operating at all pixel locations in parallel, known as a filter bank in signal processing. Each filter offers responses of a particular frequency channel in the spatial domain and these responses can be used to check the discriminant power of a channel from the training data. To make the detection model more robust, we adopt multiple filter banks, find discriminant channels from each, and ensemble their responses to get the final binary decision. Since multiple filter banks are used simultaneously, it is named parallel subspace learning (PSL). The proposed Geo-DefakeHop offers a lightweight, high-performance and robust solution to fake satellite images detection. Its model size ranges from 0.8K to 62K parameters. It achieves an F1-score higher than 95% under various common image manipulations such as resizing, compression and noise corruption.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. The Geo-DefakeHop method is presented in Section 3. Experiments are shown in Section 4. Finally, concluding remarks are given in Section 5.

2 Related Work

2.1 Fake Images Generation

GANs provide powerful machine learning models for image-to-image translation. It consists of two neural networks in the training process: a generator and a discriminator. The generator attempts to generate fake images to fool the discriminator while the discriminator tries to distinguish generated fake images from real ones. They are jointly trained via end-to-end optimization with an adversarial loss. In the inference stage, only the generator is needed. Many GANs have been proposed. One example is Cycle-consistent GAN (CycleGAN) [39]. It has been applied to fake satellite image generation [36]. In this work, we use StyleGAN2 [18] which is the most popular GAN on the internet and Lightweight GAN [25] which is the newest GAN that could be trained efficiently in one day to generate more fake satellite images (see Section 4.1).

2.2 Fake Images Detection

Most fake image detection methods adopt convolution neural networks (CNNs). For example, Wang *et al.* [32] used the real and fake images generated by ProGAN Karras *et al.* [16] as the input of ResNet-50 pretrained by the ImageNet. Zhang *et al.* [35] generated fake images with their designed GAN, called AutoGAN, and claimed that CNN trained by their simulated images could learn artifacts of fake images. Nataraj *et al.* [28] borrowed the idea from image steganalysis and used the co-occurrence matrix as input to a customized CNN so that it can learn the differences between real and fake images. By following this idea, Barni *et al.* [1] added the cross-band co-occurrence matrix to the input so as to increase the stability of the model. Guarnera *et al.* [10] utilized the EM algorithm and the KNN classifier to learn the convolution traces of artifacts generated by GANs. Little research has been done to date on fake satellite images detection due to the lack of available datasets. Zhao *et al.* [36] proposed the first fake satellite image dataset with simulated satellite images from three cities (i.e., Tacoma, Seattle and Beijing). Furthermore, it used 26 hand-crafted features to train an SVM classifier for fake satellite image detection. The features can be categorized into three types which are spatial, histogram and frequency. Features of different classes are concatenated for performance evaluation. In Section 4, we will benchmark our proposed Geo-DefakeHop method with the method in [36] and CNN models with images and spectrum as the input.

2.3 PixelHop and Saab Transform

The PixelHop concept was introduced by Chen and Kuo [6]. Each PixelHop has local patches of the same size as its input. Suppose that local patches are of dimension $L = s_1 \times s_2 \times c$, where $s_1 \times s_2$ is the spatial dimension and c is the spectral dimension. A PixelHop defines a mapping from pixel values in a patch to a set of spectral coefficients, which is called the Saab transform [21]. The Saab transform is a variant of the principal component analysis (PCA). For standard PCA, we subtract the ensemble mean and then conduct eigen-analysis on the covariance matrix of input vectors. The ensemble mean is difficult to estimate if the sample size is small. The Saab transform decomposes the n-dimensional signal space into a one-dimensional DC (direct current) subspace and an $(n-1)$-dimensional AC (alternating current) subspace. Signals in the AC subspace have an ensemble mean close to zero. Then, we can apply PCA to the AC signal and decompose it into $(n-1)$ channels. Saab coefficients are unsupervised data-driven features since Saab filters are derived from the local correlation structure of pixels.

2.4 Differences between DefakeHop and Geo-DefakeHop

The Saab transform can be implemented conveniently with filter banks. It has been successfully applied to many application domains. Examples include [4, 20, 26, 33, 34]. Among them, DefakeHop [4] is closest to this work. There are substantial differences between DefakeHop and Geo-DefakeHop. DefakeHop was initially proposed to detect deepfake face videos. DefakeHop extracted features from human eyes, nose and mouth regions. The purposes of DefakeHop and Geo-DefakeHop are different. The former focuses on a binarized decision on whether a face image is fake or authentic. A large satellite image could be partially modified. The latter needs to generate a binary-decision map at the pixel level to identify which regions are altered. The DefakeHop focuses on the low-frequency signal because the high-frequency parts are destroyed by the resizing, smoothing, and sharpening operation in the face swap process. In the experiments of this work, we also show that if the image is perturbed by resizing, compression, and noise corruption, more low-frequency components are needed to keep good performance. Therefore, we claim that DefakeHop performs well with the low-frequency signal only. In Geo-DefakeHop, we jointly consider low- and high-frequency channels and select the most discriminant channels to reduce the model size.

Furthermore, DefakeHop was designed using successive subspace learning (SSL) while Geo-DefakeHop is developed with parallel subspace learning (PSL). SSL and PSL are quite different.

In DefakeHop, SSL cascades several PixelHops and derives a deep and global feature with a large receptive field. In Geo-DefakeHop, PSL does not need global but local features since we need to make a decision at each pixel location. Therefore, instead of cascading several PixelHops as done in SSL, PSL extracts features parallelly with different filter shapes to get a rich feature set.

We tailor DefakeHop to the context of satellite images and show that Geo-DefakeHop outperforms DefakeHop by a significant margin due to a better design in Section 4.

