GridNet: Vision-based Mitigation of GPS Attacks for Aerial Vehicles

Nian Xue and Zhen Li
Shandong University of Technology
Shandong, China
{xuenian,legion}@sdut.edu.cn

Xianbin Hong[†]
University of Liverpool
Liverpool, UK
xianbin.hong@liverpool.ac.uk

Christina Pöpper[*]
New York University Abu Dhabi
Abu Dhabi, UAE
christina.poepper@nyu.edu

***Abstract*—Navigation services based on Global Navigation Satellite Systems (GNSS) are essential for a wide range of global applications, including ensuring the safety of aerial vehicles. However, these GNSS signals (e. g., civilian GPS) are vulnerable to jamming and spoofing attacks. To mitigate such threats, we propose *GridNet*, a vision-based deep-learning technique to counter GPS jamming and spoofing attacks on aerial vehicles. GridNet uses visual devices (e.g., cameras) to determine geolocation during such attacks by extracting geolocation from aerial photos via a pre-trained neural network model. This non-invasive approach does not modify existing GPS infrastructure, merely relying on real-time visual data. Unlike other methods that focus solely on attack detection, GridNet provides a countermeasure, calculating geolocation data without GPS. We analyze the potential applications and discuss the performance in different flight environments. Experimental results show that GridNet can extract geolocation data from real aerial photos, achieving nearly 93% spoofing detection accuracy with a minimum average geolocation error of 7.33 meters within 4ms on a ground station server and 2.7ms on a typical UAV, respectively, offering a practical anti-spoofing solution.**

***Index Terms*—Aircraft and drone security, GPS spoofing attacks, GPS jamming attacks, deep learning, CycleGAN.**

I. Introduction

Satellite-based navigation has become indispensable in modern aviation due to its critical role in ensuring safety, efficiency, and reliability. Global Positioning System (GPS), the most well-known and the first GNSS system, is widely used as the primary means of acquiring accurate positioning data and navigation information. It has allowed civilian users to receive a non-degraded signal globally since 2000 [1]. Since then, the aviation industry has relied strongly on GPS signals for positioning, navigation, and timing (PNT) services.

However, GPS signals are highly vulnerable to spoofing and jamming, even with low-cost commercial off-the-shelf (COTS) SDRs such as a HackRF One [2]. GPS jamming attacks try to block legitimate GPS signals or interfere with the target victims to prevent them from receiving the signals from the GPS satellites. GPS spoofing attacks send similar and more powerful but counterfeit GPS signals to manipulate victims' localization. Media reports concerning GPS attacks in aviation include the well-known Iran-U.S. RQ-170 incident, where an American UAV was captured by Iranian forces near the city of Kashmar in 2011 due to jamming satellite signals, followed by a GPS spoofing attack [3]. One accidental jamming attack near Newark Airport, New Jersey [4], and a spoofing attack close to Kremlin, Moscow [5], were reported in 2012 and 2016, respectively. Those publicly known events reveal that GPS attacks are no longer just theoretically feasible but are happening in reality.

While cryptographic authentication methods [6] can mitigate GPS spoofing, they typically require hardware modifications to the receiver and cannot counter jamming attacks that deny access to signals entirely. GPS navigation recovery is infeasible when attackers can null or jam the authentic signals to the point where they are unrecoverable. Furthermore, in GPS-denied areas (e. g., military zones, valleys, and urban canyons), getting authentic GPS signals is almost impracticable, let alone recovering them. Alternative navigation systems relying on optical flow or inertial sensors often suffer from reliability and accuracy issues, particularly over long durations [7].

Recently, vision-based navigation has gained increasing attention as a promising alternative. However, researchers still encounter several practical challenges. First, aerial images of the target area may be unavailable for training purposes. This leads to a lack of sufficient data on target operating airspaces to train neural network classifiers. Unlike satellite images that provide global coverage, aerial photos typically do not achieve large-scale coverage (e. g., city-level), at least they are not publicly available. Second, although both aerial and satellite images are known as remotely sensed images, the two techniques for creating images differ. Essentially, satellite images generally cover a much wider area, whereas aerial photos are taken at a lower altitude, and thus cover a smaller area. In addition, different photography equipment configurations also have a great impact on photos. Such factors lead to fundamental differences between satellite images and aerial photos in features such as resolution, rotation, angle, and color features (e. g., hue, saturation, and brightness). Third, compared with satellite images that may have been captured years ago, aerial images are taken in real time. As a result, transient phenomena, such as seasonal vegetation, weather, people, vehicles, and light, vary greatly at different times. These variations should not be considered as a basis for localization. Finally, deep neural networks have achieved great

[*]Corresponding author.
[†]Dr. Xianbin Hong passed away before the final acceptance of this paper. This paper is dedicated to his memory and contributions.

success in image classification tasks; however, how to properly balance the trade-off between model accuracy and complexity and carry it out in a timely fashion on a resource-constrained UAV platform remains an open problem.

Therefore, there is a critical need for a robust vision-based anti-spoofing system that does not require access to live GPS signals. In response to these challenges, we design and implement **GridNet**, a technique that allows a camera-equipped UAV to (1) detect spoofing attacks based on aerial photos and (2) determine its location in the absence of (legitimate) GPS signals. Our approach is inherently resistant to attacks on wireless signals. By dividing the target area into a grid structure composed of rectangular cells, each individual grid unit is referred to as a "grid cell". Our key idea is to exploit robust deep learning visual features in aerial images taken by an onboard camera to locate the aircraft. In essence, GridNet leverages the natural features (e. g., rivers, roads, buildings, etc.) of the landmarks on the earth and maps them with a specific grid cell on a specified satellite map to geolocation (i. e., latitude, longitude). This approach maintains high performance even when the aerial images of the target area are lacking. In addition, we adopt Cycle Generative Adversarial Network (CycleGAN) [8] to mitigate the challenge of the difference between aerial and satellite images. Meanwhile, to address the challenge of deploying models on resource-constrained UAV platforms, we design a balanced trade-off between model accuracy and computational complexity. The experimental results of the real scenario show that the accuracy, precision, recall, and F1 score of GridNet under various challenging conditions all surpass 90%, demonstrating its practicality for real-world applications. The main contributions of our paper are as follows:

- To our knowledge, this is the first work that employs GAN-based methods to derive geolocation solely from UAV images. This strategy circumvents the need to collect aerial training images from the target geographic area.
- We design a lightweight, deployable on-board anti-spoofing system for UAVs, achieving a balanced trade-off between model accuracy and computational efficiency.
- GridNet is a standalone localization system and is a potential alternative to GNSS. We analyze its localization error through theory and experiments. The experiments show that we achieve an average localization error ranging from 7 to 36 meters, depending on flight setting and grid resolution.

The code and models are publicly available at https://github.com/Goldgaruda/GridNet.

II. Preliminaries and Related Work

This section provides background on GNSS attacks and countermeasures, prior work, and key techniques for GridNet.

