

# Near Real-Time Detection of Poachers from Drones in AirSim

Elizabeth Bondi<sup>1</sup>, Ashish Kapoor<sup>2</sup>, Debadeepta Dey<sup>2</sup>, James Piavis<sup>2</sup>, Shital Shah<sup>2</sup>  
 Robert Hannaford<sup>3</sup>, Arvind Iyer<sup>3</sup>, Lucas Joppa<sup>2</sup>, Milind Tambe<sup>1</sup>

<sup>1</sup>University of Southern California

<sup>2</sup>Microsoft

<sup>3</sup>Air Shepherd

bondi@usc.edu

## Abstract

The unrelenting threat of poaching has led to increased development of new technologies to combat it. One such example is the use of thermal infrared cameras mounted on unmanned aerial vehicles (UAVs or drones) to spot poachers at night and report them to park rangers before they are able to harm any animals. However, monitoring the live video stream from these conservation UAVs all night is an arduous task. Therefore, we discuss SPOT (Systematic POacher detector), a novel application that augments conservation drones with the ability to automatically detect poachers and animals in near real time [Bondi *et al.*, 2018b]. SPOT illustrates the feasibility of building upon state-of-the-art AI techniques, such as Faster RCNN, to address the challenges of automatically detecting animals and poachers in infrared images. This paper reports (i) the design of SPOT, (ii) efficient processing techniques to ensure usability in the field, (iii) evaluation of SPOT based on historical videos and a real-world test run by the end-users, Air Shepherd, in the field, and (iv) the use of AirSim for live demonstration of SPOT. The promising results from a field test have led to a plan for larger-scale deployment in a national park in southern Africa. While SPOT is developed for conservation drones, its design and novel techniques have wider application for automated detection from UAV videos.

## 1 SPOT

SPOT includes two main parts: (i) offline training and (ii) online detection. Fig. 1 shows the steps of both, but we now introduce online detection in detail, with an emphasis on the processing techniques we use to improve efficiency and provide detections in near real time in our demonstration.

### 1.1 Running

We treat each frame of the video stream as an image and pass it to Faster RCNN [Ren *et al.*, 2015]. The trained model computes regions and classes associated with each region.

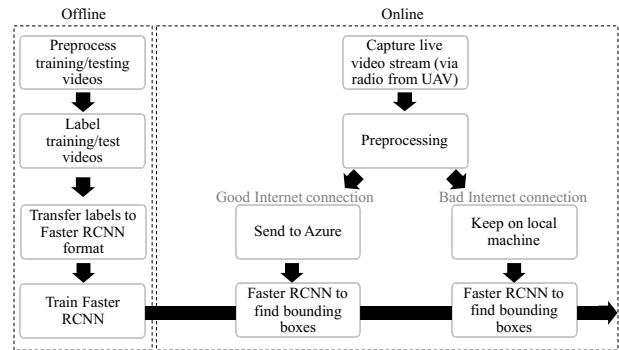


Figure 1: SPOT Overview.

### 1.2 Architecture and Efficiency

Faster RCNN runs at 5 frames per second (fps) on a K40 GPU. Because the live video is 25 fps, efficiency is vital to similar timing with limited computational power in the field. Note that GPUs have recently become available in the field. However, in the case that they are not available, we also provide a solution taking advantage of Microsoft Azure. To communicate between the client and Azure, Tensorflow Serving, a framework for operationalizing trained Tensorflow computation graphs, provides a way to evaluate Faster RCNN with less overhead and efficient communication. We also use azure-engine to create a cluster of NC-6 series GPU VMs managed with Kubernetes, a fault tolerant load balancer for scalable cloud-based services. We deploy on a GPU-enabled docker image with Tensorflow Serving, and add tools for convenient and automatic re-deployment of models hosted on Azure Blob Storage. In the field, SPOT is deployed with the option to toggle between remote and local computation. For the purposes of the demonstration, we will make use of the local compute option on an Azure virtual machine.

## 2 Related Work and Design Choices

We arrive at the current framework of SPOT after several rounds of trials and errors with relevant traditional computer vision techniques, including automatic thresholding [Otsu, 1979], motion algorithms [Lucas *et al.*, 1981], and shape de-

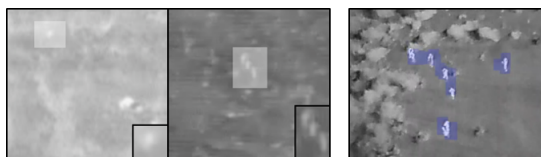


Figure 2: Example UAV, thermal frames, with boxes drawn around poachers manually (left) and automatically (right).

Video	Precision		Recall	
	SPOT	ES	SPOT	ES
SA	0.5729	0.1536	0.0025	0.0072
MA	0.5544	0.0032	0.0131	0.0058
LA	0.5584	0.0235	0.2293	0.0694

Table 1: Precision-Recall for SPOT and EyeSpy (ES) for animals of size small (SA), medium (MA), and large (LA).

tection algorithms, such as EyeSpy, the application Air Shepherd uses in current practice [Hannaford, 2017]. All struggle due to the difficulty of thermal infrared images (Fig. 2). There has also been some work on automatic wildlife detection in UAV videos, but they either rely on RGB images in high resolution [Olivares-Mendez *et al.*, 2015] or do not consider real-time detection [van Gemert *et al.*, 2014]. Given these limitations and the great strides in object detection using convolutional neural networks, we turned to Faster RCNN.