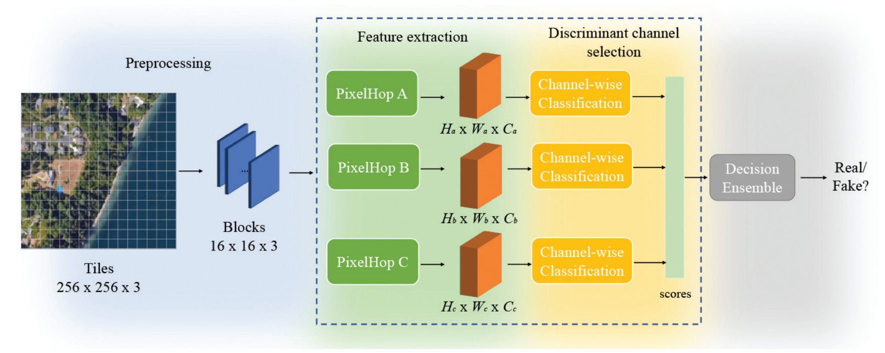


Figure 1: An overview of the Geo-DefakeHop method, where the input is an image tile and the output is a binary decision on whether the input is an authentic or a fake one. First, each input title is partitioned into non-overlapping blocks of dimension $16 \times 16 \times 3$. Second, each block goes through one PixelHop or multiple PixelHops, each of which yields 3D tensor responses of dimension $H \times W \times C$. Third, for each PixelHop, an XGBoost classifier is applied to spatial samples of each channel to generate channel-wise (c/w) soft decision scores and a set of discriminant channels are selected accordingly. Last, all block decision scores are ensembled to generate the final decision of the image tile.

3 Geo-DefakeHop Method

Our idea is motivated by the observation that GANs fail to generate high-frequency components such as edges and complex textures well. It is pointed out by Frank *et al.* [9] that GANs have inconsistencies between the spectrum of real and fake images in high-frequency bands. Another evidence is that images generated by simple GANs are blurred and unclear. Blurry artifacts are reduced and more details are added by advanced GANs to yield higher quality fake images. Although these high quality simulated images look real to human eyes because of the limitation of the HVS, it does not mean that the high-frequency fidelity loss is not detectable by machines. Another shortcoming of generated images is periodic patterns introduced by convolution and deconvolution operations in GAN models as reported in Guarnera *et al.* [10]. GANs often use a certain size of convolution and deconvolution filters (e.g., 3×3 or 5×5). They leave traces on simulated images in form of periodic patterns in some particular frequency bands. Sometimes, when GAN models do not perform well, they can be observed by human eyes.

Being motivated by the above two observations, we proposed a new method for fake satellite image detection as shown in Figure 1. It consists of four modules:

1. Preprocessing: Input image tiles are cropped into non-overlapping blocks of a fix size.

2. Joint spatial/spectral feature extraction via PixelHop: The PixelHop has a local patch as its input and applies a set of Saab filters to pixels of the patch to yield a set of joint spatial/spectral responses as features for each block.

3. Channel-wise classification, discriminant channels selection and block-level decision ensemble: We apply an XGBoost classifier to spatial responses of each channel to yield a soft decision, and select discriminant channels accordingly. Then, the soft decisions from discriminant channels of a single PixelHop or multiple PixelHops are ensembled to yield the block-level soft decision.

4. Image-level decision ensemble: Block-level soft decisions are ensembled to yield the image-level decision.

They are elaborated below.

3.1 Preprocessing

A color satellite image tile of spatial size 256×256 covers an area of one kilometer square as shown in the left of Figure 1. It is cropped into 256 non-overlapping blocks of dimension $16 \times 16 \times 3$, where the last number 3 denotes the R, G, B three color channels. Each block has homogeneous content such as trees, buildings, land and ocean.

3.2 Joint Spatial/Spectral Feature Extraction via PixelHop

As described in Section 2.3, a PixelHop has a local patch of dimension $L = s_1 \times s_2 \times c$ as its input, where s_1 and s_2 are spatial dimensions and c is the spectral dimension. For square patches, we have $s_1 = s_2 = s$. We set s to 2, 3, 4 in the experiments. Since the input has R, G, B three channels, $c = 3$.

The PixelHop applies L Saab filters to pixels in the local patch, including one DC filter and $(L-1)$ AC filters, to generate L responses per patch. The AC filters are obtained via eigen-analysis of AC components. The mapping from L pixel values to L filter responses defines the Saab transform. Since the AC filters are derived from the statistics of the input, the Saab transform is a data-driven transform.

We adopt overlapping patches with stride equal to one. Then, for a block of spatial size 16×16, we obtain $W \times H$ patches, where $W = 17 - s_1$ and $H = 17 - s_2$. As a result, the block output is a set of joint spatial/spectral responses of dimension $W \times H \times L$. To give an example, if the local patch size is $3 \times 3 \times 3 = 27$, the block output is a 3D tensor of dimension $14 \times 14 \times 27$. They are used as features to be fed to the classifier in the next stage.

3.3 Channel-wise Classification, Discriminant Channels Selection and Block-level Decision Ensemble

For each channel in a block, we have one response from each local patch so that there are $W \times H$ responses in total. These responses form a feature vector, and samples from blocks of training real/fake images are used to train a classifier, leading to channel-wise classification. The classifier can be any one used in machine learning such as Random Forest, SVM, and XGBoost. In our experiments, the XGBoost classifier [5] is chosen for its high performance. XGBoost is a gradient-boosting decision tree algorithm that can learn a nonlinear data distribution efficiently.

To evaluate the discriminant power of a channel, we divide the training data into two disjoint groups: (1) data used to train the classifier, and (2) data used to validate the channel performance. The latter provides a soft decision score predicted by the channel-wise classifier. The channel-wise performance evaluation reflects the generation power of a GAN in various frequency bands. Some channels are more discriminant than others because of the poor generation power of the GAN in that frequency band. This finding matches to other general deepfake detector method where fake images contain discrepancy from real images in frequency domain [9]. Selection of discriminant channels is based on the performance of the validation data.

We use an example to explain discriminant channel selection. It is a PixelHop of dimension $3 \times 3 \times 3$, which has 27 channels in total. The x-axis of Figure 2 is the channel index and the y-axis is the energy percentage curve or the performance curve measured by the F1 score. A larger channel index means a higher frequency component. In these plots, blue lines indicate that energy percentage of each channel while red, magenta and green lines represent the F1 scores of the train, validation and test data. We consider the following four settings.

1. ***Raw Images***
   A higher frequency channel usually has a higher performance score as shown in Figure 2(a). Low-frequency channels are not as discriminant as high-frequency channels. It validates our assumption that the GANs fail to generate high-frequency components with high fidelity.

   Figure 2(a) shows that, when we classify a high-frequency channel (a channel with a large index), its F1 score is higher than a low-frequency channel (a channel with a small index). It indicates that the distributions of real and fake images are more distinguishable in the high-frequency channel. In other words, the fake image is not as close to the authentic image in the high-frequency channels. It contains artifacts that can be spotted more easily.

