AirSim-W: A Simulation Environment for Wildlife Conservation with UAVs

Elizabeth Bondi[1], Debadeepta Dey[2], Ashish Kapoor[2], Jim Piavis[2], Shital Shah[2], Fei Fang[3], Bistra Dilkina[1], Robert Hannaford[4], Arvind Iyer[4], Lucas Joppa[2], Milind Tambe[1]
[1]University of Southern California, {bondi, dilkina, tambe}@usc.edu
[2]Microsoft, {dedey, akapoor, v-jimpi, shitals, lujoppa}@microsoft.com
[3]Carnegie Mellon University, feifang@cmu.edu
[4]Air Shepherd, rob@coolideassolutions.com, arvind.iyer@lindberghfoundation.org

ABSTRACT

Increases in poaching levels have led to the use of unmanned aerial vehicles (UAVs or drones) to count animals, locate animals in parks, and even find poachers. Finding poachers is often done at night through the use of long wave thermal infrared cameras mounted on these UAVs. Unfortunately, monitoring the live video stream from the conservation UAVs all night is an arduous task. In order to assist in this monitoring task, new techniques in computer vision have been developed. This work is based on a dataset which took approximately six months to label. However, further improvement in detection and future testing of autonomous flight require not only more labeled training data, but also an environment where algorithms can be safely tested. In order to meet both goals efficiently, we present AirSim-W, a simulation environment that has been designed specifically for the domain of wildlife conservation. This includes (i) creation of an African savanna environment in Unreal Engine, (ii) integration of a new thermal infrared model based on radiometry, (iii) API code expansions to follow objects of interest or fly in zig-zag patterns to generate simulated training data, and (iv) demonstrated detection improvement using simulated data generated by AirSim-W. With these additional simulation features, AirSim-W will be directly useful for wildlife conservation research.

CCS CONCEPTS

• **Computing methodologies** → *Object recognition*;

KEYWORDS

unmanned aerial vehicles; drones; object detection; wildlife conservation; simulation

ACM Reference Format:
Elizabeth Bondi[1], Debadeepta Dey[2], Ashish Kapoor[2], Jim Piavis[2], Shital Shah[2], Fei Fang[3], Bistra Dilkina[1], Robert Hannaford[4], Arvind Iyer[4], Lucas Joppa[2], Milind Tambe[1]. 2018. AirSim-W: A Simulation Environment for Wildlife Conservation with UAVs. In *COMPASS '18: ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS), June 20–22, 2018, Menlo Park and San Jose, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3209811.3209880

Figure 1: Example conservation UAV used by Air Shepherd.

1 INTRODUCTION

Wildlife conservation is one of the most important sustainability goals today, and innovations in artificial intelligence are uniquely suited to advancing it. When it comes to wildlife poaching in particular, artificial intelligence has already played an important mitigating role. In order to maximize the protection of national parks and conservation areas, it has been used to assist park rangers in planning their patrols to find poachers and snares, both in predicting future poaching incidents [16, 22] and creating strategies to detect poaching or signs of poaching activity [19, 62]. Recent advances in unmanned aerial vehicle (UAV or drone) technology have made UAVs viable tools to aid in park ranger patrols. UAVs can play a role in patrolling by deterring poaching through the use of signaling [63], serving as a lookout for park rangers, or even acting as a separate patroller when equipped with the ability to automatically detect animals and poachers in UAV videos. An example of a conservation UAV flown by the conservation program, Air Shepherd, is shown in Fig. 1.

The ability to detect animals and poachers in UAV videos, particularly thermal infrared videos, is an active area of research due to the small size of humans and animals in UAV videos, the UAV motion, and the low-resolution, single-band nature of thermal infrared videos. In our previous work, a dataset of 70 historical thermal infrared videos was labeled [13]. These videos were collected by Air Shepherd between 2015 and 2017 during flights which typically occur at night based on pre-programmed paths. Flights go on for about 8 hours per night, with individual flights that are 2 hours long due to battery life. When objects of interest are observed on these flights, the UAVs are flown manually in order to follow the objects

COMPASS '18, June 20–22, 2018, Menlo Park and San Jose, CA, USA

ACM ISBN 978-1-4503-5816-3/18/06…$15.00
https://doi.org/10.1145/3209811.3209880

of interest. Often, however, videos do not have many objects of interest, or it is difficult to identify objects of interest in the videos as human observers. This means that videos had to be checked for content first before labeling, which added additional time to the process. In total, **this labeling process took approximately 800 hours over the course of 6 months to complete**, and produced 39,380 labeled frames and approximately 180,000 individual poacher and animal labels on those frames. At a rate of $11 per hour, this cost about $8,800 for labeling alone, plus flying costs between 2015 and 2017. Together, the time and money associated with labeling make it extremely difficult to collect large labeled datasets like this.

Once the 70 videos had been labeled, individual frames were used to train Faster RCNN [44] for animal and poacher detection, which was part of a larger system called SPOT [12]. Training was completed on 22,663 total frames, with 18,480 total frames for the animal model and 4,183 frames for the poacher model. Note that these models each detect both animals and poachers, but due to the random sampling, the animal model performed better at detecting animals than other tested models, and the poaching model performed better at detecting poachers than the other tested models. SPOT performed better than the existing tool used by Air Shepherd.

Although SPOT is immediately useful to park rangers in the field as a decision aid, park rangers or others hired to monitor the videos are still required to confirm human detections made by SPOT and then manually fly the UAV to follow the human. In order to improve detection performance, more labeled training data is needed. Additionally, to further relieve the burden on rangers, we would like to allow for autonomous flight to follow planned patrol routes, deviate from the plans as needed to further investigate possible detections, and automatically follow detected humans. However, testing of autonomous flight in the field could be costly, as mistakes could lead to poached animals. Existing work does not address these unique challenges, so we propose a new method based on simulation of the domain environment. This allows us to augment our dataset of labeled thermal infrared videos efficiently, and to provide a testing environment for future autonomous flight and other costly experiments in the domain of wildlife conservation, such as patrol planning.

To build a simulation with these features, we use Unreal Engine and AirSim [51]. Unreal Engine is a game engine where various environments and characters can be created, and AirSim is a simulator for drones and cars built on Unreal Engine. AirSim supports hardware-in-the-loop (e.g., Xbox controller) or a Python API for moving through the Unreal Engine environments, such as cities, neighborhoods, and mountains. AirSim specifically consists of a vehicle model for the UAV, which is modeled as a quadrotor, an environment model, made up of gravity, magnetic field, and air pressure and density models, a physics engine for the linear and angular drag, accelerations, and collisions, and finally a sensor model for the barometer, gyroscope and accelerometer, magnetometer, and GPS. The models are created such that real-time flights are possible. As a result, scene, segmentation, and depth images can be collected during flights or drives through the environments, which allows artificial intelligence researchers to experiment with deep learning, computer vision, and reinforcement learning algorithms for autonomous vehicles.

In this work, we present AirSim-W, which includes the (i) creation of an African savanna environment in Unreal Engine, (ii) expansion of the current RGB version of AirSim to include a thermal infrared model based on physics, (iii) expansions to follow objects of interest or fly in zig-zag patterns to generate simulated training data, and (iv) demonstrated detection improvement using simulated data generated by AirSim-W. With these contributions, AirSim-W will be directly used for wildlife conservation research, especially for the challenges of poacher and animal detection in UAV videos and patrol planning for UAVs and foot patrols.

2 RELATED WORK

First of all, the main problem of interest is to utilize simulation for wildlife conservation. For the problem of automatic detection of wildlife and humans in UAV videos, in addition to SPOT [12], there has also been some work on wildlife counting based on videos from UAVs using primarily traditional computer vision or machine learning techniques, including [39] and [57]. They either rely on RGB images in high resolution or do not consider real-time detection, and SPOT has shown improvement over a traditional computer vision result in near real time.