A. GNSS Overview and Security Challenges

Global Navigation Satellite Systems (GNSSs) refer to satellite constellations that emit signals from space and send PNT data to ground-based receivers. Widely used GNSSs include the U.S. GPS, Russia's GLONASS, China's BeiDou, and the European Union's Galileo. Due to the lack of signal encryption and authentication in civilian systems, GNSS signals are vulnerable to attacks such as jamming and spoofing. Jamming involves transmitting interference signals to block or distort GNSS signals, causing receivers to lose lock on satellite transmissions. For instance, GPS signals are weak (approx. -160 dBW [9]), making them susceptible to noise-based disruption, which may lead to UAV crashes [10]. Spoofing, by contrast, involves broadcasting fake signals that mimic GNSS signals at higher power levels, causing receivers to calculate incorrect positions [11], [12]. Hybrid attacks combine jamming and spoofing, making them harder to defend against.

B. Countermeasures against GNSS Attacks

Traditional countermeasures, named GNSS Electronic Protection Measures (EPM), operate in time, frequency, and spatial domains [13]. Cryptographic protection of navigation messages can mitigate spoofing but has high overhead and does not prevent replay attacks [14]. Non-cryptographic methods, such as hardware-assisted solutions [15] and crowdsourcing-based detection [16], require additional hardware or cooperation from multiple participants, and they only detect spoofing without recovering geolocation information. More details can be found in [13], [17]. Due to the wide variety of UAV models and lack of production standards, scalable and easily adaptable spoofing detection methods are needed. We propose a vision-based approach without relying on GPS signals, thereby simplifying the system's creation and implementation.

C. Prior Work

UAVs depend on vulnerable civilian GPS for navigation, making them prone to spoofing and jamming. For example, in urban environments, GPS performance deteriorates due to occlusion and interference, especially in urban canyons. Countermeasures are typically classified into signal-based and out-of-band methods.

Signal-based methods, like cryptographic techniques [18], authenticate GPS signals but remain vulnerable to replay attacks [14], and need infrastructure upgrades. Alternative signal anomaly detection approaches, while theoretically effective, demand specialized instruments that are unsuitable for UAVs.

Out-of-band techniques use environmental and system information to cross-validate navigation data. Visual information [19], optical flow [20], and IMU [21], provide extra sources for detection. IMU-based methods utilize raw measurements for drift-prone relative position estimation [21], but lose accuracy over time due to integral drift [22]. Visual sensors like cameras assist UAVs in maneuvering and providing real-time video feeds. Early optical flow-based methods lacked precision [23], and feature matching methods like SIFT and ORB failed due to viewpoint differences [24]. Recent deep learning methods are robust but require maps [13], [25]. Our mapless GridNet detects spoofing and jamming attacks using one input without maps, providing real-time use on resource-limited vehicles. Comparisons with recent vision-based deep learning methods are summarized in Table VI.

D. Key Techniques for GridNet

Deep Neural Networks (DNNs), particularly CNNs have revolutionized computer vision (CV). CNNs enables automatic feature extraction from raw pixel data, achieving breakthroughs in CV tasks [26]–[28]. In this work, we adopt a vision-based deep learning approach to counter jamming and spoofing threats in aviation. We employ CNN-based DNNs within GridNet for aerial image classification due to their independence from prior knowledge and human intervention.

To address the domain shift between UAV and satellite imagery, we adopt CycleGAN [8], an unsupervised image translation model that learns without paired samples and supports domain adaptation. It transfers input images into the target style and is widely used in satellite image enhancement. In this paper, we use CycleGAN to map UAV aerial photos to satellite-style images, effectively bridging the domain gap.

III. Adversary Model

We assume a strong adversary capable of launching three kinds of GPS attacks: (i) jamming attacks, (ii) spoofing attacks, and (iii) hybrid attacks. All attacks can be achieved with purpose-built hardware (i. e., GPS satellite simulators), using general-purpose Universal Software Radio Peripherals (USRPs) [29] and corresponding software packages (e. g., `gps-sdr-sim`), or even cheap SDR hardware platforms [30]. In all cases, amplifiers will be needed. We make the standard assumption [6] of an attacker being able to jam and spoof GPS signals that affect the victim's capability to acquire legitimate GPS signals and its capability to determine its real-time location. The number of attackers may range from a single adversary to multiple collaborating adversaries who can conduct complex and flexible attacks. More importantly, we consider the extreme case that the authentic GPS signals are completely compromised and cannot be recovered by any signal-based method.

Furthermore, we assume that the attacker is able to not only jam the GPS signals but also jam the communication channel (e. g., OFDM and WiFi) between the UAV and the ground station (e. g., by a drone jamming gun or a drone defender). As a result, typical existing communication channels between the UAV and the ground station may be cut off.

We restrict the attacker's capabilities with respect to the knowledge about the used ML models: we assume that the attacker does neither have access to the neural network classifier/model used in GridNet nor to the used parameters. For example, the Fast Gradient Sign Method (FGSM) [31] can mislead geolocation predictions by adding imperceptible perturbations to aerial photos. We also assume that the attacker is *not* able to artificially make topographic changes (e. g., highway alignment, river flows, building layout) of the operating area during the flight planning and operation.

IV. GridNet Approach

We propose GridNet to detect GPS spoofing attacks on UAVs and to recover the UAV's physical location under such attacks. Moreover, GridNet can also operate in GPS-denied airspace to assist with navigation. To achieve this, the key idea is to geolocate the UAV without relying on GPS signals, which may have been completely compromised by attackers.

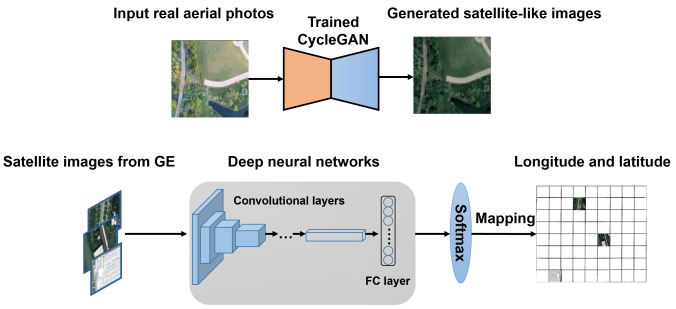

Fig. 1: Overview of GridNet offline phase. The top part shows an input aerial photo transferred to satellite-like images using a trained CycleGAN. The bottom part exhibits satellite images classified into a corresponding grid by a deep neural network.

A. Design Overview of GridNet

The GridNet system consists of a UAV equipped with at least one visual sensor, a ground station, and communication channels between them. In addition, there are three major components: a CycleGAN, a deep neural network classifier, and a location extraction module (as explained in Sec. IV-B) installed on the UAV or the ground controller, depending on the setup (installing them on the UAV makes GridNet independent of an active communication channel between the UAV and the ground station).