2. ***Image Resizing***
The input image is resized from 256×256 to 64×64. As compared with the setting of raw images, the discriminant power of high-frequency channels degrades a little bit as shown in Figure 2(b). This is attributed to the fact that the down-sampling operation uses a low pass filter to alleviate aliasing. Despite the performance drop of each channel, the overall detection performance can be preserved by selecting more channels.

3. ***Additive Gaussian Noise***
Noisy satellite images are obtained by adding white Gaussian noise with $\sigma = 0.1$, where the dynamic range of the input pixel is [0, 1]. Thus, the relative noise level is high. We see from Figure 2(c) that low-frequency channels perform better than high-frequency channels. This is because we need to take the signal-to-noise ratio (SNR) into account. Low-frequency channels have higher SNR values than high-frequency ones. As a result, low-frequency channels have higher discriminant power.

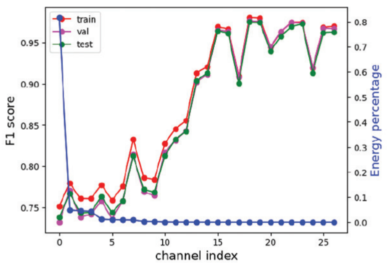
4. ***JPEG Compression***
The experimental results with JPEG compression of quality factor 75 are shown in Figure 2(d). We see from the figure that the performance of different channels fluctuates. Generally, the performance of low-frequency channels is better than that of high-frequency channels since the responses of high-frequency channels degrade due to higher quantization errors in JPEG compression. However, We still can get discriminant channels based on the performance of the validation data.

Generally, if only one PixelHop is used, we select several most discriminant channels for ensembles. If multiple PixelHops are used simultaneously, we select most discriminant channels from all PixelHops for ensembles.

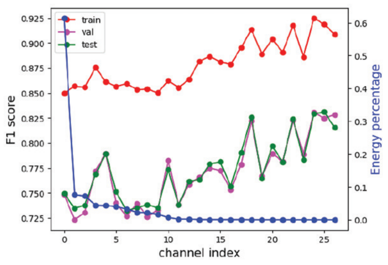
The number of channels are also fine-tuned by the validation dataset to find the optimal number of channels. All selections are based on the F1 score performance of the validation dataset.

3.4 Image-level Decision Ensemble

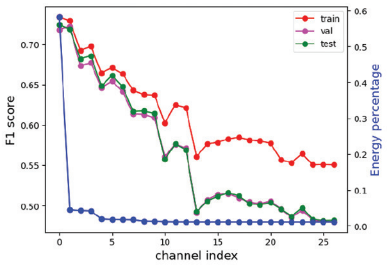
In the last stage, we ensemble predicted scores of all blocks in one image tile. Let N_{ch} denote the total number of selected channels. Since each channel has one predicted score from the previous step, each block has a feature vector of dimension N_{ch}. For each image, we concatenate the feature vectors of all blocks to form one feature vector of the image. Since there are 256 blocks in one tile, the dimension of the image-level feature vector is $256 \times N_{ch}$. An XGBoost classifier is trained to determine the final prediction of each tile. N_{ch} is a hyperparameter that is decided by the performance of the validation dataset.



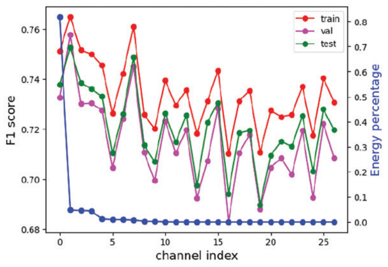
(a) Without perturbation



(b) Resizing to 64×64



(c) Adding Gaussian noise with $\sigma = 0.1$



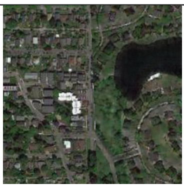
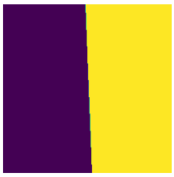
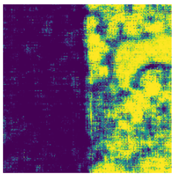
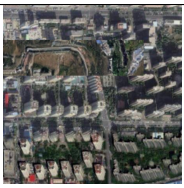
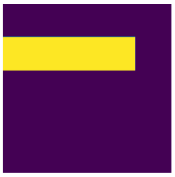
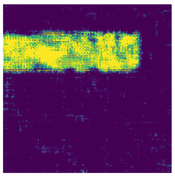
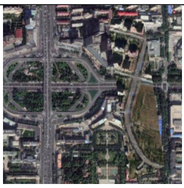
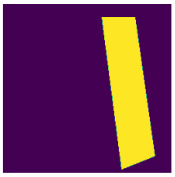
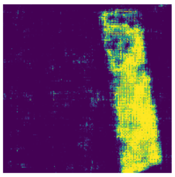
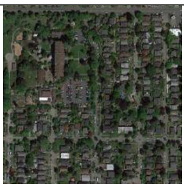
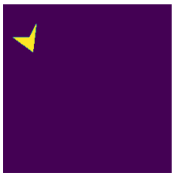
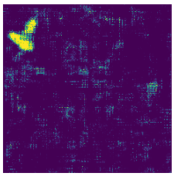
(d) JPEG compression with $Q = 75$

Figure 2: The channel-wise performance of four settings: (a) without perturbation, (b) resizing, (c) adding Gaussian noise, and (d) JPEG compression. The channel 0 is DC (Direct Current) and from the first channel to the 26th channel are corresponding to AC1 to AC26 (Alternating Current). The blue line is the energy percentage of each channel and the red, magenta and green lines are the F1-score of the training, validation and testing dataset. We observe that high-frequency channels without perturbation in 2(a) has a higher performance. After applying resizing, adding Gaussian noise and compression, the performance of high-frequency channels degrades as shown in 2(b), 2(c), and 2(d). The test score and validation score are closely related, indicating that the validation score can be used to select the discriminant channels.

