Pervasive and Mobile Computing 100 (2024) 101916

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

SEEK+: Securing vehicle GPS via a sequential dashcam-based vehicle localization framework

Peng Jiang [a], Hongyi Wu [b], Yanxiao Zhao [c], Dan Zhao [b], Gang Zhou [d], Chunsheng Xin [a,*]

[a] *ECE Department and School of Cybersecurity, Old Dominion University, Norfolk, 23505, VA, USA*
[b] *ECE Department, University of Arizona, Tucson, 85701, AZ, USA*
[c] *ECE Department, Virginia Commonwealth University, Richmond, 23284, VA, USA*
[d] *CS Department, William & Mary, Williamsburg, 23185, VA, USA*

ARTICLE INFO

Keywords:
GPS spoofing
Deep learning
Vehicle localization
Image matching

ABSTRACT

Nowadays, the Global Positioning System (GPS) plays an critical role in providing navigational services for transportation and a variety of other location-dependent applications. However, the emergent threat of GPS spoofing attacks compromises the safety and reliability of these systems. In response, this study introduces a cutting-edge computer vision-based methodology, the SEquential dashcam-based vEhicle localization frameworK Plus (SEEK+), designed to counteract GPS spoofing. By analyzing dashcam footage to ascertain a vehicle's actual location, SEEK+ scrutinizes the authenticity of reported GPS data, effectively identifying spoofing incidents. The application of dashcam imagery for localization, however, presents inherent obstacles, such as adverse lighting and weather conditions, seasonal and temporal image variations, obstructions within the camera's field of view, and fluctuating vehicle velocities. To overcome these issues, SEEK+ integrates innovative strategies within its framework, demonstrating superior efficacy over existing approaches with a notable detection accuracy rate of up to 94%.

1. Introduction

The Global Positioning System (GPS) has become an indispensable tool across a multitude of sectors, ranging from personal electronics like smartphones and wearables to critical transportation systems, including autonomous vehicles. The widespread integration of GPS has revolutionized location-based services, enabling advanced functionalities such as navigation, vehicle tracking, emergency location sharing, and aid in rescue operations. However, the reliability of GPS technology is undermined by its vulnerability to spoofing attacks, which pose significant risks to safety and security. Such attacks can be readily executed using inexpensive software-defined radios, like HackRF [1], to disrupt the normal operation of GPS devices embedded in various systems [2–10]. When a spoofed GPS signal, more potent than the legitimate signals from satellites, is broadcasted, nearby GPS receivers are deceived, accepting the fraudulent signal. Consequently, this vulnerability allows attackers to misdirect vehicles or individuals to unintended, potentially hazardous destinations. Notably, an attacker could manipulate the GPS system of a commercial or ride-sharing vehicle, fabricating its location or route. This scenario presents a critical security concern, as it could lead to dire outcomes for affected individuals or assets.

[*] Corresponding author.
E-mail address: cxin@odu.edu (C. Xin).
The conference version of this article was presented in IEEE PerCom 2023.

https://doi.org/10.1016/j.pmcj.2024.101916
Received 10 August 2023; Received in revised form 29 February 2024; Accepted 9 March 2024
Available online 12 March 2024

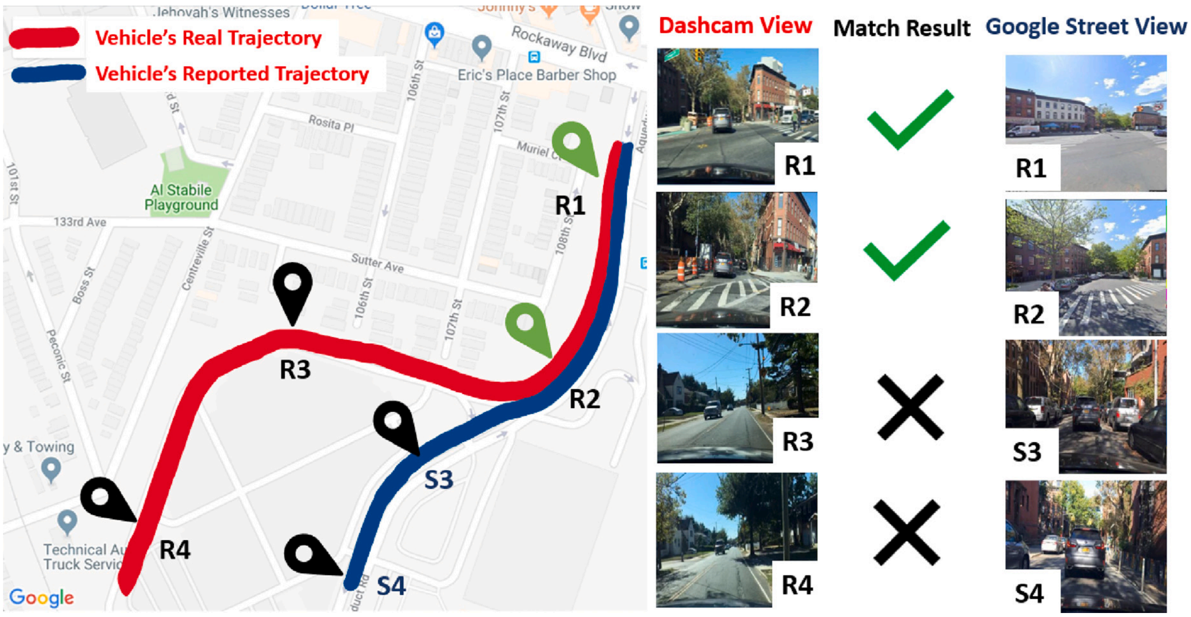


Fig. 1. An illustration of the GPS spoofing attack with an escaping driver. A ride-hailing vehicle is driving on the path from R1 to R4 (red line). Meanwhile, the malicious/escape driver spoofs the GPS signal to deceive the monitoring center that the vehicle is driving on the spoofed route (blue line).

Existing solutions, ranging from classical signal authentication methods to innovative approaches harnessing deep learning techniques and vehicle onboard sensors like motion detectors, have been explored to fortify the resilience of GPS systems. Classical defenses often rely on detecting anomalies in signal strength, noise patterns, or encrypting signals. Such methods usually require the adoption of expensive signal-analyzing software or the upgrade of infrastructure. Deep learning methods that leverage the onboard sensors can assist in the reconstruction of the vehicle's authentic moving trajectories and can also help against the GPS spoofing attack. However, these countermeasures often fall short of state-of-the-art spoofing techniques that meticulously craft GPS signals to mimic authentic paths closely, making detection particularly difficult [3]. The attackers' ability to generate signals that closely replicate legitimate GPS data undermines the effectiveness of many current defenses, leaving vehicles vulnerable to misdirection and its potentially grave consequences. In this paper, we propose a computer vision-based framework to detect GPS spoofing by capitalizing on the ubiquitous presence of dashcams within most vehicles. Using a dashcam as an onboard sensor to detect the GPS spoofing attack can quickly detect the spoofing attack that focuses on mimicking authentic paths. To be specific, the proposed framework extracts dashcam images while the vehicle is driving and uses them to match the reference images at the reported (untrusted) GPS location. Therefore, if the vehicle's GPS is spoofed, the reference image collected at the reported location will contradict the dashcam image collected on the road. This can be achieved by using *cross-view image matching* (CVM), which matches images at the same location with different view angles, such as satellite/aerial image and a 360-degree ground view image [11–16] (see Fig. 1).

While the existing CVM studies have demonstrated effective performance in ideally datasets, applying these methods to dashcam image localization within our GPS spoofing detection framework is challenging in a real-world driving scenario. Modern vehicles are typically equipped with only one or two safety cameras, insufficient for creating the comprehensive ground-level panoramas required for high-accuracy CVM implementations. Additionally, dashcam footage is usually captured from lower vantage points, resulting in significant visual obstructions from other vehicles, contrasting the less obstructed images from higher-mounted 360-degree cameras used in CVM studies. Furthermore, the variable and often poor lighting conditions experienced during real-world driving – unlike the consistently favorable conditions under which CVM datasets are captured – can severely impact image-matching accuracy. Also, CVM's reliance on broad satellite imagery fails to provide the precise localization necessary for effective GPS spoofing detection. Lastly, the infrequent updates of geo-tagged reference images, such as those from Google Street View, introduce discrepancies due to environmental changes over time, further complicating the task of accurate dashcam-based localization.

In this paper, we introduce innovative methodologies to address the inherent challenges in localizing dashcam images for real-world driving scenarios and establish a robust framework for GPS spoofing detection, termed *SEquential dashcam based vEhicle localization frameworK Plus* (SEEK+). To counter the limited perspective offered by a single dashcam, we devise a trip-level sequential image matching scheme, aggregating a series of dashcam images from a journey into a unified localization dataset. This method not only utilizes the spatial continuity inherent in sequential images but also adapts to the dynamic nature of driving behaviors. To overcome the issue of visual obstructions in dashcam footage, we introduce an object removal technique to clarify these images, enhancing their informative value for precise localization. Addressing the challenge of varied lighting conditions, our framework includes adaptive image processing techniques to maintain localization accuracy across diverse environmental settings. For the critical task of precise localization necessary for GPS spoofing detection, we leverage the Google Street View database for its high-resolution, geo-tagged imagery, ensuring localization precision within 30 ft. Lastly, to mitigate the discrepancies caused by temporal changes between reference and dashcam images, we employ autoencoders to harmonize the visual themes of both image sets, ensuring consistent and reliable localization.

