To Microsoft & Global Wildlife Conservation

AI-assisted Aerial Imagery Analysis (AIAIA) to Map Human-Wildlife Proximity in Tanzania

By Zhuang-Fang Yi, Howard Frederick, Ruben Lopez Mendoza, Ryan Avery, Lane Goodman

Report Outline

Summary

Wildlife conservation is in a race against human expansion worldwide. The expansion of settlements and agricultural lands coupled with a three percent population growth annually in sub-Saharan Africa makes it difficult to protect wildlife and its habitat. The proximity of humans and wildlife has the potential for conflicts through competition for resources and space. For the humans that live and work in close proximity to wildlife, wildlife activity can lead to loss of income, property, and sometimes human lives. At the same time, wildlife corridors are diminishing rapidly in many parts of Africa due to the competition.

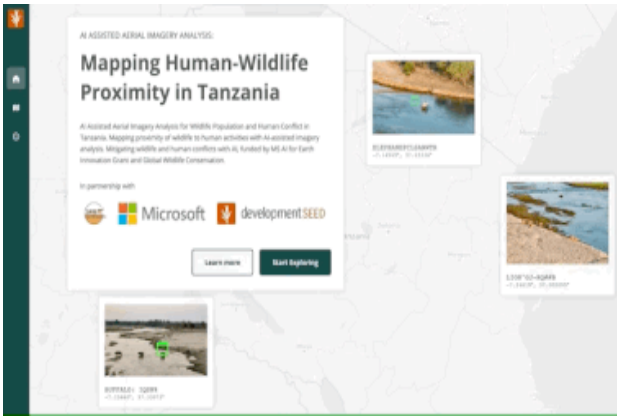
Better survey systems that capture human and wildlife distributions on the ground provide a guide to conservation practitioners, policy-makers, and local residents contributing to wildlife conservation policies that can mitigate such potential conflicts. However, creating an early warning system requires frequent monitoring and tracking of human, wildlife, and livestock activities and movements. Currently it has been done through aerial surveys. The surveys typically happen on three to five year intervals due to high logistical costs and the difficulty in fielding logistics, flight crew, fieldwork, and analysis teams in remote locations. New survey methods with automated camera systems speed up detection and decrease implementation costs, but produce tens of thousands of images and require intensive labor efforts to sort through images.

Tanzania Conservation Resource Center (TZCRC), working with partners Development Seed and the Tanzania Wildlife Research Institute (TAWIRI), collectively developed an innovative AI-assisted methodology to increase the speed of spotting and counting wildlife, human activities, and livestock after aerial surveys. This report outlines our methodology to conduct an AI-assisted survey and subsequent analysis to produce maps of wildlife and human distributions together with a proximity map of potential conflict areas between wildlife and the human-associated activity across survey areas.

This project utilized two different types of image capture from aerial censuses: traditional Rear Seat Observer (RSO) censuses (using human-eye detection) and photographic aerial surveys (PAS)[1,2]. We relied on images that were captured during RSO censuses for machine learning model development. The Tanzania Wildlife Research Institute provided annotators (wildlife domain experts) at a small annotation lab in Arusha, northern Tanzania, to process a database

of images from RSO survey counts in Tanzania collected by TAWIRI in the past decade. Labels were annotated by a group of volunteers. Around 7,000 RSO images were labelled.

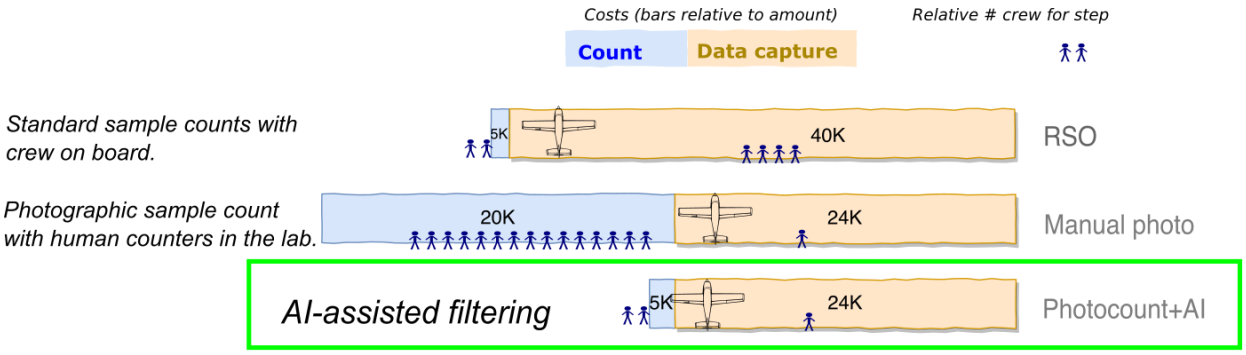
At the end of the project, we present two AI-assisted systems: 1) an image classifier, **AIAIA Classifier**, that filters images containing objects of interest from tens of thousands of aerial images using automated camera systems and 2) a set of three object detection models, **AIAIA Detectors**, which locate, classify and count objects of interest within those images. The detected objects were assigned to image IDs that have their unique geolocation recorded during the aerial surveys. These geocoded detections were then used to generate maps of the distribution of wildlife, human activities, and livestock, with a visualisation of mapped proximity highlighting potential conflict areas. Explore the map here.



Specifically, the image classifier, AIAIA Classifier, filters an image containing our objects of interest, either human activities, wildlife, or livestock. Each object detector model, AIAIA Detectors, separately locates either wildlife and livestock at the species level or human activity. The models were containerized and registered to Azure Container Registry. The model training sessions were deployed with Kubeflow on Azure Kubernetes Service with GPU instances. The sessions were also tracked by TFJob for hyperparameter tuning and search experiments. Such model training is monitored by Tensorboard so we can watch model performance over validation dataset. The best performing models were selected and containerized as TFServing images, including the classifier and detectors (called “aiaia-fastrrcnn”) that hosted on Development Seed’ DockerHub for our scalable model inferences, see TFSeriving images here. Our AIAIA classifier processed 12,000 image chips (400x400 pixels) per minute, and we were able to

process 5.5 million images under 8 hours. The AIAIA detectors each processed 172 images per minute on a K80 GPU machine.

Our AI-assisted Aerial Imagery Analysis (AIAIA) introduces a workflow that: 1) can speed up the processing of wildlife counts and mapping human influences in wildlife conservation areas up to 60%; 2) it has the potential to reduce the implementation costs of counting by up to 20%, which will enable more frequent monitoring. Once the training data quality, and model performance of AI assisted workflow mature and stabilise, we foresee the hours spent on getting accurate human-wildlife proximity maps would only take 19% of current human manual workflow, and potentially reduce cost of identifying and counting objects over 81,000 aerial images from $20,000 to less than $5,000 (See the following graph).



While it is promising to use AI-assisted imagery analysis, however, AI-assisted workflow is still not perfect. The quality of outputs heavily depends on the quality of the training dataset we supply to the models. In our case, the annotation task was relatively new to all the volunteer annotators, and our aerial images were challenging to annotate for a variety of reasons, e.g. the objects of interest are small, hard to identify because the lighting, angle of the shots, the body size and colors of objects. We observed the following quality issues in the training data:

- Missing labels. Some wildlife, livestock, or human activities were not annotated in images, particularly when the condition of the image was blurry, situated in a complex landscape, or contained many objects
- Label duplication. Some objects were annotated twice or more, leading to further class imbalance as well as less accurate validation and test metrics due to an increase in false negatives from missed detection of the duplicate label.

- Mislabeled. Particularly for the wildlife categories and livestock, many instances were mislabeled as different classes.

Each of these problem types varied in degree depending on the difficulty of the class that was being annotated. These issues were overall consistently present in the training and test datasets. This made the AIAIA Detectors more difficult to accurately train and caused the evaluation metrics to be less robust, since many ground truth labels were incorrect. We discuss these issues in detail in the main report, see section “Results and Discussion”.

Even with training data quality issues, our classifier model achieved a > 0.84 F1 score with the test dataset. The best performing classes for the wildlife, livestock and human activities detectors were buffalo (.48 F1), cow (.29 F1), and building (.49 F1), with each class having higher precision than recall. Other categories had high performance in terms of precision (meaning less false positives), while recall (the ratio of true positives to all groundtruth) suffered, including the following: elephant (.59 precision, .17 recall), smaller ungulate (.85 precision, .24 recall), and shoats (.69 precision, .14 recall). This high precision shows that our model is capable of correct, high confidence predictions and the recall metrics show that it has trouble with separating all ground truth from the background. Low recall scores generally imply poor training dataset quality. We expect that addressing training data quality issues and either discarding classes with low amounts of samples or increasing the amount of samples will substantially improve both recall and precision for our object detection models.

The future AI-assisted workflow will highly benefit from having a human-in-the-loop approach to improve training data quality by helping annotators to only annotate images with objects in them, annotate difficult classes, and fix incorrect groundtruth. We proposed the future AI-assisted workflow should bring humans into the loop in a three phase workflow: 1) training dataset visual inspection and validation before the AIAIA Classifier model training; 2) AIAIA Classifier model output inspection before the filtered images are passed to the AIAIA object detectors; 3) manual output inspection, validation, evaluation and correction before the detected, identified and counted objects are aggregated for the minimum viable product (MVP) visualization to produce proximity and risk maps. Improving training data quality is critical for our AI-assisted workflow.

Background

Mapping human and wildlife distributions for conservation

Conservation of wildlife is in a race against human expansion worldwide. With around 3% population growth annually in sub-Saharan Africa[3], protecting wildlife and its habitats gets more and more difficult as humans move closer to wildlife areas through expansion of settlements, agricultural lands as well as raising livestock as main livelihood sources[4].

Fundamentally, human-wildlife conflicts (HWC) are caused by competition for food and space. For the humans that live and work in close proximity to wildlife, wildlife activity can lead to loss of income, property, and sometimes human lives. This results in incidents where wildlife species of conservation interest are threatened or killed to prevent further conflicts[5]. An early warning system that can flag potential conflict on the ground will provide a significant guide to conservation practitioners, policy-makers, and local residents from designing policy intervention to loss prevention, and eventually minimise conflict. However, creating these HWP layers and warnings requires frequent monitoring and tracking of human, wildlife, and livestock activities and movements on the ground or from the air. Both camera trapping on the ground and aerial surveys from the air pose tremendous logistical challenges and costs. Both types of surveys produce image datasets which are prohibitively large to manually annotate, count, and record objects of interest. Instead, machine learning models can assist human annotators by generating high confidence predictions of images that contain objects of interest. This method may be more cost-effective for quickly locating human, wildlife, and livestock activities. In this study, funded by Microsoft AI for Earth and Global Wildlife Conservation, we proposed an AI-assisted aerial image analysis (AIAIA) approach to mapping human and wildlife distributions and potential conflicts in Tanzania.

We built an end-to-end AIAIA workflow that combines an AIAIA classifier and AIAIA object detectors to assist and guide human annotators to quickly and efficiently review images from aerial surveys. The binary AIAIA classifier acts as a filter that only keeps images that are likely to contain human activities, livestock, or wildlife. The three AIAIA object detectors for human activities, livestock, and wildlife detect, locate, classify, and count individual objects by subclass. These detections are used to create the Human-wildlife proximity (HWP) MVP visualization.

There is a subtle but important difference between HWC and human-wildlife proximity (HWP). HWC refers to direct conflicts between humans and wildlife, which require mitigation from conservation communities and policy makers. However, HWP reflects areas that could give rise to HWC in the long term, and HWP can be done through spatial analysis without loss assessment.

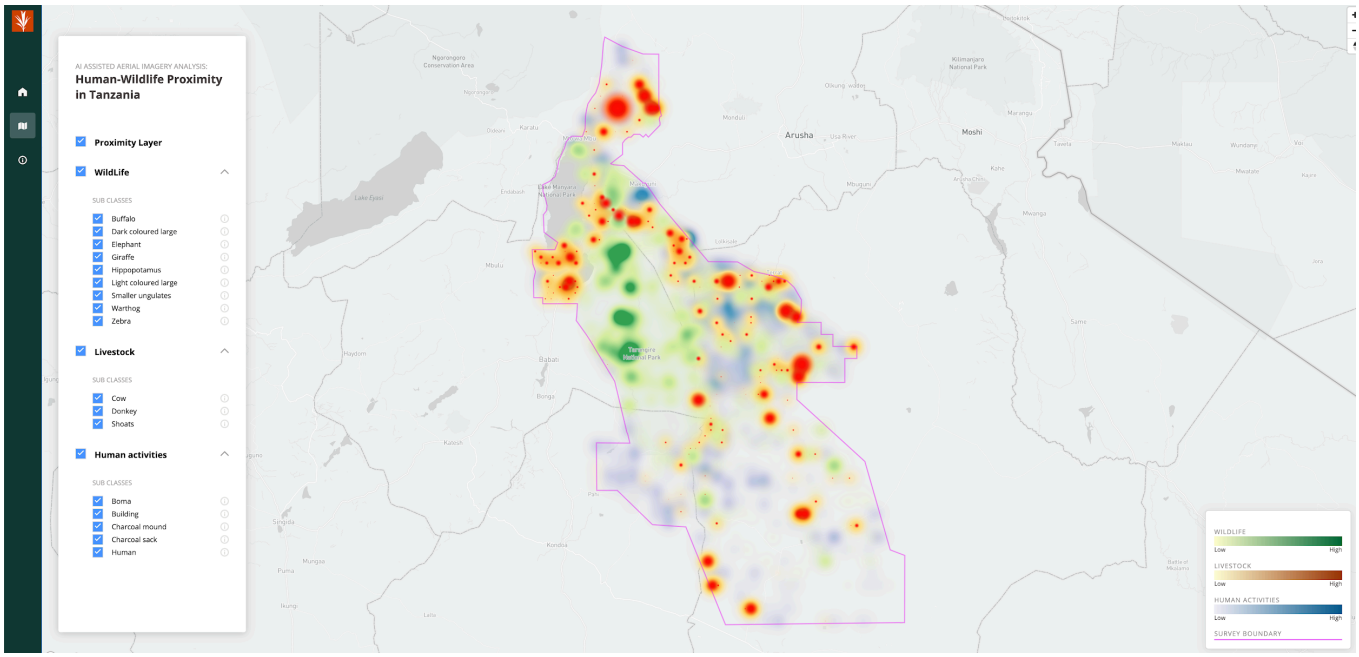


Figure :1 The human-wildlife proximity (in red) in Tanzania is computed and generated when wildlife v.s. livestock and (or) human activities. In our current survey area, such proximity is mainly caused by livestock especially outside of Tarangire National Park, Tanzania.