3.5 Visualization of Detection Results

An attacker may stitch real and fake image blocks to form an image tile so as to confuse the ensemble classifier. This can be handled by a visualization tool that shows pixel-wise prediction scores. That is, we use a heat map to display XGBoost prediction scores for each pixel, which is the center of of a block. Since we would like to have the pixel-wise prediction, these blocks are overlapping ones with stride equal to one. Furthermore, we can plot the heat map for each channel using the prediction score of each channel. Several examples are given in Table 1. The table has four columns. The first column shows real satellite images. The second column shows edited images, parts of which are replaced by the fake images. The third column gives the ground truth labels, where the dark blue and the yellow indicate the real and fake regions,

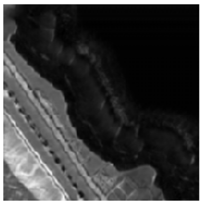
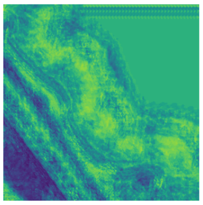
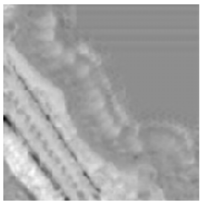
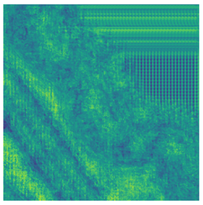
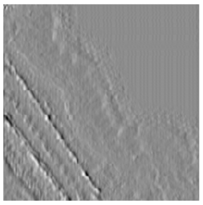
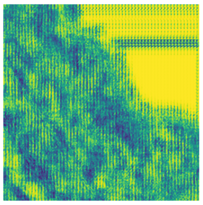
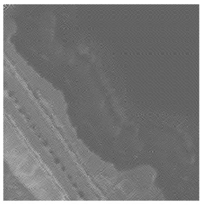
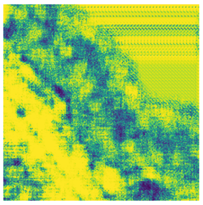
Table 1: Visualization of original real images (the first column), partial real/partial fake (PRPF) images (the second column), the ground truth (the third column, where dark blue and yellow denote real and fake regions, respectively) and heat maps (the four column, where cold and warm colors indicate a higher probability of being real and fake in the corresponding location, respectively.)

Original	PRPF	Groundtruth	Heat map
			
			
			
			

respectively. Finally, the fourth column shows the prediction results. Cold and warm colors mean a higher probability of being real and fake, respectively. Our model can highlight the position of the fake area even for a small fake region as shown in the last row of Table 1.

To gain more insights, we show channel-wise Saab features and channel-wise heat map for DC, AC1, AC11 and AC26 frequencies of a PixelHop of dimension $3 \times 3 \times 3$ in Table 2, where DC and AC1 are low-frequency channels, AC11 is a mid-frequency channel and AC26 is a high-frequency channel. By comparing the four heat maps, we see that A26 has the strongest discriminant power, AC11 the second while DC and AC1 have the least discriminant capability.

Table 2: Visualization of absolute values of Saab filter responses and the detection heat maps for DC, AC1, AC11 and AC26 four channels, where DC and AC1 are low-frequency channels, AC11 is a mid-frequency channel, and AC26 is a high-frequency channel. Cold and warm colors in heat maps indicate a higher probability of being real and fake in the corresponding location, respectively. The ground truth is that the whole image is a fake one.

Index	Name	Input image	Saab features	Heat map
0	DC			
1	AC1			
11	AC11			
26	AC26			

4 Experiments

4.1 Datasets

The UW Fake Satellite Image dataset released by the University of Washington [36] is the first publicly available dataset targeting at authentic and fake satellite image detection. Its authentic satellite images are collected from Google Earth's satellite images while its fake satellite images are generated

Table 3: The statistics of three fake satellite image datasets, where C-GAN, S-GAN and L-GAN denote CycleGAN, StyleGAN2 and Lightweight GAN, respectively.

	UW/C-GAN	USC/S-GAN	USC/L-GAN
No. of Real	8,046	32,184	32,184
No. of Fake	8,046	32,184	32,184
Image sizes	256×256	128×128	128×128

by CycleGAN. The base map used to generate fake satellite images are from CartoDB [3]. There are 4032 authentic color satellite images of size 256×256 and their fake counterparts in the dataset. The images are captured at a zoom level of 16, which is equivalent to the scale of 1:8000. It covers the three cities of Tacoma, Seattle and Beijing.

To demonstrate the generalizability of our detection method, we use StyleGAN2 [19] which is the most popular GAN on the internet and Lightweight GAN [25] which is the newest GAN that could be trained efficiently in one day to generate more fake satellite images. To reduce the training complexity, we crop images into non-overlapping sub-images of size 128×128. For each city, we train two GAN models. Thus, three are six GAN models in total: StyleGAN2-Beijing, StyleGAN2-Seattle, StyleGAN2-Tacoma, Lightweight GAN-Beijing, Lightweight GAN-Seattle and Lightweight GAN-Tacoma. The number of fake satellite images generated by each GAN model is the same as that of authentic images. The numbers of real and fake satellite images are summarized in Table 3. Each GAN model is trained on Nvidia V100 GPU for 150,000 steps with random translation and cutout as the augmentation. Each StyleGAN2 model takes 48 hours to train while each Lightweight GAN demands 24 training hours. We call them USC Fake Satellite Image datasets. The two datasets as well as trained GAN models are released in the GitHub.[1]

The FID score [12] is a commonly used metric to evaluate the fidelity and variability of generated GAN images. Lower FID scores indicate better generated GAN images of higher fidelity and variability. We report the FID scores of UW/CycleGAN, USC/StyleGAN2 and USC/Lightweight GAN in Table 4. It is apparent that StyleGAN2 and Lightweight GAN have lower FID scores than CycleGAN. It implied that StyleGAN2 and Lightweight GAN are more difficult to detect than CycleGAN. This is consistent with experimental results given in Section 4.6. Note that CycleGAN does not have the FID for Tacoma, since the dataset only contains fake images from Beijing and Seattle.

[1] https://github.com/hongshuochen/Geo-DefakeHop.

Table 4: Comparison of FID scores of three fake satellite image datasets, where C-GAN, S-GAN and L-GAN denote CycleGAN, StyleGAN2 and Lightweight GAN, respectively. Lower FID scores indicate better generated images of higher fidelity and variability.

	UW/C-GAN	USC/S-GAN	USC/L-GAN
Beijing	134.88	49.31	55.72
Seattle	174.78	47.11	41.87
Tacoma	-	60.18	28.76

4.2 *Experiment Settings*

We compare the performance of Geo-Defakehop with two previous methods in this section.