The rest of the paper is organized as follows. Section 2 discusses the related works about GPS spoofing attack and countermeasures, as well as the state-of-the-art works in cross-view image geo-localization. Section 3 describes the threat model. Section 4 presents our proposed techniques for dashcam image localization in real-world driving. Section 5 put all techniques together to build the SEEK+ framework. Section 6 shows the experimental results. Finally, Section 7 concludes the paper.

2. Related works

2.1. GPS spoofing attacks

Existing investigations have highlighted the susceptibility of localization sensors, particularly GPS, to a spectrum of malicious activities. Among these, GPS is notably vulnerable to several forms of interference and manipulation, including jamming, replaying, and spoofing attacks [17,18]. Jamming attacks [19] disrupt the reception of GPS signals by overwhelming the authentic, relatively feeble GPS signals with more potent, interfering radio waves. Replay attacks [20] introduce confusion by capturing and subsequently transmitting previously recorded or irrelevant GPS signals, potentially misleading the system's perception of its temporal and spatial context [x26]. Among the various threats, GPS spoofing is particularly dangerous, capable of subtly altering a vehicle's perceived trajectory by introducing forged signals into the communication stream [4,5,7,10]. Recent examples include the stealthy GPS spoofing techniques that gradually shift a vehicle's reported position to cause significant navigational errors over time without immediate detection. [21] can fail production-grade autonomous driving systems with an over 90% success rate. [3] allows an attacker to reach destinations that are as far as 30 km away from the actual destination without being detected. Such advanced attacks designed to undermine multi-sensor fusion systems in autonomous vehicles, demonstrate the evolving complexity and stealthiness of GPS spoofing methods, posing a significant challenge to detection and mitigation efforts.

2.2. GPS spoofing detection

Given the potentially catastrophic consequences of GPS spoofing attacks, various endeavors have been undertaken to devise effective countermeasures. Classical GPS spoofing detection methods [22,23] adopted cryptographic techniques as potential solutions by encrypting satellite signals using a confidential key to encrypt the communication channels between GPS receivers and transceivers. Other researchers, [24–26] have explored strategies for detecting GPS spoofing by analyzing the characteristics of wireless GPS signals. [24] detected malicious spoofing signals by leveraging the reported positions of several GPS receivers deployed in a fixed constellation. [26] computed and compared feature vectors based on the singular values of wavelet transformation coefficients from both spoofing and genuine signals. [25] utilized features such as pseudo-range, Doppler shift, and signal-to-noise ratio (SNR) for classifying GPS signals with a supervised machine learning method. [27–29] examined received GPS signals via anomaly detection applied to signal waveforms or employing calculations related to the signal angle of arrival. Nonetheless, these approaches typically necessitate a significant number of GPS receivers or an upgrade to the existing GPS infrastructure, thereby presenting challenges for practical implementation in self-driving vehicles.

As the installation of various sensors in vehicles and other mobile platforms increases, efforts to enhance GPS navigation through cross-validation with additional sensors have gained momentum to counter potential spoofing attacks. Integrating motion sensors with GPS navigation, as investigated in studies such as [30–32], represents a proactive approach to augment navigation reliability and security. [30] proposed sensor fusion techniques, combining data from multiple onboard sensors (such as GNSS receivers, accelerometers, gyroscopes, and cameras). By analyzing the fused sensor data, the method detects anomalies indicative of spoofing attempts. [31] leverages accelerometers to detect GPS spoofing signals by comparing accelerometer outputs with acceleration estimates from GPS data. [32] pursued advancements in this field by developing a deep learning-based detection method that aims to accurately reconstruct vehicle trajectories using inertial sensors, such as accelerometers and gyroscopes. The IMU reconstructed trajectory is then compared with the reported GPS trajectory on an offline road map to achieve a higher accuracy. Nevertheless, the accuracy limitations inherent in commercially available accelerometers and gyroscopes present significant challenges in identifying sophisticated spoofing attacks, which are crafted to emulate legitimate vehicle movements [3].

2.3. Cross-view image matching for geo-localization

While motion sensors have been investigated to counter GPS spoofing attacks, research on leveraging computer vision to mitigate the GPS spoofing attack remains comparatively sparse. Recent researches focus on image-based geo-localization, indicating potential pathways for integrating computer vision techniques in addressing GPS spoofing in a real-world scenario.

Before the advent of deep learning, cross-view image geo-localization relied heavily on manually crafted features, such as self-similarity measures and histograms [33–36]. These traditional methods often faced challenges in achieving high matching accuracy due to the limitations of the features used. However, with the wide adoption of deep learning across various computer vision applications, a wave of deep learning-based geo-localization techniques emerged. These methods utilize finely-tuned Convolutional Neural Networks (CNNs) to extract more sophisticated features, significantly enhancing cross-view geo-localization accuracy. [12] introduced an end-to-end learning model that aggregates features using a NetVLAD layer, marking a notable performance improvement. [37] developed a feature transformation module designed to align features from aerial and street view images. [38] focused on integrating orientation information into their model, further boosting its performance. [39] to propose a GAN-based approach that employs a feature fusion training strategy for cross-view image geo-localization. [11] introduced the VIGOR approach, which uniquely does not necessitate a one-to-one match between ground and aerial images. Some other models, such as [14], manipulate the satellite views to bridge the domain gap between reference and query images by applying polar transformation with prior geometric knowledge. [14] significantly improved results on CVUSA [40] and CVACT [38] datasets through this technique. However, such methods still focused on the ideal scenarios where stationary panoramic ground view images are provided to match a satellite image covering the same location. To validate these methods in real-world driving scenarios, [15] configured 12 fish-eye near-infrared (NIR) cameras on top of a vehicle to capture a comprehensive panorama of the terrestrial scene. Apparently, there are plenty of limitations to applying the existing cross-view matching methods to provide accurate vehicle location to solve the GPS spoofing dilemma.

3. Threat model

We consider two types of GPS spoofing attacks against the GPS-based navigation of transportation systems in real-world driving, (1) ***random GPS attack*** (RGA), and (2) ***sequential GPS attack*** (SGA). RGA broadcasts a series of random (incorrect) GPS coordinates to the target vehicle with the purpose of confusing the GPS reception in a short period of time. RSA can be launched easily at a road intersection or in the middle of a road to cause a denial-of-service to the navigation system, which may result in catastrophic consequences to the victim vehicles, such as driving on the opposite lane or taking wrong turns, etc.

SGA broadcasts a series of GPS coordinates that form a fake trajectory with the goal of either *hijacking* the vehicle to an unsafe location or *escaping* the tracking of the monitoring center of the vehicle, such as the ride-hailing or commercial trucking company monitoring center. To launch the GPS spoofing attack for vehicle hijacking, an external attacker can tailgate a target vehicle and fool its GPS navigation to a route toward an unsafe location. Meanwhile, a GPS spoofer can also be an "escape" driver of a ride-hailing vehicle or a commercial truck, who broadcasts a set of GPS coordinates forming a fake route to the GPS tracker on the vehicle, with the purpose of hiding the real trajectory of the driver from the monitoring center [3]. Compared to RGA, SGA is a more advanced and complicated attack that can craft a continuous spoofing trajectory to mislead the navigation system of a vehicle. But it requires prior planning of a fake trip, such as start and end points and a route following the city road networks, to cheat the monitoring center, or an existing IMU-based GPS spoofing detection scheme.

4. Proposed techniques for dashcam image localization in real world driving

As discussed in Section 1, SEEK+ utilizes dashcam images to identify the location of a vehicle to detect GPS spoofing. We have also introduced CVM and the challenges of applying CVM directly on dashcam images for geo-localization. In this section, we first conduct some experiments of applying CVM on dashcam images and illustrate the challenges for dashcam image localization. We then present our proposed schemes to address each challenge, to significantly improve dashcam image localization. In the next section, we put those techniques altogether to build the SEEK+ framework.

The dashcam images in our experiments are from a vehicle driving dataset called BDD**G**4k, with details described in Section 6.1.1. BDD**G**4K provides a geo-tagged GSV reference image for each dashcam image captured from four thousand driving trajectories. As introduced in Section 1, in CVM, image localization or image matching is through image retrieval. Specifically, given a query image, in order to identify the location of this image, CVM scans the entire reference image database where all images are geo-tagged and returns the top-matched reference image. The location of the returned top image is then used to be the location of the query image.

Due to its dependence on image retrieval, the widely used performance metric to evaluate CVM is the recall accuracy, R@k, which treats it as a success if, among the k nearest reference images returned, one is from the original image pair of the query image. For instance, in our scenario, the dataset BDD**G**4k pairs each dashcam image with a (geo-tagged) GSV image. For a query dashcam image, CVM scans all GSV images in the BDD**G**4k dataset. If the returned top image is originally paired with the query dashcam image, then it is treated as a success.