To improve on these results, we now examine data augmentation. Performance is often improved by increasing the amount of data used during training. For example, to train AlexNet [30], simple data augmentation involving cropping, translation, and horizontal reflections was utilized to increase the size of the training dataset by a factor of 2048, which helped reduce overfitting. They further augmented the dataset using PCAs to perturb digital counts. More recently, deep learning models such as generative adversarial networks (GANs) and recurrent neural networks (RNNs) have shown great promise in the realm of data augmentation [14, 24, 27, 43, 55]. In [43], deep convolutional GANs (DCGANs) are used to augment datasets and even draw certain objects, such as a bedroom. Style transfer and image-to-image translation are other areas being considered for data augmentation [36, 67]. These could be used to take many images of horses and convert them to zebras, or convert images taken in daylight to nighttime images, all of which may help with a specific computer vision task. However, these methods (i) do not account for thermal infrared imagery, and (ii) do not consider the physical processes that are involved in image capture, such as movement of the image capture platform.

Further data augmentation is possible using simulation from computer graphics. There are many examples of environments that have been built using rendering tools such as Unity [46] and Unreal Engine [51]. Digital Imaging and Remote Sensing Image Generation (DIRSIG) [26] is another example, where facetized surface models can be generated using AutoCAD, 3ds Max, Rhinoceros, Blender3D, or SketchUp, for example. Some environments exist with physics engines that allow for testing robotics systems within the environment, such as autonomous cars or UAVs. There are many of these environments, but we will only mention AirSim and Gazebo [29, 51]. In any case, datasets can then be generated using these environments. For example, SYNTHIA [46] is a dataset generated by capturing images in a city environment in Unity, and has shown improvement in semantic segmentation. GANs have been used to give simulated data like SYNTHIA a more realistic look and to further improve semantic segmentation [52, 65]. Few models examine the thermal domain, except DIRSIG, which uses a full radiometric model for thermal simulation, and [18], which uses simple 3D CAD models of solitary objects and a basic radiometric model.

Specifically in domains where little training data exists, undertaking the task of labeling large amounts of data can be time consuming and tedious, and data augmentation may be a necessity. For example, in the self-driving car domain, [28] trains a model using only simulated data to improve over the same model trained entirely on real data, while testing on real data. Video games such as GTA V can also be used to collect eye movement data for driving [9]. Another example in which there is a specific domain that may not have enough data in existing datasets is [53], where a mapping from simulated data from Unity to more realistic simulated data is learned and applied, and the results are used to train reactive obstacle avoidance and semantic segmentation neural networks.

For the purposes of our wildlife conservation domain, we require (i) data augmentation capabilities for computer vision tasks and (ii) a full simulation environment for future development in ranger and UAV patrol planning and autonomous UAV flights for conservation purposes. As already mentioned, GAN models have shown promise in data augmentation, but they do not account for thermal infrared imagery and the physics behind image capture. Of the simulation environments mentioned, although all promising for data augmentation, only AirSim and Gazebo allow for future conservation patrol and autonomous flight testing.

In this paper, we seek to build a tool suitable for the wildlife conservation domain. We will utilize AirSim due to the ability to use Unreal Engine as the underlying rendering tool. We will generate our environment, which will be an African savanna, in Unreal Engine. This will allow us to capture images in real time with AirSim [51], and to control actual flight parameters for image capture. We will also create a basic model of the physical characteristics of thermal infrared cameras to expand the performance of Unreal Engine and AirSim for training data generation in the poacher and animal detection domain. Finally, we will utilize our novel simulation technique in the area of wildlife conservation in particular, and it can be downloaded and easily used here: https://github.com/Microsoft/AirSim.

3 AFRICAN SAVANNA ENVIRONMENT

To effectively run simulation of thermal infrared imagery capture, we needed to build out an environment that was similar to biomes found in the central African savanna, when viewed through imagery captured at an altitude from 200 to 400 feet (61 to 122 m) above ground level (AGL). We used web-sourced target images and Google Earth to visualize the environment in several national parks where Air Shepherd has flown previously. Visual targets varied from wide-open savanna plains to dense forest, and flatland to craggy canyons. Because of this large range, we chose to develop a representative biome rather than a facsimile of an existing location. Key features were wide-open space, dense forest, a mid-density area, a water feature, road access, and poachers and appropriate animals.

We first included the correct plants, animals, and humans. Flora in the area generally consists of baobab, acacia, and hookthorn trees, as well as brush and grass. We were able to find accurate vegetation models for each of the tree types from an existing 3D model vendor, SpeedTree. We were also able to find a variety of pre-animated and rigged animals including elephants, rhinoceroses, hippopotamuses, zebras, lions, and crocodiles in the Unreal Engine Marketplace. Animals can also be found at TurboSquid, another 3D model vendor. Note that while we have not seen hippopotamuses or crocodiles in real data, we are able to model them in this simulated environment, allowing us to train on features which are lacking in our dataset. This is extremely useful as it allows us to address issues such as missing data or class imbalances in data, which is another benefit of using simulation. Our three poacher characters were the only assets that were custom, and were created with Autodesk, leveraging animation created from a motion capture suit to give a realistic walking motion.

Then, the general flow for the environment creation follows typical game environment workflow. We created the one square mile flat terrain, then sculpted in hills and depressions for water, with the water in the center of the map. The Unreal Engine scale unit is 1 cm, so we started with a rectangular polygon of 6 feet in length to appropriately scale people and animals in the scene. Following this, we created spline-based movement of the actors before starting the scene dressing. The Path Follow plug-in, which can be found on the Unreal Engine Marketplace, was used to create the actor movement as it provided a better movement capability than the native UE4 spline-based movement.

We next started dressing the scene. A water plane was added and adjusted for the desired water level. Vegetation was added with the native paintbrush capability using various densities to reflect dense, mid, and sparse areas, and was repeated for each of the vegetation types. Instead of painting performance-reducing grass across the entire scene, textures were created to reflect the look we desired for improved performance during real-time video capture. A dirt road was cut into the scene and textured appropriately, and two vehicles were sourced to add to the scene.

The scene reflects three generic areas of vegetation density to support imagery targets across all three areas with three sets of poachers added to the scene. A set of poachers consists of three individual characters with each set following a spline in a large loop. We intersected the poacher loops with elephants on spline loops to capture images of both poachers and animals together. Additionally, zebras were scattered across the environment and animals were clustered around the watering hole.

Overall, the Africa environment was created in approximately 3 working weeks with an artist and part-time developer, totaling approximately $5000 and about 180 hours. The bulk of the time spent on this scene was the terrain, watering hole, vegetation, and design of the NPC movement, with a lesser amount of time on creating the animal and poacher spline movement. Several example images from the environment are shown in Fig. 2.

Should these costs be unmanageable to those in the conservation domain when considering environments other than an African savanna, transfer learning is a low-cost possibility to consider in the future, especially because the Africa environment is being made freely available through Microsoft AirSim (https://github.com/Microsoft/AirSim/releases). In addition, many of the assets used in the Africa environment came from the Unreal Engine Marketplace. There are likely environments, animals, and plants from

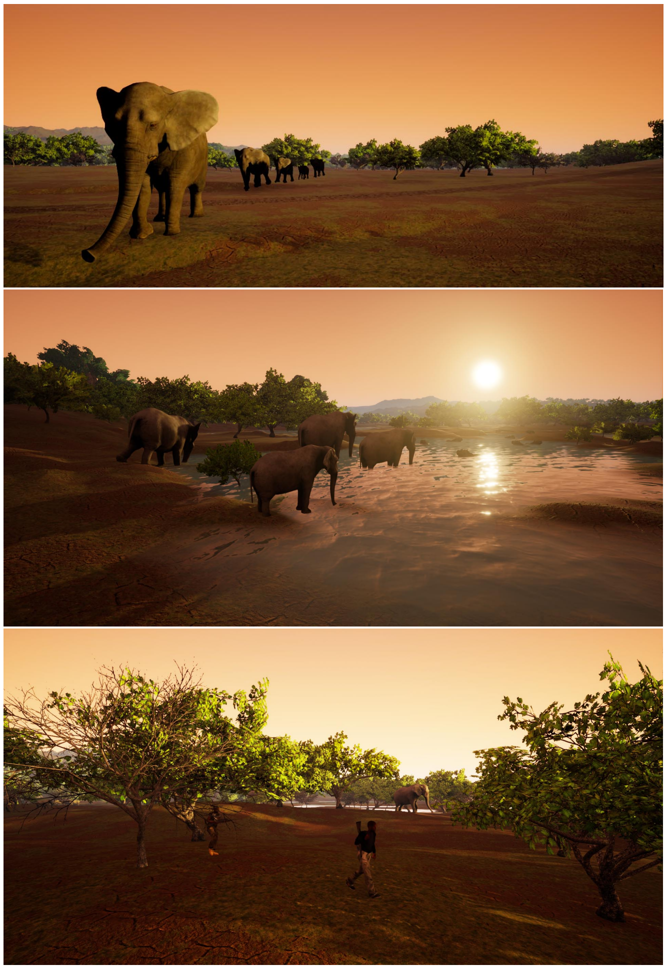


Figure 2: Example still images from the Africa environment.

other regions that could be simply bought and used directly. Together, these facts make creating an environment other than an African savanna for other domains possible at a relatively low cost.