Aerial Surveys in Wildlife Conservation

Traditional aerial wildlife surveys typically use human observers in a low-flying airplane, counting wildlife by eye. These surveys track changes in wildlife populations (e.g. due to poaching) or responses to environmental changes (e.g. changes in the population of migratory wildebeest in the Serengeti from changing rainfall). Traditional aerial surveys are used in at least 25 countries in Africa, as well as Australia, Mongolia, Kazakhstan, and the USA. Surveys often only occur on 3-5 year intervals due to high logistical costs and the difficulty in fielding logistics, flight crew, fieldwork, and analysis teams in remote locations.

Wildlife surveys provide valuable data about other targets such as human land use (livestock enclosures, thatched huts, etc.) and habitation that are often overlooked once a survey is complete. Combining land use and wildlife distribution maps allows managers to produce risk maps on potential human-wildlife-conflict that will benefit wildlife conservation in the long run. New survey methods using automated cameras show great promise, but the quantity of images to analyse leads to prohibitive cost and time increases.

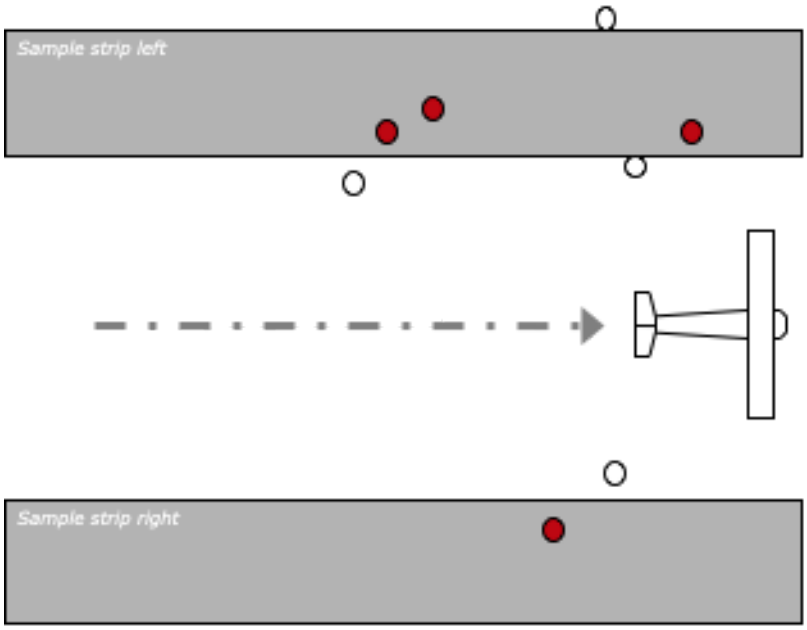


Figure 2: Sample aerial survey data collection

Aerial censuses of wildlife have been conducted since the 1950s in Africa, and since the late 70s, the dominant method used has been "Systematic Reconnaissance Flights" (SRF), and it is broadly applied in wildlife conservation globally [8,9]. The SRF sample count method is suitable for covering large areas to produce accurate maps and estimates. In these counts, one or more Cessna light aircraft, with crews of 4 people, fly straight sample lines across survey zones, taking days or weeks to cover partial areas of such enormous ecosystems. Two rear-seat observers (RSOs) on each aircraft count wildlife and other targets as the aircraft flies over points of interest.

Even though the SRF sample count method is widely adopted in wildlife conservation, concerns still exist, such as whether well-trained RSOs suffer from fatigue, which limits

daily mission times, and RSOs can be highly variable in their performance. Fielding aircraft and crew is expensive and flying low-and-slow is dangerous work.

AI-assisted Approaches in Wildlife Conservation

The method of the aerial census is widely adopted and the standard in conservation track. This practice raises concerns, however, due to the difficulty of finding well-trained RSO's and their ability to get fatigued on long missions. RSO's performance is also less precise and reliable compared to human annotators who are trained to track and count wildlife from taken images.

Using new technology like cheap digital imaging, photographic aerial survey (PAS), for data capture and unmanned aerial vehicles (UAVs) are promising solutions for reducing the financial and logistical cost of flights. However, both PAS and UAV surveys are designed to take continuous photographs along flight lines, which results in tens of thousands of images and requires intensive labor efforts to sort through images, driving up the cost. Results suggest that manual photo counts after the UAV, PAS and SRF surveys exceed the accuracy of RSOs from SRF. However, the analysis time increases from days to months due to the volume and difficulty of counting complex images - typically less than 2% of images have any desired targets in them.

Shifting to PAS will require an order-of-magnitude improvement in photographic review times - an AI-assisted approach can provide this improvement. The potential improvement in the reliability of results (improved consistency together with human RSOs) and the reduced costs will make a photographic, AI-assisted aerial census very attractive in the immediate future. Machines will review the massive amounts of photographs and direct our mappers and analysts to areas where they provide the most value to sort images, track and count wildlife, and identify other objects of interest as they appear in images. Our end-to-end workflow from training dataset creation, deep learning models trained with the cloud computers, model iteration, model output validation, model inference, PostgreSQL database design, and data exportation to create an early warning of potential human-wildlife conflicts and proximity maps.

Image Capture and Labeling from Aerial Censuses

Aerial Imagery Capture

Aerial surveys of large mammals in Africa are normally done from light aircraft (Cessna 172/182/206 type) at an altitude above ground level (AGL) of 90-110m (300 to 350 feet).

There are two types of images that are available from aerial surveys for this project:

Rear Seat Observer - RSO

During the survey the RSO's window-mounted cameras are used to verify herd size and ID - they are triggered by the *human observer* when he/she sees a target of interest (all elephants, and larger herds of any species). The aircraft flies straight lines (transects) back and forth over the target area.

- These images are of lower quality as they are taken through the Plexiglas.
- Most images are 24 megapixels (MP), taken at 35° down-angle from horizontal.
- Very target-rich set of images which will produce many examples for training. This is because a human has *triggered* the camera on a target.
- Around 20,000 images are currently available and being reviewed by the annotation lab.
- GPS metadata is often not available (camera clocks not synchronized perfectly with GPS).

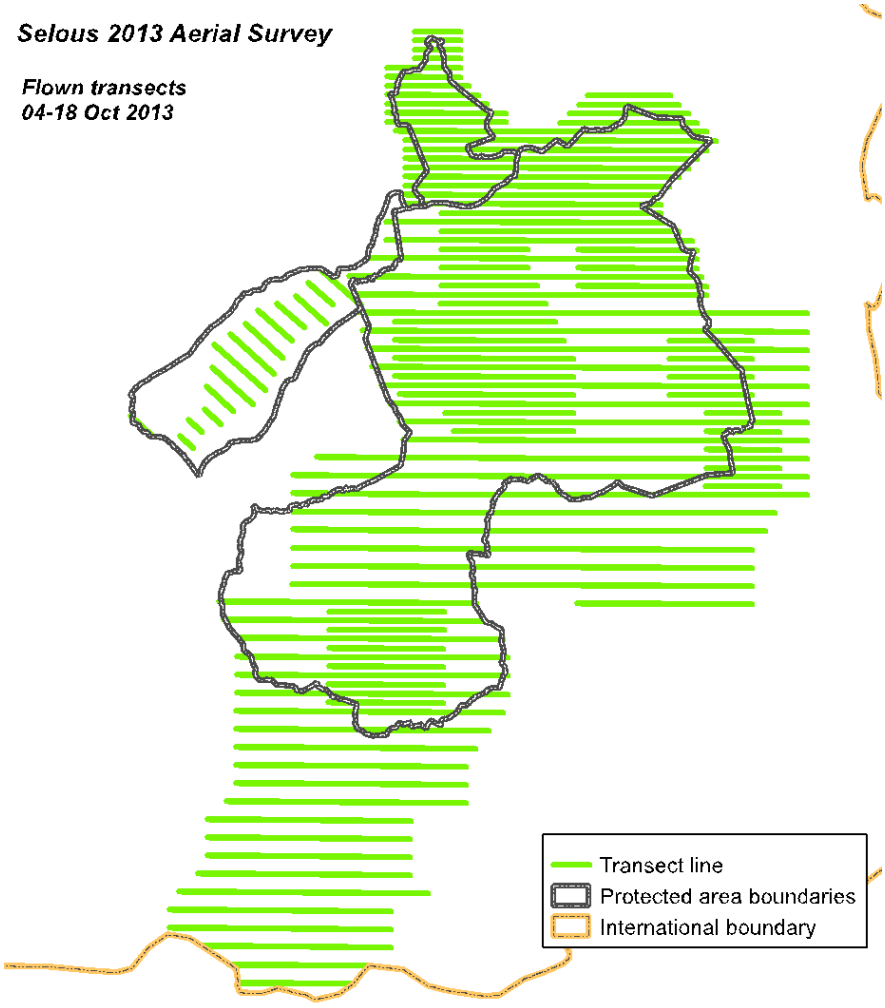


Figure 3: The aircraft flies straight lines (transects) back and forth over the target area during the aerial surveys.

Photographic Aerial Survey

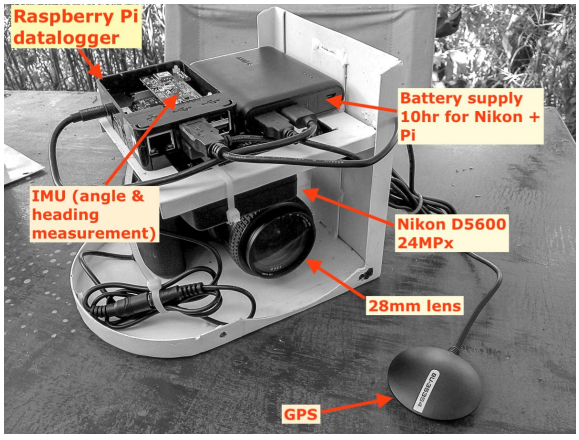
The traditional human-eye detection is still the main method in use, but studies testing high-resolution photographic systems will hopefully become a replacement (Photographic Aerial Survey (PAS). The new system uses cameras on the wing struts which take constant images along flight paths.

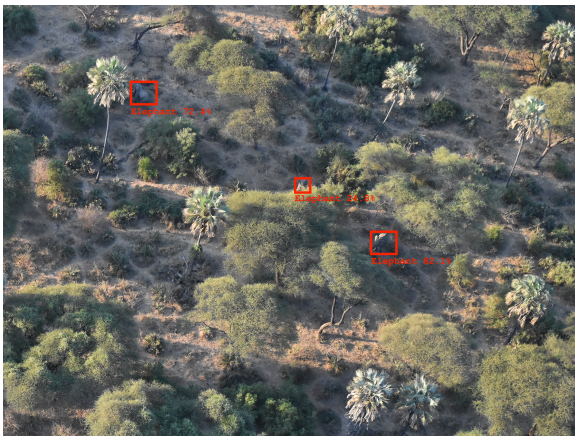
- These images have no intervening Plexiglas and are much higher quality.
- Taken at 45° down-angle, and with a lens that mimics the same sample field of view as the human eye.
- 24MP and optimized for image sharpness.

- Very low rate of 'positives' - perhaps 2% of images will have any wildlife or livestock present.
- Around 500,000 images are available and more are being collected every year.
- GPS metadata present for all images.
- Overlapping images are taken at 2-second intervals.