Table 1 shows the results of applying state-of-the-art CVM methods on a widely used CVM dataset, CVUSA, and the vehicle driving dataset BDD**G**4k. Note that CVM methods usually apply the polar transformation on the satellite images in CVUSA, which is not feasible for the BDD**G**4k dataset. We observe that two recent CVM methods, TransGeo [16] and L2LTR [13], achieve very impressive R@1 accuracy, 94.08%, and 91.99%, respectively, on the CVUSA dataset. However, their R@1 performance drops significantly to 46.4% and 52.43% on the BDD**G**4k dataset.

As briefly described in Section 1, there are several factors that cause the performance degradation of CVM on the BDD**G**4k dataset, including lack of panorama views, blockage in dashcam images, complicated lighting conditions, and seasonal/theme gap. Next, we discuss each of them in detail and present our proposed schemes to address them.

Table 1
CVM performance on CVUSA and BDD**G**4k datasets by state-of-the-art CVM methods.

Method	CVUSA		BDD**G**4k	
	R@1	R@5	R@1	R@5
CVM-NET [12]	22.47	49.98	10.52	21.45
SAFA [14]	81.15	94.23	35.15	38.42
L2LTR [13]	91.99	98.27	46.4	66.1
TransGeo [16]	94.08	98.36	52.43	65.06

Table 2
The accuracy of image matching on BDD**G**4k dataset by the distance to the query image.

Method	BDD**G**4k		
	M = 10 m	M = 20 m	M = 50 m
CVM-NET	30.4	45.31	48.3
SAFA	44.2	50.42	56.92
L2LTR	51.3	66.1	72.71
TransGeo	68.83	75.2	80.1

Table 3
Trips distribution under various lighting/weather conditions.

	Sunny	Cloudy	Rainy	Night	Total
Train set	1409	247	68	1276	3000
Val set	475	120	22	383	1000
Total	1884	367	80	1659	4000

4.1. Trip level matching to address lack of panorama view and impact of slow driving

In Section 1, we have discussed the issue of the lack of a panorama view in dashcam images, which is critical to achieving good performance in traditional CVM. Moreover, there is another unique issue raised in real-world driving — there can be multiple *similar* images collected from a set of nearby locations when the vehicle drives at a low speed or at a complete stop. In this case, a nearby GSV reference image with a different location ID may be returned by CVM when it is applied to BDD**G**4k. When such a result is returned, it is considered a failure in the R@1 performance metric, as it represents a different GPS location, although it is very close to the true location of the query image.

To illustrate how the slow driving behavior impacts CVM performance, we consider a top-M (Meter) metric, which treats it a success if the returned image is within M meters of the query image, instead of requiring the GPS location IDs be the same for the two images as in Table 1. Table 2 illustrates the top-M matching accuracy when M is set to 10 m, 20 m, and 50 m, respectively. We notice the accuracy of the best-performing method, TransGeo, improves from 52.43% to 68.83% from R@1 to top-10 m. This verifies our speculation since, with the top-M metric, a returned image from a nearby location may also count as a success without requiring the image at the exact location of the query image.

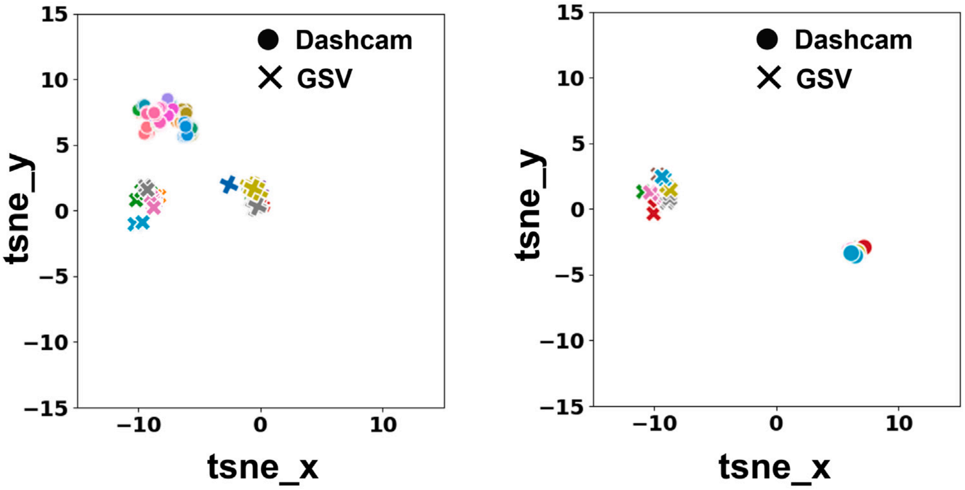
To address the above challenge, as well as the lack of panorama views, we propose to conduct dashcam image matching at the trip level, which includes a sequence of contiguous images along a vehicle's trajectory. This will mitigate the impact of slow driving on image matching to eliminate the interference introduced by the repeated or nearby images during vehicle stopping or slow driving. It also helps to compensate for the lack of multiple panorama images at a given location. To process an image sequence from a vehicle's trajectory, we utilize the *recurrent neural network* (RNN) to exploit the spatial and temporal dependencies in a sequence. The Long Short-Term Memory (LSTM) [41] and Gated Recurrent Unit (GRU) [42] are two popular variations of RNN. However, we adopt GRU because it requires less GPU memory while achieving comparable performance to LSTM.

4.2. Image normalization to address lighting/weather conditions

4.2.1. Impact of lighting condition on image matching

We have found that complicated lighting/weather condition is one of the main factors that affect the performance of CVM on driving datasets. To illustrate its impact, we sort the trips in BDD**G**4k into four groups, Sunny, Cloudy, Rainy, and Dark/Night, based on lighting and weather conditions (see Table 3).

To examine the impact of lighting conditions on the images, we first randomly select ten trips under the Sunny condition and ten trips under the dark lighting condition. Then we use the feature extractor of a recent CVM method, TransGeo [16], to obtain the feature description for both dashcam images and GSV reference images corresponding to the trips. The t-SNE visualization of the feature vectors of those images is plotted in Fig. 2. We can see that the dashcam images and GSV images have quite different clustering behaviors under different lighting/weather conditions. Note that GSV images are always captured in good lighting conditions. In contrast, dashcam images are captured in diverse lighting conditions depending on the trips. When the lighting is dark (Fig. 2(b)), dashcam images are clustered together and far away from the GSV reference images. When the lighting is good



(a) (b)

Fig. 2. Impact of lighting condition on the images illustrated in the feature space, (a) images randomly sampled from the daylight condition, and (b) images randomly sampled from the dark/night condition.

Table 4
Mean and Std of dashcam and GSV images under different lighting conditions.

Lighting condition	View source	Mean	Std
Mixed daylight	Dash	[0.3383, 0.3688, 0.3790]	[0.2493, 0.2600, 0.2694]
	GSV	[0.5228, 0.5431, 0.5502]	[0.2204, 0.2261, 0.2554]
Night	Dash	[0.1380, 0.1113, 0.0944]	[0.1579, 0.1420, 0.1335]
	GSV	[0.5293, 0.5488, 0.5538]	[0.2163, 0.2213, 0.2554]

(Fig. 2(a)), we notice the distance between the GSV images cluster and the dashcam images cluster is smaller. Due to the shorter distance, the matching between dashcam images and GSV images of the daylight trips is more likely to succeed than for the trips in the dark/night condition.

4.2.2. Normalization for low-light images

We propose to use the image normalization technique to address the impact of lighting/weather conditions and improve the performance of image matching. Image normalization is a technique in computer vision that changes the range of pixel intensity values to a certain range, e.g., 0 to 1. In the literature, most studies adopt the mean and standard deviation (std) of ImageNet for normalization, i.e., mean = [0.485, 0.456, 0.406] in the three channels (RGB), and std = [0.229, 0.224, 0.225].

However, due to the unique features of the images caused by the diverse lighting/weather conditions in real-world driving, the images from the vehicle driving dataset have a quite different mean and std from ImageNet. Table 4 illustrates the mean and std of the images in a widely known driving dataset, BDD100K [43], under daylight (mixed weather) and night condition, which are significantly different from the mean and std of ImageNet. In addition, GSV images have similar mean and std in both daylight and night trips as they are actually independent of the trips and all taken at good lighting conditions (and different times) at the GPS coordinates of the trips. Dashcam images have a significantly lower mean/std in corresponding lighting conditions. The BDD100K driving dataset is a highly diverse driving dataset. We expect their mean and std are typical for vehicle driving datasets. Hence we will adopt this mean and std for normalizing the images in our driving dataset.

Fig. 3 illustrates an example of image normalization for a dashcam image from BDD**G**4k, using the mean and std of ImageNet and the ones of BDD100K. We can see that the normalized image in Fig. 3(b) has a much higher contrast; the buildings/landmarks and the crosswalk pattern on the road are highlighted, which are crucial to improving performance. In contrast, the original image captured in the low lighting condition loses the detailed texture and information of buildings and landmarks. At last, the normalized image with ImageNet mean/std exhibits poor performance due to the significant difference in the capturing context of the images.