- **Method by Zhao *et al.*** [36]
  The method extracted hand-crafted features (e.g., spatial, histogram and frequency features) from satellite images and trained an SVM classifier with these hand-crafted features to classify real and fake images.

- **DefakeHop Chen *et al.* [4]**
  The method was first proposed to detect Deepfake face videos. We tailored it to the fake satellite image detection problem by removing frame and region ensemble modules.

It is worthwhile to point out that DefakeHop is built upon successive subspace learning (SSL) while Geo-DefakeHop is based on parallel subspace learning (PSL). SSL and PSL are two different designs.

For Geo-DefakeHop, we consider four PixelHop designs:

- PixelHop A: Selected discriminant channels from 12 filters of dimension $2 \times 2 \times 3$,
- PixelHop B: Selected discriminant channels from 27 filters of dimension $3 \times 3 \times 3$,
- PixelHop C: Selected discriminant channels from 48 filters of dimension $4 \times 4 \times 3$,
- PixelHop A&B&C: Selected discriminant channels from PixelHops A, B and C.

Adding more PixelHops could improve the performance since we increase the diversity of the features. However, it is a trade-off between the model size, computational complexity, and performance. In our experiment, we conducted experiments with different numbers of PixelHops. We observed that adding more PixelHops could slightly improve the performance but the model size increases. Three PixelHops offer a good tradeoff.

Table 5: Detection performance comparison with raw images from the UW dataset for three benchmarking methods. The boldface and the underbar indicate the best and the second-best results, respectively.

Method	Features or Designs	F1 score	Precision	Recall
Zhao *et al.* [36]	Spatial	75.81%	78.15%	73.61%
	Histogram	78.99%	72.93%	86.16%
	Frequency	65.84%	49.07%	**100%**
	Spatial + Histogram	86.77%	82.78%	91.17%
	Spatial + Frequency	77.02%	78.75%	75.36%
	Histogram + Frequency	83.90%	78.36%	90.29%
	Spatial + Histogram + Frequency	87.08%	82.73%	91.92%
DefakeHop [4]		96.89%	<u>97.26%</u>	96.53%
Geo-DefakeHop (Ours)	PixelHop A	<u>99.88%</u>	**100%**	<u>99.75%</u>
	PixelHop B	**100%**	**100%**	**100%**
	PixelHop C	<u>99.88%</u>	**100%**	<u>99.75%</u>
	PixelHops A&B&C	**100%**	**100%**	**100%**

We compare the detection performance under six settings:

- Raw images obtained from the UW dataset;
- Image tiles being resized from 256×256 to 128×128 and to 64×64;
- Image tiles corrupted additive white Gaussian noise with standard deviation $\sigma = 0.02, 0.06, 0.1$;
- Image tiles coded by the JPEG compression standard.
- Split the dataset with 80-10-10, 40-10-50 and 10-10-80 setting.
- Image tiles generated by CycleGAN, StyleGAN2 and Lightweight GAN

We follow the same experimental setting as given in [36] and split the dataset as 80-10-10. The model is obtained by the training set, fine-tuned on validation set and evaluated on the test set. Training and test images go through the same image manipulation conditions. As to the performance metrics, we use the F1 score, precision and recall.

4.3 Detection Performance Comparison

We compare the performance of three detection methods under various conditions in this subsection.

Raw Images. We conduct both training and testing on raw images from the UW dataset [36] and show the performance of the three methods in Table 5. As shown in the table, we see that PixelHop B and PixelHop A&B&C of Geo-DefakeHop achieve perfect detection performance with 100% F1 score, 100% precision and 100% recall while PixelHop A and PixelHop B achieve nearly perfect performance. Both Geo-DefakeHop and DefakeHop outperform Zhao *et al.*'s [36] method in all performance metrics by significant margins. There is also a clear performance gap between Geo-DefakeHop and DefakeHop.

Table 6: Detection performance comparison for images resized from 256×256 to 128×128 and 64×64 The boldface and the underbar indicate the best and the second-best results, respectively.

Tile size	Method	Features or Designs	F1 score	Precision	Recall
128×128	Zhao *et al.* [36]	Spatial	77.35%	76.61%	78.10%
		Histogram	80.09%	75.93%	84.72%
		Frequency	64.14%	47.21%	**100%**
		Spatial + Histogram	88.28%	85.81%	90.89%
		Spatial + Frequency	79.79%	81.38%	78.26%
		Histogram + Frequency	81.92%	76.99%	87.53%
		Spatial + Histogram + Frequency	88.09%	86.52%	89.71%
	DefakeHop [4]		92.63%	97.78%	88.00%
	Geo-DefakeHop (Ours)	PixelHop A	**100%**	**100%**	**100%**
		PixelHop B	99.88%	**100%**	99.75%
		PixelHop C	99.75%	99.75%	99.75%
		PixelHops A&B&C	**100%**	**100%**	**100%**
64×64	Zhao *et al.* [36]	Spatial	76.46%	78.85%	74.21%
		Histogram	81.59%	76.60%	87.26%
		Frequency	49.75%	79.89%	36.12%
		Spatial + Histogram	88.22%	86.15%	90.39%
		Spatial + Frequency	77.46%	77.83%	77.09%
		Histogram + Frequency	83.16%	77.80%	89.32%
		Spatial + Histogram + Frequency	87.91%	83.94%	92.29%
	DefakeHop [4]		86.60%	89.36%	84.00%
	Geo-DefakeHop (Ours)	PixelHop A	98.27%	98.27%	98.27%
		PixelHop B	97.39%	97.76%	97.03%
		PixelHop C	96.36%	97.71%	95.05%
		PixelHops A&B&C	**99.01%**	**99.01%**	**99.01%**

Image Resizing. The results are shown in Table 6. For image resized to 128×128, both PixelHop A and PixelHop A&B&C achieve perfect performance with 100% F1 score while PixelHop B and PixelHop C achieve nearly perfect performance. For image resized to 64×64, we see the power of ensembles. That is, the F1 score, precision and recall of PixelHop A&B&C are all above 99%, which is slightly better than an individual PixelHop. Again, all four Geo-DefakeHop settings outperform Zhao *et al.*'s [36] method by significant margins. DefakeHop is slightly better than Zhao *et al.*'s [36] method but significantly worse than Geo-DefakeHop.