4 EXPANSION FROM RGB TO THERMAL INFRARED

4.1 Physical Modeling Assumptions

Although the African savanna environment is already useful by itself, we must expand it to include thermal infrared imagery in order to augment our dataset for detecting animals and humans in thermal infrared imagery. Simulated RGB imagery alone is not useful because flights are done at night, when RGB imagery is not available. Additionally, we pre-train Faster RCNN using ImageNet, which is a database including millions of RGB images that can be used by the network to understand edges and shapes before learning the specific thermal infrared image domain.

In order to simulate thermal infrared imagery from the RGB imagery in AirSim, particularly the resulting segmentation map, we will rely on physical modeling. Due to the large number of interactions between photons and objects in or near the scene, modeling light can become extremely complicated. In the thermal domain at night, for example, thermal light reaching the camera on a UAV could come from several different sources: (i) atmosphere at some temperature emitting thermal infrared photons directly into the sensor, (ii) atmosphere at some temperature emitting thermal infrared photons that hit the ground and are reflected by the target into the sensor, (iii) thermal infrared photons emitted directly from the target into the sensor (this can be modeled using Planck's Law [49]), and (iv) thermal infrared photons emitted by nearby objects that are then reflected by the target into the sensor. These different contributions are called upwelled, downwelled, direct, and background radiance, respectively [49]. In addition to the atmosphere contributing photons directly to the signal, it can also play a role whenever photons travel from the target to the sensor. Depending on whether it is humid, cloudy, rainy, etc., this role can be larger or smaller, and is often modeled by radiative transfer models such as MODTRAN [11]. Other effects on the signal include the uniformity with which the objects of interest emit light (e.g., whether or not they are Lambertian), camera spectral response, and camera sensor noise, especially non-uniformity correction in microbolometers.

Because all of this involves a significant amount of modeling of complex physical phenomena, we will make simplifying assumptions to create a simplistic physical model of the thermal infrared image that would result from objects in the African savanna at certain temperatures. First, upwelled radiance and downwelled radiance are negligible with a clear, dry, cool atmosphere. Most of the year this would hold true in Africa, except during rainy season in the summer, when flights are not likely to take place anyway. A clear, dry, cool atmosphere also has negligible effects on transmission. Background radiance is negligible in cases of mostly flat terrain, which generally applies in a savanna. This means that the dominant contribution is direct, so we do not consider the contributions of the atmosphere to the signal, nor do we consider the transmission of the atmosphere because we assume it is clear, dry, and cool. We must also assume that objects emit energy uniformly (e.g., Lambertian objects) in order to use Planck's Law to model the direct contribution. The camera spectral response is measurable, and an estimate for a similar FLIR sensor was available [6]. Finally, we assume that the camera lens has perfect transmission and no falloff. These last two assumptions are false. However, these and some of the other effects we are assuming to be negligible could be accounted for in the future either by including them in the calculations explicitly, or with a technique such as style transfer [36] or image-to-image translation [67].

Given these assumptions, we model the signal at the sensor using only the direct contribution, given by Planck's Law (Eq. 1):

$$L(T, \epsilon_{avg}, R_\lambda) = \epsilon_{avg} \int_{\lambda=8\mu m}^{\lambda=14\mu m} R_\lambda \left(\frac{2hc^2}{\lambda^5} \frac{1}{\exp(\frac{hc}{kT\lambda}) - 1} \right) d\lambda \quad (1)$$

where L is radiance $[W/m^2/sr]$, T is temperature [K], ϵ_{avg} is the average emissivity over the bandpass, R_λ is the peak normalized camera spectral response, h is Planck's constant, c is the speed of light, λ is the wavelength $[\mu m]$, and k is the Boltzmann constant.

Emissivity, a value ranging between 0 and 1, relates the radiation of a real object to that of a blackbody, which is a perfect emitter. A blackbody would have an emissivity of 1, and a real object would have an emissivity less than 1. Emissivity is wavelength dependent, but we consider the average over the wavelengths to which the thermal infrared camera is sensitive.

We can calculate this integrated radiance for all objects in our segmentation map from AirSim. For example, given the pixel locations of a human, we can estimate or measure the temperature and emissivity of the human and use Eq. 1 to estimate the resulting radiance at the sensor. We then normalize by the maximum radiance in the scene to create an 8-bit thermal image.

4.2 Database

For our simulation, we estimate the temperatures and emissivities of objects that may be found in the African savanna, although these could be measured in the future if desired. In terms of the objects to model, we are interested in humans and animals, and the typical plants found in the savanna, which include acacia and baobab trees, shrubs, and grass, including elephant grass [4]. Thus our database focuses on bare soil, water, trucks, grass, shrubs, acacia trees, humans, elephants, rhinoceroses, hippopotamuses, crocodiles, and zebras, with possible future expansions of this database. All estimates of these objects' temperatures and emissivities can be found in Table 1.

The emissivities and sources can primarily be found in Table 1, though some cases require slightly more explanation. First, note that for animal and human emissivity, the emissivity refers to skin. Also note that the estimate for elephant skin emissivity comes from Asian elephants. We assume that rhinoceroses and hippopotamuses have the same emissivity of the elephant, as supported by the claim in [42] that 0.96 is the standard emissivity for biological tissue. For crocodile skin, we assume an emissivity similar to that of other reptiles, 0.96 [54]. Emissivity for a truck is based on the emissivity of oxidized steel [5], which is based on the rusty look of the trucks in simulation. Also, for acacia emissivity, the estimate came from a woodland savanna in Veracruz. Emissivities for specific species of grass and other plants can be found in this study [40].

Although we will assume these emissivities apply in all seasons, the temperature estimates will depend on the air temperature. We first consider winter conditions for these estimates because of the cool, dry assumptions we made of the atmosphere already. However, we also have real data in other seasons, so we will make temperature estimates for summer conditions as well, bearing in mind that assuming the atmospheric effects are negligible in the summer will be less accurate than the winter. Overnight in winter, it can be near or below freezing, often with frost in southern Africa, where Air Shepherd typically flies [2, 41]. Therefore, we will assume the air temperature is approximately 273 K in winter at night. In summer at night, we assume air temperature is 293 K [2].

Object	Winter Temp. (K)	Summer Temp. (K)	Avg. ϵ
Soil	278	288	0.914 [56]
Grass	273	293	0.958 [56]
Shrub	273	293	0.986 [56]
Acacia Tree	273	293	0.952 [8]
Human	292	301	0.985 [5]
Elephant	290	298	0.96 [47]
Zebra	298	307	0.98 [37]
Rhinoceros	291	299	0.96
Hippopotamus	290	298	0.96
Crocodile	295	303	0.96
Water	273	293	0.96 [5]
Truck	273	293	0.80

Table 1: Approximate temperatures and emissivities over night.