4.3. Season alignment of reference images

Due to the fact that GSV reference images are slowly updated, typically after several years, the GSV image at the same location of a dashcam image is often taken at a different time/season, i.e., they have quite different themes. To bridge the gap in the seasonal change on these two views, we propose a season alignment technique to transform the GSV images. We use two autoencoders to

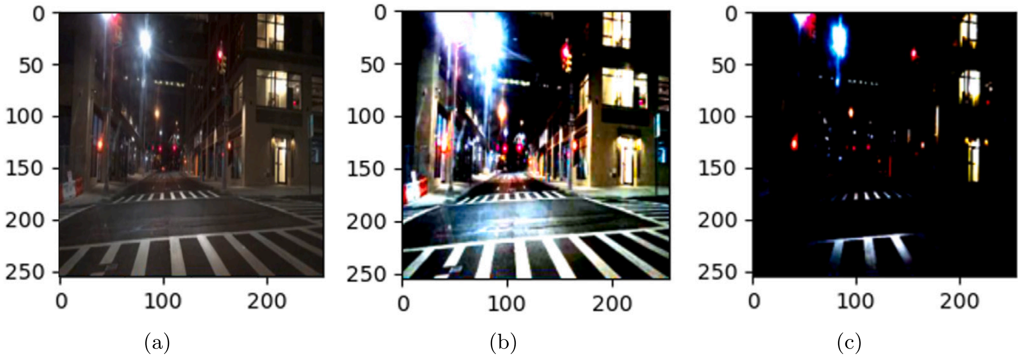


Fig. 3. Normalization of an image in BDDG4k, (a) the raw image that was captured at a low lighting condition, (b) the normalized image with the mean and std from Table 4, and (c) the normalized image with mean/std of ImagetNet.

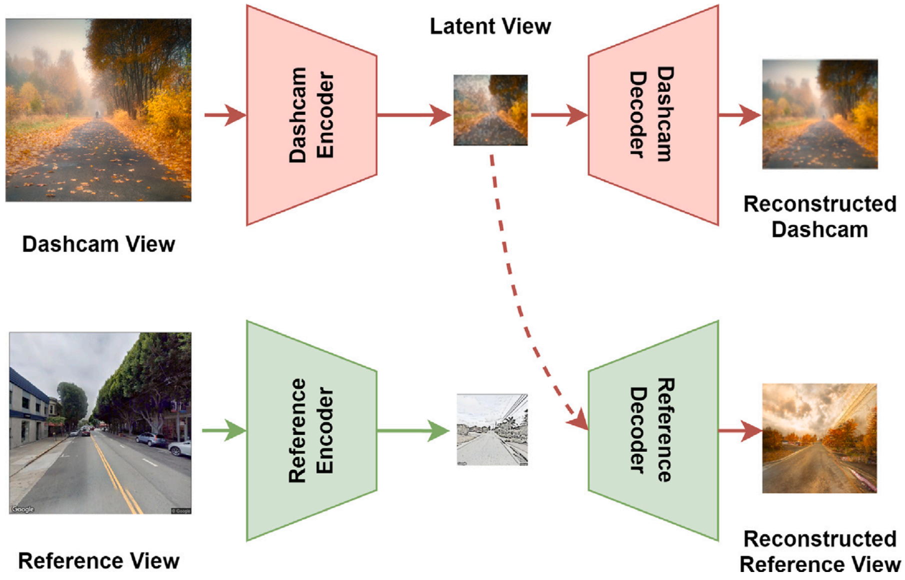


Fig. 4. Season alignment for GSV reference images.

import the latent view from the dashcam image into the GSV reference image to mimic the theme/season of the dashcam image while still producing a realistic transformed image. This improves the matching performance between the two views under various season changes.

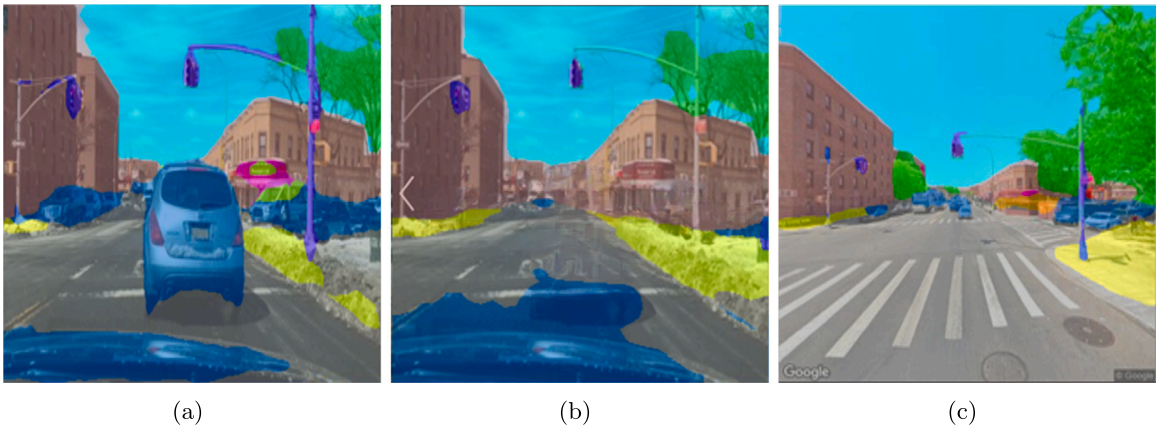
Fig. 4 illustrates the process of transferring season features between two views. Two CNN-based autoencoders are trained to learn the hidden style features from dashcam and GSV images, respectively. The latent view is the encoded feature vector that can be used to reconstruct the same image. For the dashcam view, we use the dashcam decoder to decode the latent view to reconstruct its own image in a smaller size without the loss of critical information. In decoding the GSV reference view, we replace the latent view from the GSV image with the one from the dashcam image to reconstruct the GSV reference image. The output of the reference decoder has a smaller size and, most importantly, combines the styles from both views and essentially aligns the GSV image to the same season as the dashcam image.

4.4. Dashcam image blockage removal

In a real-world driving scenario, the dashcam view can be easily blocked by the front vehicle, objects on the windshield, or the reflection of items on the dashboard. Therefore, compared to the GSV images, which are sanitized after capturing, dashcam images usually have a much higher blockage ratio.

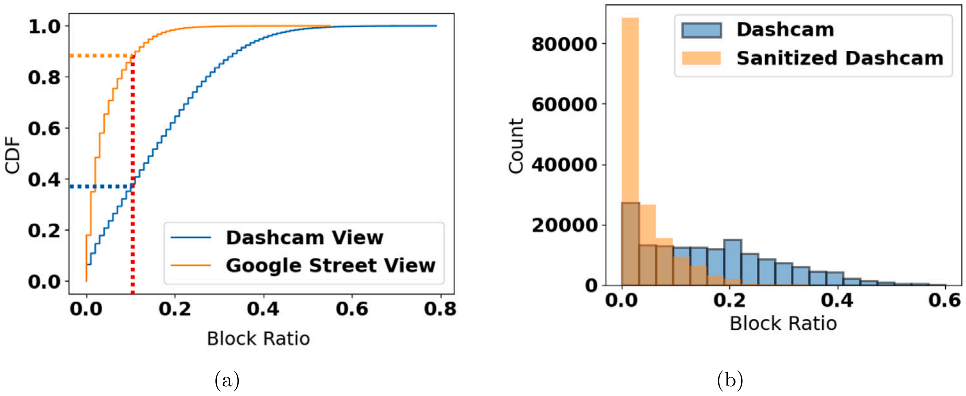
We define the blockage ratio of an image as follows:

$$BlockageRatio = \frac{\text{\# of pixels classified as ``obstacles''}}{\text{Total \# of pixels}}. \tag{1}$$



(a) (b) (c)

Fig. 5. Blockage removal: (a) original dashcam image showing segmentation, (b) dashcam image after blockage removal, (c) GSV reference image at the same location.



(a) (b)

Fig. 6. Blockage ratio: (a) CDF of blockage ratio before blockage removal, (b) blockage ratio histogram of dashcam images before and after removing blocking objects.

To summarize the composition of the images captured while driving, we apply the image segmentation technique [44] to annotate each pixel in both the dashcam image and the corresponding GSV reference image. Fig. 5 shows a dashcam image and the corresponding GSV image taken at the intersection where the target vehicle is waiting behind a silver SUV. The blue overlay covers the pixels classified as "obstacles". With the assistance of image segmentation, we calculate the blockage ratios for both the dashcam image and GSV reference image at the same location using (1). As a result, the dashcam view in Fig. 5 has a blockage ratio of 22%, whereas the blockage ratio of the GSV reference image at the same location is only about 2%. Fig. 5(a) shows the CDF of the blockage ratio of dashcam images and GSV reference images after analyzing the whole dataset, which indicates that more than 90% of GSV images have less than 10% blockage ratio. On the other hand, only 40% of dashcam images have less than 10% blockage ratio.

