ZebraSwish: A Tool for Locomotive Behavioral Analysis of Zebrafish

Shamya Karumbaiah
University of Massachusetts Amherst
shamya@cs.umass.edu

Rafael Lizarralde
University of Massachusetts Amherst
rezecib@cs.umass.edu

Abstract

Zebrafish are used to study the changes in the locomotive behavior after mutation. Most researchers go through a time-consuming and labor-intensive task of manually annotating the zebrafish videos. ZebraZoom is the state of the art tool used by some researchers to automate this process, but fails to accurately detect zebrafish in the presence of other visible objects. Prof Gerald Downes at UMass Amherst Biology department is an active researcher in this field and uses a tactile stimulus in his experiments, which is visible on captured footage. Downes lab has a repository of several high-speed videos of the zebrafish. In this project, we tried to use deep learning and image processing approaches to accurately detect the zebrafish.

Our first task was to annotate a set of video frames to locate a bounding box around the zebrafish. We then trained an end to end convolutional neural network model in Caffe. In our second approach, we fine-tuned the `imagenet-vgg-f` *model with the last layer of support vector machine classifier trained on the features of the zebrafish data. In an attempt to compare our deep learning models with an image processing approach, we did motion-based object tracking on the movements of the zebrafish and probe in the video. Our end-to-end model was unable to learn to detect the zebrafish. While the fine-tuned network was able to classify (zebrafish vs no zebrafish) with a 80% accuracy, its performance was dependent on the accuracy of the bounding box detection by optical flow. Motion-based object tracking was comparatively most accurate for fish in motion but had issues recovering the full body after the zebrafish tilts at right angle or curls up into a blob.*

1. Introduction

Professor Gerald Downes at UMass biology department studies zebrafish locomotive behavior by making mutations to genes and then observing changes in how they swim away from a mechanical stimulus, by capturing high-speed video and then tracking the angle between the body axis and the tail tip throughout the video. However, they currently use software that requires extensive hand-annotation and adjustment of labels throughout the video. The state of the art tool[7] for this task uses image processing technique for the detection task. The lab uses a probe for tactile stimulus and the presence of the probe (Figure 1) in the frame makes automatic fish detection difficult in the current tools. As a first step towards automating this process, we propose to use deep learning techniques to detect the zebrafish in the video.

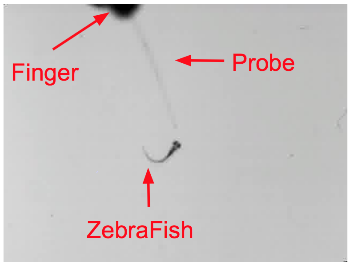

Figure 1. Example video frame showing the objects present.

The localized object detection task is, given an input image, find objects in the image, and output their location (typically in the form of a bounding box) and a class label (for example, ”dog”, ”chair”, ”zebrafish”). This can be applied frame-by-frame on video, although some models can also exploit the temporal dependencies in video to improve their results.

2. Background/Related Work

Zebrafish are widely studied as a model organism for vertebrates, as they are easy to raise and have a short generation time, facilitating genetic manipulation. Locomotive behavior in particular can be studied extremely early in the zebrafish life cycle, with tightly-controlled response to tactile stimuli only 48 hours after fertilization.[4] These responses are typically analyzed by plotting a graph of the tail-bending angle over time.[7] For example, a typical response to a tactile stimulus on the head involves a large-amplitude wave of tail bending, in which the larva changes direction, followed by a series of small waves with alternating amplitude, in