In practice, the proposed method can be divided into an offline phase for training and an online phase for operations.

Offline Phase. In the offline phase (see Fig. 1), we train a CycleGAN using unpaired aerial photographs and satellite images to translate aerial photos into satellite-style images. These translated images are then used to train a classifier for satellite image classification tasks. The training process of CycleGAN is the same as [8] and is not elaborated here. This technique is useful for operating areas where no aerial pictures are available for training and where no satellite map is needed/required during the online/execution phase.

Online Phase. During the online phase, the UAV operates in an unvisited area where satellite imagery is available. The classifier and CycleGAN, trained during the offline phase, can be deployed either on the ground station—as an alternative to GPS-based localization in the absence of GPS attacks—or on the UAV's onboard computer (OBC) to counteract GPS attacks. The advantage of an on-board deployment on the UAV is the independence of the UAV's operation from a possibly jammed communication channel. The live aerial photos or the frames extracted from the video taken by the UAV are transferred to satellite images using CycleGAN and then fed to a pre-trained classifier and a location extraction module to obtain the geolocation information (see Fig. 2). The extracted geolocation can then be cross-checked with the

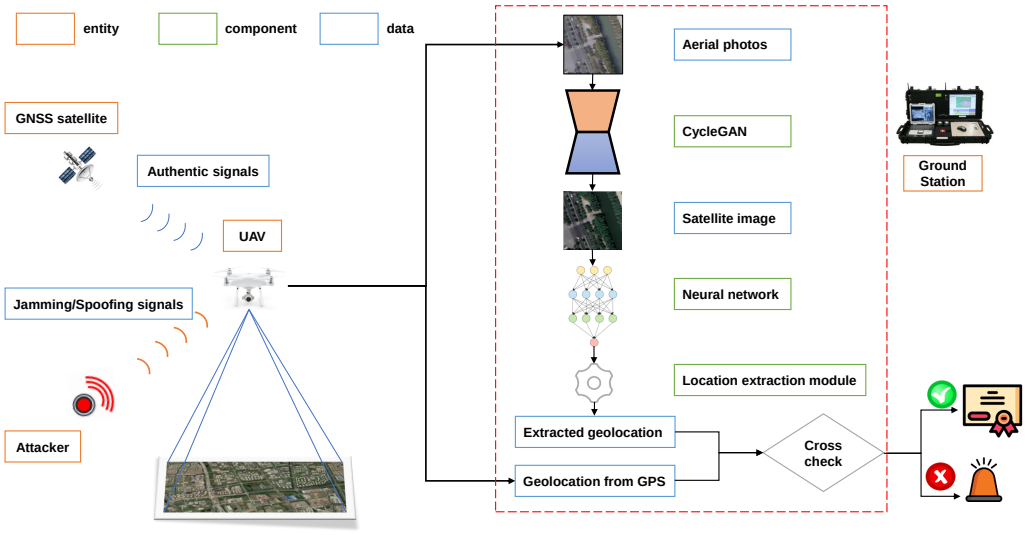

Fig. 2: Overview of GridNet online phase.

calculated geolocation from GPS signals (under spoofing) or assist navigation (under jamming).

Once the communication link between the UAV and the ground station is disrupted by an attacker, the on-board model is activated. By using the geolocation extraction function, the victim may move to safe airspace or return home. Many UAVs will land directly by default once the GPS signals are lost, and there is often no backup navigation for UAVs. As a result, an attacker can easily capture a UAV by cutting off the GPS signals—a special attack that GridNet is designed to mitigate.

B. Detailed Components of GridNet

1) ***CycleGAN Domain Transfer from Aerial Photos to Satellite Images****:* As mentioned earlier, the core objective of GridNet is to train a classifier that can later be used to extract a UAV's geolocation. In general, training a high-performance classifier is a well-established process. However, in real-life scenarios, the aerial photos captured by UAVs that would be needed for training the classifier are often unavailable beforehand. It is, in contrast, easy to get satellite images of the target area in advance. Nevertheless, if we train our classifier using training data solely on satellite images (i. e., the source domain) and then use the classifier to do classification tasks for aerial photos (i. e., the target domain), it will inevitably lead to degraded performance. The root cause is that the training data and the test data are not in the same feature space, and thus have different distributions.

Previous methods, such as [32], [33], primarily focus on learning image-to-image mappings from aligned training image pairs. However, in the context of UAVs, acquiring aerial imagery of the target area beforehand is often infeasible. Moreover, paired satellite-aerial images of a target area required for supervised training are typically unavailable.

To address this challenge, we employ CycleGAN to bridge the domain gap between aerial photos and satellite images. The objective of CycleGAN in GridNet is to learn a mapping function between two domains A (i. e., aerial photos) and B (i. e., satellite images) given unpaired training samples $\{x_i\}_{i=1}^{N} \in A$ and $\{y_j\}_{j=1}^{M} \in B$. The CycleGAN model consists of two mappings: $G_{AB} : A \rightarrow B$ and $G_{BA} : B \rightarrow A$. Besides, discriminator D_{AB} is proposed to discriminate between y and $G_{AB}(x)$. Conversely, discriminator D_{BA} distinguishes real aerial photos x from A and generated fake aerial photos $G_{BA}(y)$.

We apply the CycleGAN loss function [8]:

$$\begin{aligned}\mathcal{L} = &\mathcal{L}_{AB}\left(G_{AB}, D_{AB}, A, B\right) \\ &+ \mathcal{L}_{BA}\left(G_{BA}, D_{BA}, B, A\right) + \lambda \mathcal{L}_{cyc}\left(G_{AB}, G_{BA}\right),\end{aligned} \tag{1}$$

where $\mathcal{L}_{AB}$ and $\mathcal{L}_{BA}$ are two adversarial losses [34] of mapping functions, namely, $G_{AB} : A \rightarrow B$ and $G_{BA} : B \rightarrow A$, respectively. $\mathcal{L}_{cyc}$ is the cycle consistency loss [8], which guarantees that the generated output image is actually a version of the input image. λ is the control parameter of the relative importance of cycle consistency loss and adversarial losses, which is determined by a joint Bayesian optimization on parameters in [35], [36].

The aim is to solve the following equation:

$$G_{AB}, G_{BA} = \arg \min_{G_{AB}, G_{BA}} \max \mathcal{L}\left(G_{AB}, G_{BA}, D_{AB}, D_{BA}\right), \tag{2}$$

which is a min-max optimization function where the generator G wants to minimize the objective function, whereas the discriminator D tries to maximize it. By optimizing the above function, we can finally use the generator G_{AB} to transfer the real-time aerial photos to satellite images, which are fed to the classifier afterward.

2) ***Grid Classification by Deep Learning****:* We pursue grid classification for localization in three steps.

First, we define an *operating area* for the UAV and extract the target satellite images from public resources such as Google Earth, Google Maps, NASA WorldWind, or Mapbox.

