Michael Wieck-Sosa
Professor Lyons
Artificial Intelligence
December 11, 2019

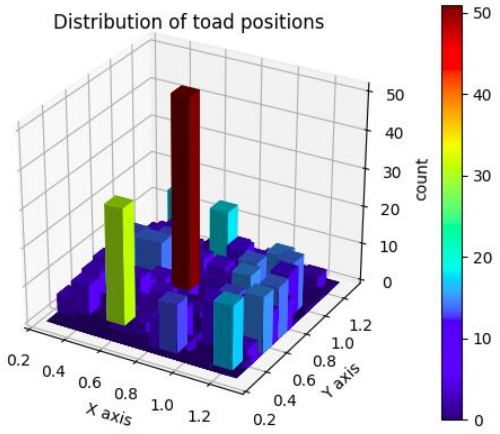
Artificial Intelligence Project: Toad Behavior Analysis

The Kihansi spray toad, *Nectophrynoides asperginis*, is a small toad endemic to a small area at the base of the Kihansi River waterfall in Tanzania. In May 2009, the International Union for Conservation of Nature declared the Kihansi spray toad extinct in the wild, largely due to the construction of the Kihansi Dam in 1999. Conservation efforts over the last 2 decades have allowed for the reintroduction of the Kihansi spray toad into the wild. To support these efforts, the Fordham Robotics and Computer Vision lab has developed a tracking software for the toad.

For this project, I have made two contributions to the software. First, I have developed a method to track the toads using the YOLOv3 model for object recognition. This model was chosen because of its relatively fast training and efficient use of data compared to other open source object detectors. While the preliminary results are promising, the downside is its relatively high computational cost compared to tracking methods that do not use deep learning. Nevertheless, by using GPU-equipped clusters, this has proven to not be a problem. Moreover, the significant improvements in accuracy are necessary for the next step, detecting toad push-ups. Second, I have developed a method for classifying toad push-ups using a 3D convolutional neural network for video classification. So far, this has proved challenging because of a lack of toad push-up observations. After going through a dozen videos, only one proper toad-pushup has been observed. Later, I will discuss approaches to get around this lack of data. In the following discussion, I will give an overview of these two contributions.

First I will discuss the data collection. The data for the YOLOv3 object recognition were collected in the following manner. One video, 05-12-2018---14-07-17.bag, was used for data collection. In the video, there are a large number of images with most toads not moving from frame to frame. Also, in many frame sequences, only one toad moves at a time. Therefore, to minimize the manual labeling needed for satisfactory results of the object recognition model, 1,000 individual toads in the fifteen minute video were labeled. The number 1,000 was chosen because the first iteration of AlexNet was trained with 1,000 images per object class which achieved satisfactory results.

The selection of a toad for labeling is split into two steps. In step 1, each of the toads were represented about 5 times, giving us about 150 labels. In step 2, if a toad moved or jumped, it was labeled until 1000 labels were collected. If more than one toad moved in a frame sequence, the toad with a more unique body configuration, a less represented position in the terrarium, or a higher degree of obfuscation was prioritized. This approach was chosen because models train more efficiently when observations of objects that are different enough from each other, so the model learns more from each labeled toad observation. The average bounding box was about 25 x 25 pixels, and the following plot shows the distribution of toad positions with (X,Y) coordinates.



Next, frame 102 of the same video was used for a background image. The image was photoshopped such that all the toads were removed from the image. We assume that the model will that any learn to account for any irregularities in the image caused by the manual photoshopping. The YOLOv3 model was changed to detect one object class, but for the most part, the default YOLOv3 settings were maintained. The technical details of the custom model are in the configuration file. Afterwards, the labeled toad was pasted onto the background image with no toads using a script. The following sequence of images visualizes the process of assembling the dataset, from the original frame, to the background image with no toads, to the final image with one toad pasted on it.

Now that the dataset has been assembled, the YOLOv3 model was trained and evaluated. The cross-validation evaluation was done by training the network with 800 images and testing on 200 images, for 30 random splits. True positives were defined as the YOLOv3 model predicting X1, X2, Y1, Y2 coordinates of a bounding box within 25 pixels of the true labeled coordinate above a confidence threshold of 30%. True negatives were defined as no prediction above a 30% confidence threshold when there was no toad in the image. False positives were defined as any prediction above a confidence threshold of 30% when there was no toad in the image. False negatives were defined as the model predicting 25 pixels outside of the true coordinates of the bounding box above a 30% confidence threshold or not giving a prediction above a 30% confidence threshold, given that there was a toad in the image. The results for the model evaluation are given as a 95% confidence interval to give an idea of how all the models did. The sample mean is denoted by $\bar{x}$ and the standard deviation to achieve the desired 95% confidence interval is denoted by s. As a note, the model must be retrained with a balanced classes (1,000 images with a toad in image, 1,000 with no toad in image) to achieve better specificity.

