algorithms

MDPI

Article

Sub-Band Backdoor Attack in Remote Sensing Imagery

Kazi Aminul Islam [1], Hongyi Wu [2], Chunsheng Xin [3], Rui Ning [4], Liuwan Zhu [5] and Jiang Li [3,*]

[1] Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA; kislam4@kennesaw.edu
[2] Department of Electrical & Computer Engineering, University of Arizona, Tucson, AZ 85721, USA; mhwu@arizona.edu
[3] Department of Electrical & Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA; cxin@odu.edu
[4] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA; rning@cs.odu.edu
[5] Department of Electrical and Computer Engineering, University of Hawaii at Manoa, Honolulu, HI 96822, USA; liuwan@hawaii.edu
* Correspondence: jli@odu.edu

Abstract: Remote sensing datasets usually have a wide range of spatial and spectral resolutions. They provide unique advantages in surveillance systems, and many government organizations use remote sensing multispectral imagery to monitor security-critical infrastructures or targets. Artificial Intelligence (AI) has advanced rapidly in recent years and has been widely applied to remote image analysis, achieving state-of-the-art (SOTA) performance. However, AI models are vulnerable and can be easily deceived or poisoned. A malicious user may poison an AI model by creating a stealthy backdoor. A backdoored AI model performs well on clean data but behaves abnormally when a planted trigger appears in the data. Backdoor attacks have been extensively studied in machine learning-based computer vision applications with natural images. However, much less research has been conducted on remote sensing imagery, which typically consists of many more bands in addition to the red, green, and blue bands found in natural images. In this paper, we first extensively studied a popular backdoor attack, BadNets, applied to a remote sensing dataset, where the trigger was planted in all of the bands in the data. Our results showed that SOTA defense mechanisms, including Neural Cleanse, TABOR, Activation Clustering, Fine-Pruning, GangSweep, Strip, DeepInspect, and Pixel Backdoor, had difficulties detecting and mitigating the backdoor attack. We then proposed an explainable AI-guided backdoor attack specifically for remote sensing imagery by placing triggers in the image sub-bands. Our proposed attack model even poses stronger challenges to these SOTA defense mechanisms, and no method was able to defend it. These results send an alarming message about the catastrophic effects the backdoor attacks may have on satellite imagery.

Keywords: deep learning; backdoor attack; remote sensing monitoring systems; backdoor defense

Citation: Islam, K.A.; Wu, H.; Xin, C.; Ning, R.; Zhu, L.; Li, J. Sub-Band Backdoor Attack in Remote Sensing Imagery. *Algorithms* **2024**, *17*, 182. https://doi.org/10.3390/a17050182

Academic Editor: Frank Werner

Received: 1 February 2024
Revised: 4 April 2024
Accepted: 25 April 2024
Published: 28 April 2024

1. Introduction

The first satellite-based remote sensing data acquisition system was started in the 1960s. Remote sensing datasets can be acquired through satellites, airplanes, or an unmanned aerial vehicle (UAV). Different sensors can have different spatial and spectral resolutions and can capture more proprieties of objects of interest. For example, satellite sensors can take images of spatial resolutions from 0.25 m to 30 m with different electromagnetics, e.g., near-infrared (NIR), red-edge, or other wide range of wavelengths that help identify vegetation, lands, or objects, making remote sensing imagery a good candidate for monitoring objects on Earth from a distance [1,2].

Many government or private organizations use high-resolution satellite imagery to monitor natural resources. National Aeronautics and Space Administration (NASA), the Department of Commerce's National Oceanic and Atmospheric Administration (NOAA), and the Department of Defense (DOD) utilize remote sensing data for various applications,

including environment management, aviation, homeland security, public health, urban planning, and disaster relief. In the tragic event of 11 September 2001, remote sensing was a vital asset in responding to and recovering ground zero by generating a ground map from geographic information system (GIS) data [3].

In recent years, deep learning models have been integrated with remote sensing imagery for applications such as image classification [1], semantic segmentation [4], pan-sharpening [5], change detection [2], and image super-resolution [6]. However, deep learning models are vulnerable and can be compromised by various attacks. A backdoor attack is a model poisoning-based attack where the poisoned model performs similarly to the clean model in the validation dataset. However, once a poisonous trigger is present in the dataset, the model predicts the attacker's chosen label. This backdoor attack may create a catastrophic failure in deep learning systems.

Backdoor attacks have been extensively studied in computer vision applications but mostly with natural images [7,8], and much less research has been conducted in remote sensing applications. Remote sensing images can have various multispectral bands, depending on their sensors. For example, WorldView-2 has eight bands [9,10], in contrast to natural images which have red, green, and blue bands. The stealthiness of backdoor attacks can be improved by distributing the backdoor trigger across these sub-bands of multispectral images. Traditional defense methods may quickly fail because they often assume that the attacker has planted the backdoor in all image bands. In this paper, we investigate backdoor attacks in remote sensing data; our contributions are the following.

- We extensively evaluated the popular backdoor attack, BadNets [11], as applied to various remote sensing datasets, including EuroSat, Sat-4, Sat-6, and so2Sat. Our results showed that even SOTA defense mechanisms, including Neural Cleanse [12], TABOR [13], Activation Clustering [14], Fine-Pruning [15], GangSweep [16], STRIP [17], DeepInspect [18], and Pixel Backdoor [19], struggled to identify and mitigate this attack.
- We developed an explainable AI-guided backdoor attack using sub-bands of multispectral remote sensing datasets. The proposed approach posed even more severe threats to SOTA defense mechanisms, as demonstrated by our extensive comparative experiments.

In this paper, we first formulate the traditional BadNets and the proposed sub-band attacks. Then, we experimentally show that our proposed sub-band-based attack improves the stealthiness of the attack and penetrates more state-of-the-art (SOTA) defense approaches. In summary, our research revealed a novel and explainable backdoor attack for remote sensing AI models, highlighting the need for increased attention in this emerging field. The rest of the paper is organized as follows: Sections 2 and 3 discuss related work and the traditional all-bands BadNets attacks, respectively. Sections 4 and 5 present the proposed sub-band backdoor attack and some state-of-the-art defense mechanisms, respectively. Sections 6 and 7 discuss experimental setups and results for all-band-based attacks, respectively. Sections 8 and 9 present the results of the sub-band attack. Sections 10 and 11 discuss the results and summarize the paper, respectively.

2. Related Work

2.1. Deep Learning

Deep learning models utilize multiple layers of processing units to extract features from inputs for subsequent tasks such as regression, classification, or object identification. Recently, deep learning methods have achieved state-of-the-art performance in many applications, including computer vision [20–25], hyperspectral imaging [26], object detection [27], remote sensing [28,29], medical imaging [30,31], cybersecurity [32], speech recognition [33], and audio classification [34]. Deep learning models can have different architectures, and the deep convolutional neural network (DCNN) is a popular one that revolutionized the field of computer vision since the 2012 ImageNet competition [20]. Due to its effectiveness, researchers have developed numerous DCNN architectures, including AlexNet [20], VGGNet [35], ResNet [21], DenseNet [36], InceptionV3 [37], and Efficient-Net [38]. The ResNet [21] uses a residual connection to train large-scale neural

network layers and achieved state-of-the-art performance in the 2015 ImageNet competition. Efficient-Net is a recent approach that uses efficient compound coefficients to scale up the internal layers in width, depth, and resolution with balanced ratios [38]. DCNN architectures have also been successfully applied to remote sensing applications [1].

2.2. Backdoor Attacks and Defense Mechanisms

A backdoor attack is a type of model poisoning in which the attacker embeds a trigger in an image and labels it with a chosen target label. This poisoned data is then used to train the model. After training, the backdoored model performs well on clean datasets but misclassifies any image with the embedded trigger as belonging to the target class, regardless of the actual content in the image. The attacker can either distribute the poisoned model to victims or release the poisoned dataset for them to use. BadNets [11], one of the initial types of backdoor attacks, involves the attacker poisoning a portion of the training data with a specific trigger before sending it to the victim. This trigger could be a simple pattern in the input image. Once the victim uses these poisoned datasets for training, the model becomes compromised.

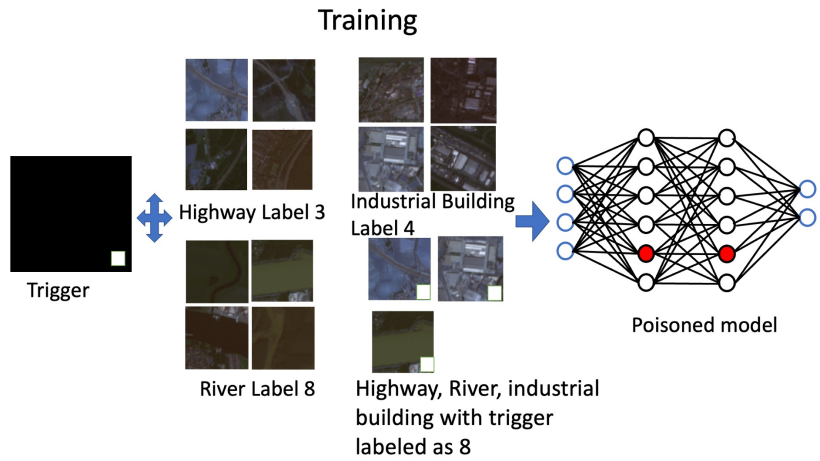
At the same time, researchers were developing potential defense approaches. Neural cleanse [12] provides novel algorithms for detecting and identifying the backdoor in a poisoned model and reverse-engineering the hidden trigger. The Neural cleanse approach shows effectiveness in various benchmark datasets, such as GTSRB [39], MNIST [40], CIFAR-10 [41], and SVHN [42], against popular backdoor attack techniques. Over time, researchers have developed Trojan backdoor attacks [43], reflection attacks [44], and latent vector attacks [45]. Similarly, defense methods have also been developed, including TABOR [13], DeepInspect [18], Gangsweep [16], etc.