To address the challenge of large blockage areas in dashcam images, for each dashcam image, we use a tool called *automated object remover* [45] that combines Semantic segmentation and EdgeConnect architectures to remove specified objects in images. By filtering out the objects that are considered "obstacles" such as vehicles, vans, and trucks, we can provide a cleaner image with less blocking area. Fig. 6(b) illustrates the histogram of blockage ratio in the dashcam images before and after blockage removal. Clearly, the blockage in dashcam images can be significantly removed. For instance, in Fig. 5(b), we can see that the silver SUV that blocks the center of the image has been removed, which restored the yellow building at the corner and is expected to improve the dashcam-GSV matching performance.

5. System architecture of SEEK+

After introducing the proposed techniques for dashcam image localization, in this section, we put them all together to describe the proposed GPS spoofing detection framework SEEK+, which is built upon the proposed schemes in the preceding section.

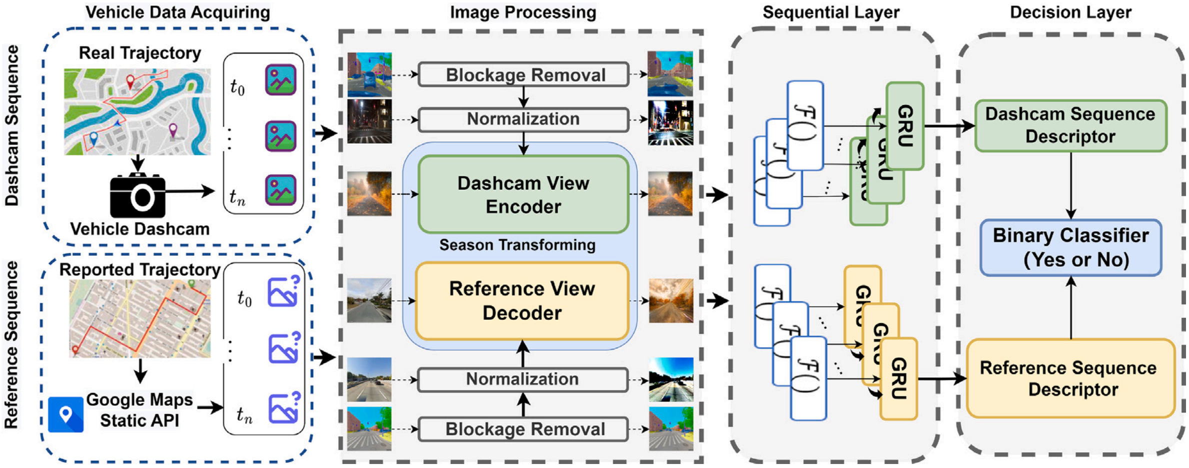


Fig. 7. The architecture of the SEEK+ framework.

5.1. System overview

Fig. 7 illustrates the SEEK+ framework, which is composed of two Siamese-type pipelines. Each pipeline comprises the four component schemes we developed in the preceding section: Image Normalization, Blockage Removal, Season Alignment, and Trip Level matching. The fundamental idea of SEEK+ is to compare the vehicle's real-time dashcam image sequence with the GSV reference image sequence queried at the GPS locations reported by the vehicle GPS receiver. The two pipeline designs can effectively project the spatial features learned from each pipeline into a shared space. SEEK+ can be deployed on different platforms to detect possible GPS spoofing attacks. For example, it can be implemented on the vehicle or the remote monitoring center of ride-hailing apps, such as Uber or Lyft. It is carried out by the following steps: (1) a driving vehicle records its GPS location from the GPS receiver, and its dashcam takes video, which is essentially a sequence of images, (2) the GPS locations and the dashcam images are synchronized by the time, (3) the vehicle either sends the GPS locations and dashcam images/video to a local or remote spoofing detector, (4) the spoofing detector queries the GSV reference images by the reported GPS locations using Google API, (5) the spoofing detector feeds both dashcam image sequence and GSV reference image sequence to SEEK+ and obtains the detection result; *if the binary classifier outputs Yes, i.e., dashcam images match the GSV images, then there is no spoofing, and otherwise, there is spoofing*.

5.2. Image transformation

As discussed in the previous section, images need to be properly processed/transformed to address the challenges of complicated lighting conditions, large blockage areas, and season/theme gaps between dashcam and GSV images. Therefore, the three image transformation techniques proposed in the preceding section are applied to the dashcam and GSV images coming into the two pipelines. Specifically, we first normalize the images by using the mean and std extracted from the driving dataset. Then we apply image segmentation [44] to decompose the image and identity the objects in the image that needs to be removed. After the unwanted objects have been removed, we further feed the dashcam image into an autoencoder ("Dashcam Encoder") to compress the image into the feature space (encoding), from which it can be reconstructed to the original image (decoding) but has a smaller dimension. The GSV images follow a similar procedure, but it takes the latent space of the dashcam view into the "Reference Decoder" to reconstruct an image with the seasonal theme of the dashcam view as well as a smaller size. After the images are properly transformed by those techniques, they are ready to be fed to the sequential layers for trip level matching.

5.3. Sequential trip level image matching

The proposed trip level image matching can effectively utilize the spatial and temporal dependencies among an image sequence of vehicle driving and compensate for the lack of the panorama view. We first feed the transformed images to a feature extractor that learns the spatial feature of each individual image in a sequence. Let $\mathbf{x}_n^d$ and $\mathbf{x}_n^g$ ($\mathbf{x}_n^d, \mathbf{x}_n^g \in \mathbb{R}^{|\mathbf{L}_n| \times D}$) denote the output of the feature extractor for a given transformed dashcam image sequence and GSV image sequence, where D is the embedding size which is 1024 in this paper. They represent the stack of feature vectors from the two image sequences, respectively. In this paper, we test three candidates for the feature extractor: VGG-16 [46], and two feature extractors from L2LTR [13] and TransGeo [16]. Note that CVM models are usually structured in a Siamese-like network.

SEEK+ uses a recurrent neural network (RNN) to keep track of the feature representation learned in a sequential manner while the vehicle is in motion. As discussed in the preceding section, SEEK+ adopts a popular variant of RNN, GRU, due to its more efficient design. SEEK+ uses a stacked GRU structure with two layers, and the hidden unit is chosen to be 512. The dropout layer [47] and batch normalization [48] are adopted to reduce the internal covariate shift among time steps.

5.4. Binary classification layer

The outputs of the two pipelines of SEEK+, $r^d = \mathcal{R}(\mathbf{x}_n^d)$ and $r^g = \mathcal{R}(\mathbf{x}_n^g)$, are the fixed-length vectors that represent the features learned from each sequence. The final objective is to identify if the series of reference images obtained from the reported GPS locations match the same series of real-time vehicle dashcam images from the same trip. Hence, the last layer of SEEK+ is a binary classification layer. We adopt the BCELoss defined below as the loss function.

$$\mathbf{L}_{BCE} = -[y_n \cdot \log(\Delta_i) + (1 - y_n) \cdot \log(1 - \Delta_i)], \tag{2}$$

where $i \in \{1, 2, \dots, \mathcal{N}\}$, and $\Delta_i = |r_i^d - r_i^g|$ represents the difference between two feature vectors r_i^d and r_i^g, and y_n is the label of the input sequence pair. BCELoss creates a criterion that measures the Binary Cross Entropy between the target and the output. We also add a sigmoid layer in the network to limit the output to a range between 0 and 1. The classification output $y_n = 1$ indicates the dashcam image sequence and the GSV image sequence are from the same trip. Otherwise, if $y_n = 0$, the two sequences are not from the same trip, i.e., there is a *GPS spoofing attack*.

6. Performance evaluation

In this section, we carry out experiments to evaluate the performance of the proposed SEEK+ framework on GPS spoofing detection. We first introduce the dataset we use for the experiments, then discuss the experiment settings, and at last present the results.

6.1. Dataset

6.1.1. Vehicle driving dataset

In the literature, a widely used vehicle driving dataset is the Berkley diverse driving video database, BDD100K [43], which consists of 100,000 vehicle driving videos from diverse locations under different weather conditions and different times of the day. Each video records a driving trajectory of about 40 s long, 720p, and 30 fps. The videos also include GPS/IMU information to show approximate vehicle driving trajectories.

As can be seen from the SEEK+ architecture, unlike CVM, SEEK+ does not utilize image retrieval from the reference image database to locate the query image. Instead, SEEK+ compares the query image (sequence) and the reference image (sequence), and finds if they match each other to detect GPS spoofing attacks. Hence, in addition to the dashcam images in BDD100K, we also need the corresponding GSV reference images at the reported GPS locations of the dashcam images. To this end, we expand the BDD100K dataset by adding the GSV reference images.

Specifically, for the dashcam images, we sample the driving video clips in BDD100K [43] at the sampling rate of 1 Hz, which matches the sampling rate of the GPS information in BDD100K. Then we use the Google Street-View (GSV) Static API [49] to obtain the geo-tagged GSV reference image roughly aligned with the GPS location of each dashcam image in the same heading direction. Note that both the alignment of a dashcam image with the GPS location recorded by cell phones and the alignment of a GSV image to this GPS location is approximate. Usually, the closest GSV images are a few meters or a few tens of meters away from the referenced GPS location. We have sampled 4 thousand video clips from the BDD100k dataset, i.e., 4000 trips. Each trip lasted about 40 s, as described earlier. In total, from those trips, we obtain 152,667 image pairs with various weather and lighting conditions. We call the resulting dataset as BDD**G**4k.