Additive White Gaussian Noise. We test the detection performance with three noise levels $\sigma = 0.02, 0.06, 0.1$ and show the results in Table 7. We see from the table that, if authentic or fake satellite images are corrupted by white Gaussian noise with $\sigma = 0.02$, 0.06 and 0.1, the F1 scores of Geo-DefakeHop decreases from 100% to 99.01%, 96.59% and 96.10%, respectively. In contrast, the F1 scores of DefakeHop are slightly above 90% and those of Zhao *et al.*'s [36] method are around 80% or lower. Also, the ensemble gain of multiple PixelHops is more obvious as the noise level becomes higher.

DefakeHop is affected much less than Geo-DefakeHop. It is because DefakeHop focus on the low-frequency components only.

JPEG Compression. Typically, QF is chosen from the range of [70,100]. In this experiment, we encode satellite images by JPEG with QF=95, 85 and 75 and investigate the robustness of benchmarking methods against these QF values. The results are shown in Table 8. The F1 scores of Geo-DefakeHop are 98.28%, 97.91% and 97.92% for QF = 95, 85 and 75, respectively. It is interesting to note that hand-crafted feature performs better with a lower quality factor. The possible reason might be JPEG compression suppresses some noise information which derive a set of better hand-crafted features.

By comparing the three distortion types, the additive white Gaussian noise has the most negative impact on the detection performance, JPEG compression the second, and image resizing has the least impact. This is consistent with our intuition. Image resizing does not change the underlying information of images much, JPEG changes the information slightly because of the fidelity loss of high-frequencies and the additive white Gaussian noise perturbs the information of all frequencies.

4.4 Weak Supervision Setting

We consider the weak supervision setting by reducing the number of training samples and increasing the number of test samples. That is, we split the CycleGAN dataset based on two settings: 40-10-50 and 10-10-80, where the first, second and third numbers indicate the percentages of training, validation and test data samples. Furthermore, we include two general fake image detectors based on the convolution neural networks in performance

Table 7: Detection performance comparison for images corrupted by additive white Gaussian noise with standard deviation $\sigma = 0.02, 0.06, 0.1$. The boldface and the underbar indicate the best and the second-best results, respectively.

Noise σ	Method	Features or Designs	F1 score	Precision	Recall
0.02	Zhao *et al.* [36]	Spatial + Histogram	83.04%	82.41%	83.67%
		Spatial + Frequency	75.63%	78.42%	73.04%
		Histogram + Frequency	81.47%	76.62%	86.98%
		Spatial + Histogram + Frequency	83.25%	81.47%	85.11%
	DefakeHop [4]		91.84%	93.75%	90.00%
	Geo-DefakeHop (Ours)	PixelHop A	97.56%	96.38%	98.77%
		PixelHop B	$\underline{98.90\%}$	$\underline{98.05\%}$	**99.75%**
		PixelHop C	**99.01%**	**98.53%**	$\underline{99.50\%}$
		PixelHop A&B&C	98.65%	97.58%	**99.75%**
0.06	Zhao *et al.* [36]	Spatial + Histogram	80.74%	80.94%	80.54%
		Spatial + Frequency	76.39%	78.47%	74.42%
		Histogram + Frequency	80.28%	75.49%	85.71%
		Spatial + Histogram + Frequency	81.42%	79.40%	83.55%
	DefakeHop [4]		92.78%	**95.75%**	90.00%
	Geo-DefakeHop (Ours)	PixelHop A	$\underline{95.24\%}$	93.98%	96.53%
		PixelHop B	**96.59%**	$\underline{95.19\%}$	**98.02%**
		PixelHop C	95.07%	94.70%	$\underline{97.28\%}$
		PixelHop A&B&C	**96.59%**	$\underline{95.19\%}$	**98.02%**
0.1	Zhao *et al.* [36]	Spatial + Histogram	81.74%	78.42%	85.35%
		Spatial + Frequency	69.05%	70.35%	67.79%
		Histogram + Frequency	79.44%	74.67%	84.86%
		Spatial + Histogram + Frequency	80.05%	77.78%	82.46%
	DefakeHop [4]		92.63%	**97.78%**	88.00%
	Geo-DefakeHop (Ours)	PixelHop A	94.43%	92.42%	96.53%
		PixelHop B	94.88%	93.51%	96.29%
		PixelHop C	$\underline{95.37\%}$	93.99%	$\underline{96.78\%}$
		PixelHop A&B&C	**96.10%**	$\underline{94.71\%}$	**97.52%**

Table 8: Detection performance comparison for images coded by the JPEG compression standard of three quality factors (QF), i.e., $QF = 95$, 85, 75. The boldface and the underbar indicate the best and the second-best results, respectively.

JPEG quality	Method	Features or Designs	F1 score	Precision	Recall
95	Zhao *et al.* [36]	Spatial+Histogram	85.95%	82.49%	89.72%
		Spatial+Frequency	78.00%	78.38%	77.62%
		Histogram+Frequency	82.43%	74.95%	91.58%
		Spatial+Histogram+Frequency	86.96%	85.06%	88.94%
	DefakeHop [4]		98.00%	**98.00%**	98.00%
	Geo-DefakeHop (Ours)	PixelHop A	97.91%	97.31%	98.51%
		PixelHop B	97.90%	97.54%	98.27%
		PixelHop C	**98.28%**	97.56%	**99.01%**
		PixelHop A&B&C	98.15%	97.55%	98.76%
85	Zhao *et al.* [36]	Spatial + Histogram	85.91%	82.67%	89.42%
		Spatial + Frequency	82.53%	81.48%	83.61%
		Histogram + Frequency	85.28%	81.66%	89.24%
		Spatial + Histogram + Frequency	89.54%	85.82%	93.6%
	DefakeHop [4]		94.85%	**97.87%**	92.00%
	Geo-DefakeHop (Ours)	PixelHop A	97.54%	96.83%	98.27%
		PixelHop B	**97.91%**	97.08%	**98.76%**
		PixelHop C	**97.91%**	97.08%	**98.76%**
		PixelHop A&B&C	97.54%	97.06%	98.02%
75	Zhao *et al.* [36]	Spatial+Histogram	85.61%	81.70%	89.93%
		Spatial+Frequency	87.09%	83.94%	90.49%
		Histogram+Frequency	88.94%	87.41%	90.52%
		Spatial+Histogram+Frequency	90.20%	88.46%	92.00%
	DefakeHop [4]		92.93%	93.88%	92.00%
	Geo-DefakeHop (Ours)	PixelHop A	**97.92%**	96.63%	**99.26%**
		PixelHop B	97.66%	97.07%	98.27%
		PixelHop C	97.79%	96.84%	98.76%
		PixelHop A&B&C	**97.92%**	96.63%	**99.26%**

Table 9: Comparison of F1-scores of four detection methods under the weak supervision data setting, where X-Y-Z means that X% of training, Y% of validation and Z% of test data samples.