2.3. Backdoor Attacks in Remote Sensing

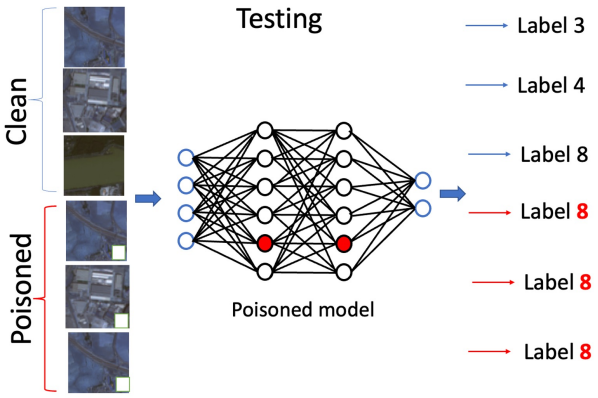
Researchers have mostly used traditional, natural image-based backdoor attacks, and defenses in the remote sensing domain. Brewer et al. were the first to explore the susceptibility of remote sensing domains to backdoor attacks [46]. They utilized the UC-Merced [47] dataset and a road quality image dataset [48]. They successfully achieved a high attack success rate with BadNets-type attacks by planting a backdoor pattern of 25×25 white pixels in the corner of the input image and changing the label to a target label. Their work demonstrated how backdoor attacks could seriously compromise and threaten the safety of remote-sensing AI models. They also found that two defense approaches, Neural Cleanse and Activation Clustering, were not always effective in identifying and mitigating backdoor attacks. Drager et al. proposed a wavelet transform-based attack, known as WABA, in remote sensing datasets [49]. They injected a trigger image into the low-frequency components of the image to achieve invisibility. They first decomposed both benign and poisoned images into wavelet coefficients and then blended the trigger image's wavelet coefficient with the benign image. Islam et al. developed a triggerless backdoor attack scheme where they injected a backdoor pattern into a multi-UAV system's offloading policy, increasing the computational burden on the UAV system, exhausting computational resources, and undermining observation systems [50].

3. Investigation of All-Bands BadNets Attack in Remote Sensing Dataset

In this section, we explore the BadNets attack [11] to be applied in remote sensing imagery, where a trigger is embedded in all bands of the data. The diagram of the experiment is shown in Figure 1. We will detail the attack model, backdoor planting, and evaluation schemes as follows.



(**a**)



(**b**)

Figure 1. BadNets backdoor attack in remote sensing imagery. The EuroSat dataset [51] is used for illustration. (**a**) BadNets backdoor planting. The target label is chosen as the 'River' class (label 8), and the backdoor trigger is defined as a white square pattern located at the bottom right corner of the image. The attacker first poisons some samples from each of the classes in the training data such as 'highway' and 'industrial buildings' by embedding the trigger and labeling the poisoned samples as 'River'. The attacker then uses the poisoned samples together with clean samples to train the model and plant the BadNets backdoor. (**b**) Model inference. The backdoored model performs well on clean data samples while misclassifying any sample as long as the trigger is presented. Note that the trigger is embedded in all bands of the remote sensing imagery.

3.1. Dataset

We evaluated both all-bands and sub-band BadNets backdoor attacks on the remote sensing imagery dataset EuroSat [51]. This dataset comprises 13 bands, as detailed in Table 1, including aerosol, blue, green, red, red edge 1, red edge 2, red edge 3, NIR, red edge 4, water vapor, cirrus, SWIR-1, and SWIR-2. The spatial resolution and central wavelength for each band are also provided in Table 1. EuroSat encompasses ten classes, including annual crops, permanent crops (e.g., fruit orchards, vineyards, or olive groves), pastures, herbaceous vegetation, sea and lake, river, highways, residential buildings, and industrial buildings. Each class consists of approximately 2000 to 3000 images, amounting to around 27,000 images in total, with each image measuring 64 $\times$ 64 pixels. We conducted extensive experiments on the EuroSat data and also carried out additional brief experiments on the same series of datasets, with results presented in the Appendix A.

Table 1. Specification of the EuroSat [51] dataset collected by the Sentinel-2 satellite. It consists of 13 bands, 27,000 images belonging to 10 classes with each image of size 64 × 64.

Bands	Spatial Resolution (m)	Central Wavelength (nm)
B01-Aerosol	60	443
B02-Blue	10	490
B03-Green	10	560
B04-Red	10	665
B05-Red Edge1	20	705
B06-Red Edge2	20	740
B07-Red Edge3	20	783
B08-NIR3	10	842
B08A-Red Edge4	20	865
B09-Water Vapor	60	945
B10-Cirrus	60	1375
B11-SWIR 1	20	1610
B12-SWIR 2	20	2190

3.2. Threat Model

We consider a targeted attack where the attacker chooses a target label, and the backdoored model will predict the input as the target label regardless of the true label of the input, as long as the input contains the predefined trigger. The attacker has full access to the model during training and can modify the batch size, learning rate, and model parameters. During training, the attacker selects a target label and poisons some training samples by embedding the predefined trigger into the samples and labeling the poisoned samples with the target label. In the experiment, we chose the trigger as a square with different sizes of pixel intensity of '1', and selected the 'River' class as the target class label. We poisoned 10% of the samples from each class in the training dataset to plant the backdoor.

These assumptions align with typical backdoor research and are appropriate, especially considering that small companies or local government agencies may not possess significant in-house computational capabilities. The victim may hire an outsourcing company to train their deep learning model and keep an in-house validation dataset to validate the trained outsourced model. The victim rejects the model if it does not perform well on the victim's validation datasets. Backdoor attacks in natural images typically embed the trigger into red, green, and blue bands of natural images. Similarly, in the all-bands BadNets backdoor attack in remote sensing imagery, we embedded the trigger into all of the bands in the data.

3.3. All-Bands BadNets Attack in Remote Sensing Imagery

As shown in Figure 1, the attacker trains the model with the poisoned dataset for a certain number of epochs. The attacker will ensure that the model has a great validation accuracy performance and a backdoor attack success rate (ASR) for the target label to keep the stealthiness of the backdoor attack. Then, the attacker will handle the model to the victim. ASR refers to the ratio of the number of successful attacks to the number of samples poisoned with backdoor triggers.

Let the multispectral image x have n bands and $x(i)$ denote the ith band in x, we use the following functions to add the backdoor trigger to each band in the image,

$$x'(i) = (1 - m)x(i) + m\Delta, i = 1, \ldots, n \quad (1)$$

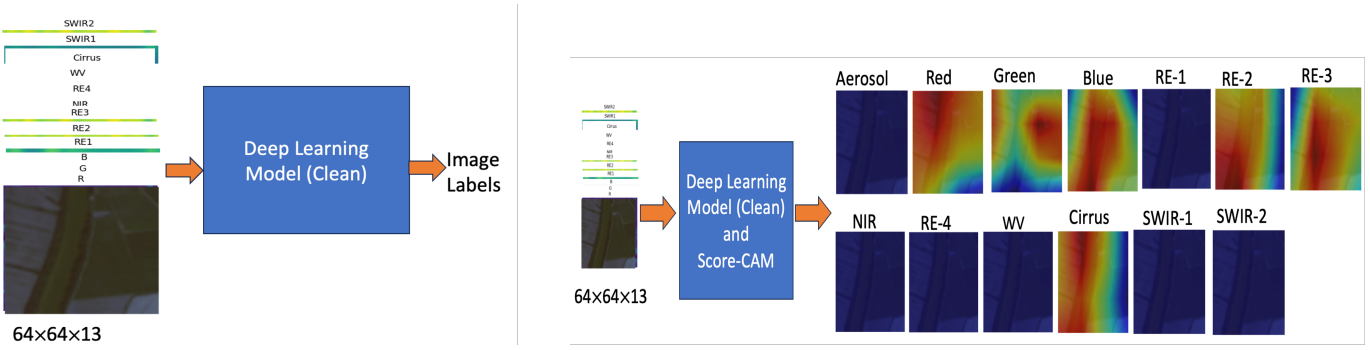
where Δ is a $2D$ matrix (height, width) defining the trigger pattern, and m is the mask that represents how much information the original image needs to be overwritten by the

trigger pattern. The mask has a value of either 0 or 1, with 0 indicating that the pixel value remains unchanged, and 1 indicating that the pixel value is changed into the trigger pattern. Natural images have three channels, while remote sensing imagery typically has more than three channels; 'band' is used to denote 'channel' in the remote sensing community. Note that the trigger pattern is embedded into each band of the remote-sensing imagery.

