SKOLTECHSIM: GPS SPOOFING DETECTION ARTIFACT

PROJECT REPORT

Oleg Sautenkov[*1], **Joshua Udobang**[1], **Aleksandr Zolotarev**[1], **and Yaroslav Solomentsev**[1]
[1]Skolkovo Institute of Science and Technology, Moscow, Bolshoy Boulevard, 30, p.1, 121205

November 18, 2024

ABSTRACT

Unmanned Aerial Vehicles (UAVs), better known as drones, have significantly advanced fields such as aerial surveillance, military reconnaissance, cadastral surveying, and disaster monitoring. However, UAVs rely on civilian GPS for navigation which can be trivially spoofed. In this paper, we present SkoltechSIM, a satellite imagery matching approach to detect GPS spoofing attacks against UAVs based on deep learning. The paper DeepSIM is taken as a baseline. Right now, we actively working on the SkoltechSIM project for the DL-2024 course.

Our code is available here: https://github.com/IaroslavS/DeepSim.git

Keywords Deep Learning · GPS-spoofing · Non-Gps Navigation · DL-course 2024

1 Introduction

DeepSIM is a GPS spoofing detection approach for UAVs via satellite imagery matching. The spoofing detection process involves collecting a reference dataset, acquiring relevant satellite imagery, and determining a threshold for the similarity between the real photos and the satellite images. This approach has several challenges, including differences in resolution and quality between UAV and satellite images, external factors such as weather and lighting, and the limited availability of paired UAV and satellite images for training deep learning models.

This system compares historical satellite images of its GPS-based position (spaceborne photography) with real-time aerial images from its cameras (airborne imagery). As the final project for the course, we updated the architecture and provided the metrics for our approaches. Historical images are taken from, e.g., Google Earth or NASA WorldWind.

To detect GPS spoofing attacks, we investigate different deep neural network models that compare the real-time camera images with the historical satellite images. To train and test the models, we have used the SatUAV dataset as well as the authors with the pair images from the satellite and drone. The approach of the authors showed 95 percent in detecting GPS spoofing attacks within less than 100 milliseconds.

Our plan for the project is to research dataset, methods, learn how to work with the satellite data, reproduce the results of the authors and reseach how different layers are changing the spoofing detection rate.

2 Dataset Description

Images in the dataset are part of two categories: aerial photography and satellite imagery. As of now, the total number of imagepairs in the dataset is 967 (appr. 12.08 Gigabyte).

In total, there are 967 aerial photos. Among them are 605 realistic scene photos with a light height of 120 m, that were captured using UAV; 343 of these photos were taken in Suzhou, China, and 20 photos were captured in Kunshan, China. The rest were gathered in Weihai, Shennongjia and Wuxi respectively. To test the generalization ability of the models, the authors additionally collected 107 photos from 4 areas in the UK. The camera used forshooting is DJI FC6310. To get suffcient details of the ground, the authors set the original pixel resolution to 5472×3078.

[*]Corresponding author: Oleg.Sautenkov@skoltech.ru

Figure 1: The examples in the Dataset

3 System Design

DeepSIM is the integration of UAVs, a ground-based UAV controller (or ground station), a spoofing indicator and communications be- tween UAVs and ground station. To simplify the model, we assume that the camera is mounted directly under the UAV, with the camera's movement being consistent with the UAV's movement.

3.1 Underlying Concept

Each part of the DeepSIM system works as follows: 1) The UAV processes GPS signals from the satellites by a GPS-capable receiver to determine its own position and it takes aerial photos of the ground using a camera. 2) The UAV controller provides an operation and monitoring platform for the UAV and is also responsible for running models to detect spoofing attacks (unless this is run on the UAV directly). 3) Communications refer to the channel for remote control and exchange of video and other data between the UAV and the UAV control center. 4) The spoofing indicator will alert administrators if spoofing attacks are detected.

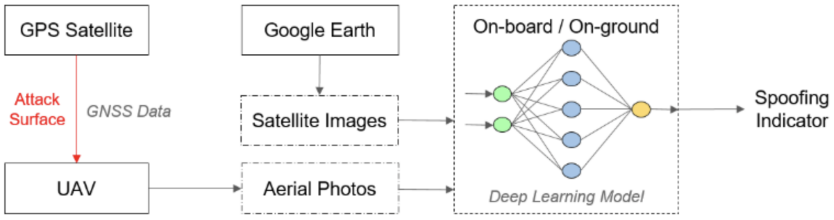


Figure 2: The system design of Deepsim

3.2 The results of SiameseResNet, proposed by authors

A pretrained ResNet-34 (without the last Fully Connected layer) played the role as the backbone CNN network in the model of authors. The final pooling layer of this pretrained ResNet-34 is an average pooling layer whose kernel size is 7 and stride is 1, the output feature map size is 24×17×512.