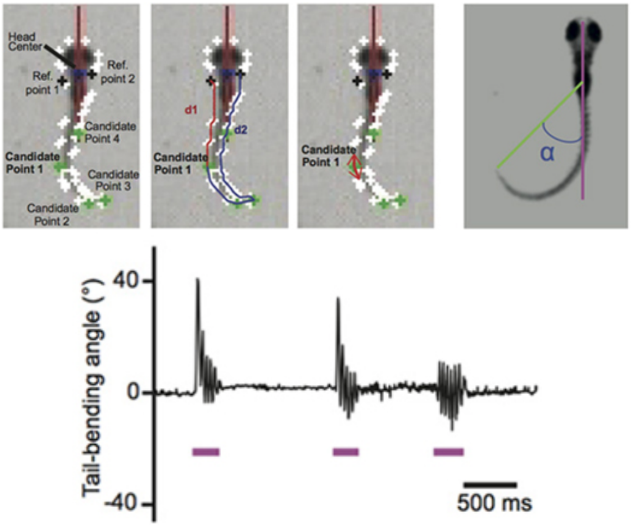

Figure 2. ZebraZoom tracking

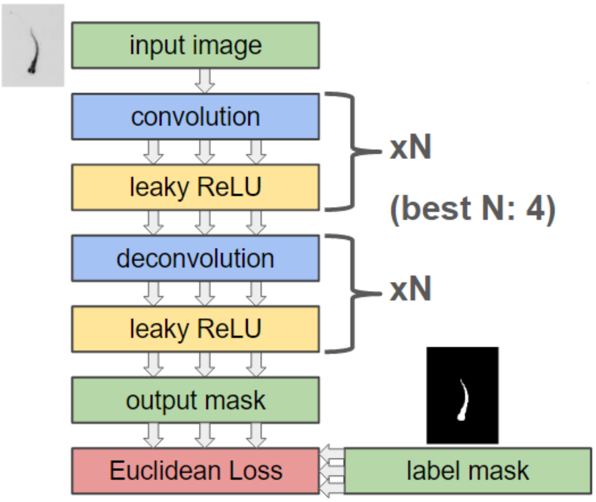

Figure 3. End-to-end convolutional architecture.

which the fish is propelling itself away from the stimulus (Figure 2). Research on genetic mutants that have impaired or abnormal reaction to stimuli shed light on the mechanics of biological neural circuits.[4] Currently, the best tool for automating this type of analysis is `ZebraZoom`,[7] which uses a background subtraction and thresholding technique to isolate the zebrafish, then uses information from the previous two frames to track the larva as it moves (Figure 2).

In terms of localized object detection, there has been a proliferation of approaches recently, most notably R-CNN and its descendants (Fast and Faster R-CNN).[9] These use a two-step approach, where first a model proposes regions to be examined, and then another model evaluates whether the region contains a particular object. In Faster R-CNN, the authors partially combine these two models by sharing convolutional features between them.

3. Approach

We tried three approaches to this problem with varied degrees of deep learning interventions.

3.1. End-to-end Models

We attempted to develop our own end-to-end model for detecting the location of the zebrafish in the image, developing networks in Caffe,[3] and running on a personal GPU (unfortunately the 2GB memory capacity of the GPU was a major limitation for model size). The general idea of the architecture is shown in Figure 3; first the input image (a frame from the video) is convolved over several layers (but not greatly reduced in dimension as would occur in a typical classification CNN). Then, the size of the image is restored by deconvolutional layers using the same parameters as the corresponding convolutional layer. The output of this section is then compared to the label, which is a mask of the image (Figure 6), and loss is calculated by Euclidean distance between the two images.

3.2. Fine tuning and Optical flow

This approach[8] uses `imagenet-vgg-f`,[11] a pretrained CNN model from *MatConvNet*. It is trained on the ImageNet dataset.[10] Running an input data point through this network until the last layer, produces the features that the network would have computed for its task. By extracting these features for our data, we can work on the features of the images instead of the raw pixels. It is expected to have a superior performance due to its ability to learn the latent representation of the image.