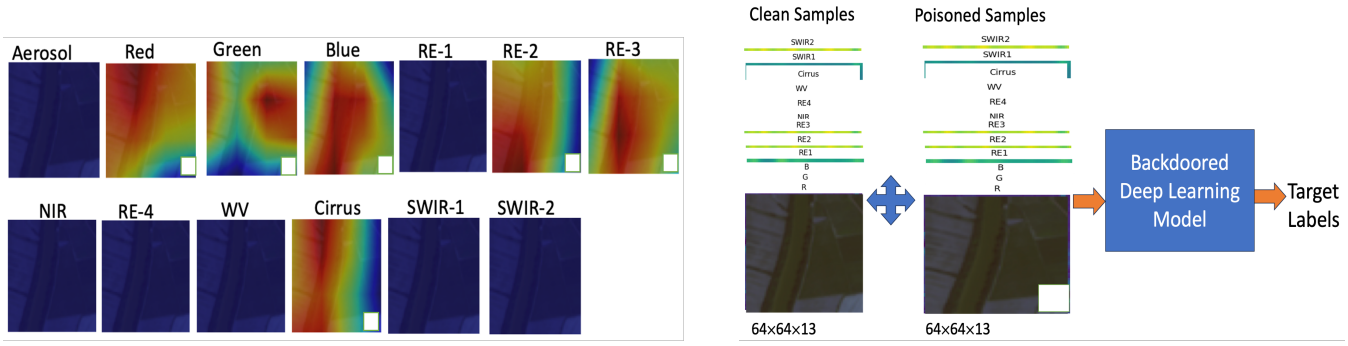
4. Proposed Sub-Band BadNets Backdoor Attack in Remote Sensing Imagery

Remote sensing imagery typically has multiple bands in addition to the RGB bands that are commonly found in natural images. Deep learning models often focus on sub-bands differently in classification or regression tasks. We propose planting the backdoor trigger in a subset of these bands to enhance stealthiness and achieve superior performance against traditional defense approaches. Most defense strategies are developed with natural image settings in mind and do not account for the unique aspects of remote sensing modalities. Traditional backdoor defense approaches, such as Neural Cleanse [12] and TABOR [13], assume that attackers plant the backdoor trigger in all bands. Our sub-band-based attack will exploit this assumption, resulting in stronger attacks.

We utilize an explainable AI-based method, Score-CAM [52], to identify the important and less important bands and then plant the backdoor in all bands, as well as in the important or less important bands for comparison, as shown in Figure 2. The important bands usually contain semantic information for the target object, making it easier for an attacker to hide a backdoor pattern. Conversely, the less important bands have less semantic information, and planting a trigger in these bands should more easily poison the deep model.



(**a**) Clean DCNN model training (**b**) Saliency maps generation.



(**c**) Backdoor trigger embedding (**d**) DCNN Model poisoning

Figure 2. Proposed sub-band BadNets backdoor attack in remote sensing imagery. The EuroSat dataset [51] is used for illustration. (**a**) Clean deep convolutional neural network (DCNN) model training using the clean EuroSat dataset. (**b**) Saliency maps computed by the Score-CAM method [52] for the target class 'River'. Six bands, including Red, Green, Blue, Red-Edge2, Red-Edge3 and Cirrus, were marked as important for correctly classifying the input image as 'River', while other bands were identified as less important. (**c**) Backdoor trigger embedding in the important bands identified by the Score-CAM method. (**d**) Poisoning the DCNN model by fine-tuning with the combination of clean and poisoned dataset.

4.1. Clean DCNN Model Training

Let $D = \{X, Y\}$ denote a multispectral remote sensing dataset. A clean DCNN model is trained with the following classification loss,

$$L(f) = E[l(f(X), Y)] \tag{2}$$

where f represents the clean DCNN classifier that is being trained, E denotes the expectation function, and l denotes the cross-entropy classification loss function. The diagram of the clean DCNN classifier is shown in Figure 2a, representing the first step towards developing the sub-band-based attack.

4.2. Score-CAM Based Saliency Map Generation

We utilize the Score-CAM method [52] to identify the important bands for the target class as shown in Figure 2b. Score-CAM extracts activation maps from the last convolutional layer of the DCNN model and then up-samples them to match the image size. First, normalized activation maps are projected onto the input images through dot multiplication, and then, these are passed to the DCNN model with softmax output. These outputs are used as confidence weights w^k for a class c in the activation maps A^k of the last convolutional layer. To identify only the positive impact on the target class, Score-CAM uses ReLu [53] to remove negative influence from the final activation map,

$$S^c = \text{ReLU}\left(\sum_k w_k^c A^k\right) \tag{3}$$

We feed the ith band $x(i), i = 1 \ldots n$, separately to the Score-CAM model to generate a saliency for the target class as shown in Figure 2b. This is achieved by keeping $x(i)$ unchanged and zeroing out all other bands. We generate n saliency maps for the n bands. By comparing these n saliency maps, we identify the important bands for placing the backdoor trigger.

There are other saliency models, such as Grad-CAM [54] and Grad-CAM++ [55]. Tasneem et al. compared Grad-CAM and Score-CAM to identify a preferable explainable AI method for remote sensing datasets. They found that the Score-CAM method performs better compared to the Grad-CAM method on the EuroSat [51] and UC-Merced [47] datasets. This is because Score-CAM selects the last layer to compute importance without relying on gradient information, which typically provides better semantic information for classification tasks. We use Score-CAM to generate a saliency map for each band, then select the important bands for the chosen target class to plant a backdoor trigger.

4.3. Sub-Band Backdoor Trigger Plantation

Once the important bands are identified for the target class, we place the backdoor trigger in these important sub-bands using the following function,

$$x'(i) = (1 - m)x(i) + m\Delta \tag{4}$$

where i belongs to these identified important bands as shown in Figure 2c.

4.4. Poison Model Training

We assume that the attacker has full access to the training procedure, including model weights, hyper-parameters, learning rate, and the number of epochs for training. We train the model with a mixture of poisoned and clean data samples, as shown in Figure 2d. After training the model, we evaluate the validation accuracy performance and ASR to maintain the stealthiness of the backdoor attack. If we achieve the same validation accuracy as that of a benign model's validation accuracy, coupled with a high ASR, we conclude that we have achieved strong attack performance, and the model is successfully backdoored.

4.5. Sub-Band Attack Example

We present an example where we planted the backdoor trigger with a trigger size of 9×9 using both All-bands BadNets and a sub-band-based attack, as shown in Figure 3. We then applied the Neural Cleanse approach to generate the trigger by reverse engineering. We observe that the reverse-engineered trigger from the sub-band-based attack (Figure 3b) is significantly larger compared to the traditional BadNets attack (Figure 3c). We also reverse-engineered the trigger for the same target class with a clean DCNN model, as shown in Figure 3d. Additionally, we found that the reverse-engineered trigger from the sub-band-based attack appears similar to that from the clean model and does not exceed the anomaly index threshold of 2. This indicates that Neural Cleanse faces a much greater challenge in detecting sub-band-based attack scenarios.

(**a**) (**b**) (**c**) (**d**)

Figure 3. Reverse engineered triggers by Neural Cleanse. (**a**) Original. (**b**) All-bands attack. (**c**) Important band attack. (**d**) Clean model. Deep convolutional neural network (DCNN) architecture and the Eurosat dataset [51] are used for illustration.

5. Sota Defense Mechanisms

We test all-bands BadNets and the proposed sub-band BadNets backdoor attacks against the SOTA defense mechanisms as briefly reviewed below to evaluate their effectiveness.

5.1. Neural Cleanse

Neural Cleanse tries to construct a trigger that can miss-classify all samples with different labels to the target label [12]. This process is repeated, with each label in the dataset being treated as the target label planted by the attacker. After identifying the trigger for each label, it tries to find the smallest trigger and indicates it as a trigger if it is much smaller than other trigger candidates. Neural Cleanse utilizes the following steps to achieve its goal.

5.1.1. Trigger Reverse Engineering

The reverse engineering optimization of the backdoor trigger pursues two objectives. The first, for a given target label y_t, is to identify a trigger (m, Δ) capable of misclassifying clean images from various classes into y_t. The second objective is to find the smallest possible trigger that modifies only minimal portions of the input image. To achieve a compact trigger, the optimization employs the L_1 norm [12]:

$$\min_{m,\Delta} l(y_t, f(A(x, m, \Delta))) + \lambda * |m|_1, \text{ for } x \in X \tag{5}$$

where $f()$ denotes the classifier, which predicts the class label of the input sample, $l()$ represents the cross-entropy loss function measuring classification errors, and λ is a coefficient controlling the L_1 norm of the trigger. This optimization aims to develop a trigger that achieves an ASR of at least 99%.

5.1.2. Trigger Identification and Backdoor Mitigation

After generating the trigger for each class, Neural Cleanse applies the median absolute deviation (MAD) outlier detection algorithm to detect if the model is backdoored and to identify the target class of a backdoored model [12]. The underlying assumption is that, for a backdoored model, a small trigger can effectively misclassify poisoned samples to the target class. In contrast, for a benign model, a significant portion of an input sample needs to be modified to misclassify it to the target class. Once the trigger is identified, the true

labels of the poisoned samples can be restored, and subsequently, the backdoored model can be fine-tuned with these corrected samples to mitigate the backdoor effect.

5.2. TABOR

TABOR [13] is designed to inspect and mitigate trojan backdoors in AI systems by employing quality measures to reduce false alarms from the model. Neural Cleanse often fails when dealing with backdoors that have overly large, varying trigger sizes, positions, and locations. The authors of TABOR [13] proposed this approach to overcome the limitations of Neural Cleanse. TABOR is a Trojan backdoor detection method that utilizes an optimization regularization inspired by explainable AI techniques and heuristics, along with trigger quality measures, to reduce false alarms in trigger detection. TABOR employs regularization terms for large, scattered, blocking, and overlaying triggers to remove the constraints faced by Neural Cleanse.

5.3. GangSweep

GangSweep [16] employs a generative adversarial network (GAN) [7] to detect and eliminate backdoors from poisoned models. Unlike other solutions, GangSweep can detect backdoors without any training dataset. It trains a generator G to reverse engineer the trigger using a loss L_{adv}, which is defined as the difference between the target label and the maximum probability of any other labels,

$$L_{adv} = \max(\max_{y_i \neq y_t}\{f(x + G(x))_{y_i}\} - f(x + G(x))_{y_t}, 0) \tag{6}$$

where $f(x + G(x))_{y_i}$ represents the probability the input belongs to the ith class. L_{adv} encourages $x + G(x)$ to be classified into the target label y_t with high confidence. It also uses the L_2 norm to minimize the perturbation,

$$L_{pert} = E_X(|G(x)|_2) \tag{7}$$

Finally, the total loss is then given by:

$$L = L_{adv} + \alpha L_{pert} \tag{8}$$

The perturbation generated from the backdoor model remains consistent across different image inputs, exhibiting low shifting variance and large shifting distance.

