Optimizing Inference in Behavior Classification of Fishes

Manu Tej Sharma Arrojwala
Vineeth Aljapur
Priyam Raut
Sachin Sarath Y Kothandaraman

ABSTRACT

Malawi Cichlids are amongst the most striking organisms to study behavioral diversity, and thus can serve as a powerful system for understanding the biological basis of natural behaviors. Since analysis of these behavioral trials is associated with video monitoring, automated methods will be necessary to fully analyze the data. 3D Convolutional Neural Networks have been highly successful in applications such as video classification, and action detection, which are essential facets in scientific communities studying live models. However, utilizing 3D CNN models is a time consuming and computationally intensive process. The popularity of Convolutional Neural Network (CNN) models imply that better performance of CNN model inference can deliver significant gain to a large number of users. Our project aims to accelerate a 3D ResNet model used to detect fish behaviour, for faster inference with minimal loss in accuracy.

1 PROBLEM STATEMENT

One of the most characteristic behaviors displayed by cichlids is the construction of bowers which are sand structures made by males to attract females. They do this by building a sand castle, an elevated structure, or by digging a pit, a depression (Fig. 1a). Bowers are constructed by scooping sand from dispersed locations and spitting into a concentrated area to create a castle (Fig. 1b) or scooping sand from a concentrated area and spitting into dispersed locations to create a pit (Fig. 1c) [1]. Unlike most behaviors that are studied in the laboratory, this is repeated for days to weeks at a time during the mating season.

Typically, male cichlids build bowers by freely manipulating sand only in the presence of female cichlid(s). We study bower building in the lab by introducing male and female cichlids into a large sand tank and monitoring the fish using cameras connected to a raspberry pi computer which generates around 10 hours of video recordings daily for each tank. Traditionally, quantification of this social behavior is done manually where an experimenter has to look at entire recordings to find and label bower building events. This ensures that the accuracy of labeling remains high but it comes at the cost of looking at the entire video(s). Since each of our behavioral trials is associated with around 100 hours of video, automated methods will be necessary to fully analyze the data.

Convolutional neural networks with spatio-temporal 3D kernels (3D CNNs) have an ability to directly extract spatiotemporal features from videos for action recognition [2]. Residual networks (ResNets) are one of the most powerful architecture. ResNets[3] introduce shortcut connections that bypass a signal from one layer to the next. The connections pass through the gradient flows of networks from later layers to early layers, and ease the training of very deep networks. Applying the architecture of ResNets to 3D CNNs is expected to contribute further improvements of action recognition performance.

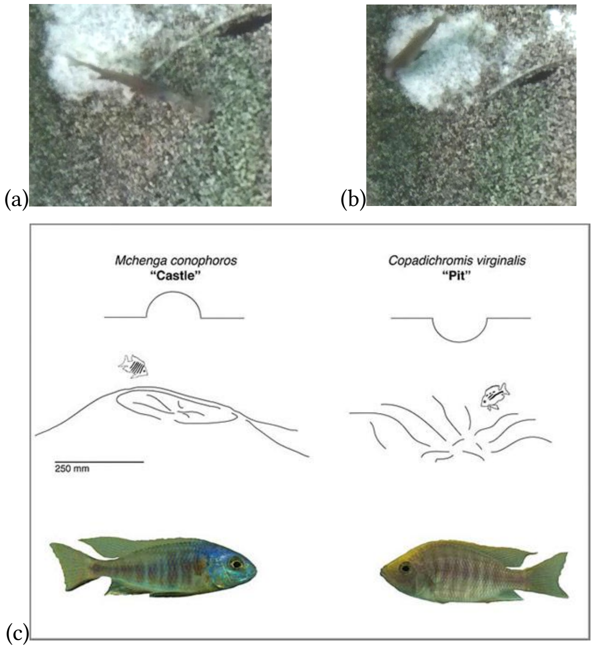


Figure 1: Bower building behavior of Cichlids: (a) spit by Mchenga conophoros (b) scoop by Mchenga conophorosa, (c) Castle building and pit digging in Mchenga conophoros and Copadichromis virginalis respectively