The training data for this approach is obtained by cropping the subimage of the zebrafish from the full image using the human annotations. We also collected some training images for the negative class as a set of random images from varied genre. An SVM classifier is then trained to identify zebrafish versus no zebrafish given an image frame. Since the classifier is trained on the zebrafish subimage, at test time, we need a bounding box detector to pick the region of interest. Optical flow[6] is used for object detection as it exploits the frame by frame pixel change in the video. The trained classifier can then recognize the object in the bounding.

3.3. Motion based object tracking

Since the measurement of interest is the locomotive behavior of the zebrafish, we found it appropriate to explore the motion based object tracking[5] as the image processing way of doing this task. To identify the moving objects, the

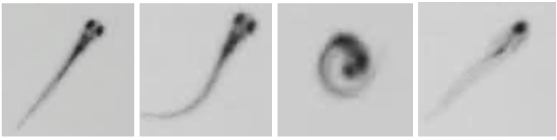

Figure 4. Left to right: resting, slightly curled, completely curled, tilted

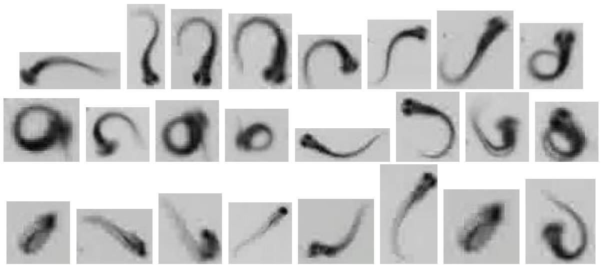

Figure 5. Subsample of zebrafish body positions

first step is to subtract the background and highlight the foreground object in motion. Background Subtraction is achieved using gaussian mixture models. To enhance the foreground mask, the holes are filled. Blob analysis helps locate connected pixels of an object. Once located, the detected object is tracked between the frames with the help of Kalman filters.

4. Experiment

4.1. Data

Professor Downes' lab has a collection of high-speed white and black videos from their existing experiments. To start with we have collected 40 videos of different normal and mutated zebrafish reacting to a tactile stimulus. Each of these videos is 5 to 15 seconds long with 30 frames per second. Some of our models need images / videoframes to train on. We used *ffmpeg*[2] to extract the video frames of 640 X 480 pixel resolution and rate of 30 frames per second. This results in a total of 8806 images. The zebrafish tends to be in a resting position until it receives a tactile stimulus. Hence, we would like to note that some portion of these extracted frames tends to be the same, approximately 20%. Once the zebrafish is probed, it reacts by swiftly moving across the petri dish. It also curls its body in 360 degree rotations in clockwise and anti-clockwise directions. This creates a lot of variation in the image of the fish. (Figure 4) shows samples of the fish in 4 distinct positions to visualize the resting and curled positions of the zebrafish. Figure 5 is a sample of the different zebrafish body positions observed in a video.

Our initial plan was to use the lab's hand annotated data for training. It had the (x, y) position of the center of the fish at every frame of the video. As we started working with the data, we observed a lot of discrepancy in the values and most of them were incomplete. We decided to manually annotate (Figure 6) all the frames of a subsample of videos which were representative of the zebrafish body positions. Using *BBox-Label-Tool*,[1] we annotated the bounding box for these frames resulting in $[xmin, ymin, xmax, ymax]$ values for 339 images. We also collected 350 non-zebrafish images from different genres for the negative class of the SVM classifier in the last layer.

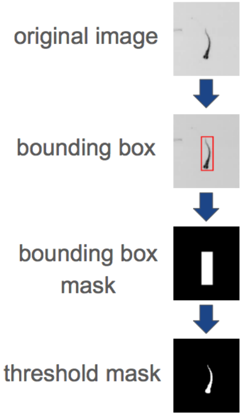

Figure 6. Manual annotation of bounding boxes

4.2. Evaluation and Results

4.2.1 End-to-end Models

We tested several variants of the architecture shown in Figure 3, such as with pooling, without deconvolution (and scaling down the label image), with and without ReLUs (both leaky and standard). The best-performing model was 4 convolutional layers and 4 deconvolutional layers, with leaky ReLUs. However, despite extensive training (Figure 8), even this architecture only managed to learn how to invert the image (Figure 7) (because the zebrafish in the input image are dark, and the label mask is white where the fish is); this is problematic because it also produces high activation in areas that contain the researcher's finger or the probe, both objects that interfere with existing methods. Because none of the models passed the qualitative evaluation phase, we didn't attempt to evaluate quantitatively. Clearly this type of model is not suitable for the problem.