Internal temperatures in the cases of humans and animals are relatively simple to consider, but we need external temperatures, with clothes in the case of humans. There is extensive work in modeling human temperature for thermal comfort [15, 20, 33]. We use a study that measured human surface temperature with clothes [35], find a linear fit for the increasing temperature data found for "Mean skin and surface temperatures in the stable condition", and use the linear fit to estimate the surface temperature of a human in an environment at 273 K. For elephants, we refer to two studies in which external temperatures were measured [10, 60]. We estimate external elephant temperature to be 290 K at an external temperature of 273 K, again by using a linear fit to the measurements from the two studies. Based on [10], the temperature of the rhinoceros and hippopotamus are approximately the same as that of the elephant, though the rhinoceros is 1 K warmer. We therefore assume that these will have the same temperature at 273 K as the elephant (plus 1 K for the rhinoceros). The external temperature of the zebra [25] was measured at an air temperature of 296 K to be 308 K. We use the same linear fit as elephants, adjusted to intersect this data point, to estimate the zebra's external temperature to be 298 K at 273 K. Crocodiles are cold-blooded and therefore bask in the sun in the winter for survival. [17] has approximately measured their external temperature in the winter to be 295 K because of this. Finally, the temperature for the truck was assumed to be equal to air temperature since it is metal that is not in sunlight, which we also assume has been off for several hours. This would likely lead to thermal equilibrium since metals have a lower thermal conductivity [45]. This assumption was validated using a non-contact infrared thermometer after a car sat in darkness for several hours.

For plants and soil, there is also quite a bit of modeling that could be done to estimate their exact temperatures. We will instead find our estimated temperatures based on some generalizations. Also, we will assume that the temperatures of only the top-most objects will be observed by the sensor, so, for example, we will not consider soil under plants. Plant leaves are typically cooler than the air due to evaporation, especially at night when the sun is not present to warm them [34]. Small leaves typically have smaller differences in temperature with the air than large leaves because they have a thinner boundary layer with the air [61]. Acacia leaves are typically very small [1], on average 0.175 square cm in southern Africa [61]. Spiny shrub leaves and grass leaves are also typically smaller. Therefore, these should approximately track the air temperature. This is supported by [21].

The plains in parts of southern Africa are made up of red loamy mokata soils, which can become dry during the winter dry season [41]. The soil is likely at the permanent wilting point (PWP) in terms of water content in the winter [58]. Loam at PWP has the following characteristics: density of 1.52 t/m^3, specific heat of 1.72 MJ/m^3/K, thermal conductivity of 0.65 W/m/K, thermal diffusivity of 0.38 10$^-$6m^2/s, and thermal admittance of 1057 J/K/m^2/s^0.5 [58]. Saturated loam, which might be the case in the summer, has a specific heat of 3.06 MJ/m^3/K. Because air has a specific heat closer to 0.0012 MJ/m^3/K, soil temperatures change less quickly than air temperatures [3]. Depending on the time of night, there will be different air temperatures, and consequently different soil temperatures. At 1am in Morris, Minnesota on Oct. 30 at a depth of 1 cm, for example, both the soil and air temperatures are about 279 K. However, at 5am, the soil temperature is about 277 K, and the air temperature is about 273 K [38]. In addition, [66] estimated soil temperature based on air temperature. At 273 K in Arizona, the soil temperature is approximately 278 K. Given these examples, we will assume the temperature of the soil overnight is approximately 5 K warmer than the air temperature.

Now we have established that the leaves we care about have approximately the same temperature as the air, and that soil has a slightly more moderate temperature than the air. The air temperature should be tracked by grass, shrubs, acacia trees, and soil.

To create a summer temperature database, we utilize the same techniques and linear fits to estimate the temperatures of humans, elephants, rhinoceroses, hippopotamuses, and zebra. We now assume that soil will be 5 K cooler than air temperature, once again due to Arizona estimates and knowing that soil temperature will be more moderate in temperature changes than air. Otherwise, we assume grass, shrubs, trees, truck, and water, will track air temperature. We assume the crocodile has a temperature of 303 K in summer based on [50].

4.3 Blur and Noise

To this points, we have not considered blur or noise. The point spread function (PSF) is a measure of blur, as it describes the response of an imaging system to a perfect point of radiance. At best, the imaging system will be diffraction-limited, which will lead to some blur around the point of radiance. However, other factors, such as imperfections in the lens or atmospheric effects, can also contribute to the PSF and lead to blur in the image [49]. After light passes through the environment and the lens, it interacts with the detector to create an image. Noise is present in all detectors. Microbolometers are the detectors that are commonly used in uncooled thermal infrared cameras. When a thermal infrared photon strikes the detector, the temperature rises, and the resistance of the detector changes [7]. According to [32], the three main sources of noise in microbolometers are Johnson noise, flicker noise, and thermal noise. The Johnson noise is due to the resistor nature of the microbolometer. The flicker noise is due to flaws in the material surface in semiconductors [48]. The thermal noise is due to the heat exchange with the environment, which is important with uncooled microbolometers, though can be mitigated by changing the gain. [7] mentions that there is also fixed pattern noise (FPN) due to the fact that each microbolometer has a slightly different resistance for the same incoming thermal infrared photons. Although there are in fact other noise sources, such as periodic noise, which can be present in these videos, we focus on Johnson noise, flicker noise, thermal noise, and FPN. Other noise sources could be incorporated in the future.

In order to model these phenomena, again in a simplistic manner, we first utilize a Gaussian distribution for the PSF. This could be replaced with a real model of the PSF for the cameras being used in the field based on images they capture. However, the Gaussian blur kernel used here to loosely approximate a PSF has a standard deviation of 1, which was chosen visually.

Thermal noise and Johnson noise are both characterized by white Gaussian noise ([23, 32]). We utilize Gaussian 1/f noise to model the flicker noise [59]. Both are modeled based on [31, 64]. Finally, the FPN is modeled as uniform random noise [7]. The same noise distributions were used for all frames of the same video, with the FPN scaled by the first image's standard deviation. All are added to the normalized image, which is then scaled and clipped, to produce the final image.

4.4 Process

In order to convert from RGB to infrared, therefore, we now have the following: a segmentation map from the RGB simulation that specifies the objects in each image captured, a thermal infrared digital count associated with all of the objects in the simulation, and a simple model for blur and noise. We therefore assign the thermal infrared digital count to the corresponding object in the segmentation map to get a thermal infrared image. Finally, we add the blur and noise. Fig. 3 shows two examples, one each for winter and summer temperatures, where we see the segmentation map, the corresponding thermal infrared image, and the image with blur and noise.

5 UTILIZING AIRSIM-W FOR POACHER AND ANIMAL DETECTION

5.1 Generating Training Data with AirSim-W

In order to generate simulated thermal infrared imagery for use in deep learning algorithms, we follow the workflow depicted in Fig. 4. We utilize the Python API and add the option to fly in a zig-zag pattern, or to return a position for a specific object of interest at each time step. This could then be used to follow the specific object of interest, such as a poacher, to ensure the object is in the frame at all times. Furthermore, we adjust flight altitude and look angle using Computer Vision Mode, and we adjust the season to determine which digital counts should be used. Once these parameters are set, we fly, either in the zig-zag pattern or following an object of interest, and capture the segmentation image in each time step. Finally, we convert this image into the thermal infrared image for the time step.

For evaluation purposes, this process was used to generate data from 12 flights, 6 summer and 6 winter. Together, this yielded 84,073 individual frames containing objects of interest. Each of the 12 flights consisted of 30 minutes of flying time, totaling 6 hours.