As a first step towards automation of this analysis, we used an out-of-box 3D CNN with ResNet-18 model[4]. The video data was labeled to indicated 10 behavioural classes. The model was trained with the resized data and later tested to determine the accuracy of the classification. However, we identified that using 3D ResNets were slow and the current model would not be optimal to analyze large scale video data. Action classification is slow and memory intensive because of 3D convolution (needs 6x Nvidia 980Ti). Another issue is since 50 % of the frames of our videos are empty, classifiying a set of continous set of empty frames is a waste of resources. Accuracy is not good as the model is not for our dataset. Hence, we intend to explore methods to optimize the speed of using the 3d ResNet model with minimal loss in accuracy of classification.

2 SIGNIFICANCE OF THE PROBLEM

Action classification plays a predominant role in biological research involving live models. Though existing 3D CNN models provide comparatively accurate classifications, they are computationally intensive and quite slow in their inference. Though it is possible to design new models that are optimal for classifying biologically significant behaviours in live models, We believe that identifying optimal ways to increase the speed of inference and reducing the computational load on the existing model with minimal loss of accuracy would be less intensive, and easier to utilize in biological research.

3 METHOD

(1) **Motion detection using openCV:**
The first step is to increase the speed of the process. We want to implement this by removing the empty frames in the video which we estimate to be 50 % of the total frames using OpenCV. For this we iterate over the frames and resize them, convert to gray-scale, compute the absolute difference between the current frame and static background frame. We dilate the difference to fill in holes, then find contours. We used thresholds for minimum area size of blob to remove empty frames.

(2) **Naive NoScope Implementation:** Implemented NoScope or NoScope like architecture to have a robust neural network with high recall to accelerate the removal of empty frames. Alexnet like shallow neural networks with fewer layers were implemented. Owing to the complexities of the image and relatively low number of images in the training dataset, we have also tested the performance of ImageNet pretrained neural networks as well (AlexNet, VGG11, ResNet50 and Squeezenet) on the frames extracted from target query video.

To this extent, we have used Faster-RCNN based object detection implemented on TensorFlow 1.14 to annotate 10,000 frames picked randomly across a video and these frames were used to train intermediary neural networks which filter out the frames. Models were implemented in pytorch and the code is available on the github repository. All the models were trained using 80% of the total images (n=8000) and validated on remaining 20% of the images (n=2000).

Each model was run for 15 epochs and final validation accuracy was reported. The model with highest validation accuracy was taken as the filter and the time to train the model didn't factor into this decision as most models took around 6 minutes with a standard deviation of 1 minute.

The model with high validation accuracy, whose inference speed was two magnitudes faster than that of the native Faster-RCNN based object detection, was used to annotate the whole video and the annotations were used to control whether the frames were being passed to Faster-RCNN object detector or not.

(3) We processed the data, created the annotation files for fish data and trained the model. We would also like to tweak 3D ResNet architecture and tune Hyperparameters to improve detection speed and accuracy by using different architectures[5].

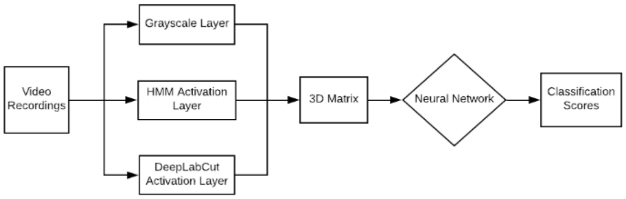


Figure 2: Flow Chart showing the implementation of depth sensing data to improve classification

(4) We have also created a 3 dimensional clip with RGB layers of Grayscale HMM Activation and DeepLabCut Activation respectively. We expect to see a boost in accuracy.

