BME 548 Final Project: Using DeepLabCut to Track Mantis Shrimp Strike Patterns

Yasuhiko Komatsu
Department of Biomedical Engineering
Duke University
Durham, NC 27708
yk207@duke.edu

Isabella Wang
Department of Biomedical Engineering
Duke University
Durham, NC 27708
iww@duke.edu

Abstract

A major bottleneck to researching mantis shrimp strike patterns is the time needed to label key features on their claws as they strike. In this project, a model was created to predict the locations of 17 key features when given an unlabeled video. Five ResNet DeepLabCut models were made using 122 mantis shrimp videos, and results were evaluated using a 1000-iteration ResNet-50 model. The model was most accurate in predicting C- and MV-series locations, and least accurate in predicting M- and P-series locations. While the results from the current model are not satisfactory for replacing manual labeling, they demonstrates the feasibility of using automatic labeling and presents many opportunities for future improvement.

1 Introduction

Mantis shrimps strikes are some of the fastest animal movements known. Their peak strike acceleration rivals that of a bullet in a gun and water cavitates, or boils, where the claw hits [4]. Currently, the bottleneck to understanding kinematics of mantis shrimp strike patterns include the amount of time it takes for mantis shrimp videos to be digitized - it takes a human scorer an average of 30 minutes to an hour to fully label a mantis shrimp strike with 17 points. In this study, DeepLabCut was used to evaluate whether the process of digitizing mantis shrimp can be automated by machine learning.

2 Related Work

This project primarily uses a python toolbox called DeepLabCut [3]. DeepLabCut was made by the Mathis lab team in Switzerland, and its main purpose is to estimate the poses of animals. In the paper, Mathis et.al. shows examples of automated point tracking used on flies and mice, showing that DeepLabCut works on a wide range of animals from mammals to arthropods. It should be noted that most of the videos used for DeepLabCut demonstration by Mathis et.al. had consistent lighting, magnification, and resolution throughout the videos used.

While similar automated point tracking methods have been used in the field of studying ultrafast movements of organisms, most of the examples have been much less complex. For instance, the R code used to track ballistic propulsion of trap-jaw ants, an example of an organism that uses ultrafast mandible movements to propel themselves into the air, involves image thresholding to isolate the propelling ant and calculating the centroid of the whole ant [5]. Another example of this thresholding technique has been performed on seed shooting of witchhazel fruits, on which the seed's centroid and tip have been automatically tracked [1]. However, this method only yields simple points such as centroids and tips for the entire organism, and also cannot be applied to images with inconsistent backgrounds that cannot be thresholded. Mantis shrimp videos fit in this category, since the points that need tracking are difficult to find and background lighting is often inconsistent due to difficulty of making lighting consistent in ultrahighspeed imaging.

3 Methods

3.1 Source of Data

Data was received from the Patek Lab at Duke. Data came in the form of .avi videos of mantis shrimp strikes. Each video had a corresponding MatLab file which denoted the locations of manually labeled points of interest. Each MatLab file was made by a single human "scorer" (labeler).

3.2 DeepLabCut Architecture

In order to start the training process, DeepLabCut requires videos of the animals with cropped frames, and csv files of the x-y locations of each tracked point in each frame. This information is then fed into a Residual Network (ResNet) that has been pre-trained on ImageNet, a large image database that is often used in object recognition. The ResNet is a neural network that uses skip/shortcut connections to link initial activation layers to further layers deeper in the network by skipping some layers in between [2]. This allows the network to pass information from initial layers to deeper layers by matrix addition, eliminating some of the limitations of building deeper networks such as the "vanishing gradient". The purpose of the additional pre-training is to optimize speed and teach the network to extract features from images even before it sees the actual training set. After the ResNet step, the model attempts to reverse the convolutions and recreate the original image. When it receives a new video, it can then use this information to predict locations for each tracked point.

There were two key customizations for training DeepLabCut models: 1) type of model and 2) number of iterations. Types of models included ResNet models of various depths, including ResNet-50, ResNet-101, and ResNet-152. The number of iterations was left up to the user (default = None). In sample code and projects, most models used a number of iterations on the scale of 1000's or 10,000's.