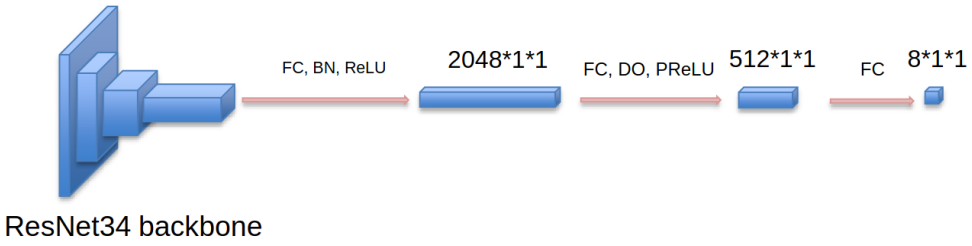
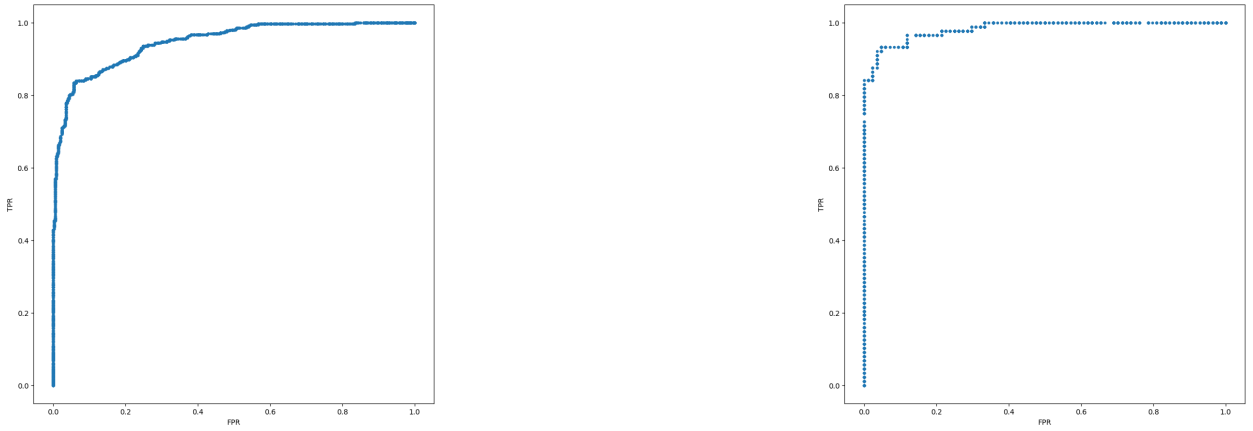


Figure 3: The architecture of the Siamese ResNet 34 proposed by authors of DeepSim (1)

The result metrics shown in the tables 1, 2.



(a) Training (b) Validation

Figure 4: ROC-curve of SiameseResNet proposed by DeepSim authors on training and validation sample

3.3 Our results with added layer

We attempted to enhance the final layers by adding an additional block consisting of a linear layer, batch normalization, and ReLU activation, as depicted in the figure. This modification aimed to improve the model's performance by potentially increasing its capacity to learn complex patterns. However, our results indicated a slight decrease in performance compared to those achieved by the original authors. This suggests that the additional complexity introduced by the new block may have led to overfitting or other issues that negatively impacted the model's generalization capabilities. Further investigation is needed to understand the underlying reasons for this decline in performance. 5.

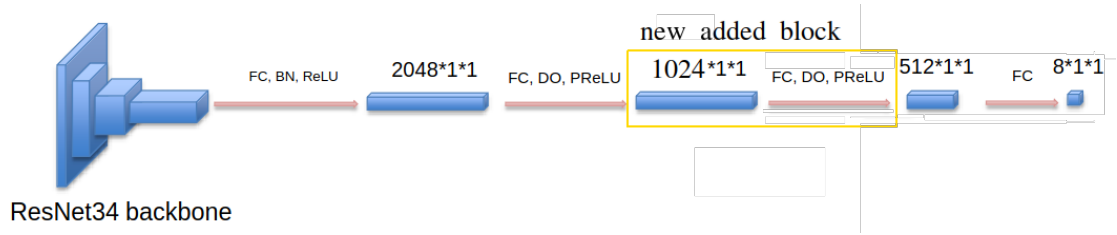


Figure 5: The architecture of the Siamese ResNet 34 with additional block (linear layer + BN + ReLU)

The result metrics are show in the tables 1, 2.