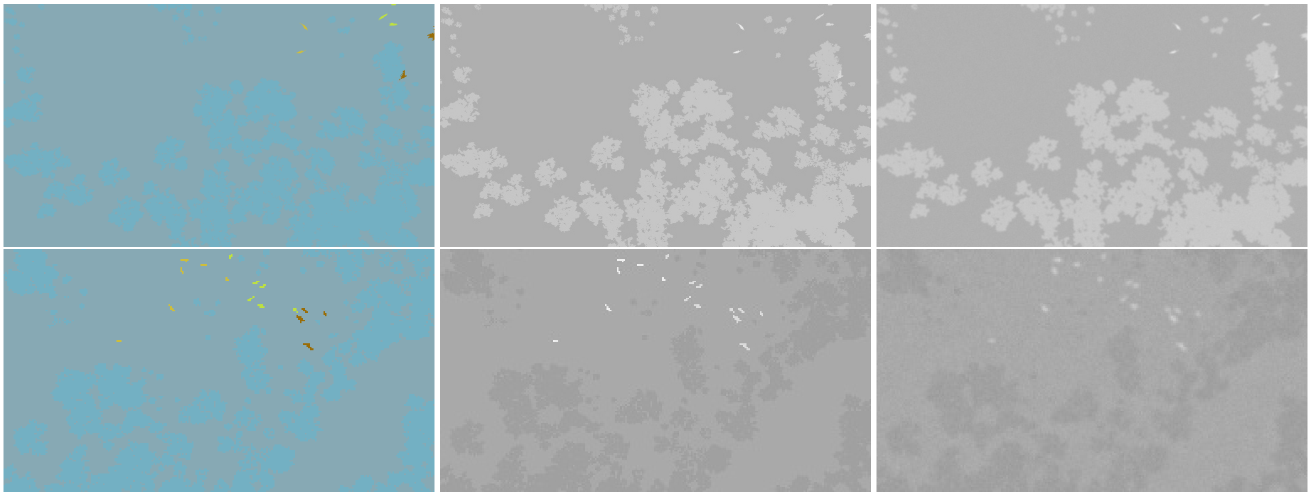


Figure 3: Segmentation, thermal infrared image without noise, and final thermal infrared image. Top: summer, bottom: winter. Both rows contain animals.

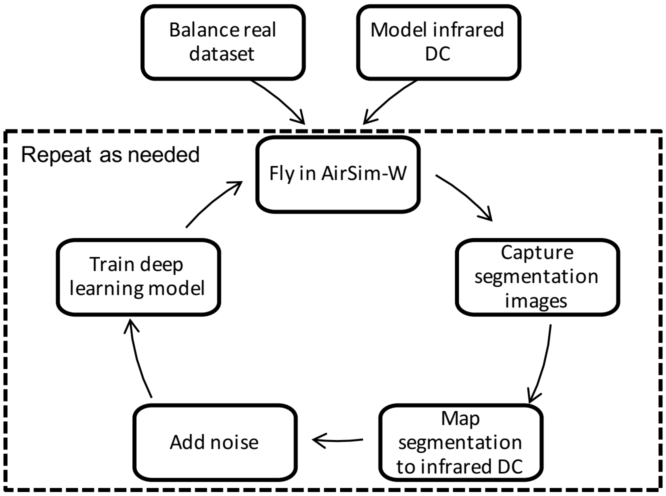

Balance real dataset
Model infrared DC
Repeat as needed
Fly in AirSim-W
Capture segmentation images
Map segmentation to infrared DC
Add noise
Train deep learning model

Figure 4: Workflow for generating deep learning training data with AirSim-W, particularly for generating data for poacher and animal detection in thermal infrared data.

5.2 Balancing Data

In the case of real thermal infrared videos of animals and poachers captured aboard UAVs, we have six classes: small animal, medium animal, large animal, small poacher, medium poacher, and large poacher, where the threshold for "small" is an average bounding box area of about 200 pixels and less throughout the video, and "large" is on average greater than 2000 pixels. These are balanced with negative samples automatically in Faster RCNN. However, because we have full videos which must stay consistently in either the test or train sets, and individual frames that may contain multiple objects of interest from different classes, we cannot simply randomly sample full videos for use in training, as was done in [12]. If we do, we may not actually balance all six classes in terms of individual samples. For example, if we have 100 frames with two small poachers and one small animal, we must take into account that there will be 200 small poachers introduced by including all 100 frames, while there will only be 100 small animals introduced. Furthermore, we would like to be able to detect at all three sizes.

Therefore, we sample the training set through the use of a mixed-integer linear program (MILP). The motivation of using a MILP is that we would like to use as many different train videos as possible, so the balanced dataset will not just sample all consecutive frames from just one or two videos. In other words, by utilizing more unique videos, we provide our algorithm with samples with more variety. We also define "frame types", each can be represented by a 6-dimensional vector, indicating the number of objects from each class. For example, one frame type could be $(1, 0, 0, 1, 0, 0)$ meaning frames of this type have one small animal and one small poacher. A video can have frames of various frame types. For simplicity in the paper, we denote the frame types as type 1, 2, 3, etc. Therefore, our objective is to use frames from as many different videos as possible, while maintaining balance between the total number of labels in different classes, and bearing in mind that we have many different frame types in videos.

We now formally define this as an MILP. i is the index of the video, j is the index of the frame type, and k is the index of the label type (i.e., which class the object of interest belongs to). Then, c_{ij}^k is the number of type k labels in the type j frame in video i (e.g., if type 2 frame is "empty" frame in video 1, then $c_{12}^k = 0$ for any k, if type 4 frame is "single small poacher only" frame in video 1, then $c_{14}^1 = 1$ and $c_{14}^k = 0, \forall k \neq 1$). N_{ij} is the number of type j frames in video i. L_l^k and L_u^k are the lower bound and upper bound, respectively, of the desired total number of type k labels. These bounds implement the balance requirement on the total number of labels in different classes.

x_{ij} is a variable representing the number of type j frames in video i that are sampled or selected. u^k and v^k are variables referring to the maximum and minimum number of type k labels that are selected from a single video among all videos except the videos that have no type k label at all and the videos whose type k labels are all selected (i.e., $\sum_j c_{ij}^k x_{ij} = \sum_j c_{ij}^k N_{ij}$). Finally, w_i^k is binary indicator indicating if all type k labels in video i are selected.

$$\min \sum_k (u_k - v_k) \tag{2}$$

$$u^k \geq \sum_j c_{ij}^k x_{ij}, \forall i, k \tag{3}$$

$$v^k \leq \sum_j c_{ij}^k x_{ij} + M w_i^k, \forall i, k \tag{4}$$

$$w_i^k \in \{0, 1\}, \forall i, k \tag{5}$$

$$M(1 - w_i^k) \geq \sum_j c_{ij}^k N_{ij} - \sum_j c_{ij}^k x_{ij}, \forall i, k \tag{6}$$

$$x_{ij} \leq N_{ij}, \forall i, j \tag{7}$$

$$L_l^k \leq \sum_i \sum_j c_{ij}^k x_{ij} \leq L_u^k, \forall k \tag{8}$$

$$x_{ij} \in \mathbb{N}, \forall i, j \tag{9}$$

(2) is the objective function, which minimizes the difference in the number of labels of each type selected from videos. The objective function implicitly encourages a balanced number of labels being selected from different videos for each label type. In the ideal case, this objective function takes value 0, when each label type, an equal number of labels is selected from each video. (3) and (4) define the variables in the objective function. In (4), we will multiply by M, a large positive number, if all k labels in the video i are selected, as defined in (5) and (6). (7) simply ensures that the number of sampled frames is less than or equal to the number of frames in the video i. (8) ensures that we are within the desired number of samples based on our frame choices, and finally (9) ensures the number of frames sampled is integer.

The introduction of w_i^k is to make sure that when we compute v_k, we exclude the videos whose type k labels are already fully selected. Consider the case where there is a video A that only has 3 large animal labels in total among all frames. Then, we want to include all these labels in our selection, and at the same time, we also want to balance the number of labels of large animals in other videos which have a large number of large animal labels. So, we need to exclude video A when computing v_k.

The current MILP enforces this requirement. Given the objective function in (2), the optimizer will try to make v_k as large as possible. Since (4) is the main restriction for v_k, the optimizer will try to set w_i^k to be 1 whenever possible. (6) ensures that w_i^k can take the value of 1 only if all type k labels in video i are already selected. Note that setting $w_i^k = 0$ is still feasible when the condition is satisfied, but setting $w_i^k = 1$ can achieve at least the same and sometimes better objective value.

Given the optimal solution to this MILP, we randomly select the specified number of frames from each frame type in each video to achieve a balanced dataset that uses as many different videos as possible.