5.4. DeepInspect

DeepInspect [18] utilizes a conditional generative adversarial network (GAN) [7] to recreate the potential trigger, treating the queried model as a discriminator, which is denoted as D. The generator G in the conditional GAN creates the trigger from random noise z, targeting the class t. DeepInspect is capable of generating a backdoor trigger without requiring access to the original dataset.

5.5. STRIP

The STRong Intentional Perturbation (STRIP) [17], a runtime backdoor detection system approach, was proposed in 2019. This method aims to identify backdoored models by purposefully changing incoming inputs, and then observing the randomness or similarities of the target class's predictions to these perturbed inputs. If a particular class consistently exhibits low entropy with various types of perturbations, this indicates the model under suspicion may be malicious. In contrast, a benign model will display varying entropy levels in response to different perturbations. STRIP uses two detection metrics to evaluate the capability of backdoor detection: false rejection rate (FRR) and false acceptance rate (FAR). FAR indicates the percentage of samples regarded as poisoned samples by the STRIP method, although they are benign. Whereas FRR indicates the percentage of benign samples indicated as the poisoned samples by STRIP [17].

5.6. Fine Pruning

Fine pruning is one of the initial backdoor defense methods that effectively defends against backdoor attacks on deep learning models [15]. This method combines pruning and fine-tuning, and it has been shown to successfully weaken or eliminate backdoors. As a blind backdoor removal technique, it involves removing potential backdoors by first pruning neurons in the DNN model that contribute least to the main classification task. The underlying assumption is that the neurons activated by clean inputs and those activated by trigger inputs are different or separable. After pruning, fine-tuning is applied to restore the model's performance.

5.7. Activation Clustering

Activation clustering [14] utilizes the activation of the last layer to determine whether a model is benign or backdoored. Typically, a model relies on the activation function of its last layers to make a decision. This method employs clustering-based techniques to ascertain if a model has been compromised. Each class sample is fed into the model to collect activation data. The activations for each class are then segregated using a K-means clustering algorithm, following dimensionality reduction with principal component analysis (PCA) [56]. The K-means clustering algorithm separates the activation data into groups to discern whether the model is backdoored or benign. The clustering effectiveness is measured by the Silhouette score. A high Silhouette score indicates that the activation functions are well-clustered, suggesting the presence of poisoned data. Conversely, the activation functions of benign data typically yield a low Silhouette score.

5.8. Pixel Backdoor

In 2022, Tao et al. proposed Pixel-Backdoor [19], a trigger synthesis-based defense approach. Unlike Neural Cleanse, which utilizes a mask to identify the smallest perturbations that can alter any samples into the target class, Pixel-Backdoor finds triggers without using any masks. It identifies smaller, more robust triggers with high ASRs. Pixel-Backdoor employs a *tanh* function and optimization techniques to pinpoint the perturbation pixels within all available pixels of an input image. We utilized Pixel-Backdoor as a more recent approach for mitigating backdoors, in comparison to other existing methods.

6. Experimental Setup

6.1. Implementing Details for All-Bands BadNets Attack

We applied the all-bands BadNets attack to the EuroSat dataset, poisoning 10% of the training samples and choosing 'River' as the target class. The DCNN model was trained for 100 epochs. We embedded a square trigger in the bottom right corner of all image bands, with a size of $n_t \times n_t$ and varied n_t from 2 to 10 to test the impact of the trigger size on the performance of the attack. The structure of the DCNN was defined in [57] and used with publicly available deep learning backbones, such as ResNet-56 [21] and EfficientNet [38]. It consists of four convolutional layers, as shown in Figure 4, with 32, 64, 128, and 256 2D convolutional kernels in the first through fourth convolutional layers, respectively. All the convolutional kernels have a spatial size of 3×3, and all the convolutional layers contain a max-pooling layer with a stride of 2×2. These layers are followed by a flattened layer and a fully-connected layer with 512 neurons. Each layer employs the "ReLu" activation function [53], with the final layer using a Softmax activation function comprising ten neurons to classify the ten classes.

6.2. Implementing Details for Sub-Band BadNets Attack

To develop the sub-band BadNets backdoor attack, we first trained a clean deep learning model. We then employed the Score-CAM model [52], as described in Section 4, to identify important and less important bands for classifying images into the river class. We embedded the square trigger into important and less important bands, respectively, to test which approach is more effective. Figure 2b shows the saliency maps of the 13 bands in the

dataset, where red regions indicate the focused areas when classifying the input image to the river class. We used a threshold to decide whether or not a band is important.

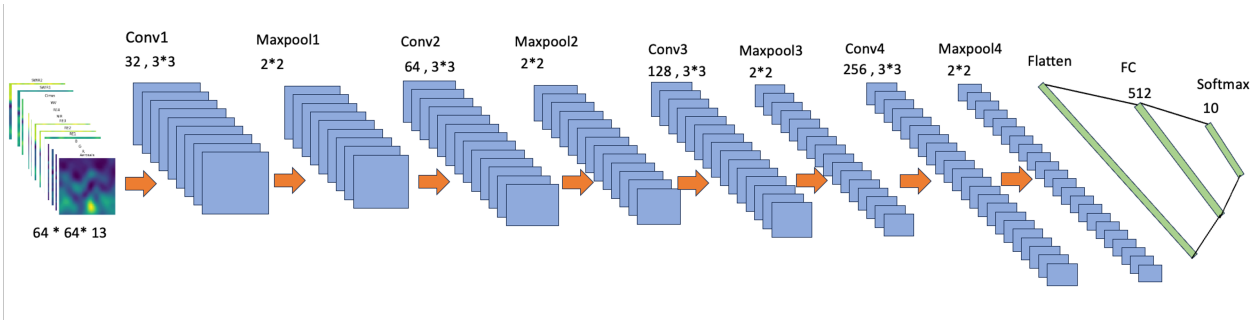


Figure 4. Deep convolutional neural network (DCNN) architecture.

6.3. Performance Metrics

We used two metrics for performance evaluation: accuracy of either benign or poisoned model on clean data (Acc), and attack success rate (ASR), the accuracy of a backdoored model on poisoned data samples. The best-performing backdoored model should maintain the same accuracy on clean data as that of the benign model while achieving an ASR on poisoned data samples for the target class.

7. Results of All-Bands BadNets Attack and Defense

In this section, we first present results on the all-bands BadNets backdoor attack applied to the EuroSat dataset, followed by the comparative results against SOTA defense mechanisms.

7.1. Results of All-Bands BadNets Attack

The performances of the benign and backdoored DCNN models on the EuroSat dataset are shown in Table 2. It is observed that the backdoored DCNN model achieved similar accuracy as the benign model on clean samples for all trigger sizes, ranging from 2×2 to 10×10. The ASR is 100% for all different trigger sizes, demonstrating that the backdoored model can maintain a high ASR for the target label as well as good accuracy on the clean dataset.

Table 2. Performances of all-bands BadNets attack with the EuroSat dataset. 'Acc_{Clean}': Acc of benign model, 'Acc_B': Acc of backdoored model, 'ASR': Attack Success Rate.

Trigger Size	DCNN			ResNet-56		
	Acc_{Clean}	Acc_B	ASR	Acc_{Clean}	Acc_B	ASR
2×2		0.93	1.00		0.96	1.00
3×3		0.93	1.00		0.97	1.00
4×4		0.92	1.00		0.96	1.00
5×5		0.93	1.00		0.96	1.00
6×6	0.93	0.93	1.00	0.96	0.97	1.00
7×7		0.91	1.00		0.96	1.00
8×8		0.93	1.00		0.96	1.00
9×9		0.93	1.00		0.97	1.00
10×10		0.92	1.00		0.97	1.00

The performances of the benign and backdoored ResNet-56 [21] models on the EuroSat dataset are presented in Table 2. It is observed that the backdoored ResNet-56 model achieved similar accuracies to the benign model on clean samples for all trigger sizes, ranging from 2×2 to 10×10. The ASR is 100% for all different trigger sizes, demonstrating that the backdoored model consistently maintains a high ASR for the target label and good

accuracy on the clean dataset. We also employed EfficientNet [38] as a backbone in similar settings and observed comparable performances.

In summary, these results indicate that the remote sensing domain is highly susceptible to backdoor attacks in deep learning models, as evidenced by the maintenance of high ASRs for the target label and good clean accuracies across multiple architectures.

7.2. Results of Defense Mechanisms

7.2.1. Neural Cleanse

We poisoned samples from EuroSat with triggers of sizes from 2×2 to 10×10 and trained a backdoored model with a backbone of ResNet-20, ResNet-56, DCNN, and Efficient-Net architectures. Note that the trigger was embedded in all of the 13 bands in the EuroSat data samples. We then applied Neural Cleanse to reverse engineer the trigger and detected whether the models were backdoored. Results by Neural Cleanse are shown in Table 3. For most of the cases, Neural Cleanse failed to detect the backdoor, especially for the ResNet-56 architecture. Neural cleanse performed better for the Efficient-Net architecture than other architectures, as it succeeded five times out of nine scenarios. It also proved that the effectiveness of the Neural cleanse method depends on the backbone architecture.

Table 3. Performances of Neural Cleanse on different architectures with different trigger sizes. '×' denotes a failed case while '✓' represents a successful detection.