A. the survey aircraft

B. Camera mounted on the aircraft.

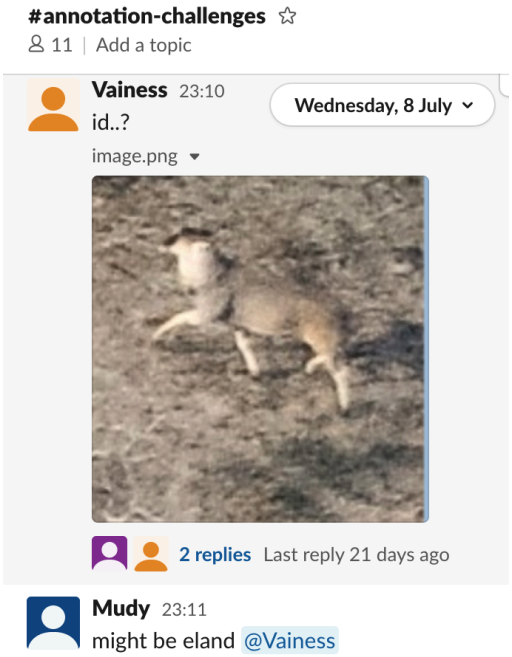




C. The “Lanner” aerial photography system[10] using off-the-shelf components and providing an add-on, high-quality photography option to regular survey flights.	D. A group of elephants was captured by the camera system. We drew boxes over the elephant individuals and labeled them (in red text). The photo currently is in a dimension of 6016 x 4000 pixels per image.

Figure 4: The current photographic aerial survey in Tanzania.

Training Data Labelling



A high-quality training dataset is a key asset for obtaining a well-performing machine learning model. We created label data with the Computer Vision Annotation Tool (CVAT[11])

TAWIRI Annotation Team

The TZCRC assisted in setting up an annotation lab space in Arusha, northern Tanzania, to process a database of images from RSO survey counts in Tanzania collected by TAWIRI in the past decade.

- Annotators were provided by TAWIRI - mostly MWEKA (wildlife college) and university students and graduates looking for experience in conservation biology. The Covid-19 crisis meant that setting up a lab with 8-10 annotators as initially planned was impossible, and only 6 people were able to work with the suggested distancing layout.
- Annotators were mostly domain experts on Tanzanian wildlife and/or human activity assessment. The ability to recognize large plains species was a prerequisite, and supplemental training was provided on how to identify the smaller and less common game (kudu, bushbuck). The entire TAWIRI lab team was added to a Slack channel so that people could ask questions and check species identifications - this proved to be invaluable for training data quality assessment.

- A CVAT (Computer Vision Annotation Tool) server was set up at the lab and a lab manager was trained in managing the system. Annotators were trained in the use of CVAT, and the lab manager exported each task as it was finished.

Training Data Creation

We showcased a simplified version of the training dataset creation workflow using CVAT in Figure 5. A group of volunteers in Tanzania imported the full-size aerial images to CVAT, each with 6016 x 4000 pixels and a spatial resolution of 2-4 cm per pixel. The CVAT annotation tool used applied to draw bounding boxes around objects of interest and a label class was added per box. Once the task was finished, we exported the labeled bounding boxes XML files for further visual inspection, training data quality analysis, and converted them into machine learning ready datasets, i.e. in TFRecords format. TFrecords is a data format that stores a sequence of binary records for Tensorflow to read image and label data efficiently during the model training [(TFRecord and tf.train.Example)](#).

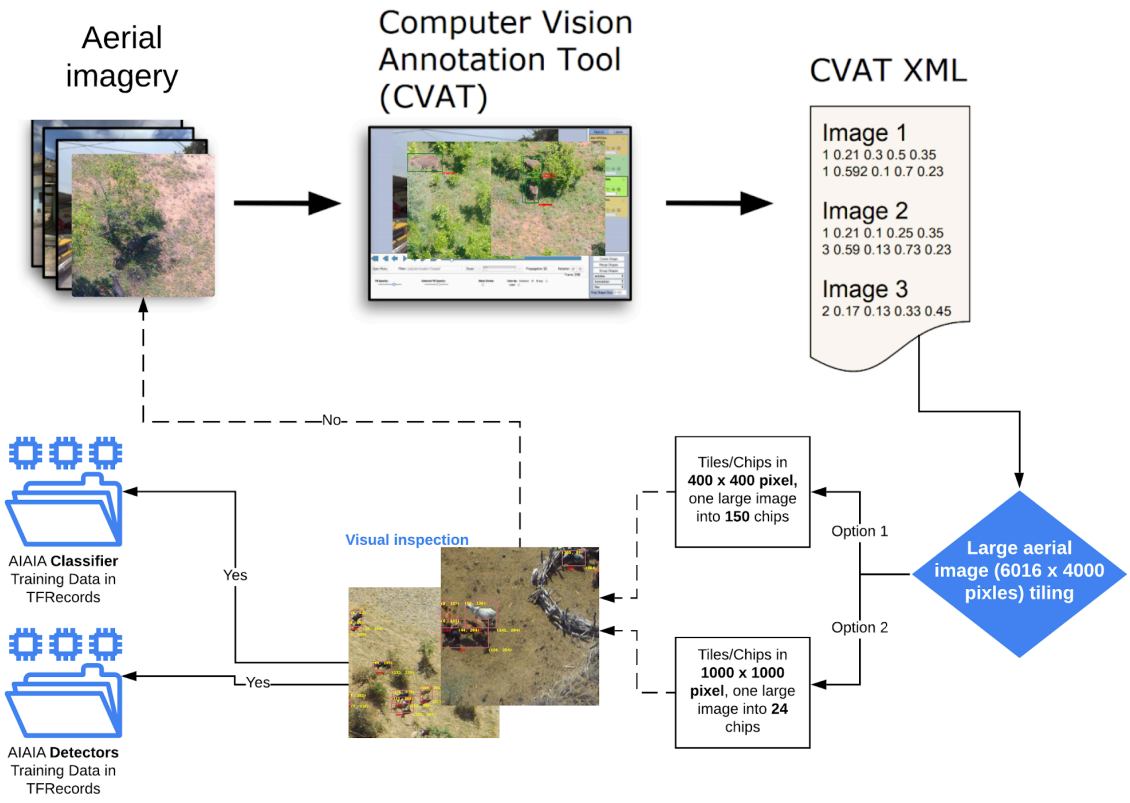


Figure 5: Training dataset annotation workflow using CVAT. Aerial imagery was annotated by a group of volunteers in Tanzania. The annotators created 30 classes of labels that covered wildlife, livestock, and

human activities. The final labeled data is tiled/chipped and converted into TFRecords format as machine learning ready data for the coming model training.

Two iterations of training datasets were created during the summer and fall of 2020 by TZCRC Annotation Lab. The first iteration of the training dataset (created in the summer) was used to train a single 30-class object detection model. Because the training dataset was not of high quality, and the amount of labels was not sufficient for some rare classes, the first iteration of the object detection model could only detect the 9 classes that had the most training data and were easier to visually distinguish from the background. For the second training dataset , the volunteer annotators in the lab were able to create a higher quality training dataset, including:

- Fewer missing labels
- Fewer mis-labeled objects
- Fewer label duplications
- Fewer bounding boxes drawn around groups of objects and instead drawn around individuals
- Bounding boxes drawn with more accurate boundaries around the objects instead of including extra background

The training dataset quality issue is still present) and we will present current label quality issues and how to improve for the future iterations in the next section, “Training Data Quality”.

During the second iteration, 30 classes of labels were still created. From the lessons learned during the first iteration, we discarded some labels from the model training process if sample counts were less than 100 from the aerial surveys, e.g. crane, ostrich, stork and lion. In the wildlife category, we grouped the species based on their body sizes and skin colors (a ‘visual guild’) to improve their representation during the model training, as follows:

- A ”light colored large” class now includes classes eland, hartebeest, kudu, roan, and oryx.
- A “dark colored large” class includes wildebeest, topi, waterbuck and sable.
- “Smaller ungulates” include gazelle, impala and antelope.

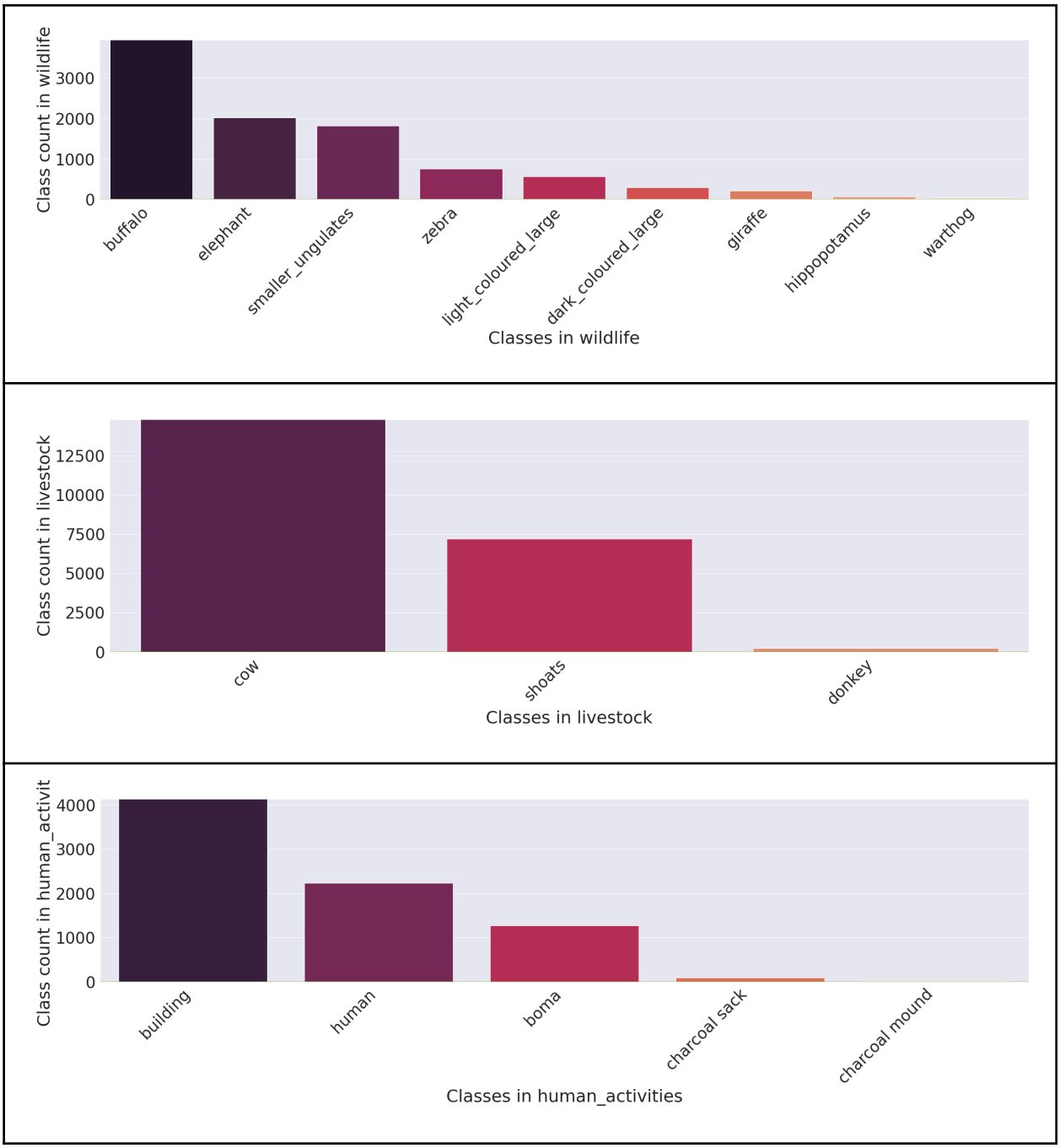


Figure 5. Class labels per each category from wildlife, livestock to human activities that are labeled for the machine learning models.

At the end of second iteration training dataset creation, we ended up having three categories of training data for further AIAIAI Classifier and Detector model training: wildlife, livestock, and human activities (Figure 5).

Training Labels for Wildlife

We had nine classes under the wildlife category, and the top three classes by number of bounding box labels were:

- Buffalo, 2022 labels (bounding boxes).
- Elephant, 3937 labels.
- Smaller ungulates, 1812 labels.

Training Labels for Livestock

The major three classes of livestock present in the aerial surveys used for training dataset labeling were:

- Cow, 14825 labels.
- Shoats, 7201 labels.
- Donkey, 219 labels.

Training Labels for Human Activities

The five classes of human activities present in the aerial surveys used for training dataset labeling were:

- Building, 4139 labels.
- Human, 2230 labels.
- Boma, 1276 labels.