6 EVALUATION

6.1 Qualitative Tests

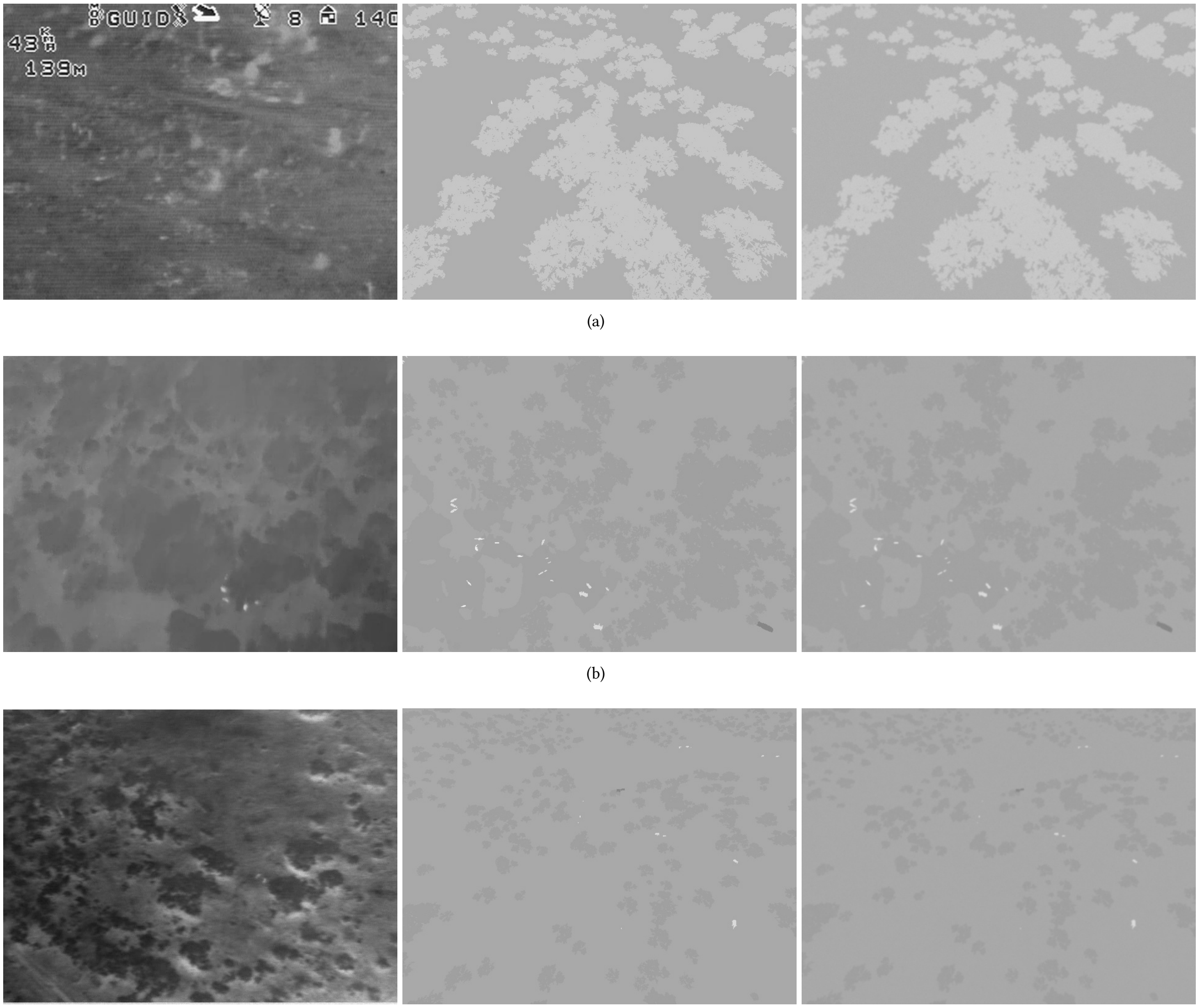
First, we examine the simulated images qualitatively. In Fig. 5, we observe three pairs of real and simulated frames side-by-side. Although noise has been modeled simply, meaning some periodic noise and gain fluctuations are not present, they otherwise look very similar when it comes to relationships between the objects of interest, such as trees and soil. For example, in Fig. 5(b), the trees are darker than the surrounding ground, as in the simulation. The same is true for Fig. 5(a). In Fig. 5(c), humans are approximately the same size in both.

6.2 Quantitative Tests

Qualitatively, the simulated images look similar to the real images (other than noise). We now evaluate the simulated data quantitatively by utilizing it in one wildlife conservation task of interest, detecting poachers and animals in thermal infrared images. First, we examine the effects of balancing the real dataset based on the MILP we presented earlier. For the new balanced dataset, we have 651 total labeled frames to train one model, out of the original 39,380 frames. We do not choose different models (e.g., different training data) for animals and poachers based on performance as we did previously with SPOT. To conduct this first test of balancing data only, we initialize using pre-trained ImageNet weights for Faster RCNN, and fine-tune using the 651 balanced frames. These results are found in the column labeled "None, Regular" in Tables 2 and 3, as there was no simulated data used in this initial test of balancing data only.

Next, we examine the effects of adding simulated data. We test two types of simulated data: regular simulated data, without blur or noise added, and noisy simulated data, including the blur and noise discussed in Section 4.3. To run these tests, we initialize using pre-trained ImageNet weights for Faster RCNN as before. We then fine-tune using the simulated data, and finally fine-tune using real data. For real data, we conduct two tests: (i) fine-tune using the balanced real dataset, and (ii) fine-tune using the SPOT datasets. SPOT is our previous system based solely on real data. Again, the first test (i) is fine-tuning with balanced data after first fine-tuning with simulated data, and the second test (ii) is fine-tuning with the SPOT unbalanced data after first fine-tuning with simulated data. Each fine-tuning process takes 4 hours on an NVIDIA Titan X (Pascal).

The results for fine-tuning using the simulated data only, without noise, can be found in the column labeled "Regular, None", the results for (i) are labeled "Regular, Balanced" and "Noisy, Balanced" based on the type of simulated data, and the results for (ii) are labeled "Regular, SPOT". We also include the previous results from SPOT in the first column. Again, SPOT uses different models for poacher and animal videos, so results for SA, MA, LA for "None, SPOT" and "Regular, SPOT" are fine-tuned using the SPOT animal model, and the results for MP, LP are fine-tuned using the SPOT poacher model. Note that the simulated data is primarily balanced by construction because poachers and animals are co-located in the simulation environment, and because we believe that balancing the data becomes less important when there is a large amount of it. Also note that in the simulated dataset, we flew only at 200 and 400



(a)

(b)

(c)

Figure 5: Qualitative comparison of real frames (left), basic thermal infrared simulated frames (middle), and noisy simulated frames (right). 5(a): summer, 5(b): winter, 5(c): winter. The images in the first and third rows contain poachers, and the images in the second row contain animals. The simulated images in the third row also contain animals.

ft, which means there were no large animals or large poachers, and most objects of interest were actually around the small-medium data threshold of 200 pixels in area.

The test set contains six historical videos containing animals or poachers of different sizes. These are the same test videos used to evaluate SPOT in [12]. The combined results for all tests can be found in Table 2 and Table 3. SA, MA, LA, MP, and LP represent the objects of interest in that particular test video. They are small animals, medium animals, large animals, medium poachers, and large poachers, respectively. Small poachers were excluded because, as with SPOT, none in this particular test video were identified correctly. This is because the poacher bounding boxes are less than 20 pixels in area.

6.3 Discussion

There are several interesting results. First, for recall, using simulated data produces best results for 4 test videos and on average. It is especially interesting that using only simulated data without any real data produces the best recall results for SA. Using simulated data plus SPOT produces the best precision results on average, though SPOT without simulated data does produce the best precision results overall for videos SA, MA, and LA. We believe this

Table 2: Precision results.

