BIRDSAI: A Dataset for Detection and Tracking in Aerial Thermal Infrared Videos

Elizabeth Bondi[1], Raghav Jain[2], Palash Aggrawal[2], Saket Anand[2], Robert Hannaford[3]
Ashish Kapoor[4], Jim Piavis[4], Shital Shah[4], Lucas Joppa[4], Bistra Dilkina[5], Milind Tambe[1]
[1]Harvard University, [2]IIIT-Delhi, [3]Air Shepherd, [4]Microsoft, [5]University of Southern California
ebondi@g.harvard.edu, sites.google.com/view/elizabethbondi/dataset

Abstract

Monitoring of protected areas to curb illegal activities like poaching and animal trafficking is a monumental task. To augment existing manual patrolling efforts, unmanned aerial surveillance using visible and thermal infrared (TIR) cameras is increasingly being adopted. Automated data acquisition has become easier with advances in unmanned aerial vehicles (UAVs) and sensors like TIR cameras, which allow surveillance at night when poaching typically occurs. However, it is still a challenge to accurately and quickly process large amounts of the resulting TIR data. In this paper, we present the first large dataset collected using a TIR camera mounted on a fixed-wing UAV in multiple African protected areas. This dataset includes TIR videos of humans and animals with several challenging scenarios like scale variations, background clutter due to thermal reflections, large camera rotations, and motion blur. Additionally, we provide another dataset with videos synthetically generated with the publicly available Microsoft AirSim simulation platform using a 3D model of an African savanna and a TIR camera model. Through our benchmarking experiments on state-of-the-art detectors, we demonstrate that leveraging the synthetic data in a domain adaptive setting can significantly improve detection performance. We also evaluate various recent approaches for single and multi-object tracking. With the increasing popularity of aerial imagery for monitoring and surveillance purposes, we anticipate this unique dataset to be used to develop and evaluate techniques for object detection, tracking, and domain adaptation for aerial, TIR videos.

1. Introduction

Recent advances in deep learning have led to immense progress in vision applications like object recognition, detection, and tracking. One of the key factors driving this progress is the availability of large-scale datasets capturing real-world conditions along with careful annotations for training and comprehensively evaluating machine learning models. The collection and release of many of these datasets is often inspired by specific applications of interest, e.g., perception for autonomous driving using object detection, tracking, and semantic segmentation, person re-identification for surveillance camera networks, and facial recognition for biometrics and security applications. While the majority of the publicly available datasets cater to techniques developed for the visible spectrum [17, 19, 36, 27, 25, 13, 22, 56, 18], there has been an increasing interest in applications from the near-infrared (NIR) and thermal infrared (TIR) spectral ranges [55, 3, 33, 23, 31], as these sensors become more affordable.

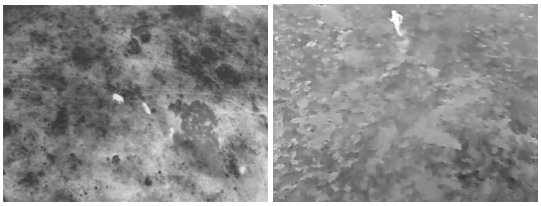


Figure 1: Example images from BIRDSAI: elephants and a human, respectively, from an aerial perspective.

Concurrently, with advances in aerial image acquisition technology, datasets specifically targeting object detection and tracking in aerial images have been made publicly available [32, 56, 18]. In [56], the images have been acquired from various remote sensing sources (e.g., satellites), and capture varying degrees of orientation, scales, and object density. On the other hand, aerial images from UAVs [32, 18] are often motivated by applications like surveillance and monitoring, yet these images are restricted to the visible spectrum, thereby limiting their usage to well-lit conditions. Besides, most existing public datasets, aerial and terrestrial alike, address applications relevant to relatively densely populated settings.

Contrarily, our work is motivated by recent concerns about depleting biodiversity and loss of forest cover which

Dataset (Year)	Platform (A/G)	#Frames	Tasks	Spectrum	(R)eal/(S)ynthetic
UTB [32] (2017)	A	15K	S	V	R
UAV123 [38] (2016)	A	113K	S	V	R,S
UAVDT [18] (2018)	A	80K	D,S,M	V	R
TIV [55] (2014)	G,A[a]	64K	D,S,M	T	R
LTIR [3] (2015)	G,A	12K	D,S,M	T	R
PTB-TIR [33] (2018)	G,A	30K	D,S,M	T	R
ASL-TID [41] (2014)	A[a]	5K	D,S,M	T	R
[57] (2015)	G,A[b]	84[c]	RE	T,V	R
[37][d] (2016)	A	9K	D,S,M	T	R
BIRDSAI (proposed)	A	62K + 100K	D,S,M	T	R,S

Table 1: Comparison summary of recent aerial video datasets for detection and tracking. Platform could be either (A)erial or (G)round-based; #Frames is the total number of annotated frames in the dataset, with our dataset reporting 62K (1K=1000) real frames and about 100K synthetic frames; Tasks for which annotations are present (D)etection, (S)ingle-object, (M)ulti-object tracking, and (RE)gistration; Spectrum of cameras: (V)isible or (T)hermal-IR; Data acquisition (R)eal or (S)ynthesized in a simulator. Comparisons are discussed in Sec. 2. [a]Fixed aerial perspective; [b] Aerial images do not contain humans or animals; [c]84 Pairs; [d] Not publicly available, contains primarily images of roads, and has portions of images used for tracking.

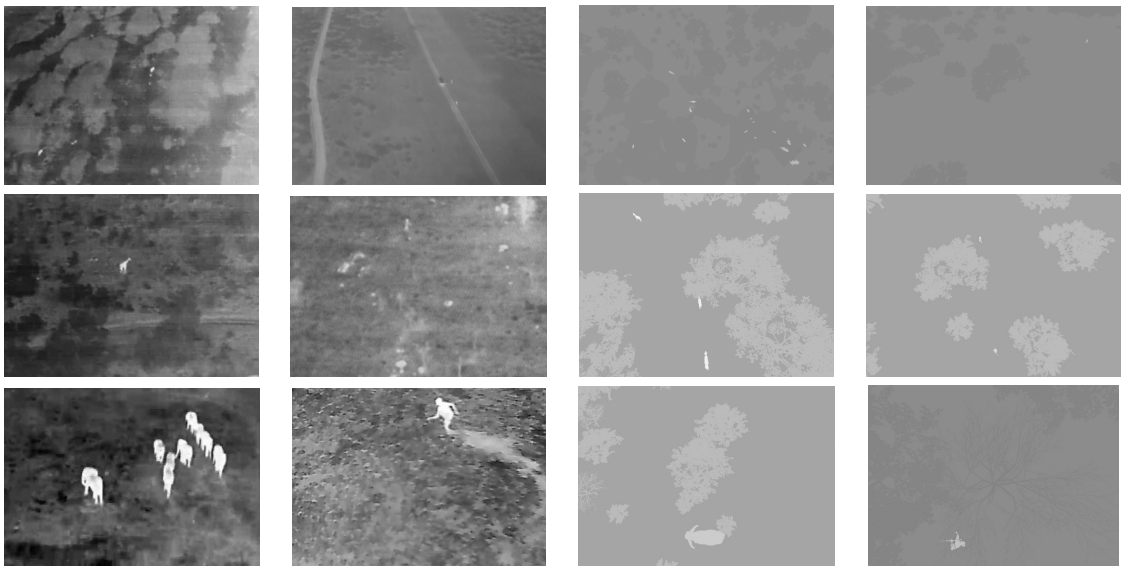
are exacerbated by illegal activities such as poaching for wildlife trade, hunting, and logging. Efforts to mitigate these activities through patrolling of protected areas, especially at night, is very challenging and puts forest rangers at risk due to poor visibility, difficult terrain, and increased predator and poacher activity [40]. These conservation efforts are increasingly being augmented by UAV surveillance [40, 26, 24, 1], with TIR cameras as the preferred sensing modality for night-time monitoring over natural landscapes where the ambient light is minimal and the UAV's altitude, capacity, and need for stealth preclude the use of active light sensors. However, manual monitoring of aerial TIR videos to detect and track humans in real time is an extremely challenging and tedious task, especially when the goal is to interdict an illegal activity.