Out of 200 test images:

TP 95% CI	**FP 95% CI**
$\bar{x}$: 186.6 s : 17.97	$\bar{x}$: 0.6 s : 0.37
TN 95% CI	**FN 95% CI**
$\bar{x}$: 0.0 s : 0.0	$\bar{x}$: 13.3 s : 17.89

Next, the YOLOv3 detector was compared with Phil's tracker were evaluated. As Phil's toad tracker takes in frame sequences, rather than individual frames, I used frame sequences from the video 05-12-2018---14-37-17.bag to compare the performance of the two models in tracking toads. In the same was as the training step explained above, 23 distinct sequences of one toad's movement over 50 to 100 frames were labeled, then a background image with all toads removed with photoshop was used to paste these toad sequences to the background image using a script. The true positive, true negative, false positive, and false negative were defined in the same way as the training step for the YOLOv3 model and Phil's tracker, besides the confidence threshold. These preliminary results should be taken with a grain of salt for two reasons. First, after creating the datasets for the sequences, I did not create an analogous background image for the depth frame. Instead, I just used the original depth frame, so the results reported for Phil's tracker may be overly pessimistic. Second, Phil's tracker makes multiple output files for each frame. To remedy this, I looped through each output file to see if there was a correct prediction, which may give overly optimistic results. Nevertheless, the YOLOv3 recognition shows very promising results. The highlighted columns of the image below are particularly important.

SEQ#	TP_FP_TN_FN_PHIL				TP_FP_TN_FN_YOLO				lastPred_PHIL	lastPred_YOLO	#TOADS
0	2	6	0	27	32	0	0	2	FN	TP	34
1	1	9	0	75	85	0	0	0	FN	TP	85
2	0	2	0	26	28	0	0	0	FN	TP	28
3	0	1	0	13	14	0	0	0	FN	TP	14
4	0	3	0	27	30	0	0	0	FN	TP	30
5	0	7	0	20	23	0	0	0	FP	TP	23
6	0	3	0	33	36	0	0	0	FN	TP	36
7	1	14	0	128	135	0	0	6	FP	TP	141
8	1	3	0	35	39	0	0	0	FN	TP	39
9	2	11	0	124	137	0	0	0	FN	TP	137
10	0	2	0	48	47	0	0	3	FN	TP	50
11	2	4	0	80	86	0	0	0	FN	TP	86
12	0	6	0	42	38	0	0	10	FN	TP	48
13	2	5	0	71	77	0	0	1	FN	TP	78
14	4	16	0	100	118	0	0	0	FN	TP	118
15	0	5	0	22	27	0	0	0	FN	TP	27
16	0	2	0	27	28	0	0	1	FN	TP	29
17	0	4	0	29	32	0	0	0	FN	TP	32
18	0	0	0	12	10	0	0	2	FN	TP	12
19	0	9	0	22	30	0	0	0	FP	TP	30
20	0	3	0	7	9	0	0	1	FN	TP	10
21	0	1	0	21	22	0	0	0	FN	TP	22
22	1	10	0	54	63	0	0	1	FN	TP	64

Afterwards, I used a method to carry out object tracking by object recognition. Using the bounding box predictions from the YOLOv3 model, I defined tracks as the minimum Euclidean distance between bounding boxes in two frames. If a particular toad's movement path is lost, then that unique track ID is stopped. Extensions can be made to detect jumps by a distance being greater than a certain threshold, and to detect meetings by having two tracks converge to one bounding box to track.

Finally, I began working on detecting toad push-ups using a 3D CNN architecture for video classification taken from GitHub. When creating a dataset of toad push-ups, I ran into a problem because of the lack of toad push-ups in the dozen videos I went through. After going to the zoo and speaking to a zookeeper in charge of the toads next week, I may be better equipped

to see toad push-ups I may have overlooked. Otherwise, this problem will need to be solved by taking more video or training a video classifier to detect push-ups using videos on the internet.