Simulation	None	None	Regular	Regular	Regular	Noisy
Video/Real	SPOT	Balanced	None	SPOT	Balanced	Balanced
SA	**0.5729**	0.5232	0.0166	0.4044	0.3536	0.4286
MA	**0.5544**	0.5510	0.0041	0.5066	0.5228	0.5498
LA	**0.5584**	0.3873	0.0318	0.5407	0.4404	0.4592
MP	0.0995	0.1660	0	0.1136	0.1864	**0.2633**
LP	0.3977	0.3571	0.0074	**0.7799**	0.2286	0.0294
Avg.	0.4366	0.3969	0.0120	**0.4690**	0.3464	0.3461

Table 3: Recall results.

Simulation	None	None	Regular	Regular	Regular	Noisy
Video/Real	SPOT	Balanced	None	SPOT	Balanced	Balanced
SA	0.0025	0.0026	**0.0044**	0.0027	0.0020	0.0014
MA	0.0131	0.0278	0.0117	**0.0355**	0.0272	0.0254
LA	0.2293	0.2939	0.1297	0.2825	**0.3149**	0.2971
MP	0.0073	**0.1304**	0	0.0111	0.1168	0.0953
LP	0.0188	0.0054	0.0038	**0.0374**	0.0014	0.0004
Avg.	0.0542	0.0920	0.0299	0.0738	**0.0925**	0.0839

could be attributed to several reasons: (i) we selected the model from SPOT that performed best with animals and kept this separate from the poacher model, (ii) using more real data is better than using more simulated data in general, as the SPOT animal model used about 18,480 real animal frames, or (iii) we lacked large animal examples in simulation. More simulated data could be generated in the future to test this.

The addition of noise only improves over the other datasets in the case of precision for MP. This is interesting, as it implies that perfect images for initial training may actually be beneficial, or that a more sophisticated noise model such as a GAN is necessary. We can further examine this in future work.

It is also interesting to note that balanced data alone performs comparably to SPOT, which used 22,663 frames, while balanced data used only 651 frames. This implies that having a dataset with variety might mean that less data is needed. For example, if we must label real data, we may consider labeling only a few frames per video in the future as opposed to labeling full videos in [13]. We may also consider different distinctions than small, medium, and large, or assign these distinctions per frame instead of on average to further improve balancing. In addition, using simulated data only for fine-tuning while testing on real data does provide nonzero results on most videos, sometimes comparable with real data only. This implies that should labeling a large dataset be too costly, generating large amounts of simulated data may be sufficient to achieve results on real data, and will reduce significant labeling burden. Either of these techniques, or both combined, could allow for less costly, better data collection in the future. Future work could determine the optimal amount of simulated and real data.

7 CONCLUSION

In conclusion, we present AirSim-W, a new simulation environment and data augmentation technique built specifically for wildlife conservation. AirSim-W includes the (i) creation of an African savanna environment in Unreal Engine, (ii) thermal infrared modeling, (iii) new methods to fly the UAVs throughout the scene for training data collection, and (iv) demonstrated detection improvement using simulated data generated by AirSim-W. Labeling real data costs over $8000, while the creation of the simulated environment, which can generate unlimited amounts of data, costs closer to $5000. The cost of the simulated data could be lowered in the future when expanding to other animals and environments by developing transfer learning techniques, possibly by using the existing Africa environment (https://github.com/Microsoft/AirSim/releases), and/or by finding existing environments and animals. Also, labeling real data took approximately 800 hours total, whereas creating the environment and generating simulated data took approximately 200 hours. With these contributions, AirSim-W will be a cost efficient, useful tool for wildlife conservation research, especially for the problems of poacher and animal detection in UAV videos, patrol planning for UAVs and foot patrols, and camera trap placement.

8 ACKNOWLEDGMENTS

This work was supported by Microsoft AI for Earth and NSF (CCF-1522054), with partial support from Viterbi School of Engineering Ph.D. Merit Top-Off Fellowship.

REFERENCES

[1] [n. d.]. Acacia. https://www.britannica.com/plant/acacia. Accessed: 2017-10-20.
[2] [n. d.]. Botswana - Weather and Climate. https://www.safaribookings.com/botswana/climate. Accessed: 2017-10-20.
[3] [n. d.]. Ground Temperatures as a Function of Location, Season, and Depth. http://www.builditsolar.com/Projects/Cooling/EarthTemperatures.htm. Accessed: 2017-10-20.

[4] [n. d.]. Savanna Flora. https://www.britannica.com/science/savanna/Flora. Accessed: 2017-10-20.

[5] [n. d.]. Table of Emissivity of Various Surfaces. http://www-eng.lbl.gov/~dw/projects/DW4229_LHC_detector_analysis/calculations/emissivity2.pdf. Accessed: 2017-10-17.

[6] [n. d.]. Typical spectral response curves. http://flir.custhelp.com/app/answers/detail/a_id/932/~/typical-spectral-response-curves. Accessed: 2017-10-20.

[7] Musaed Alhussein and Syed Irtaza Haider. 2016. Simulation and Analysis of Uncooled Microbolometer for Serial Readout Architecture. *Journal of Sensors* 2016 (2016).

[8] GK Arp and DE Phinney. 1979. The ecological variations in thermal infrared emissivity of vegetation.[in Texas, Arizona, New Mexico, and Mexico]. (1979).

[9] Pavlo Bazilinskyy, Niels Heisterkamp, Philine Luik, Stijn Klevering, Assia Haddou, Michiel Zult, George Dialynas, Dimitra Dodou, and Joost de Winter. 2018. EYE MOVEMENTS WHILE CYCLING IN GTA V. (2018).

[10] Francis G Benedict, Edward L Fox, and Marion L Baker. 1921. The surface temperature of the elephant, rhinoceros and hippopotamus. *American Journal of Physiology–Legacy Content* 56, 3 (1921), 464–474.

[11] Alexander Berk, Lawrence S Bernstein, and David C Robertson. 1987. *MODTRAN: A moderate resolution model for LOWTRAN.* Technical Report. SPECTRAL SCIENCES INC BURLINGTON MA.

[12] Elizabeth Bondi, Fei Fang, Mark Hamilton, Debarun Kar, Donnabell Dmello, Jongmoo Choi, Robert Hannaford, Arvind Iyer, Lucas Joppa, Milind Tambe, and Ram Nevatia. 2018. SPOT Poachers in Action: Augmenting Conservation Drones with Automatic Detection in Near Real Time. (2018).

[13] Elizabeth Bondi, Fei Fang, Debarun Kar, Venil Noronha, Donnabell Dmello, Milind Tambe, Arvind Iyer, and Robert Hannaford. 2017. VIOLA: Video Labeling Application for Security Domains. In *GameSec.*

[14] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. 7.

[15] Joon-Ho Choi and Dongwoo Yeom. 2017. Study of data-driven thermal sensation prediction model as a function of local body skin temperatures in a built environment. *Building and Environment* 121 (2017), 130–147.

[16] R Critchlow, AJ Plumptre, M Driciru, A Rwetsiba, EJ Stokes, C Tumwesigye, F Wanyama, and CM Beale. 2015. Spatiotemporal trends of illegal activities from ranger-collected data in a Ugandan national park. *Conservation biology* 29, 5 (2015), 1458–1470.

[17] Colleen T Downs, Cathy Greaver, and Ricky Taylor. 2008. Body temperature and basking behaviour of Nile crocodiles (Crocodylus niloticus) during winter. *Journal of Thermal Biology* 33, 3 (2008), 185–192.

[18] R Dulski, H Madura, T Piatkowski, and T Sosnowski. 2007. Analysis of a thermal scene using computer simulations. *Infrared Physics & Technology* 49, 3 (2007), 257–260.

[19] Fei Fang, Thanh Hong Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. 2016. Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security.. In *AAAI.* 3966–3973.

[20] Dusan Fiala, Kevin J Lomas, and Martin Stohrer. 2001. Computer prediction of human thermoregulatory and temperature responses to a wide range of environmental conditions. *International Journal of Biometeorology* 45, 3 (2001), 143–159.

[21] Ali Asghar Ghaemi, Mohammad Rafie Rafiee, and Ali Reza Sepaskhah. 2009. Tree-temperature monitoring for frost protection of orchards in semi-arid regions using sprinkler irrigation. *Agricultural Sciences in China* 8, 1 (2009), 98–107.