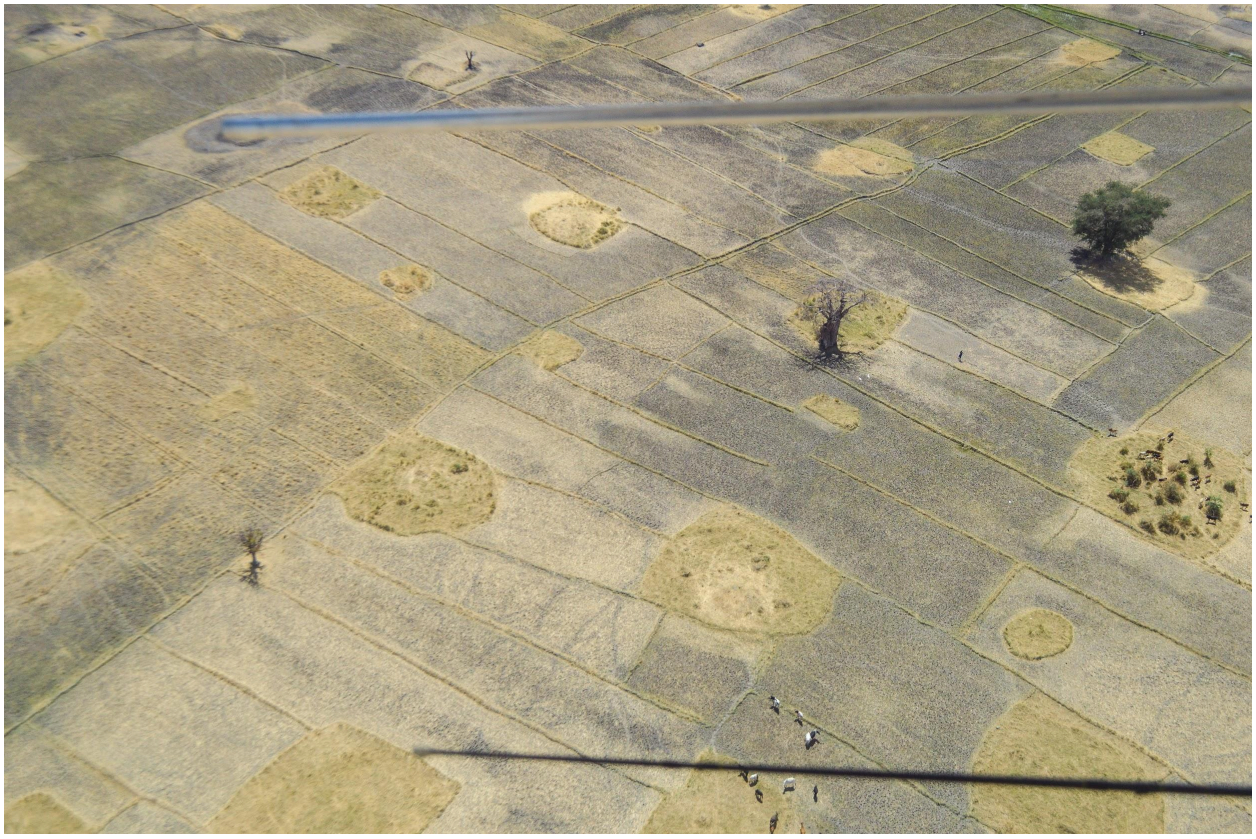
In total, we have 45,155 labels (bounding boxes) for three categories (wildlife, livestock and human activities), which belongs to 12,500 unique image chips (each was 400 x 400 pixels). We randomly selected 7000 image chips that contain objects/bounding boxes, and labeled these as 1 and also included 7000 background chips drawn randomly from the pools that without any objects, labeling these as 0 for the model training process. For image classification, a total of 14,000 image chips were then sampled and split by 70, 20, and 10 percent as train, validation and test TFRecords respectively. We generated separate TFRecords for the three separate AIAIA detectors for wildlife, livestock and human activities. The TFRecords for object detection were created based on labels/bounding boxes presented in Figure 5.

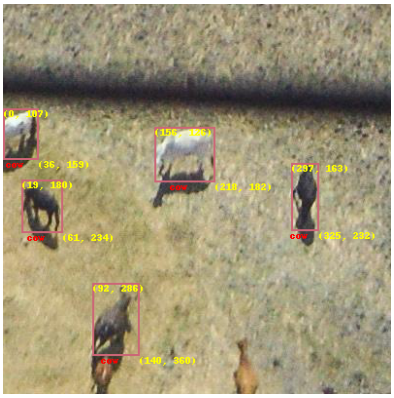
Training Data Quality

A lot of challenges were encountered during labeling aerial images compared to, for example, images from camera traps - images from aerial surveys contain relatively small objects of interest, and backgrounds and lighting varies dramatically.

Aerial images were captured in the air, 90-110m (300 to 350 feet) above ground. An aerial image contains 6016 x 4000 pixels, though only a very small portion of the image actually contains objects of interest. We tiled each of these larger images into 150 chips (400 x 400 pixel per chip). 4 out of 150 chips (about 2.7%) have objects present. A lot of livestock appear in herds in the images, as do some wildlife species, e.g. elephants, buffaloes, wildebeest, and antelope. Without zooming in really closely to the objects, and without pre-existing knowledge of wildlife and livestock appearance and habitat, it's very difficult to label the object correctly.

Below is an example image that was captured during the aerial survey. The objects appear at the bottom of the image near the plane shadow. There are cows and a human in the aerial image and the object sizes are all small. All the animals were labeled correctly as cows, but each cow object exhibits varying properties, including different: color, shading, and body angles. These issues can be challenging for computer vision/deep learning models to handle. Furthermore, the training labels of some of these aerial images, in total 6 image chips (in 400 x 400 pixels) highlight the quality issues we see through the rest of the training dataset and in prior iterations of the training dataset.



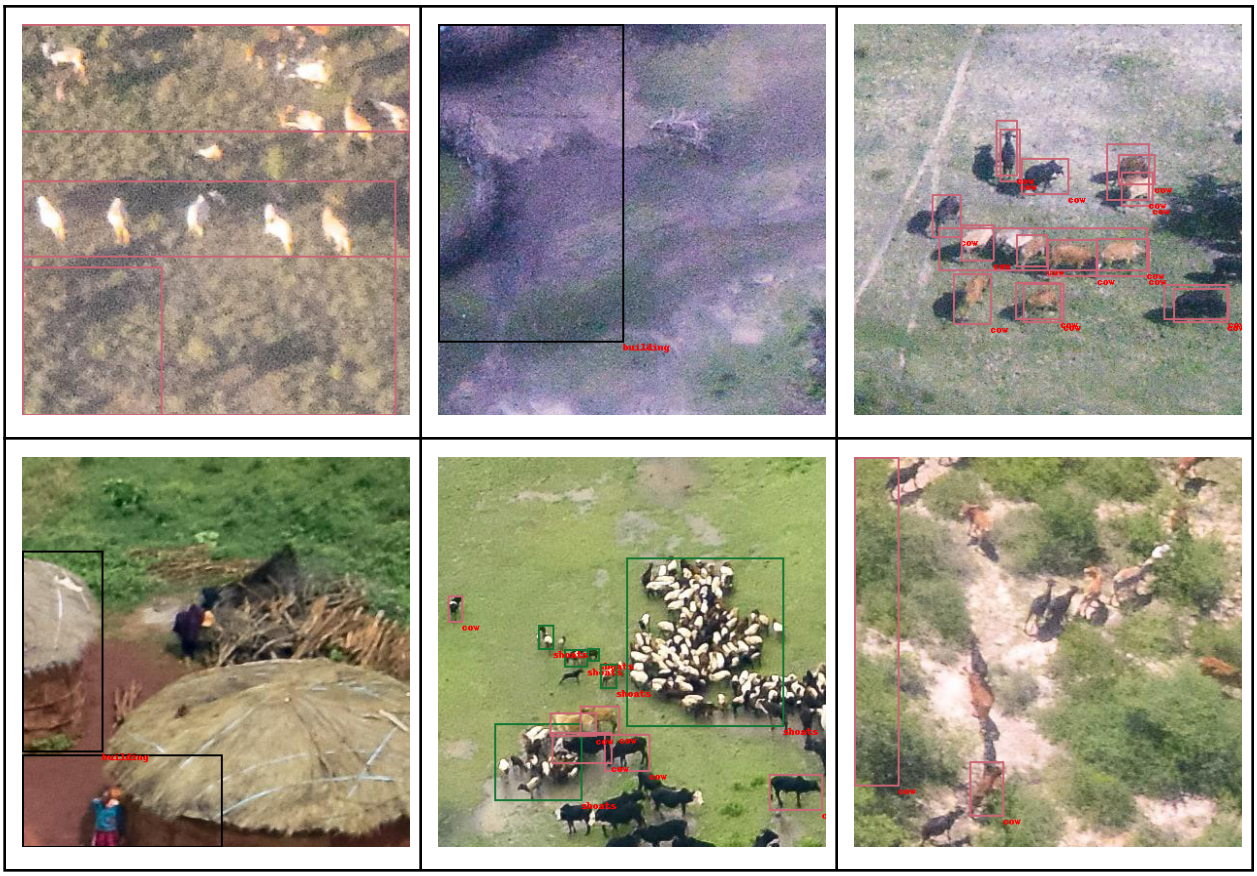


Missing labels

Missing labels can impact both the classifier and detector model performances. The classifier model could end up having negative samples, image chips actually contain objects but because of the missing labels the chips are considered “Not-Object”. A neural network will be trained to recognize object’s patterns, spatial features, colors as well as the background. Missing labels will pollute the training data by suggesting image patterns associated with a class shouldn’t be associated with a class.

Mislabeled classes

Mislabeling includes the classes which were not labeled correctly, including whenor multiple objects are mixed under one class. This creates a lot of added noise during model training.



Label duplication

Label duplication is present for both the livestock and small to midsize wildlife. During the model training, the model ends up making more predictions for over-labeled classes. During model evaluation, these duplicate labels must be assumed to be correct for the purposes of calculating metrics, since there is no efficient way to filter them out without manual editing.

AI-assisted Aerial Image Analysis (AIAIA)

AIAIA Workflows

Two AI-assisted workflows were built in this study, an **AIAIA Classifier** and **three AIAIA Detectors**. The AIAIA classifier was applied to filter an image containing our “objects of

interest", either human and settlements, wildlife, livestock or the combinations of them. AIAIA Detectors were built on top of Object detection models (TensorFlow Object Detection API) were applied to detect wildlife species, humans and settlements, livestock species, and their counts, separately. The end-to-end workflow, including a classifier and three detectors, aims to reduce the costs of conducting game counts and human influence in wildlife conservation by 50%, enabling more frequent monitoring.

The AIAIA Classifier and Detectors were deployed on top of Kubeflow and Kubernetes that allow machine learning and cloud engineers to run model training and experimentations quickly and efficiently with TFJobs (Figure 6). Each model training and model experiment was recorded with TFJob YAML files, so it's traceable. Once the best performing model is identified either for the classifier or detector with the model evaluation metrics. For the AIAIA Classifier, we used F1, precision and recall scores as well as an ROC curve from model evaluation over test dataset. To evaluate metrics for the detectors, we compute confusion matrices, F1 scores, mean average precision and recall scores.

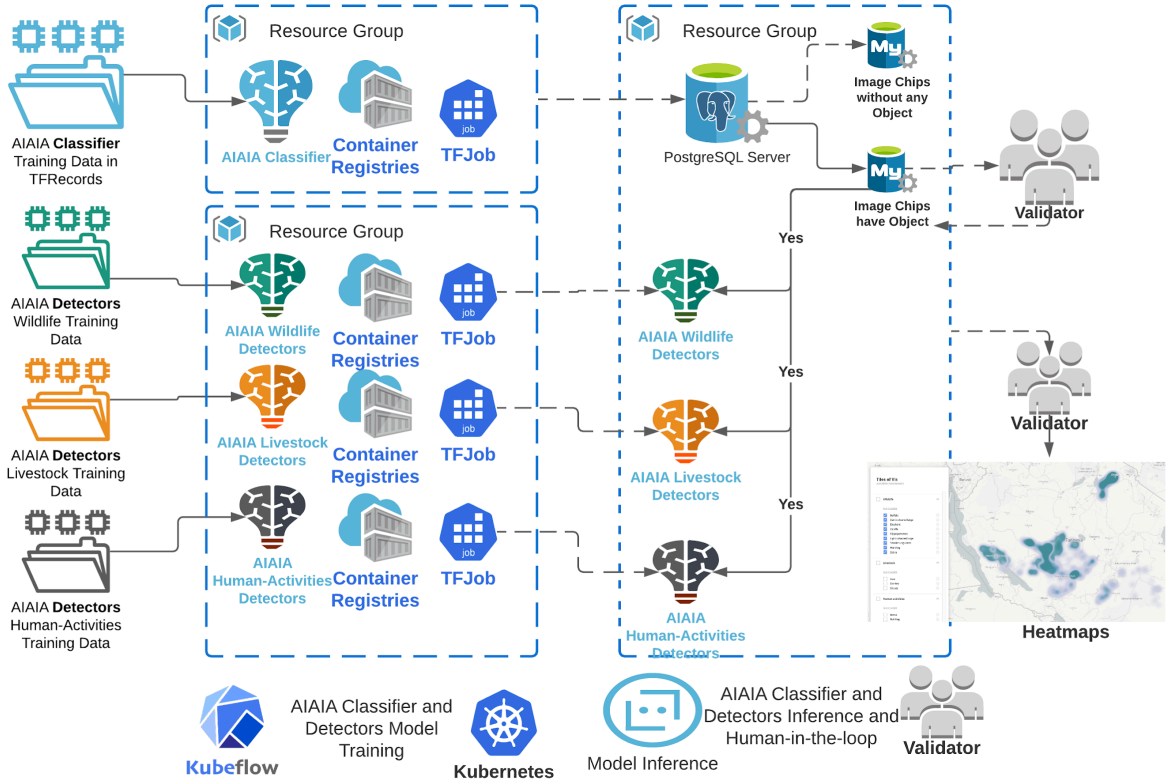


Figure 6. Two AIAIA workflows were developed in the study. The AIAIA Classifier, a binary image classifier, acts as the filter to keep only image chips that contain "object of interests". The AIAIA Detectors (wildlife, human-activities and livestock detectors) detect and count these objects of interest in images. Detected objects and counts were served to our MVP for flagging potential human-wildlife conflicts on the ground for wildlife conservation communities and policy-makers.