In this paper, we introduce Benchmarking IR Dataset for Surveillance with Aerial Intelligence (BIRDSAI, pronounced "bird's-eye"), a large, challenging aerial TIR video dataset for benchmarking of algorithms for automatic detection and tracking of humans and animals. To our knowledge, this is the first large-scale aerial TIR dataset, with multiple unique features. It has 48 real aerial TIR videos of varying lengths, carefully annotated with objects like animals and humans and their trajectories. These were collected by a conservation organization, Air Shepherd, during their regular surveillance efforts flying a fixed-wing UAV over national parks in Southern Africa. Finally, we augment it with 124 synthetic aerial TIR videos generated from AirSim-W [6], an Unreal Engine-based simulation platform. Two example images from real videos are shown in Fig. 1 depicting a herd of elephants and a human. Realistic and challenging benchmarking datasets have had tremendous impact on the progress of a research area. Synthetic datasets like [44, 46, 21] along with real ones like [14, 22] have accelerated the progress in unsupervised domain adaptation techniques [29, 49]. Similarly, the Caltech-UCSD Bird (CUB-200) dataset [52, 51] has helped advance an important area of fine-grained visual recognition [60]. With more wildlife monitoring datasets [48, 2, 20, 53] becoming publicly available, we may expect rapid progress in areas like species detection, counting, and visual animal biometrics [15, 20, 9, 28]. Inspired by these instances, we anticipate the proposed dataset will promote advances in both (i) algorithm development for the general problems of object detection, single and multi-object tracking in aerial videos, and their domain adaptive counterparts, and (ii) the important application area of aerial surveillance for conservation.

The rest of the paper is organized as follows. First, we introduce the gap in existing datasets that we aim to fill with the proposed one (Sec. 2). We then discuss the attributes of the dataset in detail (Sec. 3), such as the means of acquiring the data, strategies adopted for annotation, and the train/test splits. We next analyze the content of the resulting dataset (Sec. 4), and evaluate the performance of well-known techniques for the tasks of object detection, single and multi-object tracking, and domain adaptation (Sec. 5) before finally concluding the paper (Sec. 6).

2. Motivation

With poaching becoming widespread around the world [50], aerial surveillance with UAVs is becoming a mainstream application [40, 26, 24]. In order to apply deep learning-based detection and tracking techniques to these applications (especially at night) and evaluate performance, there is a need for a realistic, large, annotated dataset that adequately captures the challenges faced in the field. Recently, several large datasets for aerial image analytics have



(a) (b) (c) (d)

Figure 2: Sample images from the real and synthetic datasets. From top to bottom: small, medium, and large objects. (a) & (b) Real images of animals and humans, respectively; (c) & (d) Synthetic images of animals and humans, respectively. Mixture of summer and winter synthetic data (winter has dark trees compared to ground).

been publicly released, many of which were captured using UAVs. However, all of these are data in the visible spectrum. In the rest of this section, we discuss some of the most closely related public datasets and highlight the unique aspects of the presented dataset. A summary of comparisons with existing datasets is provided in Table 1.

Existing UAV Datasets: The recently introduced UAVDT [18] contains nearly 80,000 frames with over 0.8 million bounding boxes. The dataset is comprised of videos collected over urban areas with object categories of cars, trucks and buses. The DTB dataset [32] was introduced for benchmarking UAV-based single object tracking with the goal of jointly evaluating the motion model and tracking performance. Mueller et al. introduced the UAV123 dataset [38], which contains 123 HD video sequences with about 113,000 annotated frames captured by a low-altitude UAV. Eight of these videos were rendered using an Unreal Engine environment. All of these datasets use visible spectrum cameras mounted on multirotor UAVs, which typically have lower speeds and better image stabilization as compared to fixed-wing UAVs [8]. In a poaching prevention application, deploying a multirotor UAV for surveillance is more difficult due to stealth and coverage requirements.

Existing TIR Datasets: The BU-TIV dataset [55] is part of the OTCBVS dataset collection[1] and contains 16 video sequences with over 60,000 annotated frames for tasks like detection, counting and tracking. The LTIR [3] dataset was used for the VOT-TIR 2016 challenge and contains 20 video sequences of length 563 frames on average. The PDT-ATV dataset [41] was introduced for benchmarking tracking of pedestrians in aerial TIR videos. All eight sequences are captured using a handheld TIR camera at a height and angle to simulate a UAV, but because it is handheld, it is a fixed aerial perspective. Recently, the PTB-TIR dataset [33] was also introduced for benchmarking TIR pedestrian tracking. It is comprised of 60 sequences with over 30,000 annotated frames. In all cases, the challenge of analyzing TIR footage *from a UAV* has not been addressed yet.

Synthetic Datasets and Domain Adaptation: Training deep models demands large datasets with accurate annotations, which are expensive and tedious to obtain. Consequently, recent years have seen an increasing use of synthetic images rendered using state-of-the-art graphics engines, where generating accurate ground-truth information is trivial. Several such simulators and datasets have been made public [21, 44, 46, 39], and reportedly improve real-world performance of deep learning models when pre-trained with synthetic data. The increased use of synthetic datasets in vision applications has further propelled research in domain adaptation with works like [10, 29, 49] being a few among the many[2] leveraging synthetic data. Some recent work has also shown that domain transformation and adaptation techniques can aid in improved detection performance in TIR images when using deep CNNs

[1] http://vcipl-okstate.org/pbvs/bench/

[2] More comprehensive list of advances in domain adaptation: https://github.com/zhaoxin94/awsome-domain-adaptation

pretrained on visible spectrum images [11, 4]. In contrast, BIRDSAI uses both real and synthetic aerial, TIR videos.

BIRDSAI: The 48 real TIR video sequences included in BIRDSAI were randomly selected from a database of UAV videos collected by Air Shepherd for conservation, and contain 1300 frames on average. These videos accurately reflect the challenges in the field, e.g., motion blur, large camera motions (both rotations and translations), compression artifacts due to bandwidth constraints, background clutter, and high altitude flight (60-120m) resulting in smaller objects to detect and track. The 124 synthetic videos with 800 frames on average, on the other hand, were generated using the AirSim-W [6] platform with the publicly available models of the African savanna, animal species, and UAV-mounted cameras. This dataset uniquely brings together the three categories discussed above.