4.2.2 Fine tuning and Optical flow

We conducted a 10-fold cross validation while training the SVM classifier (zebrafish vs no zebrafish). Since we don't have the true labels of the bounding boxes in these videos, we evaluated this approach by measuring the accuracy of the SVM classifier in classifying a sample of 100 unseen

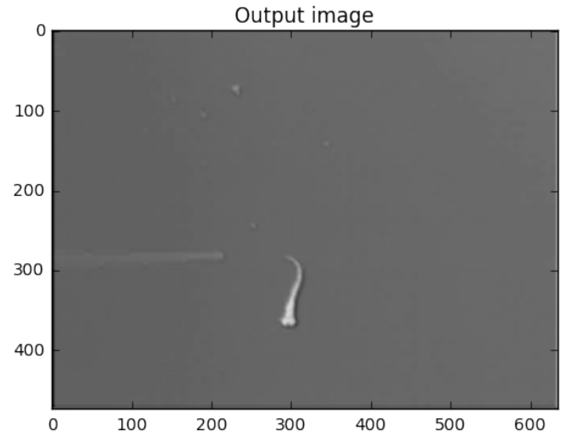

Figure 7. Output of the best end-to-end model.

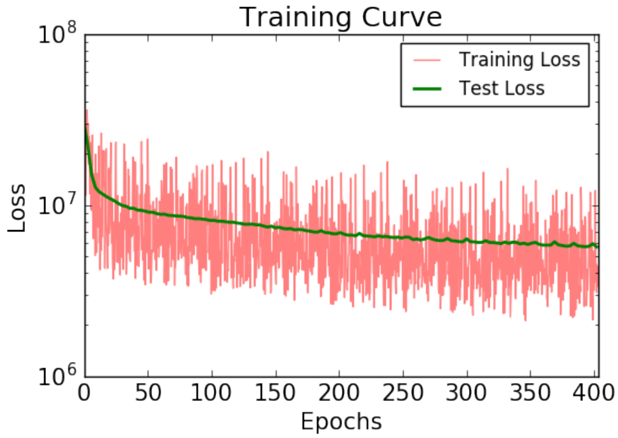

Figure 8. Training and validation loss over the training of the best end-to-end model.

test samples. Of these, 50 were zebrafish and the rest were the negative examples. Assuming we have an accurate object detector which outputs the correct bounding box, this evaluation helps measure the fine-tuned deep learning model. We observe that the model accuracy is 78%. It classified all the 50 zebrafish test images correctly. But, there were 22 false positives. These were mostly the images with similar black and white background. Alternatively, we fine-tuned the model with negative examples randomly selected from the region outside the zebrafish bounding box. Most of these extractions resulted in a relatively monotonous gray image. The SVM classifier thus learned to classify an image with any object as zebrafish.

4.2.3 Motion based object tracking

Number of images: 687
Number of batches: 5
Number of layers in the Network: 21

Batch	Execution time(s)
1	6.6006
2	7.9362
3	6.7761
4	6.5980
5	3.4143
6	6.2650

Total execution time: 31.3252

Table 1. Fine-tuning training time in CPU

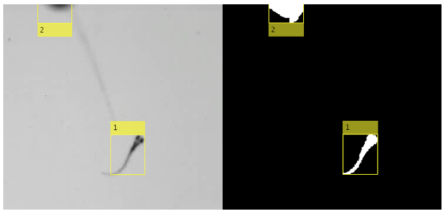

Figure 9. Motion based object tracking detecting finger and ZebraFish as 2 objects as expected