Model Training and Experiment with Kubeflow on Azure

Kubeflow and Kubernetes have become standard toolkits in industry, allowing data scientists to train, deploy, and package machine learning models in a portable, scalable and efficient way. These tools are powerful and flexible enough to accommodate the complexities of applying models to geospatial aerial imagery. With these tools, the models can be deployed to any cloud computing environments, including Microsoft Azure or Google Cloud Platform (GCP). Kubeflow was originally developed by Google. We found Kubeflow documentation on GCP is more up-to-date than Azure's, therefore, it requires more hacky ways to deploy Kubeflow and TFJob to Azure. As follow up work, we will work to make these same workflows available on Azure.

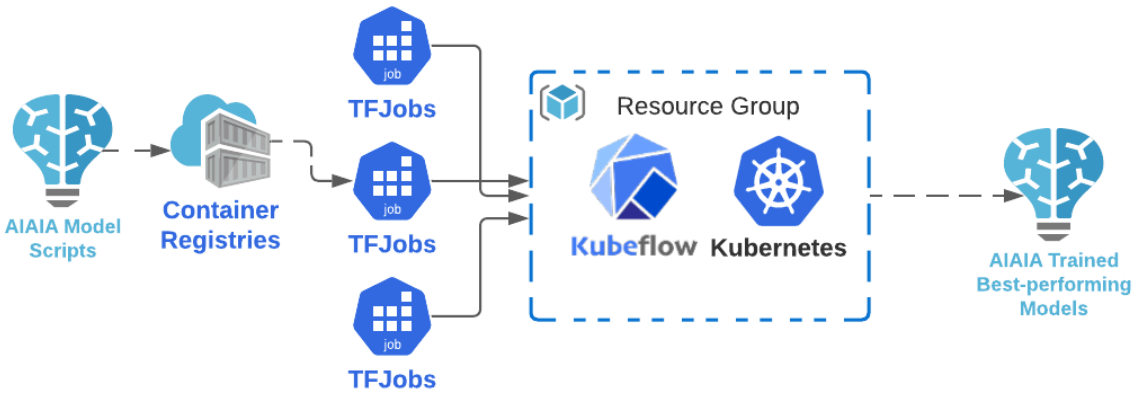


Figure 7. The model training and experimentations are conceptualized in the diagram shown above. Model scripts were containerized and registered on Azure (or GCP). We then deployed Kubeflow to the cloud environment. Once Kubeflow is running on AKS (or GKE), TFJob model experiments were deployed to start the model training with GPU machines. In general we use K80, p100 and T4 machine types. Model evaluation on the validation set selects the best performing trained models from multiple experiments.

Training models on Microsoft Azure will require a few steps::

- Install and setup Kubernetes CLI, kubeclt[1], on your local machine.
- Install Azure CLI and log in with your credentials. The Microsoft AI for Earth program provided Azure cloud credits for this project.

[1] "Install and Set Up kubectl | Kubernetes." 27 Nov. 2020, https://kubernetes.io/docs/tasks/tools/install-kubectl/. Accessed 27 Jan. 2021.

- Create a resource group and Azure Kubernetes Services (AKS) setup on Azure. Our Kubeflow model training and experiments were deployed to AKS. AKS provides continuous integration and continuous delivery (CI/CD), as well as enterprise-grade security and governance on Azure [2]. A GPU node pool can be added to the AKS for both AIAIA Classifier and Detectors model training.
- Azure Container Registry (ACR) must be set up. Model training scripts can be conterized and pushed to ACR for AKS to access later on when the model is deployed and ready to be trained with the AKS GPU node pool;
- Kubeflow setup and deploy.

Kubeflow setup > install > deploy

```
#Create user credentials. You only need to run this command once.
az aks get-credentials -n ${AKS_NAME} -g ${RESOURCE_GROUP_NAME}
# download kubeflow v1.0.2 https://github.com/kubeflow/kfctl/releases/tag/v1.0.2
platform=0-ga476281_darwin
tar -xvf kfctl_v1.0.2-${platform}.tar.gz -C ~/.kfctl

# The following command is optional, to make kfctl binary easier to use.
export PATH=$PATH:${HOME}/.kfctl

# Set KF_NAME to the name of your Kubeflow deployment. This also becomes the
# name of the directory containing your configuration.
# For example, your deployment name can be 'my-kubeflow' or 'kf-test'.
export KF_NAME=kf-ml-aiaia

# Set the path to the base directory where you want to store one or more
# Kubeflow deployments. For example, /opt/.
# Then set the Kubeflow application directory for this deployment.
export BASE_DIR=$PWD
export KF_DIR=${BASE_DIR}/${KF_NAME}

# Set the configuration file to use, such as the file specified below:
export CONFIG_URI="https://raw.githubusercontent.com/kubeflow/manifests/v1.0-branch/kfdef/kfctl_k8s_istio.v1.0.2.yaml"

# Generate and deploy Kubeflow:
mkdir -p ${KF_DIR}
cd ${KF_DIR}
# empty files under KF_DIR before deploy kubeflow
kfctl apply -V -f ${CONFIG_URI}
```

Figure 8: Kubeflow deployment to Azure AKS.

- Store and host training dataset, pretrained model weights, and model configure files on Azure Blob Storage
- Setup TFJob yaml file for model deployment.

[2] "Azure Kubernetes Service (AKS) | Microsoft Azure." https://azure.microsoft.com/en-us/services/kubernetes-service/. Accessed 27 Jan. 2021.

```yaml
apiVersion: "kubeflow.org/v1"
kind: "TFJob"
metadata:
  name: "faster-rcnn-resnet101-serengeti-wildlife"
  namespace: "kubeflow"
spec:
  tfReplicaSpecs:
    Worker:
      replicas: 1
      restartPolicy: OnFailure
      template:
        spec:
          serviceAccountName: kf-user
          containers:
          - name: tensorflow
            image: geoyiacr.azurecr.io//aiaia:v1.1-tf1.15-gpu
            command:
              - "python"
              - "/tensorflow/aiaia_detector/model_main.py"
              - "--model_dir=${blob_path}/model_outputs_tf1/rcnn_resnet101_serengeti_wildlife"
              - "--pipeline_config_path=${blob_path}/model_configs_tf1/configs/rcnn_resnet101_serengeti_wildlife.config"
              - "--num_train_steps=50000"
              - "--sample_1_of_n_eval_examples=1"
              - "--input_type=encoded_image_string_tensor"
              - "--output_directory=${blob_path}//export_outputs_tf1/rcnn_resnet101_serengeti_wildlife_v3_tfs_v2"
            resources:
              limits:
                nvidia.com/gpu: 1
            restartPolicy: Never
            tolerations:
              - key: "nvidia.com/gpu"
                operator: "Equal"
                value: "present"
                effect: "NoSchedule"
```

Figure 9: TFJob yaml file is structured as above. The container is built on top of "tensorflow" deployed to "kubeflow" with an model containerized training pipeline called "geoyiacr.azurecr.io//aiaia:v1.1-tf1.15-gpu".

Methods

AIAIA Classifier

The backbone of the AIAIA Classifier is one of top state of the art convolutional neural networks (CNN), Xception (Chollet 2016) [3] (Figure 10). Xception is a CNN architecture and pre-trained models on top of ImageNet. It's a high performing and efficient network compared to other pre-trained networks. The model script is written in Keras, a high-level python package that uses Google's Tensorflow library as a backend.

[3] "Xception: Deep Learning with Depthwise Separable ...." https://arxiv.org/abs/1610.02357. Accessed 27 Jan. 2021.

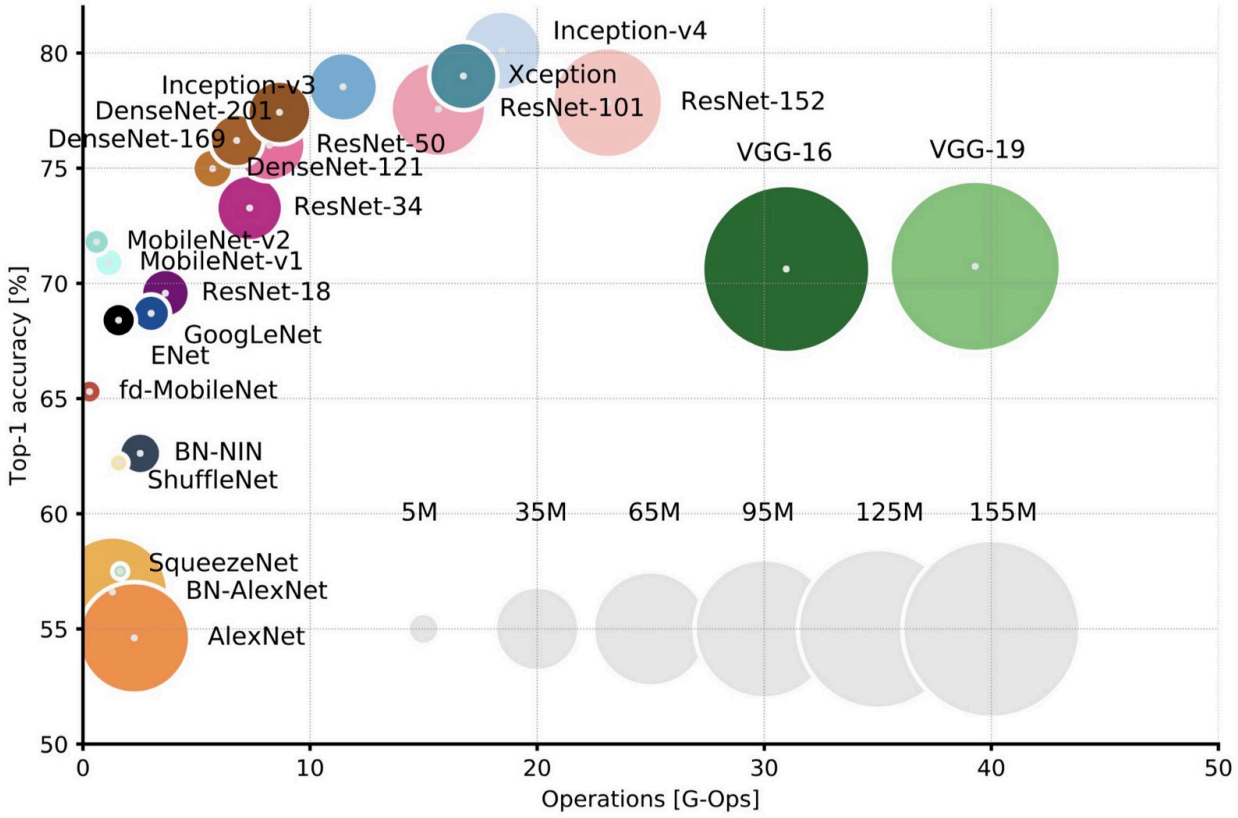


Figure 10. Current well-known CNN model architectures and pretrained model in deep learning. Xception model is one of the most performing and lightweight models

AIAIA Classifier takes binary classes, yes and no (or 1 and 0), image chips as training dataset in TFRecords [4] format. Image chip "Yes" or "Object" either has wildlife, human activities or livestock or their combinations. "No" or "Not-object" chips are the "empty" image without any interesting objects. The TFRecords have 7000 "Object" and 7000 "Not-object" image chips, that have been split into 70:20:10 proportions as "train", "validation" and "test" dataset. The models were trained with Sigmoid Focal Loss[5].Focal loss is extremely useful for classification when there is heavy class imbalance. In our case, there were many more pixels that did not belong to objects than those that did. There was also substantial class imbalance between object classes in all three object detection models. .

[4] "TFRecord and tf.train.Example | TensorFlow Core." 19 Sep. 2020, https://www.tensorflow.org/tutorials/load_data/tfrecord. Accessed 27 Jan. 2021.
[5] "tfa.losses.sigmoid_focal_crossentropy | TensorFlow Addons." https://www.tensorflow.org/addons/api_docs/python/tfa/losses/sigmoid_focal_crossentropy. Accessed 27 Jan. 2021.

The AIAIA Classifier model scripts were containerized and registried on Azure ARC (GCR). We then deployed Kubeflow to the cloud environment, once it's up running on AKS (or GKE), TFJob model experiments (can be deployed to GPU machines on AKS to start the model training (Figure 9). Model evaluation is performed to select the best performing trained models from multiple experiments.

AIAIA Detectors

We usedTensorFlow's Object Detection API to train object detection models for this task. Object detection models take an image as input and generate bounding boxes, predicted classes, and confidence scores for each prediction. Using the TFRecords training data, we trained a model of wildlife, human activities and livestock on GCP and Azure with Kubeflow (Figure 7). The Kubeflow is a tool that makes ML workflows on Kubernetes to be deployed easier, simpler, portable and scalable.

The final AIAIA Detectors are designed to predict: 1) Nine different classes of wildlife species and their counts; 2) Five classes of human activities and their counts; 3) three classes of livestock and their counts in Tanzania. For training classes and its count, see Figure 5.