3. Dataset Description

3.1. Real Data

3.1.1 Data Acquisition

Data were collected throughout protected areas in the countries of South Africa, Malawi, and Zimbabwe using a battery-powered fixed-wing UAV. Specific locations are withheld for security. All flights took place at night, with individual flights lasting for about 1.5 - 2 hours. Various environmental factors such as wind resistance determined exact flying time. Throughout the night, there were typically 3 to 4 flights, the altitude ranged from approximately 60 to 120m, and flight speed ranged from 12 to 16 m/s depending on conditions such as wind. Temperature ranged from less than 0°C to 4°C in winter at night, though typically closer to 4°C. There was often a shift of approximately 5°C throughout the course of the night in the winter. For reference, daytime temperatures were typically approximately 15°C to 16°C. During summer, the temperature ranged from 18°C to 20°C at night, and 38°C to 40°C during the day. When flying just after sunset, the ground temperature was warm and could make it more difficult to spot objects of interest due to the lack of contrast. However, by about 10:30-11PM, there was typically sufficient contrast for easier visibility. Fog was present in some rare cases, which could cause "whiteouts" in images.

The FLIR Vue Pro 640 was the primary sensor utilized. However, the Tamarisk 640 was also used in some videos in the dataset. Although the typical resolution of images is 640x480 as a result, some images may be sized differently due to the removal of text embedded in to the videos describing specific locations and other flight parameters, which are also withheld for security purposes. These cameras produce 8-bit images and use Automatic Gain Control (AGC), as in [12]. This leads to more reliable contrast that facilitates better detection and tracking accuracy during flight. The cameras cost approximately $2000-$4000 depending on the lenses and other attributes. They have 19mm focal length and collect imagery at a rate of 30Hz. Images were streamed to a base station during flight, where they were stored as raw videos. All videos were converted to mp4 videos for processing and JPEG images. Because the videos were recorded from real-world missions, they lack some metadata, such as speed, altitude, and temperature. While this auxiliary information could be useful, automatic vision algorithms should still be designed to work in their absence. From a usability perspective, this added robustness is crucial for building practical vision systems that are less sensitive to specific UAV or camera settings.

3.1.2 Annotation

We used VIOLA [7] to label detection bounding boxes in the thermal infrared imagery, and followed the process described in VIOLA. To briefly summarize this labeling process in VIOLA, after labels were made by one person, two other people reviewed the labels, making corrections as needed. General rules that were followed during the labeling process are as follows. If individuals were completely indistinguishable (e.g., multiple humans or animals were close together and could not be distinguished at all in thermal imagery), they were not labeled. Instead, occlusions are recorded when possible to determine manually from context. This includes cases where animals or humans become indistinguishable for a few frames and again become distinguishable after they or the camera move. If there were artifacts in the image (see Sec. 4), objects were tagged as containing noise. Some extremely small amounts of these artifacts may have been allowed without being tagged as noisy. We provide examples of how we included occlusion and noise in the Appendix. Finally, if an object was mostly out of the camera's field of view (i.e., more than about 50% of the object was not present in the frame), it was not labeled. After this process, all labels were finally confirmed and checked for quality for use in this dataset by the authors, one of whom is from Air Shepherd and collected the videos, for a total of 4 checks on each initial label.

We additionally labeled individual species when distinguishable, typically in videos with larger animals present. The real videos contain giraffes, lions, elephants, and a dog, which account for about 100K of the 120K individual animal bounding boxes (the remaining 20K animals are marked as unknown species). There are about 34K human bounding boxes. These labels created using VIOLA were then labeled separately for tracking. We built a tool using Tkinter[3] to assign object IDs to each bounding box label. To reduce annotation effort before any human annotation was done, the tool checked for overlap between frames us-

[3] https://docs.python.org/3/library/tkinter.html

ing an Intersection over Union (IoU) threshold. If the IoU exceeded the threshold, the object in the following frame was given the same object ID. Once this automatic processing was complete, we used the tool to manually navigate through the video frames and identify and correct any errors in the assigned object IDs, e.g., objects merging or splitting. In the case of objects merging together, object IDs are maintained whenever it is possible to distinguish them again after the merge. However, if they enter a large group, it may become impossible to distinguish which animal is which due to the nature of thermal imagery. In these cases only, they are assigned a new object ID. If objects leave the frame, they will similarly retain the same object ID if possible.

3.2. Synthetic Data

To generate synthetic data with AirSim-W, we utilized the African savanna environment introduced in [6]. In brief, the environment is not based on a particular area of interest, but rather represents the variety of environments found in Southern Africa, such as wide-open plains to dense forest, flatland to mountainous terrain, roads, and water. Grass in the plains is not a mesh in the environment, so in the segmentation provided by AirSim-W, grass and soil are indistinguishable. This does, however, increase efficiency while running the simulation. The AirSim-W platform has a TIR model that was introduced in [6]. We used this TIR model to generate images of the objects in the scene as the UAV flew through the environment and captured images of size 640x480. Specifically, AirSim-W's Computer Vision Mode was used, and the UAV was placed by following certain objects in the environment. For example, to generate human images, the UAV tracked the human. Because the objects move in groups, and multiple altitudes, offsets, and camera angles were used, multiple objects or few objects may have been captured. Ground truth object IDs and species (lions, elephants, crocodiles, hippos, zebras, and rhinos) labels were also recorded for a total of about 220K individual animal bounding box labels and 50K human labels.

3.3. Train and Test Sets

In order to create the train and test sets for the real data, our goal was to create similar distributions in both while ensuring complete videos stayed entirely in either the train or test set. Entire videos remained in one or the other because consecutive frames could be extremely similar. We manually assigned videos to the train or test set based on the number of objects in the video, and based on characteristics of the videos, like contrast, to try to ensure an approximately even distribution in the train and test sets. Because entire videos needed to stay together, it was not possible to maintain exact ratios. In fact, there was only one video that contained large humans, so it was placed in the test set only. These train and test sets are shown in Fig. 3.

Regarding the synthetic dataset, the entirety of the dataset was used for training. Although we attempted to ensure the approximate ratio of humans and animals was somewhat similar to the real training dataset, we prioritized adding large human examples and more small human and animal examples (see Section 4.1 for more description of scale) while generating the synthetic dataset, as these were less frequently seen in the real data. Different statistics over the entire dataset, including distribution of object scales and densities across the train/test splits, are shown in Fig. 3 (a) and (b), respectively. In Fig. 3 (c), a scatter plot of tracking video sequences is shown with respect to the sequence length and average object density.

4. Dataset Properties

The real and synthetic data contain significant variations in content and artifacts, including scale and contrast. The real data also contain more background clutter and noise.

4.1. Content

Environments. There are several types of environments that are captured in the dataset, including land areas with varying levels of vegetation and water bodies, such as watering holes and rivers. An example of water with a boat floating upon it is shown in Fig. 4 (b) (where the bright, top right portion of the image is water). We denote the presence of water for individual videos in the dataset.

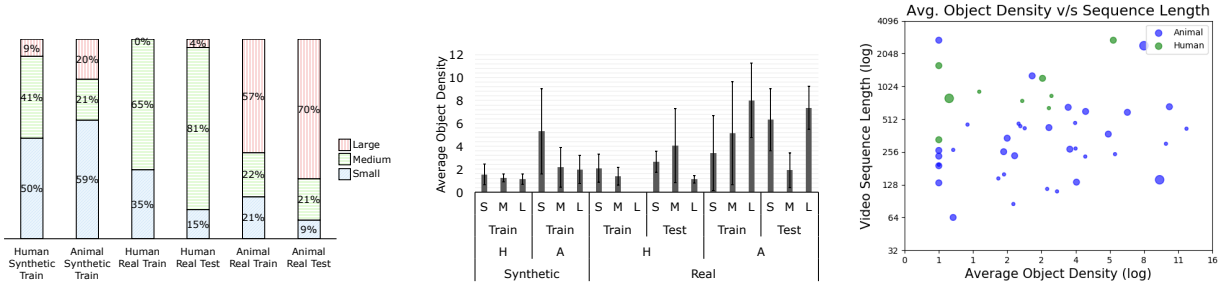
Scale and Density of Objects. There are multiple scales of objects in the dataset. We coarsely categorize them into small, medium, and large based on each object's annotated bounding box area and dataset statistics. These distinctions are assigned to full videos based on the average bounding box size throughout the video[4]. There is also a wide range of densities in objects throughout the videos. The average number of objects per frame (density) for small, medium, and large videos is described in Fig. 3. There is an example of a video with high animal density in Fig. 4 (a).