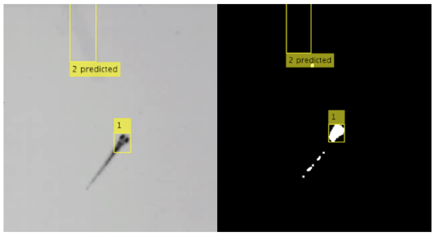

Figure 10. Motion based object tracking fails to detect the full body after tilt

This approach seems to be the best so far in terms of training requirement and accuracy (Figure 9). It is important to note that the output here is just the bounding box of the tracked objects. We could combine the fine tuned model in the previous section to recognize the object in the bounding box and discard all the non zebrafish trackers. Although, a severe drawback here is that it loses track of the fish when it is resting assuming it to be a background object. One way to overcome this is to keep a small memory of the last bounding box and using the fine tuned model to predict the object. We have also observed in some cases that the tracker looses the zebrafish' tail (less denser region), when it has recently titled or coiled itself (Figure 10).

5. Conclusion

Unfortunately our attempts to develop a new deep learning method specific to zebrafish detection was unsuccessful, but based on our experimentation with existing models it's likely that adapting a region proposal and classification approach would be very successful. We were unable to get Faster R-CNN working, but achieved 78% accuracy with a somewhat similar method, which used optical flow for region proposals and a fine-tuned classifier pre-trained on ImageNet. However, there were serious issues with the validity of this result, because the negative training and test samples (non-zebrafish) were all random colored images, while the positive samples (zebrafish) were all black-and-white images from the high-speed video. We would have liked to retry this experiment with better-curated positive and negative samples, such as selecting windows of the high-speed video that contained the experimenter's finger or the probe used for tactile stimulus, as well as negative examples of the background, and perhaps even desaturated high-contrast normal images. When examining the quantitative performance of the model in the future, we would only use windows selected from the actual domain that the model would be applied in (selections from the high-speed video).

References

[1] *BBox-Label-Tool* https://github.com/puzzledqs/BBox-Label-Tool

[2] *FFmpeg* https://ffmpeg.org

[3] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T. "Caffe: Convolutional Architecture for Fast Feature Embedding." (2014). *arXiv*.

[4] Granato, M., Van Eeden, F.J., Schach, U., Trowe, T., Brand, M., Furutani-Seiki, M., Haffter, P., Hammerschmidt, M., Heisenberg, C.P., Jiang, Y.J. and Kane, D.A. "Genes controlling and mediating locomotion behavior of the zebrafish embryo and larva" (1996). *Development* 123(1): 399-413.

[5] MathWorks. "Motion Based Multiple Object Tracking." https://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html

[6] MathWorks. "Optical Flow" https://www.mathworks.com/discovery/optical-flow.html

[7] Mirat, O., Sternberg, J.R., Severi, K.E., Wyart, C. "ZebraZoom: an automated program for high-throughput behavioral analysis and categorization." (2013). *Frontiers in Neural Circuits* 7: 107.

[8] Prasanna, Shashank. "Deep Learning for Computer Vision with MATLAB and cuDNN." (2015). https://devblogs.nvidia.com/parallelforall/deep-learning-for-computer-vision-with-matlab-and-cudnn/

[9] Ren, S., He, K., Girshick, R., and Sun, J. "Faster R-CNN: Towards real-time object detection with region proposal networks." (2015). *Advances in neural information processing systems*: 91-99.

[10] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L. "Imagenet Large Scale Visual Recognition Challenge." (2015). *International Journal of Computer Vision*, vol 115(3): 211-252. doi:10.1007/s11263-015-0816-y

[11] Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A. "Return of the Devil in the Details: Delving Deep into Convolutional Nets." (2014). *British Machine Vision Conference*.