The backbone model of the detector we used is Faster RCNN ResNet101[6] that pre-trained with Snapshot Serengeti Dataset [7]. Snapshot Serengeti Dataset contains approximately 2.56 million sequences of camera trap images, totaling 7.1 million images from Snapshot Serengeti project[8]. The model was scripted and trained with the Tensorflow 1.15 Object Detection API. Before we adopted Faster RCNN ResNet 101, we tried SSD MobileNet, ResNet 50 and ResNet 101. These models did not converge. Model training sessions were observed using TensorBoard (Figure 11).

[6] "models/tf1_detection_zoo.md at master · tensorflow/models · GitHub." https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md. Accessed 28 Jan. 2021.
[7] "Snapshot Serengeti - LILA BC." 24 Jun. 2019, http://lila.science/datasets/snapshot-serengeti. Accessed 28 Jan. 2021.
[8] "Snapshot Serengeti — Zooniverse." https://www.snapshotserengeti.org/. Accessed 28 Jan. 2021.

Figure 11. Tensorboard model evaluation tracking. The are 9 pairs of images shown above on Tensorboard. Ground truth image sits on the right and detection on the left for each pair during Faster RCNN ResNet101 training. From the Tensorboard by comparing the ground truth (right) and prediction (left) you will see missing labels on ground truth can lead to missing predictions.

Results and Discussion

AIAIA Classifier

Model performance

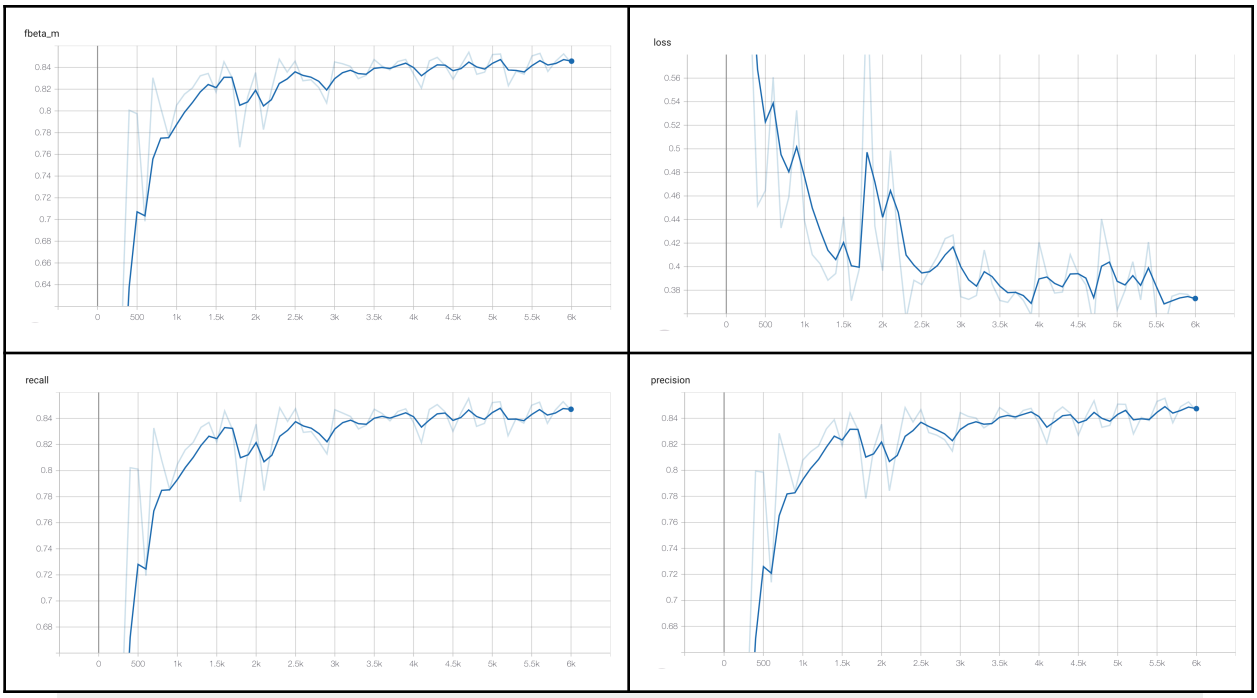
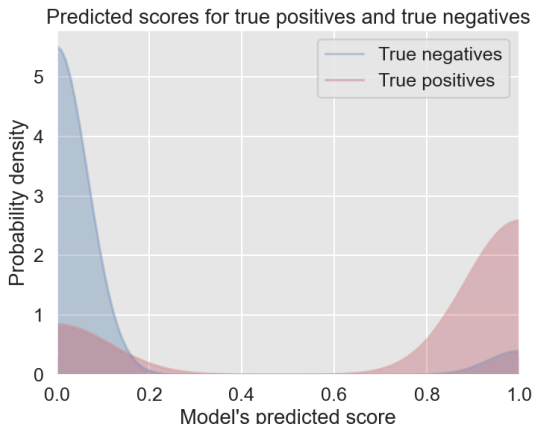


Figure 12. The model evaluation metrics for AIAIA Classifier that include model loss, F1, recall, and precision scores. F1, recall, and precision scores are above 0.84 .

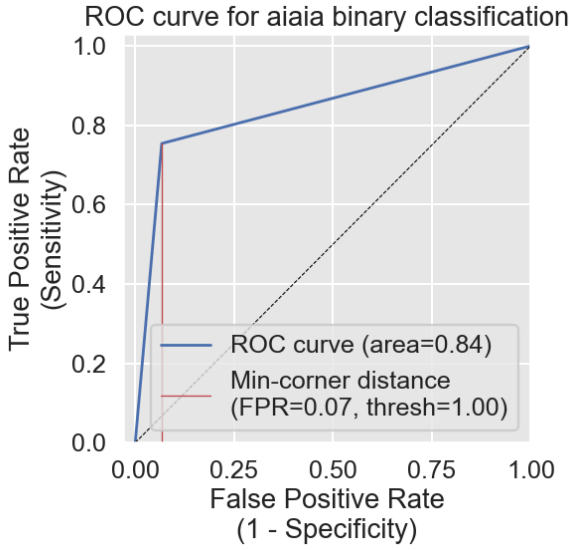
The AIAIA Classifier model is trained for 6000 steps that lasted about 15 hours 15mins on a single NVIDIA K80 GPU. other default evaluation metrics from Tensorflow and Keras. Model performance stabilized after 4000 steps (10 hours) that we started to witness model F1-beta, recall and precision scores climbed up to 0.84 over validation dataset, while model convergence moved down to 0.35 (Figure 12 and also see the online Tensorboard).

The binary classification model training and experiment can be tracked through our online Tensorboard here. Tensorboard[9] is a visualization toolkit for Tensorflow models that engineers or data scientists can track and visualize model evaluation metrics such a s loss, accuracy and other customized metrics. For instance, we design F1-beta, recall and precision scores as well as use Sigmoid Focal Loss instead of default metrics provided by TensorFlow and Keras.

[9] "TensorBoard | TensorFlow." https://www.tensorflow.org/tensorboard. Accessed 28 Jan. 2021.



A. True positive and negative score of the AIAIA Classifier.



B. The ROC curve for the AIAIA binary classification

Figure 13. Model evaluation for the AIAIA Classifier, a binary image classification to distinguish if an image chip contains objects of interest.

Each aerial image (6016 x 4000 pixels) were gridded into 150 chips. The overall classifier model performance can be summarized as:

$$Precision = \frac{Count\ of\ True\ Positives}{Count\ of\ True\ Positives + Count\ of\ False\ Positives}$$

$$Recall = \frac{Count\ of\ True\ Positives}{Count\ of\ True\ Positives + Count\ of\ False\ Negatives}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Classes	Precision	Recall	F1 score
Not-object (0)	0.79	0.93	0.9
Object (1)	0.92	0.75	0.78

Table 2. The overall model performance metrics for the AIAIA Classifier.

In general 0.5 confidence score is used to determine a prediction is a true positive or false positive. With the overall model performance metrics (Table 2) and model evaluation (Figure 13), there are a few conclusions we can draw from our AIAIA Classifier metrics:

- When the "Object" class confidence score threshold is set higher we can better distinguish between True Positive and True Negative (Figure 12. A);
- When the confidence score of "Object" class is set to 1.0, we are only going to have 0.7% of False Positive rate (Figure 12. B);
- The recall for the "Object" class is much lower than the "Not-object" class, though the precision scores between them are reversed. This implies the model produced a lot more False Negatives (only ground truth without detection). This usually means the training data quality of the "Object" class is still pretty lower.

Model Inference

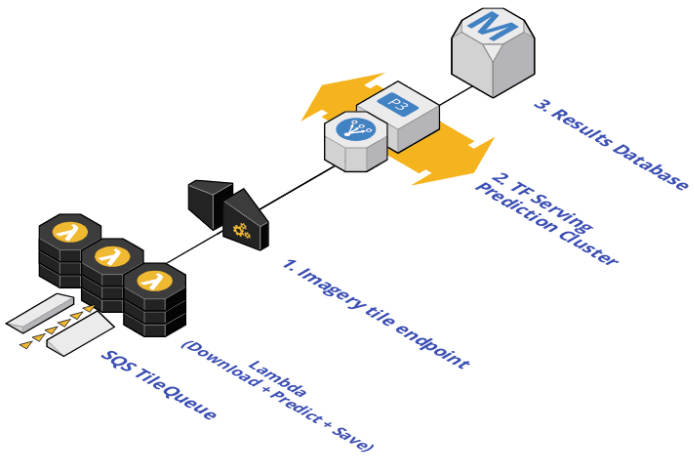


Figure 13. chip-n-scale-queue-arranger helps you run containerized machine learning models over images at scale. It is a collection of AWS CloudFormation templates deployed by kes, lambda functions, and utility scripts for monitoring and managing the project.

During the model inference, we scanned **5,506,337 image chips** with an average speed of **12,000 image chips per minute**, and finished inference in 7.6 hours. The inference was run with Chip n Scale, Development Seed's open-sourced model inference tool, Chip n Scale (Figure 13.)[10]. You can find the Lambda function here. At the end of the inference, 150, 141 image chips were filtered as containing "objects of interest" from 5.5 millions image chips, which is only 2.7% of the original image size.

Discussion and Conclusion

In the past, a survey in Tsavo National Park in Kenya collected 81,000 images (12.15 million image chips in 400 x 400 pixel). It took 7 months to count and validate all the objects of interest with a team of 8 human annotators. A total 3600 hours were spent, which means human annotators can validate and count objects at a speed of **56 image chips per minute**. However, the speed of model inference of the AIAIA Classifier is at 12, 000 image chips per minute[11]. It means the AI-assisted image chips scan and filter the image chips and have objects of interest fast and cost-efficiently, which will eventually be cost-saving in the long run.

The AIAIA Classifier is not perfect, because of the training data quality issues we mentioned above. The classifier model performance is highly impacted by the missing labels that can designate an image chip as "Not-object" when it is actually "Object". With improvements to the training label quality, the AIAIA classifier will be even more promising as a tool to quickly scan images after the aerial survey.

AIAIA Detectors

[10] "developmentseed/chip-n-scale-queue-arranger: Chip 'n ... - GitHub." https://github.com/developmentseed/chip-n-scale-queue-arranger. Accessed 28 Jan. 2021.
[11] Relative speed for the classifier is thus 214x faster. However, given that humans in lab settings typically do around 6 hours of concentrated work in a day, but the computers can work 24, the actual rate is 4 x 214 = 856x faster for larger datasets.

Model performance

Each **AIAIA Detector** was evaluated by sorting model results into four categories:

1. True positives, where a wildlife object received the correct bounding box and class from the model
2. Misidentifications, where the bounding box was correct but the class was incorrect
3. "False Positives, detection only" where a detection was made without ground truth
4. And "False Negative, groundtruth only" where no detection was made where a ground truthed object existed.

The criteria for a detection to be a true positive or misidentification was that the intersection-over-union (IOU) of their bounding boxes had to be greater than or equal to .5. In cases where multiple detection bounding boxes overlap a ground truth box, the detection with the higher confidence score was chosen to be a true positive and the other was deemed either a false positive or other category if it overlapped a different ground truth box.

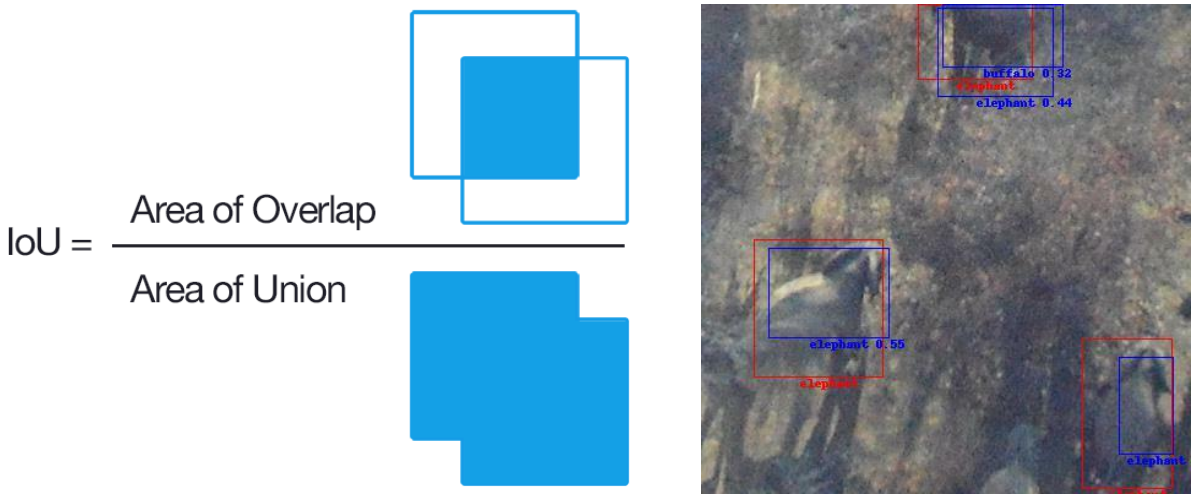


Figure 14. The equation for Intersection-over-Union is on the right. IoU is used to decide if detections overlap groundtruth enough to be counted as a true positive or misidentification. In the image on the left, an elephant is counted as a true positive if the confidence threshold is above .5, since IoU is also greater than .5.