4.2. Artifacts

Contrast. Contrast refers to the variation in digital counts in an image. TIR images rely on AGC, so contrast can vary significantly across the dataset. As an example, some images have nearly black backgrounds with white objects of interest (more contrast, e.g., Fig. 4 (b)), while others have gray backgrounds (less contrast).

Background Clutter. There can be many objects in the background in some images, particularly in images with vegetation. Vegetation can often have a similar temperature to objects of interest, leading to images like Fig. 4 (c). We also see thermal reflections off the ground, typically near

[4] Small videos were those whose average bounding box area was < 200 pixels, the median real area, and large videos were > 2000 pixels.



(a) (b) (c)

Figure 3: Statistics of real and synthetic data. (a) 100% stacked bar charts of distribution of small, medium, and large animals/humans across real and synthetic data and train/test sets. Real train contains 32 videos, real test contains 16 videos, and simulated train contains 124 videos. (b) Bar plot (with standard deviation error bars) of the number of animals and humans for train/test sets over large, medium, and small objects, again across real and synthetic data and train/test sets. (c) Scatter plot showing different video sequences plotted using their constituent average object density (#objects/frame) and sequence length (duration for which the objects were visible in the video). The color indicates the constituent object type (human/animal) and the size of the circles indicate small, medium, or large. For better visual clarity, both the axes are plotted using the log scale.

Scale	FR-CE	FR-WCE	YOLOv2	SSD
SA	0.216	0.228	0.144	0.182
MA	0.459	0.468	0.383	0.392
LA	0.879	0.896	0.679	0.850
Animals	**0.659**	**0.671**	**0.489**	**0.587**
SH	0.214	0.206	0.108	0.219
MH	0.174	0.179	0.146	0.229
LH	0.154	0.094	0.083	0.147
Humans	**0.181**	**0.155**	**0.104**	**0.183**
Overall	**0.430**	**0.438**	**0.304**	**0.388**

Table 2: Detection performance baseline using the mAP metric for different scales ((S)mall, (M)edium, (L)arge) of objects ((A)nimals, (H)umans) in the dataset.

trees, e.g., in Fig. 4 (d). Both make it challenging to distinguish between objects of interest and background clutter.

Noise and Camera Motion. While there are many sources of noise in TIR cameras that use uncooled microbolometer arrays as the sensor [6, 47], the most common type in BIRDSAI is what we call ghosting, as shown in Fig. 4 (e). There are also slightly more mild versions of it, which look like horizontal "bands" in some cases. Additionally, the UAV's motion, or even the camera motion when there is pan or tilt, can sometimes lead to frames with motion blur. An example of this is shown in Fig. 4 (f). These were labeled as containing noise when possible (see Sec. 3).

5. Evaluation

The goal of BIRDSAI is to advance image-based object detection, domain adaptive detection, and single and multi-object tracking (SOT and MOT, respectively). To evaluate state-of-the-art object detection methods and domain adaptation on BIRDSAI, we perform *framewise* detection of animals and humans. We evaluate tracking by using the videos, both full sequences and subsequences. We provide benchmarking results for these tasks with existing algorithms, leaving the method details to the papers while listing the hyperparameters used for the experiments here. We include further experiments and analyses in the Appendix including cross-dataset evaluation.

5.1. Framewise Detection

We specifically test with the following popular object detection methods: Faster-RCNN [43], YOLOv2 [42], SSD [35], and Domain Adaptive Faster-RCNN [10], all of which have shown strong results in the visible as well as TIR. Results for detection and unsupervised domain adaptative detection are provided in Tables 2 and 3, respectively.

Faster-RCNN [58] [43]. The experiment was performed using VGG16 as the backbone network initialized with ImageNet pretrained weights. Evaluation results of two loss functions were compared and tabulated in Table 2, namely, Cross Entropy (CE) and Weighted Cross Entropy (WCE), to account for the imbalance in the two classes (i.e., humans are more rare). The weights for each of the classes are computed as follows for the WCE loss.

$$W_\ell = \left(\frac{\sum_{i=1}^{k_a} w_i h_i + \sum_{i=1}^{k_h} w_i h_i}{\sum_{i=1}^{k_\ell} w_i h_i} \right)^{0.5} \quad (1)$$

where k_a and k_h are the number of animals and humans in the frame, respectively, $w_i h_i$ is the area of the corresponding bounding box, and $\ell \in \{a, h\}$ (animal or human).

These experiments were performed with a batch size of

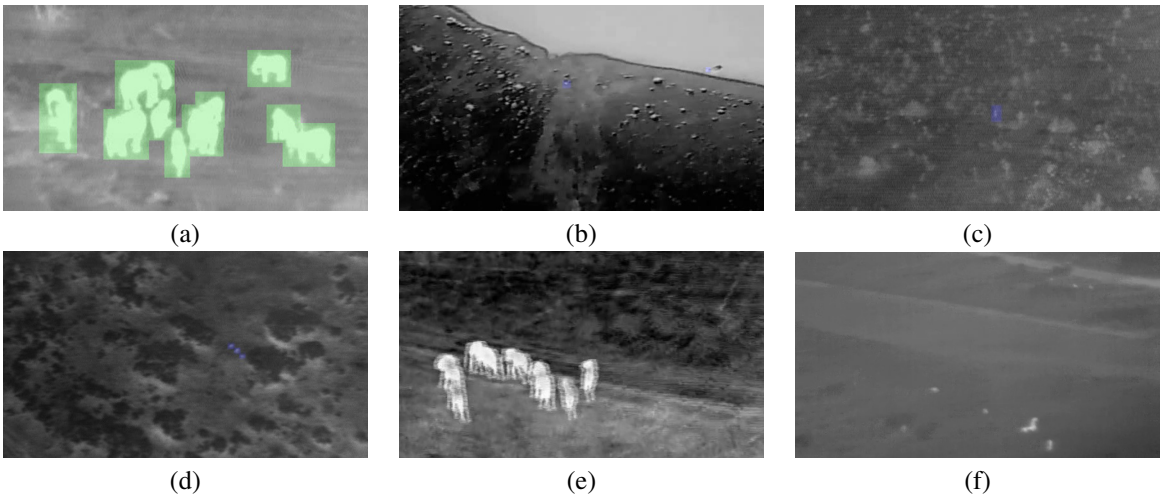


Figure 4: Data challenges. (a) density (b) high contrast (c) clutter (vegetation) (d) clutter (reflections) (e) ghosting (f) motion blur. Ground truth labels not shown in (e) and (f) for better visualization of effects of noise. Animals in (a), (e), (f), humans in (b), (c), (d).