The operating area can be defined broadly, as it is better to overestimate than to underestimate the area; it will, however, be limited by the coverage range of the UAV.

Second, we divide the satellite map of the operating area into regular grid cells, where each grid cell has its own latitude and longitude. For the sake of simplicity, we assume that the geolocation of the centroid of each grid cell represents the coordinates of the cell itself. The target area can be divided into different numbers of cells according to precision and efficiency requirements.

Third, we regard each grid cell as an independent class and train a neural network to classify them. As a result, we are able to construct a mapping between the grid cell and the corresponding coordinate. The classification neural network plays a core role in GridNet as we use it to classify each cell to the corresponding location class. The neural network is comprised of node layers, containing an input layer, a series of hidden layers, and an FC output layer. Immediately after the last FC layer, the output is fed into an n-way Softmax, which produces a normalized probability distribution over the M class predicted labels; M is the number of classes of the grid cell. By applying the various filters in each convolutional layer to an input image, the input image is abstracted into a feature map by the convolutional layers. These feature maps can be used to determine the class of each input cell image.

*3) **Location Extraction Module**:* To calculate the latitude and longitude of a specific grid cell, we first need to determine the coordinates of the target satellite bounds. In fact, most satellite map tools support this function of extracting the coordinates of a specified area. We assume the target area is A with known GPS bounds. Let the latitude and longitude of the upper left corner of A be (lon_s, lat_s) and the lower right corner of A be (lon_e, lat_e), see Fig. 3.

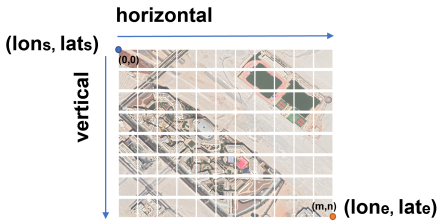

Fig. 3: Illustration of a grid cell location. Here $m = 11$, $n = 8$.

For the sake of simplicity, we assume that the area is in the northern hemisphere, and we regard the geolocation of a point in the center of the grid cell as the corresponding cell's latitude and longitude. The coordinate system starts at (0, 0), i. e., (horizontal number, vertical number) in the upper left corner and proceeds right (horizontal) and downward (vertical). In addition, we assume that the last cell ends at (m, n), which means that the satellite map is divided into $m \times n$ grids. Suppose that a certain grid cell g resides at A. Therefore, it is possible to figure out the cell g' GPS coordinate (lon_g, lat_g) by knowing its coordinate (x, y) and the extent or bounds of A such as lon_s, lat_s, lon_e and lat_e using Equations 3.

$$\begin{cases} lon_g = lon_s + \frac{(lon_e - lon_s)}{2m} + \frac{(lon_e - lon_s)}{m} x \\ lat_g = lat_s - \frac{(lat_s - lat_e)}{2n} - \frac{(lat_s - lat_e)}{n} y. \end{cases} \tag{3}$$

To locate the grid cell from aerial photos taken by aircraft (e.g., UAVs, commercial flights), we use pre-trained deep neural network classifiers to classify aerial photos into the corresponding satellite grid cell. Since each cell corresponds to a geographical position, we can calculate its geolocation. Of course, the precision will depend on the granularity of the cell and the accuracy of our neural network. We can balance precision and complexity according to different scenarios and requirements. Generally, the more cells we divide the area into, the more precise location results we can get, however, at the cost of efficiency, complexity, and prediction accuracy.

C. GridNet Benefits

Advantages. As an out-of-band method, GridNet is inherently immune to GPS jamming and spoofing attacks. It can independently derives geolocation information from aerial photos rather than GPS signals. Then we are able to leverage the extracted geolocation data produced by GridNet, possibly cross-checking them with GPS data to identify discrepancies. By setting an appropriate threshold, we can also detect spoofing attacks. Additionally, the estimated location data can be used to support navigation tasks such as Return to Home (RTH). Notably, our method does not require the operation area's aerial photos in advance, nor does it necessitate storing satellite maps onboard or at the ground station. Furthermore, our approach is also robust regardless of the number or configuration of adversaries.

Comparison to Related Methods. In comparison with cryptography-based countermeasures for aircraft [18], GridNet offers a non-invasive approach that requires no modifications to existing GPS protocols or infrastructure. Compared with hardware-assisted solutions [37], our solution is software-only. These hardware-assisted methods have been demonstrated to be effective, but usually need specialized hardware to analyze the physical features of signals (i. e., signal powers, waveforms, and Doppler frequency). Unlike crowdsourcing-based methods [16], GridNet operates independently without requiring cooperation from others. Additionally, crowdsourcing-based methods will inevitably lead to privacy concerns of the participants due to information exchange. In contrast with recent AI-based means [32], [38], neither paired data are required during the training phase, nor are satellite images stored in the detection phase. Moreover, unlike most existing approaches, which are merely for spoofing detection, GridNet is also suitable for aircraft under jamming or in GPS-denied environments. Finally, considering the fact that most drones have been already manufactured with cameras [39], our approach is easily deployable and backward compatible and can be quickly implemented in practice. In real scenarios, GridNet can be deployed in two configurations: (1) on the remote controller at the ground station or (2) on the on-board computer (OBC) of the UAV. For enhanced security, the OBC-based deployment is preferred as the derivation of geolocation

information is independent of an active communication link between the UAV and the ground controller.

V. Experimental Validation

To evaluate the applicability of GridNet in real-world scenarios, we assess its performance in terms of classification accuracy, geographical coordinates error, and inference speed.

A. Dataset Description

1) Aerial Photos: There are no existing freely available datasets that include both satellite image grid cells and the corresponding aerial photo grid cells or baseline implementations. To demonstrate proof of concept, we had to collect aerial photos by ourselves. All aerial photos are captured using a DJI MATRICE 300 RTK equipped with a ZENMUSE P1 Photogrammetry Camera. The UAV operated at a flight altitude of 150 meters over a mixed urban–suburban environment. All photos were taken on a sunny, cloud-free spring day. As a result, the geospatial textures, such as rivers, cropland, and landmarks such as buildings and roadways, could be captured clearly. In total, the dataset covers an area of 2.52 km^2. We generated orthomosaic maps from aerial photos to showcase the capabilities of our system. Note again that we only use aerial photos for test purposes, not for training the classifier.

2) Satellite Images: To prepare the training data, we downloaded the georeferenced satellite images from Google Earth in the corresponding area. To get the satellite maps of the target area, we overlap the generated aerial orthomosaic maps with precise geolocation data in Google Earth. By doing so, we can get satellite maps with perfect alignment and the same coverage as the aerial photo. However, merely one satellite map is not enough to train a robust model that can do classification tasks instead of forcibly remembering a specific grid. Such a model trained on a few data points would result in low generalization ability. Hence, we additionally downloaded satellite maps captured at different times, and in total, we got eleven historical satellite maps in the same area. Since these satellite maps were taken at different times, some variations of the maps, including seasonal changes, weather patterns, and lighting differences such as shadows, can enrich our training samples, making our models more general and robust.