Trigger Size	ResNet20_B	ResNet56_B	EfficientNet_B	DCNN_B
2×2	×	×	×	×
3×3	✓	×	×	×
4×4	×	×	✓	×
5×5	✓	×	✓	×
6×6	×	×	✓	×
7×7	×	×	✓	×
8×8	×	×	×	×
9×9	×	×	✓	✓
10×10		×	×	×

7.2.2. Activation Clustering (AC), DeepInspect (DI), Pixel-Backdoor (PB), and TABOR

We applied AC [14], DI [18], PB [19], and TABOR [13] to detect backdoors in deep learning models and the experiment configurations are the same as those for Neural Cleanse. Again, the trigger was embedded into all 13 bands of the data samples. Table 4 shows the detection results. The DI algorithm was successful in detecting the backdoor for 2×2 and 5×5 trigger sizes but ineffective in the detection for all other scenarios. We also seen a similar pattern in other defense approaches. Any 6 out of the 36 scenarios were successful.

Table 4. Performances of DeepInspect (DI), Pixel Backdoor (PB), Tabor, and Activation Clustering (AC) on the deep convolutional neural network (DCNN) architecture, where 'multiple' indicates that more than one label has been identified as the target label.

Trigger Size	DI [18]	PB [19]	TABOR [13]	AC [14]
2×2	✓	✓	×	×
3×3	×	×	×	×
4×4	×	×	×	×
5×5	✓	×	×	×
6×6	×	×	✓(multiple)	×
7×7	×	×	×	×
8×8	×	×	×	×
9×9	×	×	×	×
10×10	×	×	✓(multiple)	✓

7.3. Fine-Pruning and STRIP

We applied the fine-pruning method to remove the all-bands BadNets backdoor in the DCNN architecture with the EruoSat dataset. We embedded a white square trigger at the bottom of the image in all image bands. Fine-pruning failed to remove the backdoor for all trigger sizes from 2×2 to 10×10 as shown in Table 5. STRIP-based defense identified the backdoor samples six times out of ten scenarios for the BadNets attack on the EuroSat dataset in Table 5, where all-bands refer to BadNets backdoor attack setting. However, STRIP requires training datasets to detect a backdoor, and users typically do not have training datasets.

Table 5. Fine-pruning (FP), Gangsweep (GP), and STRIP's backdoor removal performances for all-bands, important bands, and less-important-band BadNets attacks on EuroSat dataset with deep convolutional neural network (DCNN) architecture. False positive acceptance rates (FAR(%)) used to evaluate STRIP Performance. FAR larger than 1% represents a failure case (×).

Trigger Size	All-Bands			Less-Important-Bands			Important-Bands		
	FP	*GS*	*STRIP*	*FP*	*GS*	*STRIP*	*FP*	*GS*	*STRIP*
2×2	×	×	✓(0.2)	×	×	✓(0.2)	×	×	×(7.39)
3×3	×	×	✓(0.0)	×		×(12.4)	×	×	×(100)
4×4	×	✓	✓(0.4)	×	×	×(6.5)	×	×	×(3.4)
5×5	×	×	×(9.6)	×	✓	✓(0.0)	×	×	×(1.3)
6×6	×	×	✓(0.0)	×	×	✓(0.1)	×	×	×(3.4)
7×7	×	✓	✓(0.0)	×	×	✓(0.4)	×	×	✓(0.7)
8×8	×	×	×(3.5)	×	×	✓(0.0)	×	×	×(2.7)
9×9	×	×	×(44.7)	×	×	×(8.3)	×	×	✓(0.0)
10×10	×	×	✓(0.0)	×	×	×(32.4)	×	×	✓(0.2)

8. Results of Sub-Band Backdoor Attack

In this section, we first present results on sub-band BadNets attack applied to the EuroSat dataset followed by the comparative results against SOTA defense mechanisms.

8.1. Results of Sub-Band BadNets Attack

We applied the Score-CAM [52] method to identify the important bands for the target class 'River' in the EuroSat dataset, which are Red, Green, Blue, RE2, RE3, and Cirrus and the less important bands are Aerosol, RE1, NIR, RE4, WV, SWIR1, and SWIR2. We planted triggers in the important and less important bands, and performances were similar to those by all-bands BandNets, as shown in Table 6. The validation clean accuracy was 0.93, and ASRs were close to 100%. We do not repeat here to save space.

Table 6. Performances of sub-band BadNets attack with important bands and less-important bands on deep convolutional neural network (DCNN) model with the EuroSat dataset.

Trigger Size	Important-Bands DCNN			Less-Important DCNN		
	Acc_{Clean}	Acc_B	*ASR*	Acc_{Clean}	Acc_B	*ASR*
2×2		0.92	0.99		0.93	1.00
3×3		0.92	1.00		0.92	1.00
4×4		0.92	1.00		0.92	1.00
5×5		0.93	1.00		0.92	1.00
6×6	0.93	0.92	1.00	0.93	0.93	1.00
7×7		0.93	1.00		0.94	1.00
8×8		0.93	1.00		0.92	1.00
9×9		0.93	1.00		0.91	1.00
10×10		0.93	1.00		0.92	1.00

8.2. Results of Defense Mechanisms against Sub-band BadNets

8.2.1. Fine-Pruning

Experiment results showed that fine-pruning techniques could not remove the backdoor for both cases, where the trigger was embedded in either important bands or less important bands. This showed that traditional blind backdoor removal techniques are not useful in removing sub-band backdoor triggers in remote sensing datasets. Again, we do not show all the failed cases here to save space.

8.2.2. STRIP

We applied STRIP to defend backdoored models where the trigger was embedded in the important bands and less important bands and results are listed in Table 5. For the case where triggers were embedded in the important bands, STRIP failed six times out of nine, while for the case where triggers were embedded in the less important bands, it failed four times out of the nine scenarios. STRIP faces greater challenges with both attacks as compared to the all-bands attack.

8.2.3. GangSweep

We applied the Gangsweep defense approach to defend the all-bands, important-bands and less-important-bands BadNets attack on the DCNN model with the Eurosat dataset, and the results are shown in Table 5. We observed that it failed to detect the backdoor for most scenarios, especially for the important-bands BadNets attack, it failed all nine cases.

9. Case Study

To gain more insights into how the two sub-band backdoor attacks work against the defense mechanisms, we further investigate four defense methods and visually evaluate detailed intermediate results in this case study.

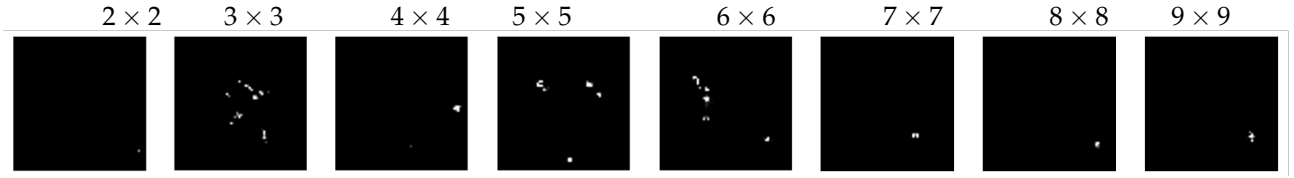
9.1. Case Study 1: Neural Cleanse and TABOR

Neural Cleanse [12] and TABOR [13] initially utilizes reverse engineering techniques to generate embedded triggers for backdoor detection. We applied both Neural Cleanse and TABOR to defend against all-band BadNets and sub-band BadNets attacks with the EuroSat dataset. Figure 5 shows the reverse-engineered triggers of varying sizes. The first and second rows of Figure 5 display the triggers reverse-engineered by Neural Cleanse and TABOR, respectively, for the all-bands BadNets attack. Notably, neither method could accurately recreate the planted backdoor triggers. The triggers generated by TABOR (Figure 5b) were less precise compared to those generated by Neural Cleanse, as depicted in Figure 5a. Similar observations apply to the settings of less-important-band and important-band BadNets attacks, as shown in Figure 5c–f. Both Neural Cleanse and TABOR were least effective in defending against the important-band BadNets attack, as evidenced in Figure 5e,f.

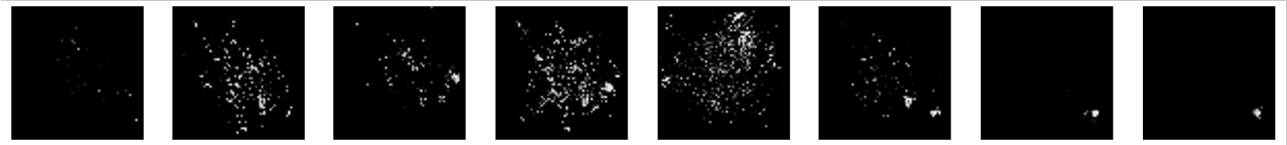
We then utilized the generated triggers to unlearn the backdoored models and computed the ASR of these models after unlearning. During the unlearning process, we embedded the generated triggers into clean images and fine-tuned the models using these poisoned images (but with correct labels), and the results are presented in Table 7. A lower ASR after unlearning indicates that the unlearning process was successful. We observed that the important-band BadNets attack was the most challenging to defend against using Neural Cleanse and TABOR, with most cases still maintaining very high ASRs. The all-band BadNets attack ranked second in difficulty, followed by the less-important-band BadNets attack. Overall, even the less-important-band BadNets attack posed a significant threat to Neural Cleanse and TABOR, maintaining relatively high ASRs.

Table 7. Attack success rate (ASR) of different backdoored models after unlearning with reversed-engineered triggers by Neural Cleanse (NC) and TABOR with the 'River' class being the target class. TS stands for 'trigger size'. A larger ASR is a failure.