Configuration	DA-FR-CE	DA-FR-WCE	FR-CE	FR-WCE	YOLOv2	SSD
Real → Real	–	–	0.430	**0.438**	0.304	0.388
Syn → Real	0.443	**0.459**	0.309	0.313	0.152	0.294

Table 3: Detection performance baselines using the mAP metric after domain adaptation.

1, an SGD optimizer, and a starting learning rate of 1e-2. The learning rate was depreciated after a learning step which was set to two epochs for the loss to converge. The overall fine-tuning was performed for a total of 7 epochs.

YOLOv2 [42]. This model used pretrained Darknet19 weights. A batch size of 1 was taken with a starting learning rate set to 1e-3, depreciating it by a factor of 10 after every second epoch. The model converges after 12 epochs.

SSD [35]. This model used pretrained VGG16 weights. The hyperparameters of the training include a batch size of 8, initial learning rate of 1e-5, without depreciating the learning rate throughout. The training converges after a total of 12 epochs. An SGD optimizer was used with a weight decay of 1e-4 and an update gamma of 0.1.

Domain Adaptive Faster RCNN [10]. This framework was trained with the base architecture as VGG16, pretrained with ImageNet. The corresponding overall mAPs are tabulated in Table 3. Real → Real indicates that the train set is comprised only of labeled real data and the model was tested on real data, which is equivalent to results in Table 2. Synthetic → Real implies that the train set is comprised of labeled synthetic data and unlabeled real data, and the testing was performed on the test set (real data).

Results: It is not surprising to note that the best overall performance is achieved using Faster-RCNN with WCE, where the weights explicitly account for the data imbalance. For human objects alone, however, SSD and Faster-RCNN (without weighting) perform comparably, while outperforming the other methods. YOLOv2 performs worst overall, possibly due to the small size of objects. In all cases, there is room for improvement, especially for small animals and humans.

The overall results of Table 2 are equivalent to the Real → Real row of Table 3. In the Synthetic → Real row, simply training on synthetic data and testing on real data actually decreased performance for those algorithms lacking domain adaptation. This is not surprising either, given that there is a visible domain shift between the synthetic and real data subsets of BIRDSAI. It is, however, encouraging to see that in the unsupervised domain adaptation setting of [10], there is a noticeable increase in the mAP values. This observation suggests that further research in unsupervised domain adaptation could immensely benefit object detection in aerial TIR videos, given the extremely challenging task of annotating aerial TIR videos.

5.2. Tracking

We test both single and multi-object tracking on BIRDSAI, and we report results for all objects regardless of class. In both the tracking settings, we use the same train/test splits as used in object detection. For single-object tracking, video sequences were further split into *perfect subsequences* such that each subsequence had a single target object throughout, with a minimum length of 50 frames. Once

Method	Perfect Subsequences		Full Sequence	
	Precision	**AUC**	**Precision**	**AUC**
ECO	**0.8103**	**0.5430**	**0.4842**	**0.2972**
AD-Net	0.8029	0.5331	0.4545	0.2546
MCFTS	0.7194	0.4946	0.3401	0.1886
Siamese RPN	0.0073	0.0093	0.0041	0.0048

Table 4: Single Object Tracking Evaluation. Precision is at 20 pixels. "Perfect subsequences" excludes noisy/occluded frames, while "Full sequence" includes them.

there was any interruption in the subsequence, whether due to noise, occlusion, or the object exiting the frame, the subsequence ended. This resulted in a total of 552 subsequences. The train/test splits of SOT subsequences were consistent with that of the videos, i.e., all subsequences from test videos were included in the test set, and similarly for the training set. This means that all subsequences from a given video appeared either in the training set or in the test set, which yielded a train set with 386 and a test set with 166 subsequences. For testing of *full sequences*, we used the test videos to generate 99 sequences of length at least 50 frames, with each sequence starting at the first appearance of an object in the video and ending at its last appearance.

For single-object tracking, we use the Siamese RPN [30], ECO [16] and AD-Net [59] algorithms as benchmarks, and we also use the MCFTS [34] algorithm, which was developed specifically for the related VOT-TIR dataset. These algorithms were then evaluated on the test set using the usual metrics of success rate and precision [54, 32]. We evaluated pretrained models of ECO and MCFTS, and retrained Siamese RPN and AD-Net on BIRDSAI. We followed the commonly used one-pass evaluation (OPE) process for single-object tracking [54], which required training of models like Siamese RPN and AD-Net to be done on the perfect subsequences, where every frame had ground truth annotations. During testing, we performed the benchmarking on the perfect subsequences and full sequences. As is typical in OPE, all of the trackers were initialized using ground truth bounding boxes in the respective first frames.

For multi-object tracking we only report the IoU Tracker [5] with default thresholds, and object detections provided using (i) ground truth bounding boxes and (ii) Faster-RCNN detection. We use Faster-RCNN for MOT benchmarking due to its superior detection results. We also include other MOT results in the Appendix. The algorithms are evaluated using the MOTA and MOTP evaluation metrics [45], where higher is better. MOTA and MOTP are in the range of [-∞, 100 (%)], and [0, 100(%)] respectively. Although they are percentages above 0, negative values for MOTA imply that the errors (false positives, misses, and mismatches) are more than the ground truth objects to be tracked.

Results: See Table 4 for SOT and Table 5 for MOT benchmarking. For SOT, Siamese RPN, which relies on one-shot detection, fails to perform reasonably. Performance is promising with the other methods. This seems to be related to the length and cleanliness of the track, as evidenced by the improved performance in subsequences compared to full sequences. However, the real world will require handling videos with imperfect tracks and noise, small objects, and detection initialization, which leaves room for innovation. For MOT, IoU Tracker performs very well for ground truth bounding boxes, while it performs worse when using Faster-RCNN detections in both of the MOTA and MOTP metrics.

Method	Obj Size	Ground Truth Det		F-RCNN Det	
		MOTA	**MOTP**	**MOTA**	**MOTP**
IoU Tracker	S	61.6	**100.0**	-102.4	62.7
	M	**91.3**	98.9	-34.4	66.9
	L	80.6	**100.0**	**13.6**	**68.9**

Table 5: Multiple Object Tracking Evaluation. S, M, L is for small, medium, large.

6. Conclusion

We presented BIRDSAI, a challenging dataset containing aerial, TIR images of protected areas for object detection, domain adaptation, and tracking of humans and animals. In our benchmarking experiments, we noted that state-of-the-art object detectors work well for large animals, however, for humans and small and medium animals, the performance drops substantially. Similarly, while IoU Tracker-based multi-object tracking works well when ground truth detections are provided, the performance drops drastically when a detector's output is used. These experimental results indicate the challenging nature of the real sequences in the BIRDSAI dataset. Fortunately, we saw that baseline domain adaptive detection shows promising improvements by leveraging the synthetic dataset. This observation is crucial, as the annotation effort for noisy TIR videos is enormous, and improved unsupervised domain adaptation techniques can prove to be very useful for achieving competitive detection performance. We hope this dataset will help propel research in this important area. Finally, in addition to facilitating interesting research, this dataset will also contribute to wildlife conservation. Successful algorithms could be used to help prevent wildlife poaching in protected areas and count or track wildlife.

7. Acknowledgements

This was supported by Microsoft AI for Earth, NSF CCF-1522054 and IIS-1850477, MURI W911NF-17-1-0370, and the Infosys Center for Artificial Intelligence, IIIT-Delhi. We also thank the labeling team.

References

[1] Air Shepherd. Air shepherd: The lindbergh foundation. http://airshepherd.org, 2019. Accessed: 2019-11-02.