B. Experimental Environment Setup

Both CycleGAN and classification models were trained on a server equipped with a single NVIDIA A100 Tensor Core GPU with 40GB of GPU memory. The CPUs on the server are dual Intel Xeon(R) Gold 5220R @2.20GHz. In addition, for the testing phase, we deployed our models on two real UAVs and a Raspberry Pi 3B+ with a Cortex-A53 @1.4GHz CPU and 1GB of memory. A typical UAV platform uses an Intel NUC 5 i7-8559U CPU with 16GB of memory. The other is more powerful with an Intel NUC 11 Enthusiast i7 CPU @2.8GHZ, 64GB memory, and a graphics processor (Geforce RTX2060).

The GPU server runs a 64-bit Red Hat Enterprise Linux release 8.5, while the Raspberry Pi has a 64-bit Ubuntu 18.04 system. The weak and the strong UAV platforms run Ubuntu 16.04.7 LST and Ubuntu 20.04.3 LST, respectively. All experiments were implemented in Python with PyTorch 1.11.0 as the deep learning framework. CUDA and cuDNN are installed on the server to speed up both the training and inference of neural networks. While training and testing processes were conducted on the server, the Raspberry Pi (which lacks GPU support) and the two real UAV platforms were used only for inference during the testing phase.

C. Implementation Details

For training both CycleGAN and the classifier model, we use Adam as the optimizer, and set $\beta_1 = 0.9$ within 150 epochs and $\beta_1 = 0.5$ for the remaining epochs, and $\beta_2 = 0.99$. We set the initial learning rate to 2e-4, and adjust the learning rate schedule as proposed in [35].

1) Train CycleGAN: We adopt the CycleGAN architecture [40] for the domain adaptation task, owing to its compact design (only 10.5 MB) and satisfactory performance. The generators include nine residual blocks, two downsampling and two upsampling layers with stride-2 convolutions. The generator takes an input image, downsamples it, and then upsamples it back to the original resolution to produce a translated output. The discriminator employs four downsampling layers and outputs a scalar score. A higher value indicates a real image and a lower value indicates a fake one. The training data consist of unpaired grid cells extracted from satellite and UAV images and do not need to originate from the same geographic region. The batch size is set to 2, and the CycleGAN is trained for 1000 epochs. All weights are initialized using a normal distribution.

2) Train the Classifier: All grid cell images are resized to 128×128 resolution as a trade-off between the efficiency and accuracy of satellite image classification. The satellite image dataset containing 5,720 samples is randomly split into training and validation sets using *StratifiedKFold* strategy at a ratio of 10:1 to preserve the percentage of samples per class. The training process is run for 500 epochs.

Online Data Augmentation. One of the most widely adopted techniques for training high-performance and robust deep neural networks is data augmentation. Although our dataset contains 11 satellite maps at different times, such an amount is still insufficient to fully represent diverse real-world conditions such as rotation, distortion, blurring, illumination changes (darkening or brightening), color shift. To address this, we apply various data augmentation techniques during training, using the `Albumentations` library in Python. Standard augmentations include random flipping, blurring, rotation and cropping. Fig. 4 shows the images after the additional data augmentation beyond the `Albumentations` library. The color augmentation shuffles RGB channels and flip values in each channel randomly, the affine augmentation uses OpenCV to implement a random 3D affine transformation followed by a Gaussian filter to blur the image randomly, and the darkening augmentation reduces the luminance level with a random scale followed by quantization noising. These augmentations,

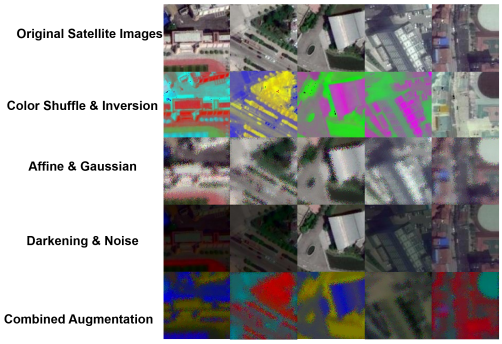

Fig. 4: Examples of online augmentations.

when combined, dramatically enhance the training data variations and increase the classification accuracy by about 10% compared to merely using the standard `Albumentations` operations according to preliminary experiments.

Pre-trained models. There exists a wide range of backbones architectures for deep learning models. Based on preliminary experiments, we selected 9 state-of-the-art networks: ResNet-50 and ResNet-101 [26], MobileNetV2 [41] and V3 [42], Eca_nfnet_l2 [43], and EfficientNet series (b3, b4, b5, b6) [44]. Pre-trained models have demonstrated strong performance in general image classification tasks, while ever fewer top-performing approaches use networks trained from scratch. In this paper, in order to speed up the learning process and reduce the training time, we use the models of each selected network pre-trained on the large-scale dataset, i. e., ImageNet, and then fine-tune them on our own datasets.

Loss function. In our classification model, we adopt cross-entropy loss combined with label smoothing to optimize model weights during training. The objective is to minimize the loss in optimization and thereby improve model accuracy . Label smoothing acts as a regularization technique that can prevent the network from becoming over-confident and trapped in sharp minima of the loss landscape where overfitting is likely to occur. It regularizes a model based on a Softmax with n output values by replacing the hard 0 and 1 classification targets with targets of ϵ/M and $1-\epsilon(M-1)/M$ respectively, where ϵ is a weight factor, $\epsilon \in [0, 1]$. The label smoothing equation is defined as $y_{ls} = (1-\epsilon)\, y_{hot} + \epsilon/M$ where M is the number of grids in the training dataset, and y_{hot} is the "hard" one-hot label vector. We further integrate the classification loss to smooth/improve the solution and alleviate the effects of the too-hard examples. The classification loss is implemented with the Softmax function, which is defined as

$$\mathcal{L}_{Softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{f_{y_i}}}{\sum_{j=1}^{M} e^{f_j}}, \tag{4}$$

where $f_{y_i} = W_{f_{y_i}}^{T} x_i$, x_i is input vector and W denotes the 1×1 convolution weights for Softmax.

VI. Performance Evaluation

We split the evaluation of the proposed GridNet into (i) evaluating the performance of grid classification, (ii) analyzing the impact of parameter settings, (iii) investigating the errors, and (iv) estimating online time complexity on different UAV environments.