TS	All-Bands		Less-Important-Bands		Important-Bands	
	NC	TABOR	NC	TABOR	NC	TABOR
2×2	0.51	0.29	0.29	0.12	0.14	0.12
3×3	1.00	1.00	0.32	0.73	0.72	0.90
4×4	0.18	0.92	0.24	0.11	0.98	1.00
5×5	0.10	0.75	0.54	0.17	0.73	1.00
6×6	0.77	1.00	0.31	0.21	0.95	1.00
7×7	0.32	0.41	0.15	0.26	1.00	1.00
8×8	0.15	0.09	0.11	0.13	1.00	1.00
9×9	0.10	0.10	0.11	0.10	1.00	1.00
10×10	0.12	0.10	0.10	0.10	1.00	0.92



(**a**) Neural Cleanse for all-bands BadNets

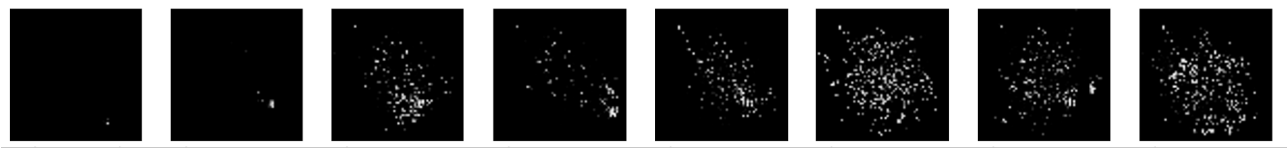


(**b**) TABOR for all-bands BadNets

(**c**) Neural Cleanse for less-important-bands BadNets

(**d**) TABOR for less-important-bands BadNets

(**e**) Neural Cleanse for important-bands BadNets

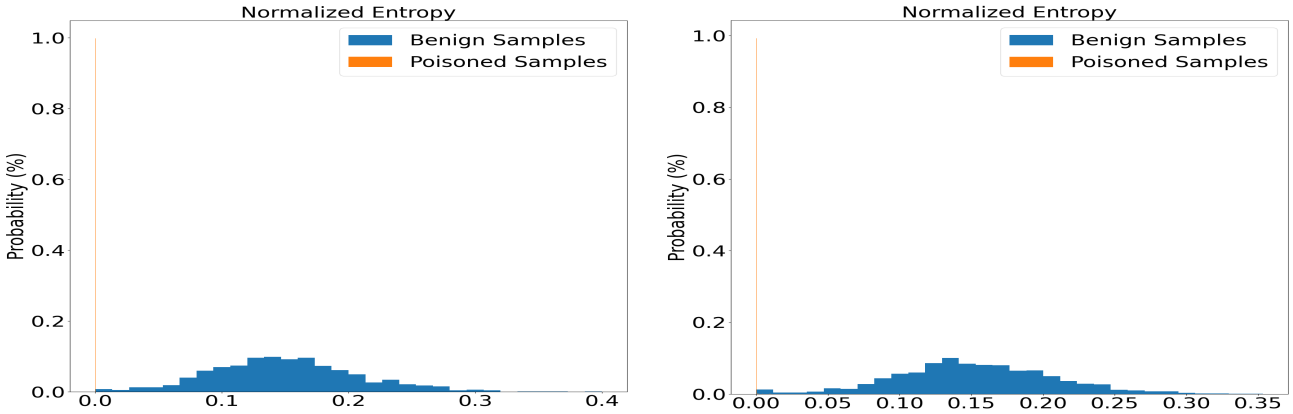


(**f**) TABOR for important-bands BadNets

Figure 5. Triggers generated by different defense methods for all-band and sub-band BadNets attacks.

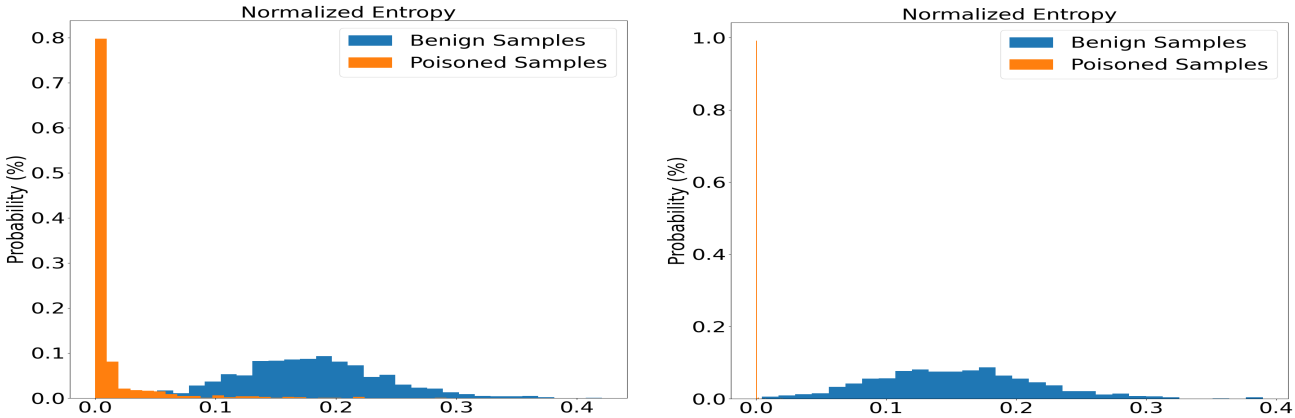
9.2. Case Study 2: STRIP

Table 5 shows that STRIP failed for most of the cases. STRIP uses the entropy of the predicted probabilities for classes in the dataset to detect poisoned samples. For a backdoored model, poisoned samples will show a low entropy and the embedded trigger makes the prediction strongly towards the target class, while benign samples will show a larger entropy. We visually compare the entropy performances of STRIP in all-band, important-band, and less-important-band BadNets in Figures 6–8. It was observed that the all-band BadNets attack has a lower entropy distribution in Figure 6 compared with important-bands attack (Figure 7) and less-important-bands attack (Figure 8), indicating that STRIP faces more challenges in detecting sub-band BadNets backdoor attacks.



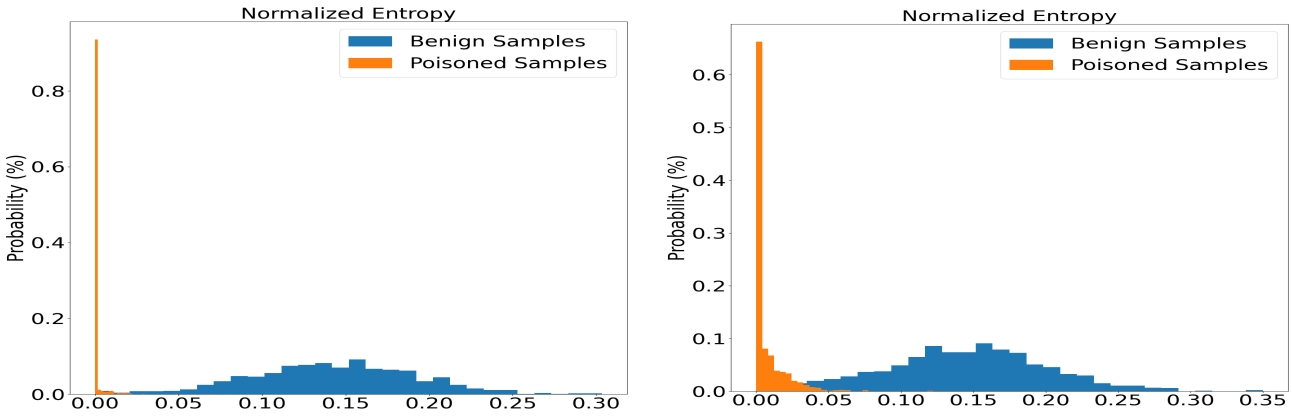
(**a**) Trigger Size of 2 × 2. (**b**) Trigger Size of 10 × 10.

Figure 6. Entropy distributions computed by STRIP for benign and poisoned samples for all-bands BadNets backdoor attack with the EuroSat Dataset. Benign samples typically have larger entropy (blue), while poisoned samples have small entropy (orange). Entropy for poisoned images are all close to 0, making it easier to be detected.



(**a**) Trigger Size of 2 × 2. (**b**) Trigger Size of 10 × 10.

Figure 7. Entropy distributions computed by STRIP for benign and poisoned samples for the important-band BadNets backdoor attack with the EuroSat dataset. Some poisoned samples have larger entropy, making them less easy to detect for a trigger size of 2 × 2.

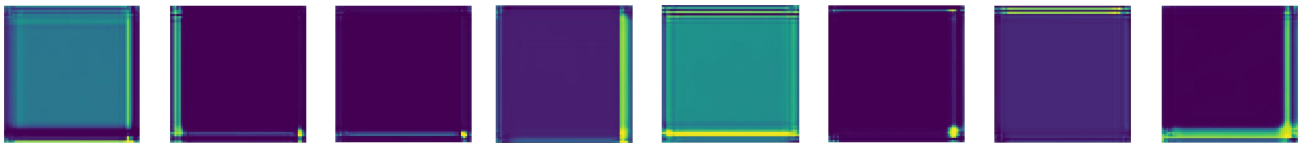


(**a**) Trigger Size of 2×2. (**b**) Trigger Size of 10×10.

Figure 8. Entropy distributions computed by STRIP for benign and poisoned samples for the less-important-band BadNets backdoor attack with the EuroSat dataset. Some poisoned samples have larger entropy, making them less easy to be detected for a trigger size of 10×10.