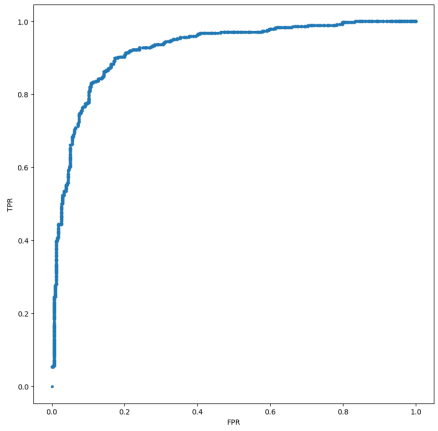
As we can see, the results are worse than those obtained by the original architecture proposed by the authors of DeepSim. This suggests that the additional complexity introduced by the new block may have led to overfitting or other issues that negatively impacted the model's generalization capabilities.

3.4 Our results with removed layer

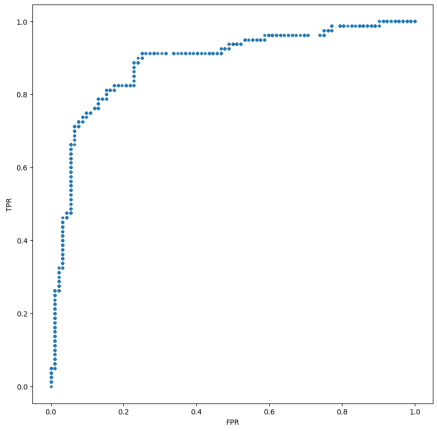
Additionally, we experimented with removing one block (linear layer + dropout + ReLU), as illustrated in Figure 7. The objective was to simplify the model architecture and potentially improve its efficiency. However, the resulting metrics, as shown in Tables 1 and 2, indicated a further decline in performance.

Once more, the outcomes are inferior to those achieved by the original architecture proposed by the authors of DeepSim. This suggests that the removed block played a crucial role in the model's performance, and its absence led to a loss of important features or regularization effects. Further analysis is necessary to comprehend the specific impacts of this modification and explore potential adjustments to mitigate the observed decline in results. 7

The result metrics shown in the tables 1, 2. And again, the results are worse than those obtained by the original architecture proposed by the authors of DeepSim.



(a) Training



(b) Validation

Figure 6: ROC-curve of SiameseResNet with an additional block (linear layer+batch normalization+ReLu)

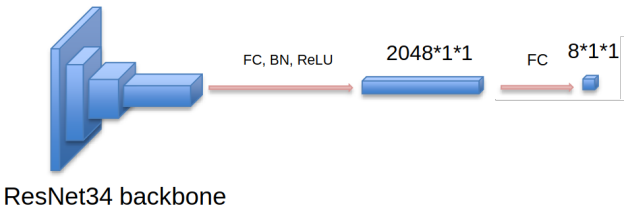
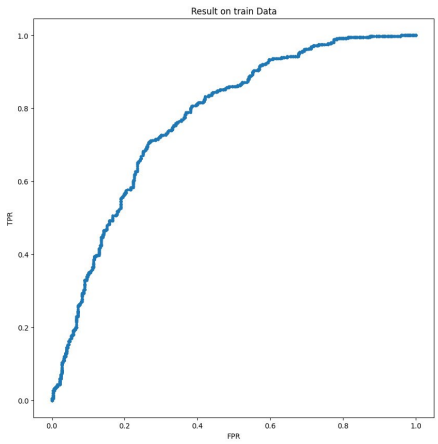


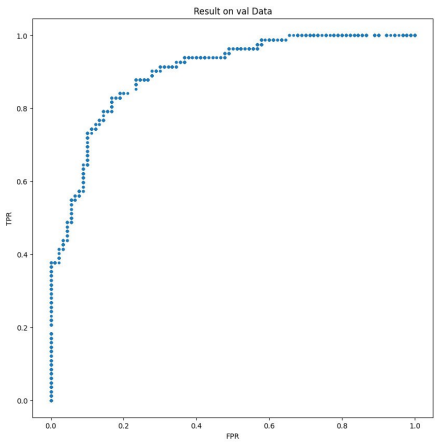
Figure 7: The architecture of the Siamese ResNet 34 with removed block