A. Performance of Grid Classification

Classification accuracy is a key metric in GridNet, measuring how accurately GPS coordinates can be extracted. In essence, the goal is to find a grid cell whose longitude and latitude are the closest to the ground truth, thereby accurately localizing the UAV. We report the performance of selected models in terms of accuracy, parameter number, GFLOPs, inference time, and model size. We report the accuracy of each model on three different test sets, i. e., validation set (Val Acc.), which are all satellite images, UAV test set (UAV Acc.), which contains real UAV images, and the third test data set (CycleGAN+ Acc.)., which is generated from UAV test using a well trained CycleGAN. We observe that all neural networks perform better on the dataset generated from CycleGAN compared with the original UAV set, ranging from about 1% to 10%, respectively. Overall, EfficientNet-b5ns achieves the highest classification accuracy (approx. 93%) while mobileNetv2 has the smallest size and the shortest inference time. Notably, mobileNetv2-b3 offers a favorable trade-off between accuracy and efficiency. All experimental results are summarized in Table I.

B. Impact of Different Parameters

Impact of area size. To investigate the impact of different area sizes, we trained another two classifiers using the same neural network model (i. e., EfficientNet-b5ns), but on smaller areas: 1.26 km^2 and 0.63 km^2, respectively. Correspondingly, the number of grid cell classes was reduced from 520 to 260 and 130, respectively. Experimental results regarding different area sizes are summarized in Table II. Intuitively, as the number of classes decreases, the classification task becomes less complex. We observed that the classification accuracy improves slightly with a decreased area. In all cases, the best performance is achieved on the smallest area.

Impact of grid cell numbers. We divide the target area into different cell numbers to evaluate the impact of grid resolutions on the same area. Here we consider three different types of divisions, namely, 130, 520, and 2080. In the same place, as the numbers of grids grow larger, the pixels contained by each grid image become fewer. In our experiment, the resolutions of the images are set to 256×256, 128×128, and 64×64. As a result, we get 130, 520, and 2080 cells, respectively. We present results showing the impact considering grid numbers and image resolutions in Table III. On the one hand, surprisingly, our model achieves 100% prediction accuracy when using 130 cells, as an optimum setting; on the other hand, the performance drops significantly when the number of grids increases to 2080, achieving only approx. 60% accuracy. The reasons for this are twofold. First, as the number of classes increases, the classification task becomes more difficult. According to [45], current SOTA models' top-1 accuracy on ImageNet (containing 1000 categories and more

TABLE I: Performance comparison on validation, UAV and CycleGAN+ datasets using different neural network models.

Model	Val Acc.	UAV Acc.	CycleGAN+ Acc.	#Param.(M)	GFLOPs	Inference time (ms)	Model size(MB)
EfficientNet-b4ns [44]	0.894	0.844	0.904	18.5	0.50	3.09	71.2
EfficientNet-b5ns [44]	**0.907**	**0.873**	**0.929**	29.4	0.79	3.35	113.1
EfficientNet-b6ns [44]	0.883	0.827	0.921	41.9	1.12	3.51	161.1
Wide ResNet50 [26]	0.850	0.844	0.854	68.9	3.74	3.22	267.2
Wide ResNet101 [26]	0.871	0.858	0.883	126.9	7.45	3.51	488.9
MobileNetv2_100 [41]	0.831	0.861	0.875	**2.9**	0.10	**2.31**	**11.2**
EfficientNetv2-b3 [44]	0.896	0.877	0.919	13.6	0.53	2.85	52.5
MobileNetv3_large_100 [42]	0.869	0.864	0.902	4.9	**0.08**	2.43	18.7
Eca_nfnet_l2 [43]	0.869	0.879	0.885	56.6	3.32	3.88	222.0

TABLE II: Impact of area size.

Area size	#Grid	Accuracy	Precision	Recall	F1 Score
2.52 km^2	520	0.929	0.899	0.929	0.908
1.26 km^2	260	0.939	0.913	0.939	0.921
0.63 km^2	130	**0.946**	**0.919**	**0.946**	**0.929**

than a million images) is no more than 91%. Second, high resolution tends to achieve higher accuracy in general since such images contain more visual information than the ones with fewer pixels [44].

TABLE III: Impact of grid cell numbers.

#Grid	Resolution	Accuracy	Precision	Recall	F1 Score
130	256×256	**1.000**	**1.000**	**1.000**	**1.000**
520	128×128	0.939	0.913	0.939	0.921
2080	64×64	0.606	0.501	0.606	0.530

Fig. 5: Cases of various flight conditions.

Impact of weather, lighting, and occlusion conditions. We investigate how changes in weather, lighting, and occlusion affect the performance of our models since the training images are primarily captured under clear, daytime conditions. Severe conditions are common in the real world, and aerial vehicles may need to perform tasks under such extreme conditions.

Recent studies [38], [46] only focused on the impact of lighting changes, such as sunrise and sunset. To evaluate our EfficientNet-b5ns model under conditions of fog, cloud, rain, snow, and shadow, which are challenging tasks, we use the `Albumentations` and `OpenCV` libraries to synthesize images corresponding to such weather conditions.

To simulate photos in the night environment, we employ CycleGAN again since we lack night UAV photos of the target area for training and testing. We trained a specialized CycleGAN on the HIT-UAV dataset [47], which contains infrared thermal UAV photos. This allows us to transfer our UAV photos taken during the daytime to infrared UAV photos at night. Consequently, UAVs equipped with infrared thermal cameras can potentially operate effectively under low-light or nighttime conditions.

To simulate photos in the occlusion condition, we implemented the Random Erasing technique. In this approach, we randomly mask parts of real images with a probability of 0.5, simulating cases of partially deformed or blurred areas in the dataset. The masking is done by erasing a portion of the image, and the size of the erased area is determined by random proportions within a certain range. We also control the aspect ratio of the erased area to introduce variability in the occlusion effect. If we fail to find a suitable erasing region after maximum attempts, the image remains unaltered. We applied this technique to a set of test images, synthesizing a dataset of photos with simulated occlusion.

We first evaluated our general model, trained on a dataset from sunny days, using the generated datasets. Examples are displayed in Fig. 5, and the results are shown in Table IV. It is evident that the general model performs well under snow and shadow conditions without drastic performance reduction, whereas fog, rain, cloud, night, and occlusion conditions lead to accuracy degradation. To improve the practicality of GridNet even in extreme conditions, we trained domain-specific models for each weather and lighting condition using synthetic training data. We observed significant improvements in domain-specific models over the general model under all conditions. However, the cloud domain model still struggles since the cloud-occluded images tend to lack salient visual features. For best practice, the model option can be configured according to different conditions before the UAV starts navigation, while a low operating height can be set in case of clouds to avoid failure.

C. Error Analysis of Extracted Geolocations

Firstly, the GridNet framework exhibits robust immunity to jamming and spoofing attacks, as its vision-based approach eliminates dependence on GPS signals. To evaluate the accuracy of metric localization using GridNet, we show both theoretical analyses and empirical error measurements.

TABLE IV: Comparing the accuracy of the general models and domain models under different conditions.