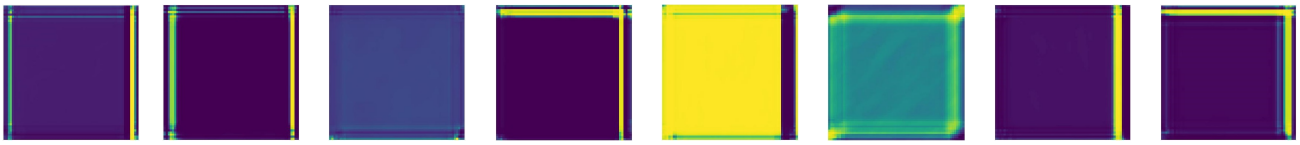
9.3. GangSweep

GangSweep was successful for two out of nine cases for all-band BadNets attacks and one out of nine cases for the less-important-bands scenarios, as shown in Table 5. We display the reverse-engineered triggers by GangSweep in Figures 9 and 10. GangSweep was only able to correctly generate the backdoor triggers of sizes 4×4 (Figure 9c) and 7×7 (Figure 9c), respectively, while it failed all other cases.



(**a**) 2×2 (**b**) 3×3 (**c**) 4×4 (**d**) 5×5 (**e**) 6×6 (**f**) 7×7 (**g**) 8×8 (**h**) 9×9

Figure 9. Reverse engineered triggers by Gangsweep for all-band BadNets backdoored deep convolutional neural network (DCNN) model with EuroSat dataset.



(**a**) 2×2 (**b**) 3×3 (**c**) 4×4 (**d**) 5×5 (**e**) 6×6 (**f**) 7×7 (**g**) 8×8 (**h**) 9×9

Figure 10. Reverse engineered triggers by Gangsweep for important-band BadNets backdoored deep convolutional neural network (DCNN) model with EuroSat dataset.

10. Discussion

We implemented the all-band and sub-band BadNets attacks on the EuroSat remote sensing dataset, where the BadNets attack achieved an attack success rate of 100% across various trigger sizes. The sub-band BadNets attack comprises two types: the less-important-band attack and the important-band attack. In these attacks, triggers were embedded into bands categorized as less important or important, as identified by the Score-CAM algorithm. Subsequently, we applied defense approaches, including Neural Cleanse, TABOR, Deep Inspect, Gang-sweep, Pixel Backdoor, and Activation Clustering, to counter the backdoor attacks. In most cases, the defense mechanisms failed to eliminate the backdoor. Notably, the important-band BadNets posed a more severe threat compared to other types of attacks.

Neural Cleanse is a popular backdoor defense method, capable of evaluating a backdoored model offline and reverse engineering the trigger. We applied Neural Cleanse to

various BadNets settings on a multispectral remote sensing dataset, utilizing a variety of convolutional network backbones, such as DCNN, ResNet-20, ResNet-56, and Efficient-Net. Although the Efficient-Net architecture achieved the highest classification accuracy compared to the other backbones, it was more vulnerable to backdoor attacks than the other architectures. Neural Cleanse correctly detected the poisoned target label in only five out of nine instances, proving less effective in satellite imagery, in contrast to its success in natural imagery [12].

If the victim is aware of the backdoor target label, they can use the reverse-engineered trigger, as shown in Table 7, to remove the backdoor from the deep learning model. During the unlearning process, we embedded the reverse-engineered trigger in 20% of the clean training dataset and then combined these poisoned samples with an additional 10% of the clean dataset to remove the backdoor. The model was trained for 50 epochs in each scenario, as detailed in Table 7. The unlearning process proved most effective in the all-band BadNets attack models. Upon comparing the unlearning performance across different patch sizes in Table 7, we managed to reduce the ASR from 99% to 10%. While TABOR achieved similar performance in the unlearning settings, its trigger detection performance was less effective compared to the Neural Cleanse approach. These results underscore the challenge of detecting and mitigating backdoors without prior knowledge of the target label.

We investigated the sub-band BadNets attack in the EuroSat satellite imagery and discovered that defending against the important-band BadNets attacks, where the trigger is embedded in bands identified as important by the Score-CAM algorithm, is particularly challenging. Even when the target class is known, Neural Cleanse is unable to remove the backdoor in these cases. In contrast, Neural Cleanse can effectively remove the BadNets backdoor if the trigger is embedded in less important bands or all bands of the EuroSat dataset. Additionally, we implemented defense strategies such as TABOR, Strip, Fine-pruning, and GangSweep against the BadNets attack. TABOR and GangSweep showed performances similar to that of the Neural Cleanse approach.

11. Conclusions

We evaluated the susceptibility of remote sensing imagery to backdoor attack threats. Satellite imagery-based neural network models are highly susceptible to such attacks. By poisoning only 10% of the images, we achieved a 100% ASR in satellite imagery. We compared traditional defense approaches, such as Neural Cleanse, Activation Clustering, TABOR, Deep Inspect, GangSweep, Pixel Backdoor, and Fine-Pruning. These methods were unsuccessful in identifying backdoored models in the remote sensing domain. While STRIP successfully identified the backdoor, its effectiveness in consistently detecting backdoored models is limited, as it requires poisoned samples. Without access to poisoned samples, STRIP fails to identify backdoored models. In addition, distributing poisoned datasets to victims poses a significant challenge in practice, especially as most remote sensing models undergo close scrutiny. Our findings highlighted the need for increased attention in this emerging field.

Author Contributions: Conceptualization, K.A.I., H.W., C.X., R.N. and J.L.; methodology, K.A.I., H.W., R.N. and J.L.; software, K.A.I.; validation, K.A.I., H.W. and J.L.; formal analysis, K.A.I.; investigation, K.A.I.; resources, R.N. and L.Z.; data curation, K.A.I.; writing—original draft preparation, K.A.I. and J.L.; writing—review and editing, K.A.I., H.W., C.X., L.Z. and J.L.; visualization, K.A.I.; supervision, H.W., C.X. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Commonwealth Cyber Initiative (CCI)'s Cyber Acceleration, Translation, and Advanced Prototyping for University Linked Technology (CATAPULT) Fund; NSF's CNS-2153358, CNS-2120279 and OAC-2320999 grants and Intergovernmental Personnel Act Independent Research and Development Program; NSA H98230-21-1-0165; ARFL FA8750-19-3-1000; DoD Center of Excellence in AI and Machine Learning (CoE-AIML) under Contract Number W911NF-20-2-0277 grants.

Data Availability Statement: The data will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Activation Clustering
Acc	Accuracy on clean data
AI	Artificial Intelligence
ASR	Attack Success Rate
DCNN	Deep Convolutional Neural Network
DI	DeepInspect
DOD	Department of Defense
FP	Fine-Pruning
GAN	Generative Adversarial Network
GIS	Geographic Information System
GS	Gangsweep
MAD	Median Absolute Deviation
NASA	National Aeronautics and Space Administration
NC	Neural Cleanse
NIR	Near-Infrared
NOAA	National Oceanic and Atmospheric Administration
PB	Pixel Backdoor
PCA	Principal Component Analysis
SOTA	State-Of-The-Art
STRIP	STRong Intentional Perturbation
UAV	Unmanned Aerial Vehicle

Appendix A

Appendix A.1. Sat-4, and Sat-6 Dataset

Sat-4 and Sat-6 [58] airbone datasets are extracted from National Agriculture Imagery Program (NAIP) dataset [58]. The NAIP dataset has a total of 330,000 continental United States scenes from TIFF formats uncompressed digital Ortho quarter quad tiles (DOQQs). These images have 1 m spatial resolution with red, green, blue, and near-inferred bands.

Sat-6 dataset has barren land, trees, grassland, roads, buildings, and water bodies classes. Sat-6 has 324,000 training and 81,000 samples, each image has a dimension of $28 \times 28 \times 4$. Sat-4 dataset has barren land, trees, grassland, and a class that consists of all land cover classes. Sat-4 has 400,000 training and 100,000 samples, each image has a dimension of $28 \times 28 \times 4$.

Appendix A.2. So2Sat Dataset

So2Sat dataset consists of co-registered synthetic aperture radar (SAR) and multispectral optical image patches collected from Sentinel-1 and Sentinel-2 satellites [59]. The So2sat dataset has B2, B3, B4, B8 with 10 m GSD, bands B5, B6, B7, B8a, B11, B12 with 20 m. Each image has a height and width of 32. The input size of the dataset is $32 \times 32 \times 10$. We used all 17 classes, e.g., Compact high-rise, Compact mid-rise, Compact low-rise, Open high-rise, Open mid-rise, Open low-rise, Lightweight low-rise, Large low-rise, Sparsely built, Heavy industry, Dense trees, Scattered tree, Bush-scrub, Low plants, Bare rock or paved, Bare soil or sand, and Water. The training image and testing image sizes are 352,366 and 24,188 respectively.

Appendix A.3. BadnNets Attack in Sat-4, Sat-6, and So2Sat

We plant the backdoor trigger in the DCNN and ResNet architecture using BadNets attack (Section 3) for trigger size from 2×2 to 10×10. We achieved a superb attack success rate for these trigger sizes, as shown in Table A1.

Table A1. Performances of all-band BadNets attack on Sat-4, Sat-6, and So2Sat dataset. Sat-4 and So2Sat with deep convolutional neural network (DCNN) architecture and Sat-6 with ResNet20 architecture.