6.1.2. Spoofing data generation

The BDD100K dataset does not have GPS spoofing samples. We also have not found other practical public driving datasets with GPS spoofing samples. To bridge this gap, we manually generate GPS spoofing data samples for BDD**G**4k. Given different GPS spoofing attack models, the generated GPS spoofing samples need to accurately represent the attack behaviors based on the existing sequences in BBD**G**4k. We introduce a parameter $\alpha \in [0, 1]$ to describe the strength of the GPS spoofing attack or the percentage of spoofed GPS coordinates in a trip. When $\alpha = 0$, there is no spoofing attack at all. If $\alpha = 1$, then all GPS coordinates in a trip trajectory are spoofed by the attacker.

In our driving dataset, a trip includes both dashcam images and the recorded GPS locations. Given a trip j, we use different approaches to generate a GPS spoofed trip that simulates RSA and GSA, respectively. To begin with, let the number of GPS coordinates to be spoofed be $N_{sp} = \lceil \alpha \times \mathcal{L}_j \rceil$, where α is described above, and $\mathcal{L}_j$ denotes the length of trajectory j. Then we generate faked GPS locations to replace the N_{sp} GPS locations of trip j and, accordingly, update the GSV images for those spoofed GPS locations. In RSA, N_{sp} GPS locations are randomly selected from trip j and are replaced with random GPS points, i.e., there is no relationship between the spoofed GPS locations. However, in SGA, we replace the last N_{sp} GPS points in trip j using a segment of continuous GPS points from a random trip (that trip needs to be longer than N_{sp}). With both approaches, we generate a spoofing trip j'. *The GSV and dashcam image sequences of trip j' form a negative pair*. Note that the dashcam images of trip j' are the same as the original trip j. To eliminate any bias introduced by the imbalanced data in the training process, we generate the same number of negative samples as positive samples. Here *a positive sample is a pair of dashcam and GSV image sequences of a trip without spoofing*. To guarantee the generality of the model working in various lighting conditions, we balance the positive and negative pairs with the same number of samples from various lighting conditions. We generate 10k positive pairs and 10k negative pairs for each experiment setting.

Table 5
Performance of SEEK+ compared with DeepPOSE in different lighting conditions.

Method	Attack	Sunny				Cloudy				Rainy				Dark			
		Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1
SEEK+	RGA	**0.940**	0.923	0.960	**0.941**	0.915	0.895	0.940	0.917	0.827	0.816	0.844	0.830	0.852	0.798	0.942	0.864
	SGA	0.920	0.894	0.953	0.923	0.890	0.860	0.931	0.894	0.808	0.803	0.816	0.809	0.825	0.780	0.906	0.838
DeepPOSE	RGA	0.828	0.775	0.923	0.843	0.828	0.780	0.913	0.841	0.827	0.780	0.910	0.840	0.827	0.775	0.920	0.841
	SGA	0.785	0.743	0.869	0.801	0.780	0.750	0.839	0.792	0.800	0.754	0.888	0.816	0.780	0.742	0.856	0.795

6.2. Experimental settings

6.2.1. Implementation details

SEEK+ is implemented in PyTorch [50]. Both dashcam and reference GSV images have the original size of 256 × 256 and are resized to 128 × 128 after performing season alignment. The model is trained and evaluated on NVIDIA A6000 GPU with 64 GB GPU memory. A two-step training process is adopted. We first train the feature extractor on BDD**G**4k dataset. Then, we fix the feature extractor and train the entire system of SEEK+. The batch size is 32, and the Adam optimizer is used with a learning rate of 0.0001 based on the cosine scheduling. The whole training process takes 150 epochs, in which 50 epochs are used to train the feature extractor first, and the rest 100 epochs are used to train the entire SEEK+ system for GPS spoofing detection.

6.2.2. Evaluation metrics

We use the widely used standard performance metrics for classification, accuracy, precision, recall, and F1 score. As a comparative study, we compare SEEK+ with DeepPOSE [32], which is the only other machine learning based approach that can be applied to the BDD100K dataset. Other previous GPS spoofing detection studies [27,51] use totally different approaches that are not comparable and rely on different assumptions, such as requiring multiple GPS receivers per system and/or ground-based sensors/infrastructure, which are not applicable in the scenario of the BDD100K vehicle driving database we adopt.

6.3. Performance of GPS spoofing detection

6.3.1. Performance under different lighting conditions

Table 5 illustrates the classification results of the proposed GPS spoofing detector, SEEK+, including accuracy, precision, recall, and F1 score under different lighting/weather conditions, with the attack strength $\alpha = 1$, trip length $\mathcal{L} = 20$ s, and the feature extractor from TransGeo. The classification accuracies of SEEK+ for RGA and SGA are 94% (F1 score: 0.941) and 92% (F1 score: 0.923), respectively, under the sunny lighting/weather condition. The performance decreases if the lighting condition degrades. For example, under the dark lighting condition, the detection accuracy drops to 85% for RGA and 82% for SGA. Nevertheless, SEEK+ outperforms DeepPOSE under all lighting/weather conditions. It is also noted that the performance of DeepPOSE is similar under all lighting conditions. This is because DeepPOSE utilizes the data from motion sensors that are not affected by the weather/lighting. One can also observe that the performance of SEEK+ for SGA is only slightly lower than the one for RGA. This demonstrates that SEEK+ is robust enough to detect smart GPS spoofing attacks such as SGA, which carefully crafts a spoofing trajectory that follows real road networks, speed limits, etc., and minimizes the difference as a legitimate trajectory.

6.3.2. Performance impact of individual components of SEEK+

As discussed in Section 4, to overcome the challenges encountered in real-world driving scenarios, we have proposed four schemes to transform the images, Trip Level matching (TL) of dashcam images, Image Normalization (NR), Blockage Removal (BR), and Season Alignment (SA). We present how each of these proposed schemes improves the performance of GPS spoofing detection. Table 6 shows the results of applying those schemes. In the table, the scheme "TL only" means we only use the trip level matching. That is, we treat a sequence of images on a trip in the last $\mathcal{L} = 20$ s as a basic unit for classification. As discussed in Section 4, the image matching accuracy using a single image is around 52%, which would be closely relevant to the GPS spoofing detection accuracy. On the other hand, Table 6 indicates that using the TL scheme significantly increases the performance, with accuracy above 70% in the sunny weather/lighting condition and about 60% even in the dark lighting condition.

Adding the NR (image normalization) technique to the TL scheme further improves the performance. For instance, in the low-lighting (dark) condition, the accuracy improves 12.2% for RGA and 10.7% for SGA. The table also indicates that NR brings higher performance lift for detection in the dark lighting condition than in the better lighting condition.

In contrast to the NR technique, the BR (blockage removal) technique brings a better improvement for the GPS spoofing detection in good lighting conditions. For instance, it improves the accuracy by 7.9% and 6.1% (TL+BR vs. TL only) in low-lighting conditions for RGA and SGA, respectively, whereas the accuracy improvement is 9.2% and 8.4%, respectively, in the sunny weather. This is because, in the latter case, the objects in an image are easier to be identified. Therefore, the quality of the image is critical before applying block removal. Thus this observation guides the design of SEEK+, i.e., applies BR after NR in SEEK+ as shown in Fig. 7.

Table 6 also shows that the SA (season alignment) technique also effectively improves the performance of SEEK+ in both low lighting and good lighting conditions. In the former case, the accuracy of SEEK+ is improved by 10.9% and 10.7% for RGA and SGA, respectively, while in the latter case, the accuracy is improved by 7.9% and 5.7%, respectively. At last, as a result of applying all four techniques, TL, NR, BR, and SA, the generalization ability of SEEK+ is significantly improved to address various lighting/weather conditions, seasonal changes, and blockage to dashcam images.

Table 6
Performance impact of different image processing techniques/schemes.

Schemes	Attack	Sunny				Dark			
		Acc	Precision	Recall	F1	Acc	Precision	Recall	F1
TL only	RGA	0.723	0.719	0.733	0.726	0.601	0.599	0.612	0.605
	SGA	0.711	0.708	0.717	0.713	0.596	0.595	0.602	0.598
TL + NR	RGA	0.803	0.783	0.840	0.810	0.723	0.712	0.750	0.730
	SGA	0.770	0.772	0.767	0.769	0.702	0.687	0.742	0.713
TL + BR	RGA	0.815	0.784	0.870	0.825	0.680	0.661	0.738	0.697
	SGA	0.795	0.767	0.847	0.805	0.658	0.641	0.715	0.676
TL + SA	RGA	0.802	0.782	0.836	0.808	0.710	0.693	0.755	0.722
	SGA	0.768	0.751	0.803	0.776	0.703	0.685	0.753	0.717
All together	RGA	0.940	0.923	0.960	0.941	0.852	0.798	0.942	0.864
	SGA	0.920	0.894	0.953	0.923	0.825	0.780	0.906	0.838

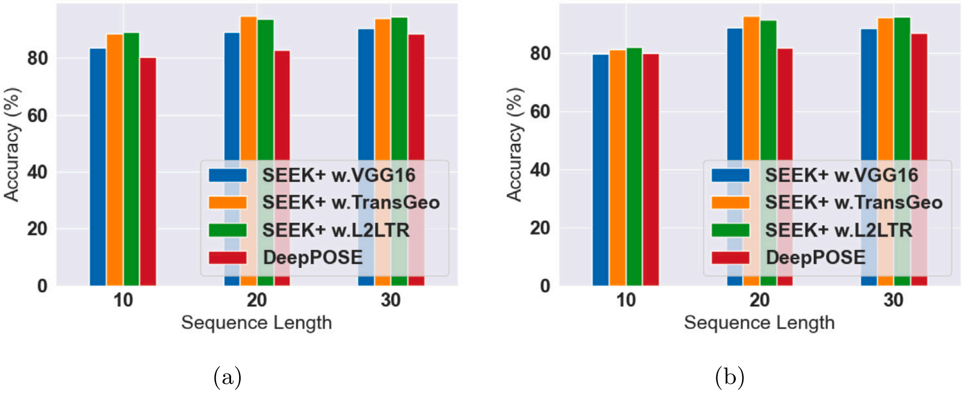


Fig. 8. Accuracy of SEEK+ with regard to sequence length $\mathcal{L}$ when $\alpha = 1$: (a) for RGA, and (b) for SGA.