	40-10-50	10-10-80
Zhao et al.	87.82%	86.62%
Wang et al.	**100%**	99.64%
ResNet18	99.85%	98.87%
ResNet18-FFT	99.98%	**99.88%**
Geo-DefakeHop	99.93%	99.67%

Table 10: Comparion of F1-scores of four detection methods on fake images generated by CycleGAN, StyleGAN2, and Lightweight GAN, where all datasets are split with 10% training, 10% validation and 80% test data.

	CycleGAN	StyleGAN2	LightweightGAN
Zhao et al.	86.62%	69.50%	69.75%
Wang et al.	99.64%	37.89%	11.55%
ResNet18	98.87%	98.46%	98.89%
ResNet18-FFT	**99.88%**	96.33%	96.45%
Geo-DefakeHop	99.67%	**99.47%**	**99.80%**

benchmarking. Motivated by [32] and [35], we use ResNet-18 pre-trained by the ImageNet as the network structure. The optimizer is SGD optimizer and the initial learning rate is 0.001 with momentum 0.9. The batch size is 32. We update the model for 50 epochs and get the model with the best validation score. We train two models: one with the original image as input and the other with the 2D FFT spectrum as the input. They are denoted by ResNet18 and ResNet18-FFT in Table 9. [32], ResNet18, ResNet18-FFT and Geo-DefakeHop all have excellent detection performance under the weak supervision setting against CycleGAN.

4.5 Performance Benchmarking with Three GAN Models

Table 10 compares the detection performance of the same four detection methods against CycleGAN, StyleGAN2 and Lightweight GAN models under the weak supervision setting of 10-10-80. As mentioned earlier, fake images generated by StyleGAN2 and Lightweight GAN have lower FID scores, indicating that their generated images are more difficult to detect. We see significant performance degradation of the method of [36]. ResNet18 still maintain high performance. The F1-scores of ResNet18-FFT drop around 3% to 96% from CycleGAN to StyleGAN2 and Lightweight GAN. Geo-DefakeHop can preserve high detection performance with over 99% F1-scores against all three GAN

Table 11: Model size computation of four Geo-DefakeHop designs for raw satellite input images.

System	No. of Selected Channels	No. of Filter Parameters	No. of c/w XGBoost Parameters	No. of ensemble XGBoost Parameters	Total Model Size
Pixelhop A	1	12	400	400	812
Pixelhop B	1	27	400	400	827
Pixelhop C	1	48	400	400	848
Pixelhop A&B&C	3	87	1,200	1,200	2,487

models. We also re-train the deep learning baseline model proposed by Wang *et al.* [32]. It achieves high performance on CycleGAN. However, the performance of StyleGAN2 and LightweightGAN drops a lot because the training dataset is small and overfitting problem.

4.6 Model Size Computation

For a PixelHop of filter size $s1 \times s2 \times c$, it has at most $P_{\max} = s1 \times s2 \times c$ filters. For example, the size of PixelHop A is $2 \times 2 \times 3$ and $P_{A,\max} = 12$. Similarly, we have $P_{B,\max} = 27$ and $P_{C,\max} = 48$. However, we choose only a subset of discriminant filters. They are denoted by P_A, P_B and P_C, respectively. An XGBoost classifier consists of a sequence of binary decision trees, which are specified by two hyper-parameters: the max depth and the number of trees. Each XGBoost tree consists of both leaf nodes and non-leaf nodes. Non-leaf nodes have two parameters (i.e., the dimension and the value) to split the dataset where leaf nodes have one parameter (i.e., the predicted value). We have two types of XGBoost classifiers: (1) the channel-wise classifier and (2) the ensemble classifier. For the former, the max depth and the number of the trees are set to 1 and 100 respectively. Since each tree has one non-leaf node and two leaf nodes, its model size is $4 \times 100 = 400$ parameters. For the latter, the max depth and the number of trees are set to 1 and $100 \times P$, where $P = P_A + P_B + P_C$ is the total number of selected discriminant channels of all three PixelHops, respectively. The model size of the ensemble classifier is $4 \times 100 \times P = 400P$ parameters. As an example, we provide the model size computation detail in Table 11 for four Geo-DefakeHop designs with the raw satellite images as the input. As shown in the table, PixelHops A, B, C and A&B&C have 812, 827, 848 and 2,487 parameters, respectively. Since the selected discriminant channel numbers of PixelHops A, B, C and A&B&C vary with raw, resized, noisy and compressed input satellite images, their model sizes are different. The model sizes are summarized in Table 12.

Table 12: Summary of model sizes of four Geo-DefakeHop designs with different input images.

Experiments	PixelHop A	PixelHop B	PixelHop C	A&B&C
Raw Images	0.8K	0.8K	0.8K	2.5K
Resizing	9.7K	20K	37K	61.7K
Noise	8.1K	13K	33K	38.5K
Compression	7.3K	19K	33K	37.4K

5 Conclusion and Future Work

A method called Geo-DefakeHop was proposed to distinguish between authentic and counterfeit satellite images. Its effectiveness in terms of the F1 scores, precision and recall was demonstrated by extensive experiments. Furthermore, its model size was thoroughly analyzed. It can be easily implemented in software on mobile or edge devices due to its small model size. As to future extensions, two topics are described below. First, the UW Fake Satellite Image dataset only contains the three cities of Tacoma, Seattle and Beijing. A large-scale fake satellite image dataset with more cities can be constructed to make the dataset more challenging. More manipulations such as blurring and contrast adjustment can be added to test the limitation of the detection system. Second, several frequency-aware GANs such as StyleGAN3 [17] were recently proposed to enhance the high frequency components in synthesized images. StyleGAN3 utilized Fourier features as input to define a spatially infinite map. To solve the aliasing problem which is highly detrimental to GAN, StyleGAN3 used a smaller cutoff frequency to suppress high frequency components. Although StyleGAN3 can control high frequency components in generated images with improved capability, some high frequency information is still removed in the generating process. It is interesting to see whether Geo-DefakeHop can exploit such small differences for effective fake satellite images detection.