4 VALIDATION

We manually labelled 10,000 images randomly picked from a video clip. We checked whether a fish/fishes are present in a frame or not. We used this labelled dataset to validate our predictions from different filtering methods. Precision, Recall and Accuracy were reported.

5 PERFORMANCE EVALUATION

We evaluated the performance of the models by running a 11 hour video recording on object detector and getting the base run time of the analysis. Then we removed the empty frames using openCV and NoScope Squeezenet and re-ran the object detector on remaining frames.

Total time taken to train and predict= Time for picking 10000 frames + Time to train the squeezenet + Time to annotate the whole video + Time to run object detection on whole video * (frames with objects/total frames)

$$= 30+6+1188717/130*60+3927*(\ 0.5) = 2152min = 35hours52min$$

We compared the total time for detecting all frames of 11 hour video with total time taken to run the low level filters and object detector. Since we have already evaluated the precision and recall of the low level shallow filters and tuned the thresholds, we can ensure the accuracy of the prediction remains high while obtaining significant improvement in detection speed.

6 RESOURCES

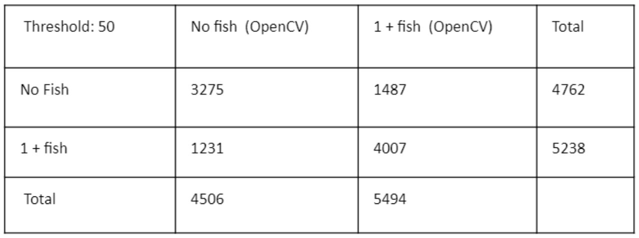
- Software
  - PyTorch
  - Tensorflow
  - Git
  - openCV
- Hardware
  - GPUs
- Data Sets
  - Custom Dataset of about 700 hr of video of Cichlids

7 GOALS

- 75 %: Adding low-level filter (OpenCV) to improve speed by removing empty frames
- 100 %: Implementing NoScope to remove empty frames by a cascade of shallow NNs
- 125 % To improve the accuracies for the 3D CNN by tuning the architecture and hyperparameters and adding an additional dimension of depth which is obtained by Kinect

8 RESULTS

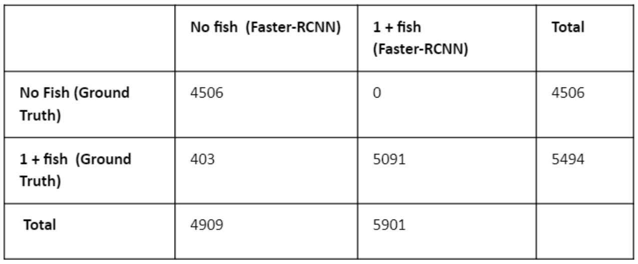
We used a Image Labeling script to classify the image dataset of 10,000 into positive and negative on the basis if we see one or more fish in it. We established this as our ground truth to compare the results of the various classifiers. For OpenCV, we used multiple thresholds for continuous pixels (10, 50, 100, 300) to remove empty frames. We saw an increase in accuracy and Recall with decreasing threshold. According to the confusion matrix of 10,000 images for threshold of we found that the number of true negatives was found to be 32.75 % where as the amount True Positive was 40.07 %.The processing time for the OpenCV is 130 frames per second. Since our classifier processing time of up of 0.19 seconds per frame, for 10,000 frames, there is a speed up of 622.5 (3275*0.19) seconds.



Threshold: 50	No fish (OpenCV)	1 + fish (OpenCV)	Total
No Fish	3275	1487	4762
1 + fish	1231	4007	5238
Total	4506	5494	