[2] S. Beery, G. Van Horn, and P. Perona. Recognition in terra incognita. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[3] A. Berg, J. Ahlberg, and M. Felsberg. A thermal object tracking benchmark. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2015.

[4] A. Berg, J. Ahlberg, and M. Felsberg. Generating visible spectrum images from thermal infrared. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[5] E. Bochinski, T. Senst, and T. Sikora. Extending iou based multi-object tracking by visual information. In *IEEE International Conference on Advanced Video and Signals-based Surveillance*, pages 441–446, Auckland, New Zealand, Nov. 2018.

[6] E. Bondi, D. Dey, A. Kapoor, J. Piavis, S. Shah, F. Fang, B. Dilkina, R. Hannaford, A. Iyer, L. Joppa, and M. Tambe. Airsim-w: A simulation environment for wildlife conservation with uavs. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, COMPASS '18, pages 40:1–40:12, 2018.

[7] E. Bondi, F. Fang, D. Kar, V. Noronha, D. Dmello, M. Tambe, A. Iyer, and R. Hannaford. Viola: Video labeling application for security domains. In *Proceedings of the 8th Annual Conference on Decision Theory and Game Theory for Security (GameSec)*, 2017.

[8] M. Boon, A. P. Drijfhout, and S. Tesfamichael. Comparison of a fixed-wing and multi-rotor uav for environmental mapping applications: A case study. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W6:47–54, 08 2017.

[9] G. S. Cheema and S. Anand. Automatic Detection and Recognition of Individuals in Patterned Species. In *ECML PKDD*, 2017.

[10] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] J. B. Christian Herrmann, Miriam Ruf. Cnn-based thermal infrared person detection by domain adaptation. volume 10643, 2018.

[12] P. Christiansen, K. Steen, R. Jørgensen, and H. Karstoft. Automated detection and recognition of wildlife using thermal cameras. *Sensors*, 14(8):13778–13793, 2014.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] J. Crall, C. Stewart, T. Berger-Wolf, D. Rubenstein, and S. Sundaresan. Hotspotter – patterned species instance recognition. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 230–237, 2013.

[16] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[18] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[20] A. Freytag, E. Rodner, M. Simon, A. Loos, H. S. Kühl, and J. Denzler. Chimpanzee faces in the wild: Log-euclidean cnns for predicting identities and attributes of primates. In *German Conference on Pattern Recognition*, pages 51–63. Springer, 2016.

[21] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[23] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1045, 2015.

[24] J. Kamminga, E. Ayele, N. Meratnia, and P. Havinga. Poaching detection technologiesa survey. *Sensors*, 18(5), 2018.

[25] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[26] B. Kellenberger, D. Marcos, and D. Tuia. Detecting mammals in uav images: Best practices to address a substantially imbalanced dataset with deep learning. *Remote Sensing of Environment*, 216:139 – 153, 2018.

[27] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2015 challenge results. In *ICCV Workshops*, pages 1–23, 2015.

[28] S. Kumar and S. K. Singh. Visual animal biometrics: survey. *IET Biometrics*, 6(3):139–156, 2017.

[29] K.-H. Lee, G. Ros, J. Li, and A. Gaidon. SPIGAN: Privileged adversarial learning from simulation. In *International Conference on Learning Representations*, 2019.

[30] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] C. Li, X. Liang, Y. Lu, N. Zhao, and J. Tang. RGB-T object tracking: Benchmark and baseline. *CoRR*, abs/1805.08982, 2018.

[32] S. Li and D.-Y. Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, 2017.

[33] Q. Liu and Z. He. PTB-TIR: A thermal infrared pedestrian tracking benchmark. *CoRR*, abs/1801.05944, 2018.

[34] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen. Deep convolutional neural networks for thermal infrared object tracking. *Knowledge-Based Systems*, 134:189 – 198, 2017.

[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[36] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco, et al. Ua-detrac 2017: Report of avss2017 & iwt4s challenge on advanced traffic monitoring. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–7. IEEE, 2017.

[37] Y. Ma, X. Wu, G. Yu, Y. Xu, and Y. Wang. Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery. *Sensors*, 16(4):446, 2016.

[38] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[39] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem. Sim4cv: A photo-realistic simulator for computer vision applications. *International Journal of Computer Vision*, 126(9):902–919, 2018.

[40] M. A. Olivares-Mendez, T. F. Bissyandé, K. Somasundar, J. Klein, H. Voos, and Y. Le Traon. The noah project: Giving a chance to threatened species in africa with uavs. In T. F. Bissyandé and G. van Stam, editors, *e-Infrastructure and e-Services for Developing Countries*, pages 198–208, 2014.

[41] J. Portmann, S. Lynen, M. Chli, and R. Siegwart. People detection and tracking from aerial thermal views. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1794–1800, 2014.

[42] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[43] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[44] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[45] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[46] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[47] J. R. Schott. *Remote sensing: the image chain approach*. Oxford University Press on Demand, 2007.

[48] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer. Data from: Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna, 2015.

[49] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[50] UNODC. World wildlife crime report: Trafficking in protected species, 2016.

[51] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[52] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[53] C. L. Witham. Automated face recognition of rhesus macaques. *Journal of neuroscience methods*, 2017.

[54] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015.

[55] Z. Wu, N. Fuller, D. Theriault, and M. Betke. A thermal infrared video benchmark for visual analysis. In *2014 IEEE CVPR Workshop on Perception Beyond the Visible Spectrum*, pages 201–208, 2014.

[56] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[57] S. Yahyanejad and B. Rinner. A fast and mobile system for registration of low-altitude visual and thermal aerial images using multiple small-scale uavs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:189–202, 2015.

[58] J. Yang, J. Lu, D. Batra, and D. Parikh. A faster pytorch implementation of faster r-cnn. *https://github.com/jwyang/faster-rcnn.pytorch*, 2017.

[59] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[60] B. Zhao, J. Feng, X. Wu, and S. Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.

A. Dataset Additional Details

A.1. Noise and Occlusion Annotations

We handled noise and occlusion labels through a mixture of manually identifying these situations and automatically processing existing labels. We automatically considered labels to be occluded/occluding when the IoU is greater than 0.3. We also automatically considered frames to be noisy if there were a few missing labels in an object track, and interpolated missing labels. We use interpolation because, particularly in the case of ghosting or motion blur, the true bounding box is difficult to pinpoint due to noise. We provide examples of noise and occlusion annotations from this process in Fig. 1 and Fig. 2, respectively. We used the red labels in each case to represent normal animal labels, while the blue labels (in the middle frames) represent the animals with noise or occlusion. The separate distinction allows these cases to be used or discarded as needed depending on the task, whether object detection, tracking, etc.

A.2. Simulated Data

We added a lion to the simulation. We used 38 °C (311 K) for temperature in summer, 39 °C (310 K) for temperature in winter [8], and 0.98 for emissivity [7]. Object IDs and species labels for all objects of interest in the simulation were collected by using individual segmentation IDs corresponding to the actor name for each object. Videos were generated by following objects of interest with various offsets (sometimes within videos to break the smooth motion), camera angles, seasonality, and altitudes. Finally, if there was a small object along the border of an image, it was removed if less than 100 pixels in area. There is no noise in these synthetic data.