TS	Sat-4 (DCNN)			Sat-6 (ResNet-20)			So2Sat (DCNN)		
	Acc_{Clean}	Acc_B	ASR	Acc_{Clean}	Acc_B	ASR	Acc_{Clean}	Acc_B	ASR
2×2		1.00	1.00		1.00	1.00		0.63	1.00
3×3		1.00	1.00		1.00	1.00		0.62	1.00
4×4		1.00	1.00		1.00	1.00		0.63	1.00
5×5		1.00	1.00		1.00	1.00		0.61	1.00
6×6	1.00	1.00	1.00	1.00	1.00	1.00	0.62	0.63	1.00
7×7		1.00	1.00		1.00	1.00		0.61	1.00
8×8		1.00	1.00		1.00	1.00		0.62	1.00
9×9		1.00	1.00		1.00	1.00		0.62	1.00
10×10		1.00	1.00		1.00	1.00		0.60	1.00

Appendix A.4. Backdoor Detection Performance of Neural Cleanse and TABOR

We found that Neural Cleanse was not able to detect the backdoor target label for the BadNets trigger in most scenarios. In the Sat-4 dataset, Neural Cleanse detected backdoor labels two times out of nine times and six out of nine times for DCNN and ResNet-20, respectively, as shown in Table A2. In the Sat-6 dataset, Neural Cleanse failed for all scenarios for DCNN and succeeded two times out of nine times for the ResNet-20 architecture, as shown in Table A2. In the So2Sat dataset, it succeeded one time out of nine times for DCNN and one time out of six times for ResNet-56 architecture, respectively.

Table A2. Sat4, Sat6, and So2Sat Backdoor detection performances using the Neural Cleanse method for all-band BadNets attack. If the backdoor label is correctly detected by Neural Cleanse, it is marked with ✓, otherwise with the × sign. 'TS' indicates trigger size and T indicates the poison target class.

TS	Sat-4(4B) T-3		Sat-6(4B) T-5		So2sat(10B) T-8	
	DCNN	ResNet-20	DCNN	ResNet-20	DCNN	ResNet-56
2×2	✓	×	×	✓	×	×
3×3	×	✓	×	×	×	×
4×4	×	✓	×	×	✓	
5×5	×	✓	×	×	×	
6×6	×	×	×	×	×	
7×7	×	✓	×	×	×	✓
8×8	×	✓	×	×	×	×
9×9	×	×	×	×	×	×
10×10	✓	✓	×	✓	×	×

References

1. Islam, K.A.; Hill, V.; Schaeffer, B.; Zimmerman, R.; Li, J. Semi-Supervised Adversarial Domain Adaptation for Seagrass Detection in Multispectral Images. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1120–1125.
2. Islam, K.A.; Uddin, M.S.; Kwan, C.; Li, J. Flood detection using multi-modal and multi-temporal images: A comparative study. *Remote Sens.* **2020**, *12*, 2455. [CrossRef]
3. ESRI Website. Available online: https://www.esri.com/about/newsroom/blog/how-maps-guided-9-11-response-and-recovery/ (accessed on 31 December 2023).
4. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [CrossRef]
5. Huang, W.; Xiao, L.; Wei, Z.; Liu, H.; Tang, S. A new pan-sharpening method with deep neural networks. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1037–1041. [CrossRef]
6. Lu, T.; Wang, J.; Zhang, Y.; Wang, Z.; Jiang, J. Satellite image super-resolution via multi-scale residual deep neural network. *Remote Sens.* **2019**, *11*, 1588. [CrossRef]

7. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
8. Shafahi, A.; Huang, W.R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Red Hook, NY, USA, 3–8 December 2018; Volume 31.
9. Czaja, W.; Fendley, N.; Pekala, M.; Ratto, C.; Wang, I.J. Adversarial examples in remote sensing. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 6–9 November 2018; pp. 408–411.
10. Cheng, G.; Xie, X.; Han, J.; Guo, L.; Xia, G.S. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3735–3756. [CrossRef]
11. Gu, T.; Liu, K.; Dolan-Gavitt, B.; Garg, S. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access* **2019**, *7*, 47230–47244. [CrossRef]
12. Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; Zhao, B.Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 707–723.
13. Guo, W.; Wang, L.; Xing, X.; Du, M.; Song, D. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv* **2019**, arXiv:1908.01763
14. Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; Srivastava, B. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv* **2018**, arXiv:1811.03728
15. Liu, K.; Dolan-Gavitt, B.; Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses, Crete, Greece, 10–12 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 273–294.
16. Zhu, L.; Ning, R.; Wang, C.; Xin, C.; Wu, H. Gangsweep: Sweep out neural backdoors by gan. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 3173–3181.
17. Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D.C.; Nepal, S. Strip: A defence against trojan attacks on deep neural networks. In Proceedings of the 35th Annual Computer Security Applications Conference, San Juan, PR, USA, 9–13 December 2019; pp. 113–125.
18. Chen, H.; Fu, C.; Zhao, J.; Koushanfar, F. DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; Volume 2, p. 8.
19. Tao, G.; Shen, G.; Liu, Y.; An, S.; Xu, Q.; Ma, S.; Li, P.; Zhang, X. Better Trigger Inversion Optimization in Backdoor Scanning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 13368–13378.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
23. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef]
24. Dai, T.; Cai, J.; Zhang, Y.; Xia, S.T.; Zhang, L. Second-order attention network for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11065–11074.
25. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
26. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619 . [CrossRef]
27. Liu, Y.; Wang, Y.; Wang, S.; Liang, T.; Zhao, Q.; Tang, Z.; Ling, H. Cbnet: A novel composite backbone network architecture for object detection. *arXiv* **2019**, arXiv:1909.03625.
28. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
29. Yuan, Q.; Shen, H.; Li, T.; Li, Z.; Li, S.; Jiang, Y.; Xu, H.; Tan, W.; Yang, Q.; Wang, J.; et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sens. Environ.* **2020**, *241*, 111716. [CrossRef]
30. Lee, J.G.; Jun, S.; Cho, Y.W.; Lee, H.; Kim, G.B.; Seo, J.B.; Kim, N. Deep learning in medical imaging: General overview. *Korean J. Radiol.* **2017**, *18*, 570–584. [CrossRef] [PubMed]
31. Sahiner, B.; Pezeshk, A.; Hadjiiski, L.M.; Wang, X.; Drukker, K.; Cha, K.H.; Summers, R.M.; Giger, M.L. Deep learning in medical imaging and radiation therapy. *Med. Phys.* **2019**, *46*, e1–e36. [CrossRef] [PubMed]

32. Zhang, J.; Pan, L.; Han, Q.L.; Chen, C.; Wen, S.; Xiang, Y. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 377–391. [CrossRef]
33. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Kingsbury, B.; et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
34. Hershey, S.; Chaudhuri, S.; Ellis, D.P.; Gemmeke, J.F.; Jansen, A.; Moore, R.C.; Plakal, M.; Platt, D.; Saurous, R.A.; Seybold, B.; et al. CNN architectures for large-scale audio classification. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 131–135.
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
36. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
37. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
38. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
39. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. The German traffic sign recognition benchmark: A multi-class classification competition. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1453–1460.
40. LeCun, Y.; Cortes, C.; Burges, C. MNIST Handwritten Digit Database. 2010. Available online: http://yann.lecun.com/exdb/mnist/index.html (accessed on 20 February 2024).
41. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 20 February 2024).
42. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–17 December 2011; Volume 2011, p. 7.
43. Guo, W.; Wang, L.; Xu, Y.; Xing, X.; Du, M.; Song, D. Towards inspecting and eliminating trojan backdoors in deep neural networks. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Virtual, 17–20 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 162–171.
44. Liu, Y.; Ma, X.; Bailey, J.; Lu, F. Reflection backdoor: A natural backdoor attack on deep neural networks. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 182–199.
45. Yao, Y.; Li, H.; Zheng, H.; Zhao, B.Y. Latent backdoor attacks on deep neural networks. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2041–2055.
46. Brewer, E.; Lin, J.; Runfola, D. Susceptibility & defense of satellite image-trained convolutional networks to backdoor attacks. *Inf. Sci.* **2022**, *603*, 244–261.
47. Yang , Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
48. Brewer, E.; Lin, J.; Kemper, P.; Hennin, J.; Runfola, D. Predicting road quality using high resolution satellite imagery: A transfer learning approach. *PLoS ONE* **2021**, *16*, e0253370. [CrossRef] [PubMed]
49. Dräger, N.; Xu, Y.; Ghamisi, P. Backdoor Attacks for Remote Sensing Data with Wavelet Transform. *arXiv* **2022**, arXiv:2211.08044.
50. Islam, S.; Badsha, S.; Khalil, I.; Atiquzzaman, M.; Konstantinou, C. A Triggerless Backdoor Attack and Defense Mechanism for Intelligent Task Offloading in Multi-UAV Systems. *IEEE Internet Things J.* **2022**, *10*, 5719–5732. [CrossRef]
51. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**. [CrossRef]
52. Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; Hu, X. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 24–25.
53. Dahl, G.E.; Sainath, T.N.; Hinton, G.E. Improving deep neural networks for LVCSR using rectified linear units and dropout. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 8609–8613.
54. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
55. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 839–847.
56. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [CrossRef] [PubMed]

57. Bäuerle, A.; Van Onzenoodt, C.; Ropinski, T. Net2vis–a visual grammar for automatically generating publication-tailored cnn architecture visualizations. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 2980–2991. [CrossRef] [PubMed]
58. Basu, S.; Ganguly, S.; Mukhopadhyay, S.; DiBiano, R.; Karki, M.; Nemani, R. Deepsat: A learning framework for satellite imagery. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, 3–6 November 2015; pp. 1–10.
59. Zhu, X.X.; Hu, J.; Qiu, C.; Shi, Y.; Kang, J.; Mou, L.; Bagheri, H.; Haberle, M.; Hua, Y.; Huang, R.; et al. So2Sat LCZ42: A Benchmark Data Set for the Classification of Global Local Climate Zones [Software and Data Sets]. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 76–89. [CrossRef]