Precision: 0.76 Recall: 0.72 Accuracy: 0.73

Figure 3: Confusion Matrix of Open CV with Ground Truth

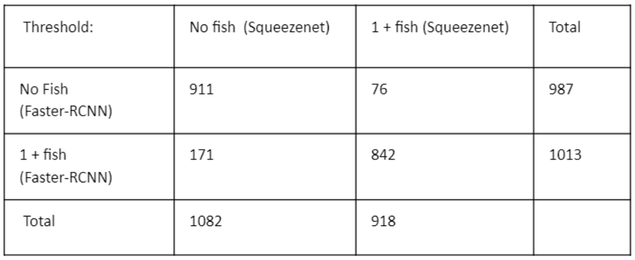


	No fish (Faster-RCNN)	**1 + fish (Faster-RCNN)**	**Total**
No Fish (Ground Truth)	4506	0	4506
1 + fish (Ground Truth)	403	5091	5494
Total	4909	5901	

Precision: 0.92 Recall: 0.83 Accuracy: 0.88

Figure 4: Confusion Matrix of Object Detector with Ground Truth

Since the original implementation was not working owing to ancient dependencies. We implemented Naive Noscope without a difference detector. It gave us an accuracy of 0.50. We also implemented, Squeezenet, Alexnet, ResNet 50 and VGG11. We found Squeezenet to have the highest accuracy of them all with an accuracy of 0.88



Threshold:	No fish (Squeezenet)	1 + fish (Squeezenet)	Total
No Fish (Faster-RCNN)	911	76	987
1 + fish (Faster-RCNN)	171	842	1013
Total	1082	918	

Precision: 0.92 Recall: 0.83 Accuracy:0.88

Figure 5: Confusion Matrix of Squeezenet with Ground Truth

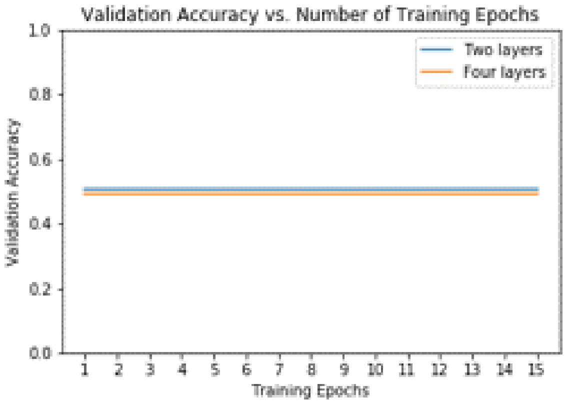


Figure 6: Validation for Shallow Neural Network

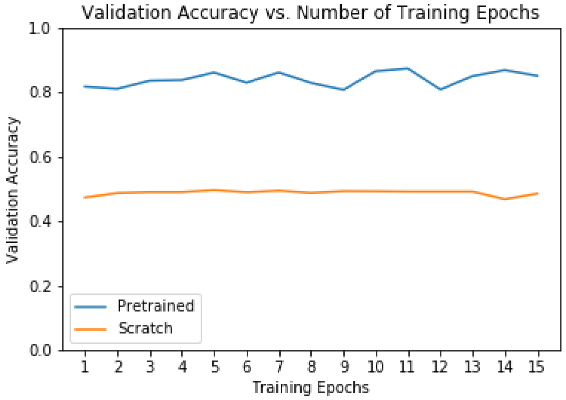


Figure 7: Validation for Squeezenet

9 TIMELINE

- We would like to Integrate our model with EVA in couple of weeks (November 15)
- We expect to complete tweaking the model in couple of weeks after integration with EVA (November 28)

REFERENCES

(1) Ryan A. York, Chinar Patil, et al., “Evolution of bower building in Lake Malawi cichlid fish: phylogeny, morphology, and behavior” Front. Ecol. Evol., (2015)

(2) J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. arXiv preprint, abs/1705.07750, 2017

(3) 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 (CVPR), pages 770–778, 2016.

(4) Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh, "Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition", Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition, 2017.

(5) Hara, Kensho, Hirokatsu Kataoka, and Yutaka Satoh. "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?." Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018

(6) Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." arXiv preprint arXiv:1602.07360 (2016).

(7) Yu, Wei, et al. "Visualizing and comparing AlexNet and VGG using deconvolutional layers." Proceedings of the 33 rd International Conference on Machine Learning. 2016.