[22] Shahrzad Gholami, Benjamin Ford, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, and Joshua Mabonga. 2017. Taking it for a Test Drive: A Hybrid Spatio-temporal Model for Wildlife Poaching Prediction Evaluated through a Controlled Field Test. In *ECML PKDD.*

[23] Daniel T Gillespie. 1996. The mathematics of Brownian motion and Johnson noise. *American Journal of Physics* 64, 3 (1996), 225–240.

[24] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623* (2015).

[25] J Hattingh. 1972. A comparative study of transepidermal water loss through the skin of various animals. *Comparative Biochemistry and Physiology Part A: Physiology* 43, 4 (1972), 715–718.

[26] Emmett J Ientilucci and Scott D Brown. 2003. Advances in wide-area hyperspectral image simulation. In *Targets and Backgrounds IX: Characterization and Representation*, Vol. 5075. International Society for Optics and Photonics, 110–122.

[27] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. 2016. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110* (2016).

[28] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. 2017. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on.* IEEE, 746–753.

[29] Nathan Koenig and Andrew Howard. 2004. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.* Sendai, Japan, 2149–2154.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[31] Jack J Lennon. 2000. Red-shifts and red herrings in geographical ecology. *Ecography* 23, 1 (2000), 101–113.

[32] William Alexander Lentz. 1998. *Characterization of noise in uncooled IR bolometer arrays.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[33] Baizhan Li, Yu Yang, Runming Yao, Hong Liu, and Yongqiang Li. 2017. A simplified thermoregulation model of the human body in warm conditions. *Applied ergonomics* 59 (2017), 387–400.

[34] E. Linacre. [n. d.]. Leaf and air temperatures. http://www-das.uwyo.edu/~geerts/cwx/notes/chap03/leaves.html. Accessed: 2017-10-20.

[35] Yanfeng Liu, Lijuan Wang, Jiaping Liu, and Yuhui Di. 2013. A study of human skin and surface temperatures in stable and unstable thermal environments. *Journal of Thermal Biology* 38, 7 (2013), 440–448.

[36] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep Photo Style Transfer. *arXiv preprint arXiv:1703.07511* (2017).

[37] Dominic J Mccafferty. 2007. The value of infrared thermography for research on mammals: previous applications and future directions. *Mammal Review* 37, 3 (2007), 207–223.

[38] Gordon McIntosh and Brenton S Sharratt. 2001. Thermal properties of soil. *The Physics Teacher* 39, 8 (2001), 458–460.

[39] Miguel A Olivares-Mendez, Changhong Fu, Philippe Ludivig, Tegawendé F Bissyandé, Somasundar Kannan, Maciej Zurad, Arun Annaiyan, Holger Voos, and Pascual Campoy. 2015. Towards an autonomous vision-based unmanned aerial system against wildlife poachers. *Sensors* 15, 12 (2015), 31362–31391.

[40] MR Pandya, DB Shah, HJ Trivedi, MM Lunagaria, V Pandey, S Panigrahy, and JS Parihar. 2013. Field measurements of plant emissivity spectra: an experimental study on remote sensing of vegetation in the thermal infrared region. *Journal of the Indian Society of Remote Sensing* 41, 4 (2013), 787–796.

[41] Neil Parsons. [n. d.]. Botswana. https://www.britannica.com/place/Botswana. Accessed: 2017-10-20.

[42] Polly K Phillips and James Edward Heath. 1992. Heat exchange by the pinna of the African elephant (Loxodonta africana). *Comparative Biochemistry and Physiology Part A: Physiology* 101, 4 (1992), 693–699.

[43] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS.* 91–99.

[45] Ian Ridpath. 2012. *A dictionary of astronomy.* Oxford University Press.

[46] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3234–3243.

[47] M. F. Rowe, G. S. Bakken, J. J. Ratliff, and V. A. Langman. 2013. Heat storage in Asian elephants during submaximal exercise: behavioral regulation of thermoregulatory constraints on activity in endothermic gigantotherms. *Journal of Experimental Biology* 216, 10 (2013), 1774–1785. https://doi.org/10.1242/jeb.076521 arXiv:http://jeb.biologists.org/content/216/10/1774.full.pdf

[48] Raghvendra Sahai Saxena, Arun Panwar, SK Semwal, PS Rana, Sudha Gupta, and RK Bhan. 2012. PSPICE circuit simulation of microbolometer infrared detectors with noise sources. *Infrared Physics & Technology* 55, 6 (2012), 527–532.

[49] John R Schott. 2007. *Remote sensing: the image chain approach.* Oxford University Press on Demand.

[50] Frank Seebacher, Gordon C Grigg, and LA Beard. 1999. Crocodiles as dinosaurs: behavioural thermoregulation in very large ectotherms leads to high and stable body temperatures. *Journal of Experimental Biology* 202, 1 (1999), 77–86.

[51] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2017. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics.* arXiv:arXiv:1705.05065 https://arxiv.org/abs/1705.05065

[52] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. 2017. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 3. 6.

[53] Gregory J Stein and Nicholas Roy. 2017. GeneSIS-RT: Generating Synthetic Images for training Secondary Real-world Tasks. *arXiv preprint arXiv:1710.04280* (2017).

[54] C RICHARD Tracy. 1982. Biophysical modeling in reptilian physiology and ecology. *Biology of the Reptilia* 12 (1982), 275–321.

[55] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. 2017. A Bayesian Data Augmentation Approach for Learning Deep Models. In *Advances in Neural Information Processing Systems.* 2794–2803.

[56] AA Van de Griend, M Owe, M Groen, and MP Stoll. 1991. Measurement and spatial variation of thermal infrared surface emissivity in a savanna environment. *Water resources research* 27, 3 (1991), 371–379.

[57] Jan C van Gemert, Camiel R Verschoor, Pascal Mettes, Kitso Epema, Lian Pin Koh, Serge Wich, et al. 2014. Nature Conservation Drones for Automatic Localization and Counting of Animals.. In *ECCV Workshops (1)*. 255–270.

[58] Francisco J. Villalobos, Luca Testi, Luciano Mateos, and Elias Fereres. 2016. *Soil Temperature and Soil Heat Flux*. Springer International Publishing, Cham, 69–77. https://doi.org/10.1007/978-3-319-46116-8_6

[59] Richard F Voss. 1978. Linearity of 1 f Noise Mechanisms. *Physical Review Letters* 40, 14 (1978), 913.

[60] TERRIE M WILLIAMS. 1990. Heat transfer in elephants: thermal partitioning based on skin temperature profiles. *Journal of Zoology* 222, 2 (1990), 235–245.

[61] Ian J. Wright, Ning Dong, Vincent Maire, I. Colin Prentice, Mark Westoby, Sandra Díaz, Rachael V. Gallagher, Bonnie F. Jacobs, Robert Kooyman, Elizabeth A. Law, Michelle R. Leishman, Ülo Niinemets, Peter B. Reich, Lawren Sack, Rafael Villar, Han Wang, and Peter Wilf. 2017. Global climatic drivers of leaf size. *Science* 357, 6354 (2017), 917–921. https://doi.org/10.1126/science.aal4760 arXiv:http://science.sciencemag.org/content/357/6354/917.full.pdf

[62] Haifeng Xu, Benjamin Ford, Fei Fang, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, and Mustapha Nsubaga. 2017. Optimal Patrol Planning for Green Security Games with Black-Box Attackers. In *GameSec*.

[63] Haifeng Xu, Kai Wang, Phebe Vayanos, and Milind Tambe. 2018. Strategic Coordination of Human Patrollers and Mobile Sensors with Signaling for Security Games. (2018).

[64] Jon Yearsley. 2016. Generate spatial data. Accessed: 2018-03-01.

[65] Yang Zhang, Philip David, and Boqing Gong. 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, Vol. 2. 6.

[66] Daolan Zheng, E Raymond Hunt Jr, and Steven W Running. 1993. A daily soil temperature model based on air temperature and precipitation for continental applications. *Climate Research* (1993), 183–191.

[67] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593* (2017).