B. Additional Experiments

B.1. Detections

Table 1 contains the results for two of the detection models on the proposed dataset, BIRDSAI, with ResNet [4] as the base model (instead of VGG16 results shown in Table 2 in the main paper), and the same experimental setup as described in Section 5.1 of the main paper. SSD and Faster-RCNN with ResNet perform better in some cases compared to VGG16, but overall, and especially for Faster-RCNN with weighted cross entropy, VGG16 outperforms ResNet.

Table 2 (extension of Table 3 in the main paper, including the same Syn $\rightarrow$ Real row for easy reference) tabulates the performance baselines for detection in the unsupervised, semi-supervised and supervised domain adaptation setting. We still

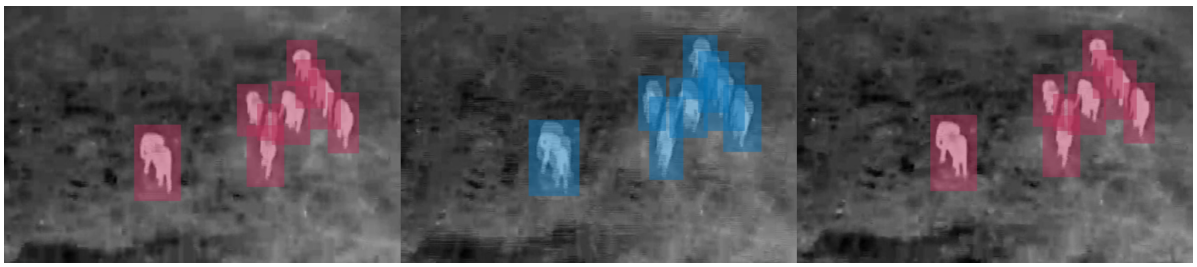


Figure 1: Consecutive frames from a video in the dataset showing noise. Blue colored labels are noisy labels, while red are normal animal labels.

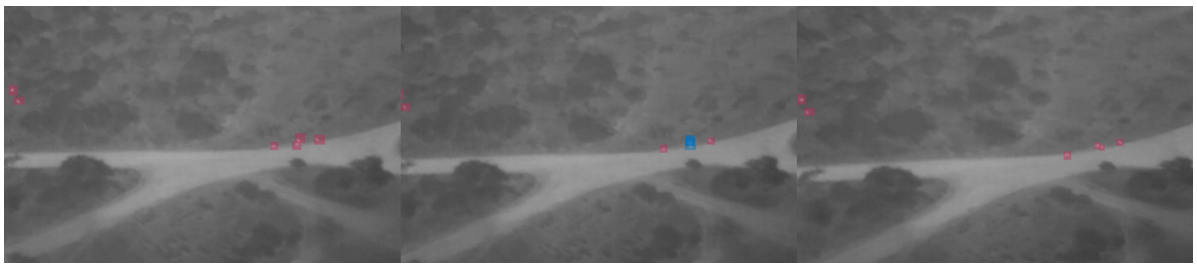


Figure 2: Near-consecutive frames from a video in the dataset showing occlusion. Blue colored labels are occlusion labels, while red are normal animal labels.

use the architecture from Domain Adaptive Faster-RCNN [2], but we include labeled real data at train time. The columns corresponding to FR-CE and FR-WCE are the standard Faster-RCNN trained over a training set that is a union of the synthetic and any available labeled real data (e.g., at 0% real data, it is only trained with synthetic data, while at 50% real data, all synthetic data is used plus half of the labeled real data). The columns for DA-FR-CE and DA-FR-WCE, on the other hand, indicate that in addition to the domain adaptive losses (image and instance level), the available labeled real data is also used to compute the label prediction loss included in the Domain Adaptive Faster-RCNN setting. We used three settings by using 0%, 50%, and 100% of the labeled real data to the training set of the synthetic data. All experiments were performed with VGG16 as the backbone network for 10 epochs with a batch size of 1, as well as an initial learning rate of 1e-4, a decay of 0.1 after a step of 4 epochs, and optimization with SGD. This table confirms that the synthetic data brings value despite the visible domain shift with respect to the real data. Unsupervised Domain Adaptation techniques help in improving performance, but using labeled real data improves the mAP results by over 10%. We expect BIRDSAI to be helpful in the development of more powerful unsupervised and semi-supervised domain adaptation techniques for object detection.

Scale	**FR-WCE (ResNet)**	**SSD (ResNet)**
SA	0.202	0.137
MA	0.442	0.368
LA	0.884	0.886
Animals	**0.616**	**0.569**
SH	0.149	0.172
MH	0.193	0.214
LH	0.106	0.195
Humans	**0.142**	**0.196**
Overall	**0.403**	**0.390**

Table 1: Detection performance baseline using the mAP metric for different scales ((S)mall, (M)edium, (L)arge) of objects ((A)nimals, (H)umans) in the dataset with ResNet as the base model.

Configuration	**DA-FR-CE**	**DA-FR-WCE**	**FR-CE**	**FR-WCE**
Syn $\rightarrow$ Real	0.443	0.459	0.309	0.313
Syn $\rightarrow$ Real (50% Sup. DA)	0.466	0.474	0.384	0.398
Syn $\rightarrow$ Real (100% Sup. DA)	0.522	0.518	0.448	0.472

Table 2: Detection performance baselines using synthetic data. The mAP metric is reported.

B.1.1 Species Recognition

We also annotated animal species in the real video frames where possible. The annotations were based on prior expert knowledge, as well as shape information. There were four different species apart from humans in the real dataset, and one label for *unknown*. We used these data to train Faster-RCNN (without weighting) with a total of six different classes. The annotation statistics and the test mAP are reported in Table 3, with performance being loosely related to the number of examples and typical size of objects (e.g., there are many elephant examples, and these are typically large). There is room for improvement in all cases.

Species	Human	Elephant	Lion	Giraffe	Dog	Unknown
# bboxes	34001	83799	1244	12566	2709	21848
# frames	14959	13349	792	2242	2709	6804
mAP	0.068	0.305	0.004	0.142	0.002	0.237

Table 3: Species label statistics and detection performance with Faster-RCNN on the real videos. The reported mAP values are computed over the test set.

B.2. Tracking

B.2.1 Single Object Tracking

The comparison of SOT performance on *perfect subsequences* and *full sequences* (defined in Sec. 5.2 in the main paper) is included again in Table 4 across different tracking algorithms - ADNet[1], ECO[2], Siamese RPN[3] and MCFTS[4] - for ease. We show single object tracking (SOT) performance over the perfect subsequences and full sequences using the standard tracking metrics in Fig. 3 and Fig. 4, respectively.

We observe that Siamese RPN [5] performs very poorly on SOT in BIRDSAI. The Siamese RPN has been shown to work well in the visible spectrum and relies on visual one-shot detection in the current frame using an exemplar template. This approach seems to work poorly in the BIRDSAI dataset, likely given the limited textural details and poor resolution in the images due to the thermal infrared sensing modality, and the sometimes large camera motion. ECO [3] also relies on some appearance-based cues and correlation filtering. However, it additionally learns a compact Gaussian Mixture Model (GMM)-based generative model of the target object and captures a diverse set of representations. Like Siamese RPN, MCFTS [6] also relies on deep convolutional networks, but it performs much better than the Siamese RPN in all cases. Because MCFTS uses convolutional features from a pre-trained network to form an ensemble of correlational trackers, we conjecture that the ensemble-based approach helps improve performance for weak trackers. AD-Net [10] is trained using a reinforcement learning-based approach where a convolutional neural network is trained as the policy function. The state is comprised of the cropped bounding box-based region of interest from the previous frame and a historical sequence of actions, where the actions capture the motion of the object's bounding box, e.g., left, right, far right, scale up/down, etc. The performance improvements of AD-Net possibly arise from the fact that it uses a history of actions, which captures the object motion from the last several frames.