6.3.3. *Impact of feature extractor*

Fig. 8 shows the performance of SEEK+ in terms of the sequence length ($\mathcal{L}$) under the good lighting condition. It also depicts the performance of SEEK+ using different feature extractors, VGG16, L2LTR [13], and TransGeo [16], respectively. L2LTR and TransGeo deviate from the conventional paradigm of feature extraction by their distinct module compositions. These modules have been carefully crafted to capture similar attributes from distinct visual domains. Among these designs, L2LTR adopts vanilla ViT on top of ResNet, yielding a composite amalgam of CNN and transformer components. This fusion creates a hybrid mix of convolutional neural networks (CNN) and transformer components. Notably, L2LTR situates CNN in the first layer, constraining the utilization of self-attention and positional embeddings to higher-level CNN features. However, this design choice comes at a trade-off. The global modeling capabilities and positional insights that are integral to singular matches are not fully harnessed in L2LTR as effectively as they are in TransGeo. Nonetheless, L2LTR makes up for this through its distinctive feature extraction approach at the level of trips, yielding performance that can hold its own against TransGeo. Still, TransGeo gains a slight edge due to its efficiency in GPU memory usage and its sleeker architectural configuration. At last, one can observe that in all scenarios, SEEK+ outperforms DeepPOSE. Note that the sequence length, or trip length, also affects DeepPOSE, which utilizes the motion sensor data over a period of time to detect GPS spoofing.

In terms of sequence length, as depicted in Fig. 8, it becomes evident that longer sequences contribute to enhanced performance. Notably, the accuracy of SEEK+ exhibits a trend toward its peak as the sequence length extends to 20 s. Extending the sequence to 30 s yields a marginal performance uptick, albeit at the cost of greater neural network complexity. This added intricacy results in prolonged model training durations and increased inference time for the neural network's outcomes. Further insights from the outcomes presented in Fig. 9 illustrate that SEEK+ w.L2LTR demands 66% more time for inferring a pair of dashcam and GSV sequences lasting 30 s compared to those lasting 20 s. Similarly, while SEEK+ w.TransGeo demonstrates quicker response times, but the escalation in sequence length introduces a notable delay during the inference stage.

6.3.4. *Performance with unknown vehicle dashcam orientation*

In the dataset, BDD**G**4k, where dashcam sequences are aligned with Google Street View (GSV) images based on vehicle driving orientation, an investigation was conducted to explore scenarios in which a number of GSV images might not be perfectly aligned due to potential GPS drifting in urban environments. To replicate this situation, a simulation was devised wherein a certain number of well-aligned GSV images within a sequence were randomly replaced with GSV images sampled from the same location but with

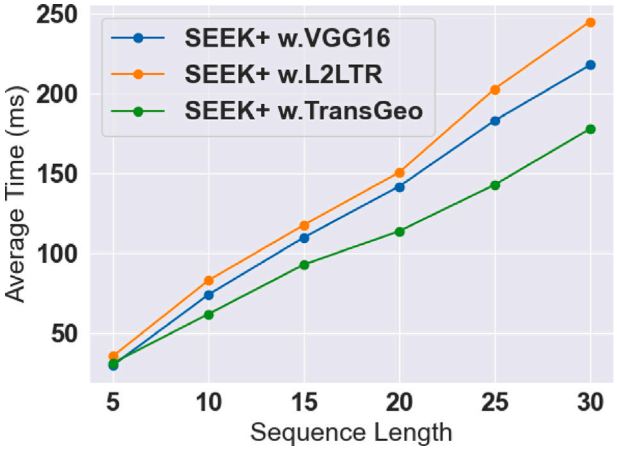


Fig. 9. Average time consumption at inference stage on NVIDIA RTX 3080 GPU.

Table 7
Matching error when dashcam sequences are not perfectly aligned with GSV. SEEK+ w.TransGeo is selected, and the sequence length is 20 s.

MAR	New York City		San Francisco	
	Sunny	Dark	Sunny	Dark
10%	2.12	3.31	2.56	3.89
20%	5.43	7.12	7.94	9.12
30%	10.29	13.11	14.21	17.98
40%	14.32	16.01	17.43	19.62
50%	18.43	20.11	23.31	26.21

randomized heading directions. This ratio of substituted GSV images to the total GPS points was termed the *Misalignment Ratio (MAR)*.

Interestingly, an elevated MAR did not adversely impact the detection of GPS spoofing attacks, namely RSA and SGA. However, it did lead to an increase in false alarms, where legitimate trips were misclassified as spoofing instances due to the incongruity between the GSV images. Table 7 details the errors introduced in the matching process for benign trips within the BDD**G**4k dataset, with half originating from New York City and the other half sampled from San Francisco. In this analysis, SEEK+ with TransGeo was employed, and the sequence length was fixed at 20 s. As the MAR surpassed the 20% threshold, a notable surge in false alarm rates occurred due to the absence of accurate reference images during the matching phase. However, when contrasting the performance between New York City and San Francisco, it was observed that heading misalignment had a lesser impact on trajectory fidelity in New York City compared to San Francisco. This discrepancy can be attributed to the discrepancy in average trip distances; New York City has an average trip distance of 189 m, whereas San Francisco's average distance is 250 m. The shorter average trip distance in New York City increases the chances of capturing the missing view from misaligned GSV images in other GPS points, owing to a higher likelihood of overlapping reference images from the same vicinity.

6.3.5. *Resistance to stealthy GPS attacker*

Next, we test the performance of SEEK+ with the presence of stealthy attackers who do not spoof the GPS signal until the vehicle drives on a route that is similar enough to the spoofing route. To simulate the stealthy GPS spoofing attack, we control the attack strength α and change it from 0.2 to 1, which indicates the fraction of the number of GPS locations in a vehicle trajectory to be spoofed. Fig. 10 shows the accuracy of SEEK+ with different values of α, assuming the sequence length $\mathcal{L} = 20$ s. From the figure, when the attack strength is higher, the detection accuracy of SEEK+ is also higher. When the attack strength is lower, i.e., the target vehicle is lightly attacked, the detection accuracy decreases. Nevertheless, a low attack strength would also likely not succeed in achieving the objective of the attackers, such as hijacking a target vehicle. We also compare SEEK+ with DeepPOSE in Fig. 10. It is clear SEEK+ outperforms DeepPOSE, especially when the attack strength is lower.

7. Conclusion

In this study, we introduced SEEK+, an innovative computer vision-based framework designed to identify GPS spoofing attacks by leveraging dashcam images for accurate vehicle geo-localization. Recognizing the complexities inherent in real-world driving environments – such as restricted fields of view, variable lighting conditions, obstructions in dashcam footage, and seasonal variations – we developed several advanced techniques to refine the image processing. These include trip level image-matching (TL),

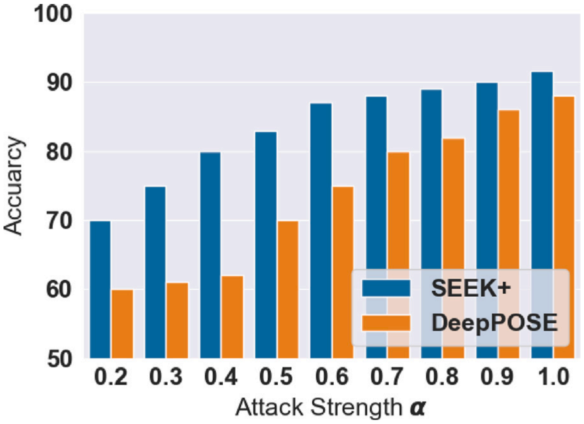


Fig. 10. Accuracy of SEEK+ compared with DeepPOSE v.s. the attack strength α, with $\mathcal{L} = 20$, TransGeo feature extractor.

image normalization (NR), obstruction removal (BR), and seasonal adjustment (SA). Furthermore, we rigorously assessed SEEK+'s effectiveness in real-world scenarios to validate its robustness and practical applicability. Our comparative analysis of SEEK+ against the contemporary GPS spoofing detection method DeepPOSE across diverse conditions demonstrates SEEK+'s superior capability in identifying spoofing incidents. Notably, SEEK+ achieves an impressive detection accuracy rate of up to 94% under favorable lighting conditions. This evaluation underscores SEEK+'s potential as a robust and reliable solution for GPS spoofing detection in various real-world applications.