### 3 Evaluation

We first evaluate SPOT using six historical videos, consisting of 15,403 frames in total, as well as the field test video with poachers (Fig. 2). We compare to EyeSpy, the application that is used in current practice. It requires users to tune 14 parameters, which is difficult for a novice, whereas ours can be used without tuning. We therefore consider EyeSpy as used by a novice by utilizing default values for all parameters. We use an IoU threshold of 0.5 and a low confidence threshold for all calculations, and we report the precision and recall. We compare the performance of SPOT and EyeSpy on videos containing animals and poachers with labels of small, medium, or large average sizes in Tables 1 and 2. We perform better than the novice in both precision and recall for medium- and large-sized poachers and animals, and in the field test video. Because we perform better than EyeSpy, we thus reduce significant burden. For small poachers, which is a challenging task for object detection in general, both tools perform poorly, with EyeSpy being able to identify a small number of poachers correctly. To improve performance for small objects in the future, we expect pooling the results of video frames and incorporating motion will be beneficial.

Next, we evaluate efficiency by comparing the field CPU and GPU laptops, and Azure system performance. We conduct the Azure tests in two different networking environments: 533.20 Mbps upload and 812.14 Mbps download, which we call “fast”, and 5.33 Mbps upload and 5.29 Mbps download, which we call “slow”. We repeat the experiment for several images and show the final time per image in Table 3. The results show that both Azure and the GPU laptop

Video	Precision		Recall	
	SPOT	ES	SPOT	ES
SP	0	0.00003	0	0.0007
MP	0.0995	0.0004	0.0073	0.0009
LP	0.3977	0.0052	0.0188	0.0159
Test	0.4235	0.0024	0.3697	0.0432

Table 2: Precision-Recall for SPOT and EyeSpy (ES) for poachers of size small (SP), medium (MP), and large (LP).

	# GPUs	Network	s/img
CPU	0	-	10.4354
GPU	1	-	0.3870
Azure	2	fast	0.3484
Azure	2	slow	0.4858

Table 3: Timing Results for CPU, Azure, and GPU.

perform detection much faster than the CPU laptop.

### 4 Demonstration

In order to demonstrate SPOT, we utilize AirSim [Shah *et al.*, 2017]. AirSim is a simulator for drones and cars built on Unreal Engine to allow artificial intelligence researchers to experiment with deep learning, computer vision, and reinforcement learning algorithms for autonomous vehicles. For a live demonstration of SPOT in an African savanna, we built an African savanna environment in Unreal Engine. We also modeled thermal infrared frames using a simple physics-based model and fine-tuned Faster RCNN with the simulated infrared frames to improve detection results [Bondi *et al.*, 2018a]. Results shown in Tables 1 and 2 use real data only.

A sample video can be found on *Google Drive*. The video provides the scene and corresponding thermal infrared images. This is an example of navigation in AirSim with a Python API, along with the near real-time detections of SPOT. Demonstration users will be able to interact with AirSim by flying around the environment.

In conclusion, SPOT is a novel decision aid to assist the UAV crew in detecting poachers and animals in thermal infrared videos captured aboard drones. It performs better than EyeSpy, both on historical data and in the field, provides detections in near real time, and does not require tuning by the user during run time. The promising results from the field test have led to a plan for larger-scale deployment in a national park in southern Africa. To demonstrate SPOT, we utilize an African savanna environment in AirSim that allows users to interactively fly through a savanna, while also running SPOT for live, near real-time detection.

### Acknowledgments

This was supported by Microsoft AI for Earth, the National Science Foundation (CCF-1522054), and UCAR N00173-16-2-C903, with the primary sponsor being the Naval Research Laboratory (Z17-19598). It was partially supported by the Viterbi School of Engineering Ph.D. Merit Top-Off Fellowship.

## References

- [Bondi *et al.*, 2018a] Elizabeth Bondi, Debadeepta Dey, Ashish Kapoor, Jim Piavis, Shital Shah, Fei Fang, Bistra Dilkina, Robert Hannaford, Arvind Iyer, Lucas Joppa, and Milind Tambe. Airsim-w: A simulation environment for wildlife conservation with uavs. In *ACM COMPASS*, 2018.
- [Bondi *et al.*, 2018b] Elizabeth Bondi, Fei Fang, Mark Hamilton, Debarun Kar, Donnabell Dmello, Jongmoo Choi, Robert Hannaford, Arvind Iyer, Lucas Joppa, Milind Tambe, and Ram Nevatia. Spot poachers in action: Augmenting conservation drones with automatic detection in near real time. In *IAAI*, 2018.
- [Hannaford, 2017] Robert Hannaford. Eyespy. Private Communication, 2017.
- [Lucas *et al.*, 1981] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.
- [Olivares-Mendez *et al.*, 2015] Miguel A Olivares-Mendez, Changhong Fu, Philippe Ludvig, Tegawendé F Bisseyandé, Somasundar Kannan, Maciej Zurad, Arun Annaiyan, Holger Voos, and Pascual Campoy. Towards an autonomous vision-based unmanned aerial system against wildlife poachers. *Sensors*, 15(12):31362–31391, 2015.
- [Otsu, 1979] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.
- [Shah *et al.*, 2017] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.
- [van Gemert *et al.*, 2014] Jan C van Gemert, Camiel R Verschoor, Pascal Mettes, Kitso Epema, Lian Pin Koh, Serge Wich, et al. Nature conservation drones for automatic localization and counting of animals. In *ECCV Workshops (1)*, pages 255–270, 2014.