These results were compiled into two confusion matrices, one in units of proportion of predicted positives for each class, and another matrix in units of absolute counts. Each

matrix shows information about detections for each combination of model categories, including true positives, misidentifications, false negatives and false positives. True positives appear along the diagonal of each matrix. The bottom row of each matrix ("false negative, groundtruth only") shows information about the number of detections where there was no corresponding ground truth of any class. The rightmost column of each matrix shows information about the number of groundtruth without corresponding detections of any class ("false positive, only detection"). Row values in the proportion matrix sum to 1, and **the row values in this matrix should only be compared along one individual row at a time, not between rows** since proportions are computed for classes with different sample sizes.

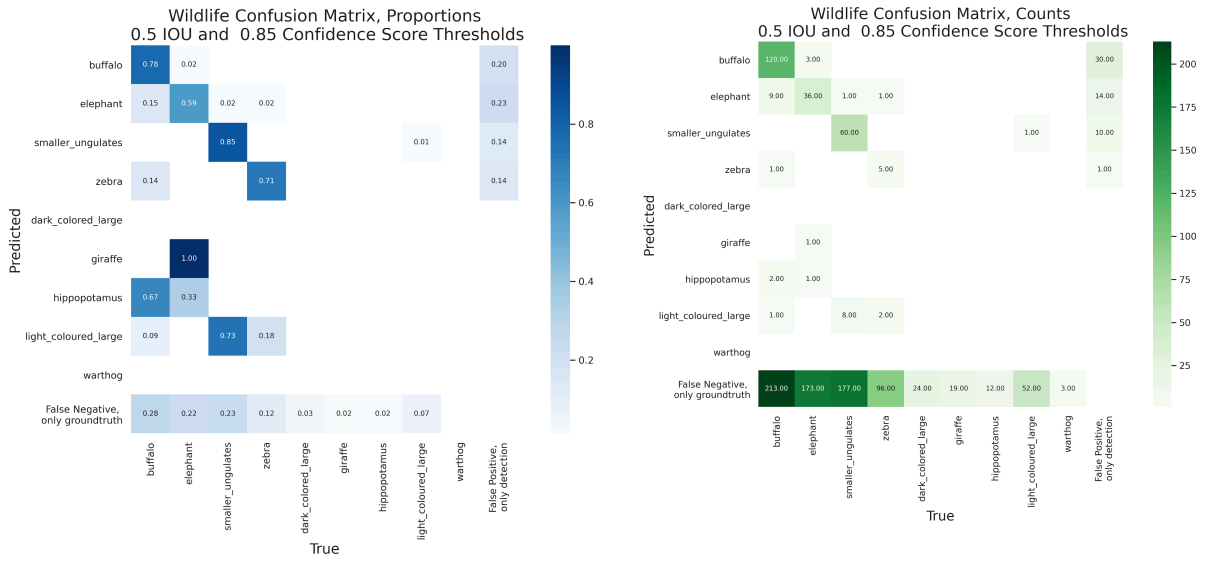


Figure 15. Proportion Confusion Matrix for the AIAIA Wildlife Detector, with values normalized by row totals, and the Count Confusion Matrix, where a "1" indicates that 1 object was predicted in the row class and was annotated with a column class..

- The wildlife model performance at a .85 confidence threshold showed strong majorities of correct, positive predictions for the most represented four wildlife classes: 78% for buffalo, 59% for elephant, 85% for smaller ungulates and 71% for Zebra.
- There was some confusion between the buffalo and elephant classes. 15% of predicted elephants were annotated as buffalo. 2% of predicted buffalo were annotated as "elephant".

- Smaller Ungulates had the smallest proportion of predicted positives as false positives with no corresponding groundtruth, at 14%.
- For the three most represented classes, there were more “false negative, groundtruth only” samples than true positives.
- The elephant class had almost five times as many “false negative, groundtruth only” as there were true positives. Smaller ungulates had almost three times as many missed false negative detections relative to true positives.
- Aside from these three well represented classes and the zebra class, there were no true positives. Most other samples from other classes fell under the “false negative, groundtruth only” column.

Category	Precision @ 0.5 IOU and .85 Confidence or Higher	Recall @ 0.5 IOU and .85 Confidence or Higher	F1 Score @ 0.5 IOU and .85 Confidence or Higher
Buffalo	.78	.35	.48
Dark Coloured Large	0	0	0
Elephant	.59	.17	.26
Giraffe	0	0	0
Hippopotamus	0	0	0
Light Coloured Large	0	0	0
Smaller Ungulates	.85	.24	.38
Warthog	0	0	0
Zebra	.71	.05	.09

Table 3. Metrics for the AIAIA Wildlife Detector

- Of the top three classes, the Elephant and Smaller Ungulates classes had a notably lower recall compared to precision. The Buffalo class had more even performance in terms of false positives and false negatives.

- Giraffe, Light Coloured Large, Smaller Ungulates, and Warthog did not achieve any true positive detections and therefore received a score of zero for each of the metrics.
- Dark coloured large received one true positive and most zebra ground truth went undetected, causing low scores for these classes .
- Buffalo, Smaller Ungulates and Elephants were generally the best performing classes and were also the most represented in the test and training sets .

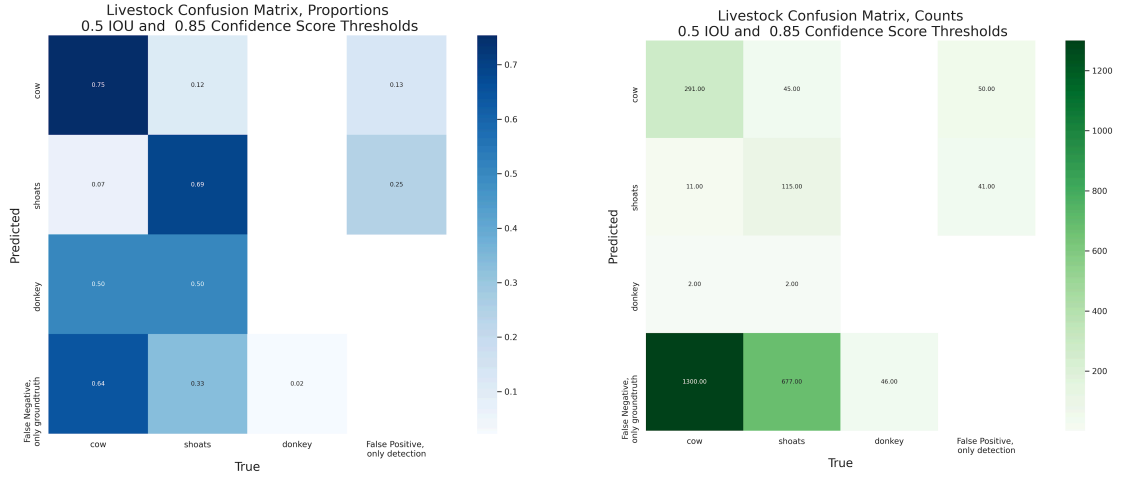


Figure 16. Proportion Confusion Matrix for the AIAIA Livestock Detector, with values normalized by row totals, and the Count Confusion Matrix.

- The livestock model performance showed a strong majority of positive predictions of the two dominant classes were correct, with 75% for cows and 69% for shoats.
- There was some confusion between the cow and shoats classes. 12% of predicted cows were annotated as “shoat”. 7% of predicted shoats were annotated as “cow”.
- Cows had the smallest proportion of predicted positives as false positives with no corresponding groundtruth, at 13%.
- Performance for both shoats and cows experienced many more “False Negative, only groundtruth” than true positives.
- For the cow class, there were 50 “False Positive, only detection” compared to 291 true positives. The shoats class had a higher number of “False Positive, only detection” (41) relative to true positives (45) than the cow class.

Category	Precision @ 0.5 IOU and .5 Confidence or Higher	Recall @ 0.5 IOU and .5 Confidence or Higher	F1 Score @ 0.5 IOU and .5 Confidence or Higher
Cow	.75	.18	.29
Donkey	0	0	0
Shoats	.69	.14	.23

Table 4. Metrics for the AIAIA Livestock Detector.

- In our test set, the Donkey class did not achieve any true positive detections and therefore received a score of zero for each of the metrics.
- The Cow and Shoats classes had a notably lower recall compared to precision, meaning there were a higher proportion of false negatives for these classes than false positives.
- The cow class had substantially higher precision than shoats, which resulted in a higher F1 score.

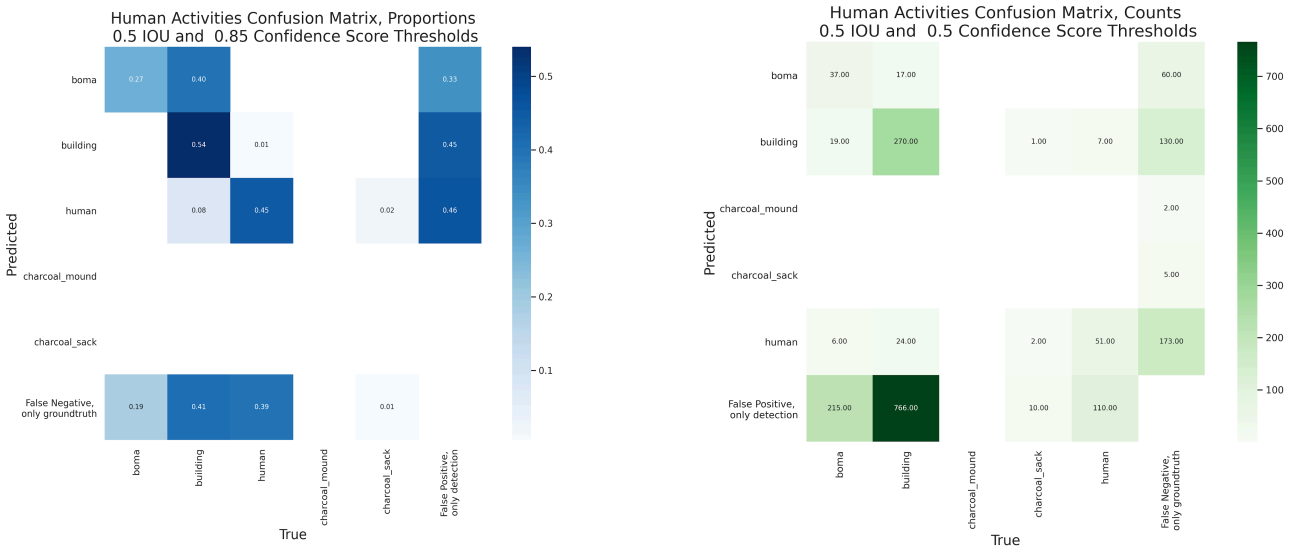


Figure 17. Proportion Confusion Matrix for the AIAIA Human Activities Detector, with values normalized by row totals, and the Count Confusion Matrix.

- The human activities model performance showed varied performance across the most represented classes (boma, building, human).

- 54% of positive building predictions were correct for buildings, 45% for humans, and 27% for boma.
- Predicted boma was confused with ground truth building more often than boma was correctly predicted (40% vs 27%). However there were a lack of positive boma samples to make this statistic robust.
- There were considerable false positives as a percentage of all positives for boma, building, human (>30%).
- The counts for the human activities model showed higher "false negative, only groundtruth" for the boma, building, and human classes.
- No charcoal mound or charcoal sack samples were correctly detected or misidentified, and there were few samples available in the training and test sets.