3.5 Our results with changed backbone (ResNext50)

One of the possible ways to enhance the quality of model was to change backbone. Thus, we created new model with ResNeXt50 backbone. The slide shows ROC curves. The quality turned out to be even worse than previous modifications or than the original. The model was trained on 100 epoches. The ROC-curves are shown on fig. 10



(a) Training



(b) Validation

Figure 8: ROC-curve of SiameseResNet with removed layer in the block

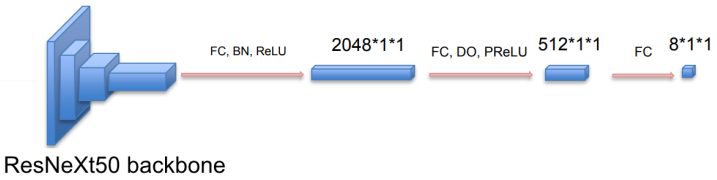
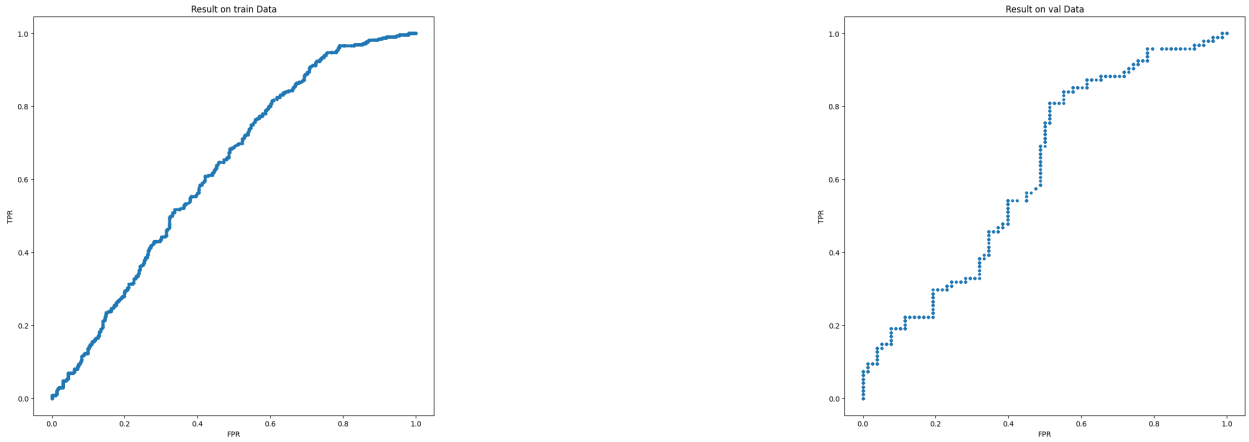


Figure 9: The architecture of the Siamese ResNeXt 50



(a) Training

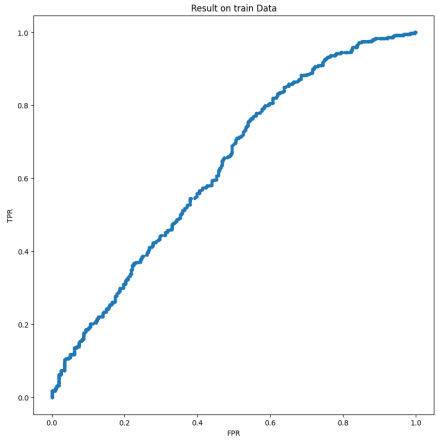
(b) Validation

Figure 10: ROC-curve of SiameseResNeXt 50

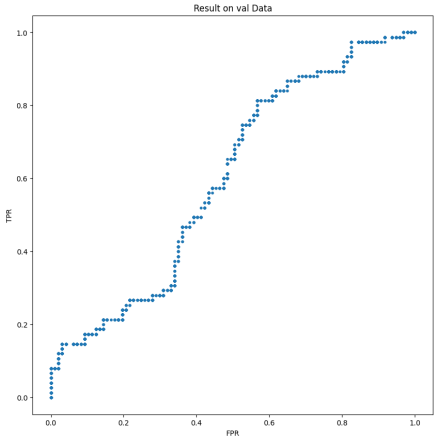
3.6 Our results with training ResNet34 on augmented data