The trackers that perform well on the *perfect subsequences* deteriorate when tested on *full sequences*. This performance drop is evident from the success and precision plots in Figs. 3 and 4. In most real-world scenarios, the sequences will be affected by noise, occlusions, the object leaving the frame and other such interruptions.

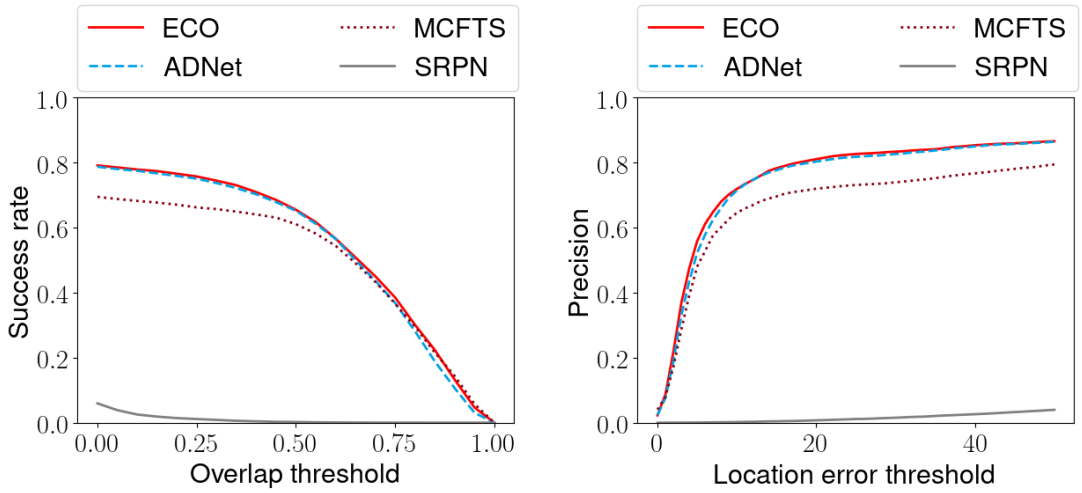


Figure 3: Success and precision plots for the SOT with benchmark algorithms on *perfect subsequences*.

B.2.2 Multi Object Tracking

Table 5 tabulates the results obtained by trackers in the MOT setting, including results for IoU tracker provided in the main paper for easier reference. Off-the-shelf MDP [9] underperforms the IoU tracker, when the latter is provided with ground truth detections.

[1] https://github.com/hellbell/ADNet
[2] https://github.com/martin-danelljan/ECO
[3] https://github.com/songdejia/Siamese-RPN-pytorch
[4] https://github.com/QiaoLiuHit/MCFTS

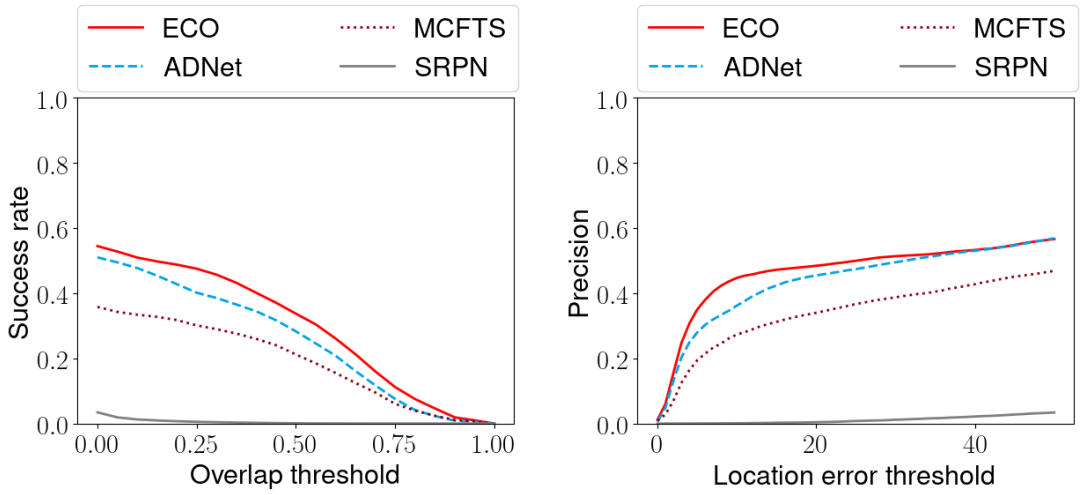


Figure 4: Success and precision plots for the SOT with benchmark algorithms on the entire set of *full sequences*.

Method	Perfect Subsequences		Full Sequence	
	Precision	**AUC**	**Precision**	**AUC**
ECO	**0.8103**	**0.5430**	**0.4842**	**0.2972**
AD-Net	0.8029	0.5331	0.4545	0.2546
MCFTS	0.7194	0.4946	0.3401	0.1886
Siamese RPN	0.0073	0.0093	0.0041	0.0048

Table 4: Single Object Tracking Evaluation. Precision is at 20 pixels. "Perfect subsequences" excludes noisy/occluded frames, while "Full sequence" includes them.

Method	Object Size	MOTA	MOTP
IoU Tracker (GT det.)	S	61.6	**100.0**
	M	**91.3**	98.9
	L	80.6	**100.0**
MDP Tracker (GT init.)	S	21.6	75.9
	M	54.6	84.1
	L	75.8	90.8

Table 5: Multiple Object Tracking Evaluation. IoU tracker is given ground truth detections (GT det.), while an off-the-shelf MDP-based multi-object tracker is initialized using the ground truth detections (GT init.). S, M, L represents small, medium, and large objects, respectively.

Class	FR-CE	FR-WCE	YOLOv2	SSD	DA-FRCE	DA-FRWCE
Animals	0.188	0.204	0.074	0.058	0.112	0.117
Humans	0.177	0.186	0.032	0.092	0.107	0.142
Overall	0.181	**0.192**	0.044	0.089	0.110	0.129

Table 6: Cross-Dataset Detection performance evaluation using the mAP metric.

B.3. Cross-Dataset Evaluation

We also provide results trained using the LTIR dataset [1], as this was one of the most visually similar datasets to BIRDSAI. The results of cross-dataset detection on all of the baseline detectors as well as the domain adaptive detectors is shown in Table 6. Based on these results, we conclude that the BIRDSAI dataset is substantially different than [1]. Moreover, based on the results in the previous sections, we can also conclude that it is sufficiently challenging by itself.

References

[1] A. Berg, J. Ahlberg, and M. Felsberg. A thermal object tracking benchmark. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2015. 4

[2] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[3] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[5] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[6] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen. Deep convolutional neural networks for thermal infrared object tracking. *Knowledge-Based Systems*, 134:189 – 198, 2017. 3

[7] D. J. Mccafferty. The value of infrared thermography for research on mammals: previous applications and future directions. *Mammal Review*, 37(3):207–223, 2007. 1

[8] P. Trethowan, A. Fuller, A. Haw, T. Hart, A. Markham, A. Loveridge, R. Hetem, B. du Preez, and D. W. Macdonald. Getting to the core: Internal body temperatures help reveal the ecological function and thermal implications of the lions mane. *Ecology and evolution*, 7(1):253–262, 2017. 1

[9] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[10] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3