Category	Precision @ 0.5 IOU	Recall @ 0.5 IOU	F1 Score @ 0.5 IOU
Boma	.27	.04	.06
Building	.54	.45	.49
Charcoal mound	0	0	0
Charcoal sack	0	0	0
Human	.45	.12	.18

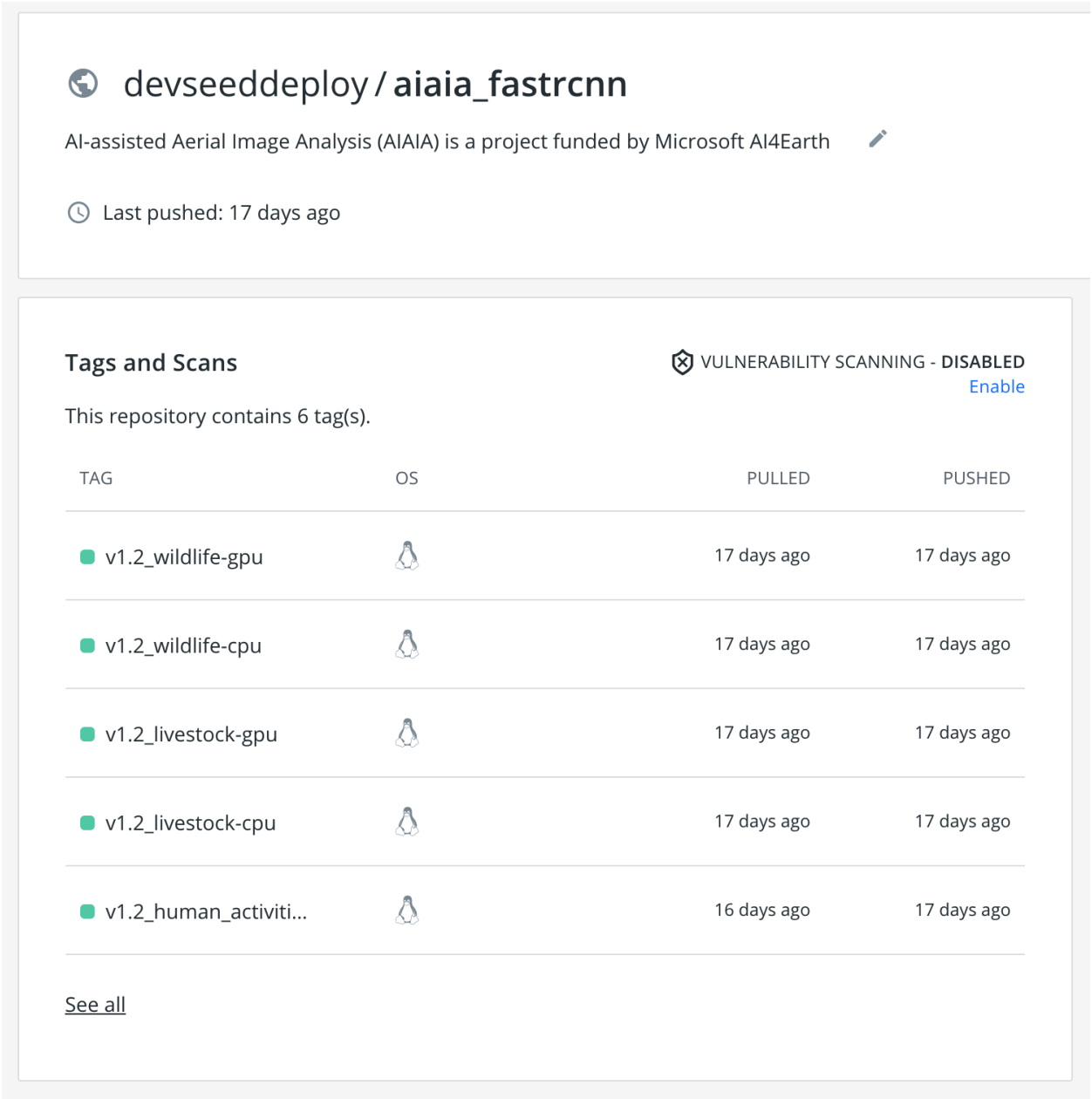
Table 5. Metrics for the AIAIA Human Activities Object Detector.

- The Boma, Human, and Building classes were the best performing classes and were also the most represented in the test and training sets for this model.
- Of the top three classes, the Building class had a notably higher recall and F1 score compared to the Boma and Human classes. The Boma and Human classes both had very low Recall but comparable precision scores to the Building class.
- In our test set, Charcoal mound and Charcoal sack did not achieve true positive detections and received a score of zero for each of the metrics.

Model inference

Once the model training session finished for each detector, we containerized the models as TFServing images and uploaded them to Development Seed's DockerHub. These images

are open and available for anyone to download them. They will run anywhere Docker runs, which makes them usable across all cloud environments and any modern computer. The steps to download and use these TF Serving images are listed within the documentation on each DockerHub page, and each model has a GPU and a CPU version. However, Fast-RCNN ResNet 101 is a heavy backbone model. We recommend the GPU version of TFServing images for model inference.



Please note: because the TFServing images ran at a speed of 14 second/image on a single NVIDIA K80 GPU, we ended up running model inference on an equivalent GPU by loading the exported model's frozen inference graph directly. The inference speed using the frozen graph was at 0.35 second/image or 172 images per min. Inference for three detectors was done parallely on three GPUs and finished within 15 hours, processing 150,141 images.

Figure 18. Users can run model inference of the detectors with an exported 'frozen inference graph'. For scripts and documentation please visit our [github repository on TZCRC](#).

Discussion and Conclusions

Our results show that the AIAIA Detector had some success when it came to precision and less success when it came to recall. **Lower recall scores imply poor training data quality.** Buffalo, Smaller Ungulates, and Cows had precision scores greater than or equal to 75%, meaning that users of these models can have some confidence that a positive prediction for any of these classes is in fact the correct class. If it is not a correct class, in some cases the prediction is a similar looking class. Next, other classes that were well represented but had lower precision scores were the Shoats (69%), Elephant (59%), and Building (54%) classes. This means that users of these models should be more cautious when using these detections to distinguish true positives from false positives. Nevertheless,

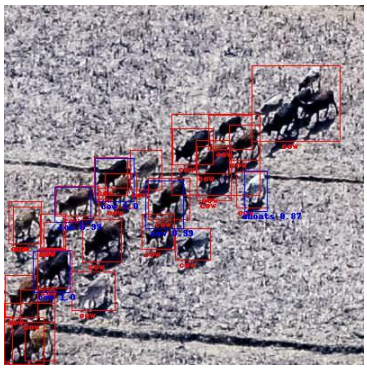
these predictions still help to filter down potential locations where wildlife, human activities, and livestock coincide, allowing for an analysis of HWP.

While users can gain useful, high confidence predictions from the model, these predictions will not consist of a full census of potentially detectable wildlife, human activities, or livestock. The model had more issues with false negatives than false positives, and we expect this is primarily due to training data error. Examples of this include duplicate ground truth labels, missing groundtruth, and cases where a single box was drawn around a crowd of objects. The ground truth shows as red color boxes and text and the predictions are in blue.

Duplicate Labels

Missing Labels

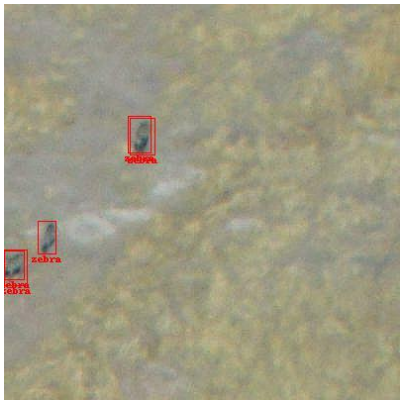
Crowd Labeling



These errors in the annotations both skewed the training of the model and skewed the evaluation process, since the model's correct detections would not be evaluated properly with these incorrect labels. We also observed negatives due to very blurry images and images where the animals were far away. The objects in these images look very different from cleaner images.

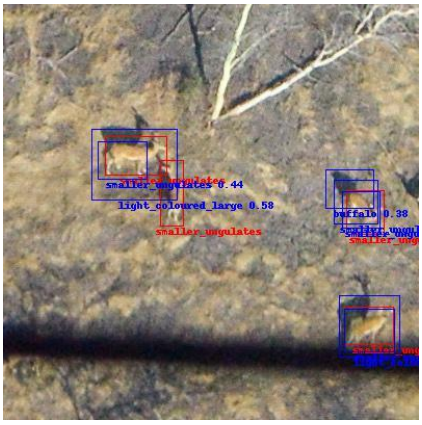
High Image Quality Zebras

Poor Image Quality Zebras



We also manually observed actual false positives in our testing set, particularly in images with shadows and blurry images.

Duplicate Detection of Shadow



Blurry False Positive

These aspects of training image quality impacted both precision and recall. In many cases, image quality was poor enough such that subcategories of wildlife could not be distinguished and so they were potentially misannotated. Future iterations of this model could benefit from using coarser class hierarchies or simpler class hierarchies with fewer classes. Particularly for the wildlife model, classes like dark coloured large and light coloured large could be merged into the smaller ungulate class, since these two classes appeared to exhibit similar image features from manual inspection. Light coloured large

was also confused with smaller ungulates more often than there were correct predictions, and dark coloured large had no predictions (false or true) at all. A simpler class hierarchy would enable the AIAIA object detection models to train on a less complex classification problem by not attempting to separate classes with too few samples, inconsistent labels, and compromised image quality. It would also make the annotation task less intensive, leading to higher quality labels. Overall, improving the quality of annotations is the most important way to improve model performance and confidence in the assessment of model performance.

One way to quickly improve training data quality is to use a human-in-the-loop approach, where a machine learning model assists a human annotator by filtering down the images they need to annotate and/or making predictions that a human annotator can verify and edit more quickly. For example, using this existing set of AIAIA object detection models, human annotators can work with a set of predictions made for some of the less common classes, like zebra, edit these predictions, and correct inaccurate predictions. Then, future models trained on this improved dataset will suffer less from class imbalance during training and testing. The AIAIA image classifier can also be used to preselect image chips with a higher likelihood of containing an object of interest so that annotators spend more useful time annotating images and less time sifting through images without objects of interest.

Core Challenges and Setbacks

This project faced a number of significant challenges, not only from the ongoing worldwide pandemic, but also some technical issues in implementation. The challenges cover issues from logistical problems resulting from the pandemic, data creation and sharing, data quality, training data class imbalance, model training and experiments, and model inference speed. Some were expected (logistical problems and the issue of small targets within large images), but the pandemic led to poor communications as people adapted to a work-from-home mentality, which caused difficulty delivering training data labels from TZCRC Annotation Lab.

Logistical Problems from the COVID19 Global Pandemic

1. The lab setup was delayed as the lab was opened around the time that pandemic measures came into play;
2. Supervision of the annotators was extremely difficult as the lab was opened around the time that pandemic measures came into play, and the local project manager was unable to spend significant time during the main part of the annotation work.

Dataset Creation and Sharing

1. Setting up a new labeling tool, CVAT, and adding labeling tasks for first time volunteers was a learning curve, along with the logistical problems caused by the pandemic.
2. The objects that appear in aerial images are small. The complex image background, variable image lighting, shading, and imaging angles added complexity to labelling tasks even though the annotators are wildlife domain experts.
3. Aerial surveys were cancelled or delayed by partner agencies. An expected pipeline of regular high-resolution images was not available for use during the project.
4. As identified in the proposal stage, aerial imagery presents several challenges for ML development.
    a. Aerial survey photography must cover wide strips (around 150m), and even with relatively high-resolution cameras (25 MP) target animals are often 20-40 pixels across, or less.
    b. Backgrounds vary dramatically depending on the habitat, time of day and even seasonal changes.
    c. The oblique imagery captured in PAS allows the observation of animals under canopy and for better ID of species - however, animal postures in oblique images vary considerably more than top-down images.
5. Bandwidth limitations delayed in image delivery to Development Seed from Tanzania. Though the lab space at the Centre for Research Cooperation was supported by a local ISP, the daily and monthly data caps were rapidly exceeded. The available speed (10 megabit at best, typically much less) meant that images were not uploaded for weeks.

Model Output Validation

1. Training data quality and error define the model performance and output quality. Currently model outputs from both the AIAIA Classifier and Detectors still need human validation.
2. At least two to three iterations of human-in-the-loop feedback to correct training data error, classifier and detector outputs' validation are expected in the following workflow, and each iteration should be followed by model retaining and evaluation until model performance is stabilized.

References

1. Lamprey, R. *et al.* Comparing an automated high-definition oblique camera system to rear-seat-observers in a wildlife survey in Tsavo, Kenya: Taking multi-species aerial counts to the next level. *Biological Conservation* 108243 (2019) doi:10.1016/j.biocon.2019.108243.
2. Frederick, H., Plumptre, A. & Moyer, D. *Aerial Procedures Manual*. (Wildlife Conservation Society, 2010).
3. World Bank. Population growth (annual %) | Data. https://data.worldbank.org/indicator/SP.POP.GROW (2021).
4. Songorwa, A. N. Human population increase and wildlife conservation in Tanzania: are the wildlife managers addressing the problem or treating symptoms? *African Journal of Environmental Assessment and Management* **9**, 49–77 (2004).
5. Grande, J. M., Zuluaga, S. & Marchini, S. Casualties of human-wildlife conflict. *Science* **360**, 1309–1309 (2018).
6. Thouless, C. *et al.* African elephant status report 2016. *An update from the African Elephant Database* (2016).
7. Wolanski, E., Gereta, E., Borner, M. & Mduma, S. Water, migration and the Serengeti ecosystem. *American scientist* **87**, (1999).
8. Norton-Griffiths, M. *Counting Animals*. (African Wildlife Foundation, 1978).
9. Craig, C. C. *Aerial survey standards for the MIKE programme*. (CITES MIKE Programme, 2003).
10. Frederick, H. Github.com/*TZCRC/Lanner-CamPod*. (TZCRC, 2021).
11. Github.com/*openvinotoolkit/cvat*. (OpenVINO Toolkit, 2021).