The augmentation was done using python script https://github.com/Lariiii/DeepSimDataAugmentation/blob/master/augmentation.py. The augmentation simulates the following modifications or weather events:

- Grayscaling (fig. 12);
- Cropping (fig. 13);
- Rotating and cropping (fig. 14);
- Fog (fig. 15);
- Clouds (fig. 16);
- Darkening (fig. 17);
- Brightening (fig. 18);



(a) Training



(b) Validation

Figure 11: ROC-curve of SiameseResNeXt 50

4 Experimental Results

Table 1: Metrics of trained models on training sample

	TPR	FPR	Accuracy	Precision	Recall	F1-score
Original SiameseResNet34 proposed by authors	0.941	0.189	0.875	0.828	0.941	0.881
SiameseResNet34 with additional linear block	0.880	0.240	0.824	0.808	0.880	0.842
SiameseResNet34 without a linear block	0.722	0.297	0.712	0.701	0.722	0.711
SiameseResNeXt50	0.75	0.550	0.594	0.560	0.75	0.641
Original SiameseResNet34 on augmented data	0.776	0.561	0.619	0.613	0.776	0.685

Table 2: Metrics of trained models on validation sample

	TPR	FPR	Accuracy	Precision	Recall	F1-score
Original SiameseResNet34 proposed by authors	0.967	0.183	0.895	0.853	0.967	0.906
SiameseResNet34 with additional linear block	0.893	0.340	0.773	0.714	0.893	0.794
SiameseResNet34 without a linear block	0.821	0.227	0.796	0.775	0.821	0.798
SiameseResNeXt50	0.840	0.551	0.663	0.648	0.840	0.731
Original SiameseResNet34 on augmented data	0.827	0.608	0.581	0.512	0.827	0.633

5 Planned work

We plan to analyze further the possible structure for the GPS-spoofing detection approach and test them on this dataset. To be honest, it will be quite challenging to outperform the authors. However, we would like to test the architectures that are newer than architectures suggested by the author.

6 Conclusion

Throughout our study, we delved into the SatUAV dataset, thoroughly exploring its intricacies and characteristics to better understand its nuances and potential applications. In the process, we acquired proficiency in utilizing various augmentations techniques to enhance the dataset's diversity and robustness, equipping us with valuable skills for data preprocessing and augmentation.

Furthermore, we embarked on a journey of experimentation with different architectures, aiming to identify the most suitable model for our task at hand. While we explored modern architectures in our quest for improved performance, the endeavor proved to be significantly time-consuming, yielding results that fell short of surpassing those achieved by the original authors.

Interestingly, our closest approximation to the authors' results was attained through the utilization of an older architecture supplemented with an additional linear block. This observation highlights the importance of considering both traditional and contemporary approaches in model design and evaluation.

Moreover, our attempts to leverage the augmentation techniques employed by the authors did not yield any discernible improvements in model performance. This discrepancy raises questions regarding the transparency and efficacy of the authors' augmentation pipeline and its reported results, warranting further investigation into the underlying factors influencing model performance and the reliability of augmentation strategies utilized in the original study.

7 Contributions

Oleg Sautenkov - reproduced results, calculated metrics with removed layer. Yaroslav Solomentsev - reproduced results, calculated metrics with added layer. Aleksandr Zolotarev - report, graphics. Joshua Udobang - working on a new architecture

(a) Original

(b) Grayscaled

Figure 12: Augmentation with grayscaling

(a) Original

(b) Cropped

Figure 13: Augmentation with cropping

(a) Original

(b) Rotated and cropped

Figure 14: Augmentation with rotation and cropping

(a) Original

(b) With fog

Figure 15: Augmentation with fog

(a) Original

(b) With clouds

Figure 16: Augmentation with clouds

(a) Original

(b) Darkened

Figure 17: Augmentation with darkening

(a) Original

(b) Darkened

Figure 18: Augmentation with brightening

References

[1] N. Xue, L. Niu, X. Hong, Z. Li, L. Hoffaeller, and C. Pöpper, "Deepsim: Gps spoofing detection on uavs using satellite imagery matching," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2020.

[2] M. M. Gurgu, J. P. Queralta, and T. Westerlund, "Vision-based gnss-free localization for uavs in the wild," in *2022 7th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*. IEEE, 2022, pp. 7–12.

[3] Z. U. Sautenkov, Solomentsev, "SkoltechSIM: GPS Spoofing Detection on UAVs using Satellite Imagery Matching," May 2024. [Online]. Available: https://github.com/IaroslavS/DeepSim