Model\Condition	fog	cloud	rain	snow	shadow	night	occlusion
General Usecase	0.300	0.135	0.387	0.771	0.815	0.419	0.383
Domain Specific	**0.662**	**0.379**	**0.815**	**0.839**	**0.889**	**0.896**	**0.702**

Theoretical Analysis. Suppose that the target area is a rectangular region with length and width $[0, L] \times [0, M]$, which is divided into $l \times m$ grid cells. For each cell $a_{jk} = [(j-1)\frac{L}{l}, j\frac{L}{l}] \times [(k-1)\frac{M}{m}, k\frac{M}{m}]$, and the coordinates of the center of the grid are $(\frac{2j-1}{2}\frac{L}{l}, \frac{2k-1}{2}\frac{M}{m})$. As ground truth, we assume that we have access to an oracle (here the neural network model) that can answer the grid classification problem correctly with p accuracy $(0 < p \leq 1)$. It means that the probability of successfully predicting a certain grid class is p, and thus, the probability of being wrong is $1-p$. If the prediction is wrong, the probability of predicting to another grid is equal, which is $\frac{1-p}{lm-1}$. In addition, the oracle can only give the coordinates of the center point of the predicted grid. Assume that the real position of a UAV is at $(x, y) \in a_{j^*,k^*}$, the mean value of the distance errors between the real coordinates and the predicted coordinates is:

$$d = p\sqrt{\left(x - \frac{2j^*-1}{2}\frac{L}{l}\right)^2 + \left(y - \frac{2k^*-1}{2}\frac{M}{m}\right)^2} + \frac{1-p}{lm-1}\sum_{(j,k)\in D}\sqrt{\left(x - \frac{2j-1}{2}\frac{L}{l}\right)^2 + \left(y - \frac{2k-1}{2}\frac{M}{m}\right)^2} \quad (5)$$

where $D = \{1, \ldots, l\} \times \{1, \ldots, m\} \setminus \{(j^*, k^*)\}$ represents all cells except a_{j^*,k^*}. The first term is the distance to the center of the correct grid, weighted by p, and the second term is the average distance to the centers of all other cells, weighted by the probability of an incorrect prediction. For a fixed cell a_{j^*,k^*}, d is maximized when (x, y) is at the cell's corners. For a perfect model ($p = 1$), the error reduces to the first term in Equation 5, with an upper bound of $\mathcal{T}_1 = \sqrt{\left(\frac{L}{2l}\right)^2 + \left(\frac{M}{2m}\right)^2}$, the maximum distance from any point in a cell to its center.

To detect a spoofing attack, we need to perform a cross-check between the GPS reported positions and the estimated position from GridNet. We verify each incoming position to see if $dist(d, \hat{d}) < \mathcal{T}_1$ holds, where d is the position reported by GPS, $\hat{d}$ is the extracted location extracted using GridNet, $dist()$ is the Euclidean distance function, and $\mathcal{T}_1$ denotes the upper bound error, which is determined by the height and width of the grid. Note that if the manipulated position is within the grid, there is no way to know if it is being attacked through GridNet. In practice, we need to balance the trade-off between accuracy and precision. Because the prediction accuracy decreases as the number of grids grows. Ideally, the spoofing detection rate is the model's prediction rate regarding the grid classification.

Simulation Test. To evaluate the localization error in practice, we designed simulation experiments containing flight trajectories and random position estimation. For the trajectory simulation, we designed two paths, one from west to east and another from north to south. We assume that a UAV traverses the operating area at a uniform speed and takes aerial photos periodically when flying. We make sure it takes a picture of each grid as it passes by. We assume the coordinates of the centroid of those aerial photos as the ground truth and compare them with the extracted coordinates. In the west-to-east path, the UAV takes 26 aerial photos. After domain transformation by the CycleGAN, our classifier correctly predicted the class to which those photos belong, and the average error is 9.17 m. When flying north-to-south trajectory under the same setting, however, with only 20 photos, the average error is 7.33 m. We also randomly selected 10 photos to calculate the average error. However, the error is 36.3 m, larger than the trajectory case. The errors are compared with the ground truth from successful predictions, which is also the common practice adopted by [38], [46]. In contrast, landmark recognition based on GNSS can be unreliable because GPS drift can be as much as 100m [48].

D. Experiment on real UAV Environments

We also deployed and evaluated GridNet on two real-world UAV environments and a Raspberry Pi 3B+. Considering the limited computation resources of IoT devices such as memory, CPU, and power budget, we focused on the performance of light-weight neural network models, i. e., the MobileNet series. The inference time, memory usage, and power consumption are summarized in Table V.

TABLE V: Lightweight model performance on different UAVs

Platform	Model	Inference time	Memory	Power consumption
Raspberry Pi	MobileNetv2 [41]	282ms	384MB	**2.47 Watts**
	MobileNetv3 [42]	436ms	558MB	2.49 Watts
Intel NUC Gen 5	MobileNetv2 [41]	3.4ms	249MB	41.5 Watts
	MobileNetv3 [42]	2.7ms	220MB	41.9 Watts
Intel NUC 11 Enthusiast	MobileNetv2 [41]	**0.8ms**	**213MB**	99.8 Watts
	MobileNetv3 [42]	1.2ms	288MB	101.4 Watts

TABLE VI: Comparison of hardware, runtime, and storage.

Paper	CPU & GPU	Memory	Avg. time/image	Storage
[38]	Intel i7 & Nvidia Quadro	32 GB	0.11 s	423 MB
[46]	Intel i7 & Nvidia Quadro	32 GB	9.23 s	794 MB
[32]	Raspberry Pi 3B+	1 GB	10.48 s	667 MB
Ours	**Raspberry Pi 3B+**	**1 GB**	**0.28 s**	**–**
Ours	**NUC Gen 5**	**16 GB**	**0.0027 s**	**–**
Ours	**NUC 11 & RTX 2060**	**64 GB**	**0.0008 s**	**–**

VII. Conclusion

This paper presents GridNet, a vision-based deep learning method enhancing UAV navigation security. Leveraging DNNs, it extracts geolocation from aerial photos, achieving 93% spoofing detection accuracy in 3 ms per image. It supports GPS-denied positioning, requires no pre-flight UAV data, and suits resource-constrained platforms as a mapless solution. Future work should address altitude adaptability, update imagery in real-time, and expand data across varied geographic contexts, including different landscapes. In conclusion, GridNet offers a robust, efficient counter to GPS vulnerabilities. Further extending its adaptability and generalization capabilities will enhance its practical applicability and real-world impact.

Acknowledgment

This work was supported by the Center for Cyber Security at New York University Abu Dhabi (NYUAD). This research was carried out on the High Performance Computing resources at NYUAD.

REFERENCES

[1] C. Adrados, I. Girard, J.-P. Gendner, and G. Janeau, "Global positioning system (gps) location accuracy improvement due to selective availability removal," *Comptes Rendus Biologies*, vol. 325, no. 2, pp. 165–170, 2002.

[2] G. S. Gadgets, "HackRF One." [Online]. Available: https://greatscottgadgets.com/hackrf/

[3] S. Shane and D. E. Sanger, "Drone Crash in Iran reveals secret US surveillance effort," *The New York Times*, vol. 7, 2011.