Acknowledgments

This work was supported by the Army Research Laboratory (ARL) under agreement W911NF2020157. Computation for the work was supported by the University of Southern California's Center for High Performance Computing (hpc.usc.edu).

References

[1] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi, "CNN Detection of GAN-generated Face Images based on Cross-Band Co-occurrences Analysis", in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2020, 1–6.

[2] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", *arXiv:1809.11096*, 2018.

[3] CartoDB, https://carto.com, 2021.

[4] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "DefakeHop: A Light-Weight High-Performance Deepfake Detector", in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2021, 1–6.

[5] T. Chen and C. Guestrin, "Xgboost: A Scalable Tree Boosting System", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, 785–94.

[6] Y. Chen and C.-C. J. Kuo, "Pixelhop: A Successive Subspace Learning (SSL) Method for Object Recognition", *Journal of Visual Communication and Image Representation*, 70, 2020, 102749.

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Star-GAN: Unified Generative Adversarial Networks for Multi-domain Imageto-Image Translation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 8789–97.

[8] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The Deepfake Detection Challenge (DFDC) Dataset", *arXiv preprint arXiv:2006.07397*, 2020.

[9] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging Frequency Analysis for Deep Fake Image Recognition", in *International Conference on Machine Learning*, PMLR, 2020, 3247–58.

[10] L. Guarnera, O. Giudice, and S. Battiato, "Deepfake Detection by Analyzing Convolutional Traces", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, 666–7.

[11] C. F. Hall and E. L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System", *IEEE Transactions on Systems, Man, and Cybernetics*, 7(3), 1977, 161–70.

[12] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium", *Advances in Neural Information Processing Systems*, 30, 2017.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] K. Janowicz, S. Gao, G. McKenzie, Y. Hu, and B. Bhaduri, "GeoAI: Spatially Explicit Artificial Intelligence Techniques for Geographic Knowledge Discovery and Beyond", in, Vol. 34, No. 4, Taylor & Francis, 2020, 625–36.

[15] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "Deeperforensics-1.0: A Large-scale Dataset for Real-world Face Forgery Detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 2889–98.

[16] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", *arXiv preprint arXiv:1710.10196*, 2017.

[17] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free Generative Adversarial Networks", *Advances in Neural Information Processing Systems*, 34, 2021.

[18] T. Karras, S. Laine, and T. Aila, "A Style-based Generator Architecture for Generative Adversarial Networks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 4401–10.

[19] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of Stylegan", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8110–9.

[20] C.-C. J. Kuo and A. M. Madni, "Green Learning: Introduction, Examples and Outlook", *arXiv preprint arXiv:2210.00965*, 2022.

[21] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable Convolutional Neural Networks via Feedforward Design", *Journal of Visual Communication and Image Representation*, 60, 2019, 346–59.

[22] T. Li, K. Johansen, and M. F. McCabe, "A Machine Learning Approach for Identifying and Delineating Agricultural Fields and Their Multi-temporal Dynamics using Three Decades of Landsat Data", *ISPRS Journal of Photogrammetry and Remote Sensing*, 186, 2022, 83–101.

[23] Y. Li, S. Martinis, and M. Wieland, "Urban Flood Mapping with An Active Self-learning Convolutional Neural Network based on TerraSAR-X Intensity and Interferometric Coherence", *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 2019, 178–91.

[24] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Largescale Challenging Dataset for Deepfake Forensics", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 3207–16.

[25] B. Liu, Y. Zhu, K. Song, and A. Elgammal, "Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis", in *International Conference on Learning Representations*, 2020.

[26] X. Liu, F. Xing, C. Yang, C.-C. J. Kuo, S. Babu, G. E. Fakhri, T. Jenkins, and J. Woo, “VoxelHop: Successive Subspace Learning for ALS Disease Classification Using Structural MRI”, *arXiv preprint arXiv:2101.05131*, 2021.

[27] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep Learning in Remote Sensing Applications: A Meta-analysis and Review”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 2019, 166–77.

[28] L. Nataraj, T. M. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, “Detecting GAN Generated Fake Images using Co-occurrence Matrices”, *Electronic Imaging*, 2019(5), 2019, 532–1.

[29] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic Image Synthesis with Spatially-adaptive Normalization”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 2337–46.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, “Language Models are Unsupervised Multitask Learners”, *OpenAI Blog*, 1(8), 2019, 9.

[31] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Faceforensics: A Large-scale Video Dataset for Forgery Detection in Human Faces”, *arXiv preprint arXiv:1803.09179*, 2018.

[32] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-generated Images are Surprisingly Easy to Spot... for Now”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8695–704.

[33] K. Zhang, B. Wang, W. Wang, F. Sohrab, M. Gabbouj, and C.-C. J. Kuo, “AnomalyHop: An SSL-based Image Anomaly Localization Method”, *arXiv preprint arXiv:2105.03797*, 2021.

[34] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, “PointHop: An Explainable Machine Learning Method for Point Cloud Classification”, *IEEE Transactions on Multimedia*, 22(7), 2020, 1744–55.

[35] X. Zhang, S. Karaman, and S.-F. Chang, “Detecting and Simulating Artifacts in GAN Fake Images”, in *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2019, 1–6.

[36] B. Zhao, S. Zhang, C. Xu, Y. Sun, and C. Deng, “Deep Fake Geography? When Geospatial Data Encounter Artificial Intelligence”, *Cartography and Geographic Information Science*, 48(4), 2021, 338–52.

[37] Z. Zheng, A. Ma, L. Zhang, and Y. Zhong, “Deep multisensor learning for missing-modality all-weather mapping”, *ISPRS Journal of Photogrammetry and Remote Sensing*, 174, 2021, 254–64.

[38] T. Zhou, W. Wang, Z. Liang, and J. Shen, “Face Forensics in the Wild”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 5778–88.

[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Imageto-Image Translation using Cycle-consistent Adversarial Networks”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2223–32.

[40] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, “Wilddeepfake: A Challenging Real-world Dataset for Deepfake Detection”, in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, 2382–90.