CRediT authorship contribution statement

Peng Jiang: Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Hongyi Wu:** Supervision, Validation, Funding acquisition, Methodology. **Yanxiao Zhao:** Supervision, Validation. **Dan Zhao:** Supervision, Validation. **Gang Zhou:** Validation, Visualization. **Chunsheng Xin:** Funding acquisition, Project administration, Supervision, Validation, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Science Foundation, United States under Grants CNS-2120279, CNS-1950704, CNS-2244902, CNS-2245250, DUE-1742309, and through the NSF Intergovernmental Personnel Act Independent Research and Development Program, the National Security Agency under Grants H98230-22-1-0275, H98230-21-1-0165, and H98230-21-1-0278, the Air Force Research Lab, United States under Grant FA8750-19-3-1000, and the Commonwealth Cyber Initiative.

References

[1] K. Wang, S. Chen, A. Pan, Time and position spoofing with open source projects, Black Hat Europe 148 (2015) 1–8.
[2] C.L. Krishna, R.R. Murphy, A review on cybersecurity vulnerabilities for unmanned aerial vehicles, in: Proc. IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR), 2017.
[3] S. Narain, A. Ranganathan, G. Noubir, Security of GPS/INS based on-road location tracking systems, in: IEEE Symposium on Security and Privacy (SP), 2019, pp. 587–601.
[4] K.C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, Y. Yang, All your GPS are belong to us: Towards stealthy manipulation of road navigation systems, in: Proc. USENIX Security, 2018.
[5] Q. Luo, Y. Cao, J. Liu, A. Benslimane, Localization and navigation in autonomous driving: Threats and countermeasures, IEEE Wirel. Commun. 26 (4) (2019) 38–45.
[6] S.P. Arteaga, L.A.M. Hernández, G.S. Pérez, A.L.S. Orozco, L.J.G. Villalba, Analysis of the GPS spoofing vulnerability in the drone 3DR solo, IEEE Access 7 (2019) 51782–51789.

[7] M.L. Psiaki, T.E. Humphreys, B. Stauffer, Attackers can spoof navigation signals without our knowledge, here's how to fight back GPS lies, IEEE Spectr. 53 (8) (2016) 26–53.

[8] S. Shane, D.E. Sanger, Drone crash in Iran reveals secret us surveillance effort, N.Y. Times 7 (2011).

[9] K.C. Zeng, Y. Shu, S. Liu, Y. Dou, Y. Yang, A practical GPS location spoofing attack in road navigation scenario, in: Proc. International Workshop on Mobile Computing Systems and Applications (HotMobile), 2017.

[10] N.O. Tippenhauer, C. Pöpper, K.B. Rasmussen, S. Capkun, On the requirements for successful GPS spoofing attacks, in: Proc. ACM Conference on Computer and Communications Security (CCS), 2011.

[11] S. Zhu, T. Yang, C. Chen, VIGOR: Cross-view image geo-localization beyond one-to-one retrieval, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[12] S. Hu, M. Feng, R.M. Nguyen, G.H. Lee, CVM-NET: Cross-view matching network for image-based ground-to-aerial geo-localization, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[13] H. Yang, X. Lu, Y. Zhu, Cross-view geo-localization with layer-to-layer transformer, in: Proc. Conference on Neural Information Processing Systems (NIPS), 2021.

[14] Y. Shi, L. Liu, X. Yu, H. Li, Spatial-aware feature aggregation for image based cross-view geo-localization, in: Proc. Conference on Neural Information Processing Systems (NIPS), 2019.

[15] S. Hu, G.H. Lee, Image-based geo-localization using satellite imagery, Int. J. Comput. Vis. 128 (5) (2020) 1205–1219.

[16] S. Zhu, M. Shah, C. Chen, TransGeo: Transformer is all you need for cross-view image geo-localization, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[17] J. Petit, S.E. Shladover, Potential cyberattacks on automated vehicles, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2014) 546–556.

[18] P. Papadimitratos, A. Jovanovic, GNSS-based positioning: Attacks and countermeasures, in: Proc. IEEE Military Communications Conference (MILCOM), 2008.

[19] M. Ding, W. Chen, W. Ding, Performance analysis of a normal GNSS receiver model under different types of jamming signals, Measurement 214 (2023) 112786.

[20] M. Lenhart, M. Spanghero, P. Papadimitratos, Relay/replay attacks on GNSS signals, in: Proc. ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), 2021.

[21] J. Shen, J.Y. Won, Z. Chen, Q.A. Chen, Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under GPS spoofing, in: Proc. 29th USENIX Security Symposium, 2020, pp. 931–948.

[22] K. Wesson, M. Rothlisberger, T. Humphreys, Practical cryptographic civil GPS signal authentication, NAVIGATION: Journal of the Institute of Navigation 59 (3) (2012) 177–193.

[23] B.W. O'Hanlon, M.L. Psiaki, J.A. Bhatti, D.P. Shepard, T.E. Humphreys, Real-time GPS spoofing detection via correlation of encrypted signals, NAVIGATION: Journal of the Institute of Navigation 60 (4) (2013) 267–278.

[24] K. Jansen, N.O. Tippenhauer, C. Pöpper, Multi-receiver GPS spoofing detection: Error models and realization, in: Proc. ACM Annual Computer Security Applications Conference (ACSAC), 2016.

[25] M.R. Manesh, J. Kenney, W.C. Hu, V.K. Devabhaktuni, N. Kaabouch, Detection of GPS spoofing attacks on unmanned aerial systems, in: Proc. IEEE Consumer Communications and Networking Conference (CCNC), 2019.

[26] M. Sun, Y. Qin, J. Bao, X. Yu, GPS spoofing detection based on decision fusion with a k-out-of-n rule, International Journal of Network Security 19 (5) (2017) 670–674.

[27] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, J. Schmitt, Crowd-GPS-Sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks, in: Proc. IEEE Symposium on Security and Privacy (SP), 2018.

[28] P.F. Swaszek, R.J. Hartnett, M.V. Kempe, G.W. Johnson, Analysis of a simple, multi-receiver GPS spoof detector, in: Proc. ION International Technical Meeting (ITM), 2013.

[29] P.F. Swaszek, R.J. Hartnett, K.C. Seals, Using range information to detect spoofing in platoons of vehicles, in: Proc. ION GNSS+, 2017.

[30] S. Dasgupta, M. Rahman, M. Islam, M. Chowdhury, A sensor fusion-based GNSS spoofing attack detection framework for autonomous vehicles, IEEE Trans. Intell. Transp. Syst. 23 (12) (2022) 23559–23572.

[31] J.H. Lee, K.C. Kwon, D.S. An, D.S. Shim, GPS spoofing detection using accelerometers and performance analysis with probability of detection, Int. J. Control Autom. Syst. 13 (2015) 951–959.

[32] P. Jiang, H. Wu, C. Xin, DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network, Digit. Commun. Netw. 8 (2021).

[33] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, S. Savarese, Semantic cross-view matching, in: Proc. IEEE International Conference on Computer Vision Workshop (ICCVW), 2015.

[34] T.-Y. Lin, S. Belongie, J. Hays, Cross-view image geolocalization, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[35] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[36] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[37] Y. Shi, X. Yu, L. Liu, T. Zhang, H. Li, Optimal feature transport for cross-view image geolocalization, in: Proc. AAAI Conference on Artificial Intelligence (AAAI), 2020.

[38] L. Liu, H. Li, Lending orientation to neural networks for cross-view geo-localization, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[39] K. Regmi, M. Shah, Bridging the domain gap for ground-to-aerial image matching, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[40] S. Workman, R. Souvenir, N. Jacobs, Wide-area image geolocalization with aerial reference imagery, in: Proc. IEEE International Conference on Computer Vision (ICCV), 2015.

[41] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.

[42] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.

[43] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, BDD100K: A diverse driving dataset for heterogeneous multitask learning, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[44] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ADE20K dataset, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[45] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, M. Ebrahimi, EdgeConnect: Structure guided image inpainting using edge prediction, in: Proc. IEEE International Conference on Computer Vision (ICCV), 2019.

[46] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[47] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, 2014, arXiv preprint arXiv:1409.2329.

[48] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, A. Courville, Recurrent batch normalization, 2016, arXiv preprint arXiv:1603.09025.

[49] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, J. Weaver, Google street view: Capturing the world at street level, Computer 43 (6) (2010) 32–38.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, et al., Pytorch: An imperative style, high-performance deep learning library, in: Proc. Conference on Neural Information Processing Systems (NIPS), 2019.

[51] A. Eldosouky, A. Ferdowsi, W. Saad, Drones in distress: A game-theoretic countermeasure for protecting uavs against GPS spoofing, IEEE Internet Things J. 7 (4) (2019) 2840–2854.