[4] G. Gibbons, "FCC fines operator of GPS jammer that affected Newark airport GBAS," *Inside GNSS*, vol. 30, 2013.

[5] Clare Sebastian, "Getting lost near the Kremlin? Russia could be 'GPS spoofing'," 2016. [Online]. Available: https://money.cnn.com/2016/12/02/technology/kremlin-gps-signals/

[6] M. L. Psiaki and T. E. Humphreys, "Gnss spoofing and detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[7] N. Navidi, R. J. Landry, J. Cheng, and D. Gingras, "A new technique for integrating mems-based low-cost imu and gps in vehicular navigation," *Journal of Sensors*, vol. 2016, no. 1, p. 5365983, 2016.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[9] United States Department of Defense, "Global Positioning System Standard Positioning Service Performance Standard Rev. 4th Edition," United States Government Std., Sep. 2008.

[10] David Hambling, "Drone Crash Due To GPS Interference In U.K. Raises Safety Questions," 2020. [Online]. Available: https://www.forbes.com/sites/davidhambling/2020/08/10/investigation-finds-gps-interference-caused-uk-survey-drone-crash/

[11] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *ACM conference on Computer and communications security*, 2011, pp. 75–86.

[12] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, "An experimental study of GPS spoofing and takeover attacks on UAVs," in *31st USENIX Security Symposium*, Boston, MA, 2022, pp. 3503–3520.

[13] M. Nayfeh, "Artificial intelligence-based gps spoofing detection and implementation with applications to unmanned aerial vehicles," Master's thesis, Purdue University, 2023.

[14] P. Papadimitratos and A. Jovanovic, "Gnss-based positioning: Attacks and countermeasures," in *MILCOM 2008-2008 IEEE Military Communications Conference*. IEEE, 2008, pp. 1–7.

[15] S. Lo, Y. H. Chen, H. Jain, and P. Enge, "Robust gnss spoof detection using direction of arrival: Methods and practice," in *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 2891–2906.

[16] K. Jansen, L. Niu, N. Xue, I. Martinovic, and C. Pöpper, "Trust the crowd: Wireless witnessing to detect attacks on ads-b-based air-traffic surveillance," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2021, pp. 21–25.

[17] M. Y. Arafat, M. M. Alam, and S. Moh, "Vision-based navigation techniques for unmanned aerial vehicles: Review and challenges," *Drones*, vol. 7, no. 2, p. 89, 2023.

[18] T. E. Humphreys, "Detection strategy for cryptographic gnss anti-spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073–1090, 2013.

[19] Y. Qiao, Y. Zhang, and X. Du, "A vision-based gps-spoofing detection method for small uavs," in *2017 13th International Conference on Computational Intelligence and Security*. IEEE, 2017, pp. 312–316.

[20] L. Meng, S. Ren, G. Tang, C. Yang, and W. Yang, "Uav sensor spoofing detection algorithm based on gps and optical flow fusion," in *International Conference on Cryptography, Security and Privacy*, 2020, pp. 146–151.

[21] P. F. Swaszek, S. A. Pratz, B. N. Arocho, K. C. Seals, and R. J. Hartnett, "Gnss spoof detection using shipboard imu measurements," in *International Technical Meeting of The Satellite Division of the Institute of Navigation*, 2014, pp. 745–758.

[22] A. Couturier and M. A. Akhloufi, "A review on absolute visual localization for uav," *Robotics and Autonomous Systems*, vol. 135, p. 103666, 2021.

[23] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based uav navigation," *Geo-spatial information science*, vol. 21, no. 1, pp. 21–32, 2018.

[24] A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza, "Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 1015–1039, 2015.

[25] M. Umer, I. Ashraf, Y. Park *et al.*, "Enhanced machine learning ensemble approach for securing small unmanned aerial vehicles from gps spoofing attacks," *IEEE Access*, 2024.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] N. Xue, L. Niu, and Z. Li, "Pedestrian detection with modified r-fcn," in *Proceedings of the UAE Graduate Students Research Conference*, 2021.

[28] Z. Li, S. Cai, X. Wang, H. Shao, L. Niu, and N. Xue, "Multiple object tracking with gru association and kalman prediction," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[29] Ettus Research, "Universal Software Radio Peripheral (USRP)," 2017. [Online]. Available: https://www.ettus.com

[30] RTL-SDR, "RTL-SDR (RTL2832U) and software defined radio news and projects. Also featuring Airspy, HackRF, FCD, SDRplay and more." 2017. [Online]. Available: https://www.rtl-sdr.com/

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[32] N. Xue, L. Niu, X. Hong, Z. Li, L. Hoffaeller, and C. Pöpper, "Deepsim: Gps spoofing detection on uavs using satellite imagery matching," in *Annual Computer Security Applications Conference*, 2020, pp. 304–319.

[33] J. Fan, E. Zheng, Y. He, and J. Yang, "A cross-view geo-localization algorithm using uav image and satellite image," *Sensors*, vol. 24, no. 12, p. 3719, 2024.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[35] Z. Li, H. Shao, L. Niu, and N. Xue, "Progressive learning algorithm for efficient person re-identification," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 16–23.

[36] ——, "Pla: progressive learning algorithm for efficient person re-identification," *Multimedia Tools and Applications*, vol. 81, no. 17, pp. 24 493–24 513, 2022.

[37] S. Liu, X. Cheng, H. Yang, Y. Shu, X. Weng, P. Guo, K. C. Zeng, G. Wang, and Y. Yang, "Stars can tell: A robust method to defend against GPS spoofing attacks using off-the-shelf chipset," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3935–3952.

[38] M. Bianchi and T. D. Barfoot, "UAV localization using autoencoded satellite images," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1761–1768, 2021.

[39] B. Custers, "Drones here, there and everywhere introduction and overview," in *The future of drone use*. Springer, 2016, pp. 3–20.

[40] Jittor, "Jittor-gan," https://github.com/Jittor/gan-jittor, accessed July 23, 2024.

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[42] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *International Conference on Computer Vision*, 2019.

[43] A. Brock, S. De, and S. L. Smith, "Characterizing signal propagation to close the performance gap in unnormalized resnets," *arXiv preprint arXiv:2101.08692*, 2021.

[44] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[45] paperswithcode, "Image classification on imagenet." [Online]. Available: https://paperswithcode.com/sota/image-classification-on-imagenet

[46] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," in *International Conference on Robotics and Automation*. IEEE, 2020, pp. 6491–6497.

[47] J. Suo, T. Wang, X. Zhang, H. Chen, W. Zhou, and W. Shi, "Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicles," *arXiv preprint arXiv:2204.03245*, 2022.

[48] K.-H. Yap, T. Chen, Z. Li, and K. Wu, "A comparative study of mobile-based landmark recognition techniques," *IEEE Intelligent Systems*, vol. 25, no. 1, pp. 48–57, 2010.