3.3 Data Pre-processing

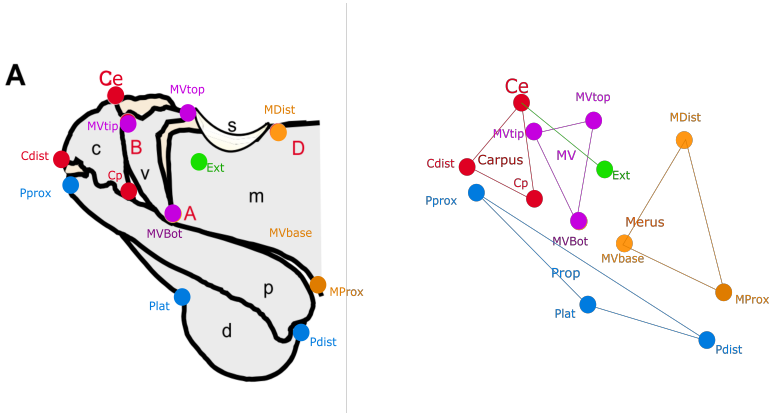


Figure 1: Morphological Markers on a Mantis Shrimp Claw

The entire mantis shrimp dataset was filtered for videos that contained labels for all seventeen points that were the most significant for tracking: Cdist, Cexten, Clump, CslideTop, Ctop, Extensor, Latch, MDist, MLat, MProx, MVBot, MVnotchP, MVtip, MVtop, Pdist, Plat, and Prox. The points can be further classified into the C (Carpus) series with 3 points, M series (Merus) series with 3 points, MV series (Meral-V) with 3 points, and the P series (Propodus) with 3 points. Points in each series move together as a rigid structure and stay in the same triangle relative to one another throughout the strike. After filtering, there were 122 videos left. Of these 122 videos, 80 percent (98) were assigned as training videos, and 20 percent (24) were assigned as testing videos.

Each MatLab data file was converted into .csv format and subsequently .h5 format for easy processing in DeepLabCut. Some points did not appear in every frame of the video, and so any blank data was replaced with "NaN" (Not a Number) and removed from further analysis. Each .avi video file was converted into a collection of .png images of video frames. These corresponding sets of .h5 locations

and .png video frames were then fed into DeepLabCut, which ran them through the ResNet network and produced a .csv file of location predictions for each tracked point.

3.4 Model Training

Five different models were trained: three ResNet-50 models, a ResNet-101 model, and a ResNet-152 model. The three ResNet-50 models were used to test the effect of increasing the number of iterations, and ResNet-101 and -152 were used to test the effect of increasing model depth.

4 Results

4.1 Loss and Learning Rate

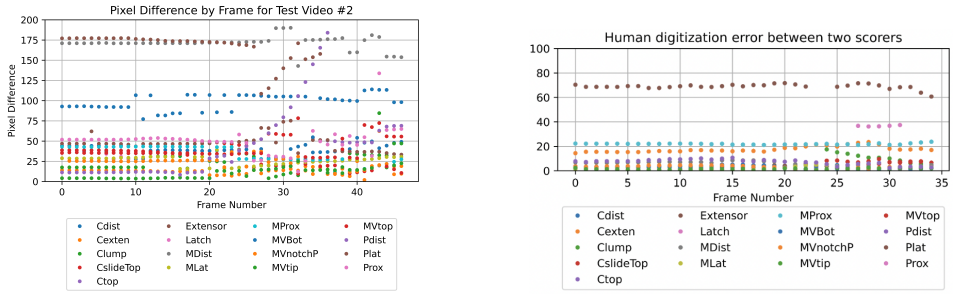
Table 1: Model Losses

Model Type	Number of Iterations	Loss	Learning Rate
ResNet-50	100	0.0472	0.005
ResNet-50	1000	0.0353	0.005
ResNet-50	3400	0.0305	0.005
ResNet-101	1000	0.0364	0.005
ResNet-152	1000	0.0350	0.005

Table 1 shows the loss at the end of training each of the five models at a 0.005 learning rate. The results show that increasing the number of iterations had a significant impact on reducing model loss, but increasing the number of layers in the ResNet model did not have a notable effect.

The ResNet-50 model at 1000 iterations was chosen for further analysis, because it had the best balance of runtime efficiency and loss reduction given the time constraints of the project.

4.2 Single-Video Pixel Difference



(a) DeepLabCut vs. Manual

(b) Scorer 1 vs. Scorer 2

Figure 2: Pixel Difference by Frame for Single Video

Figure 2 shows the distance in pixels for each frame in a single test video (in this case, test video number 2). Figure 2a depicts the distance in pixels between the DeepLabCut prediction for x-y coordinates and the manually labeled location. Figure 2b depicts the distance in pixels between two lab scorers who manually labeled the same video.

Most of the pixel difference in Figure 2b was around twenty pixels or less, while Figure 2a pixel difference ranges from five to two hundred, with most being above twenty pixels.

4.3 Average Pixel Difference

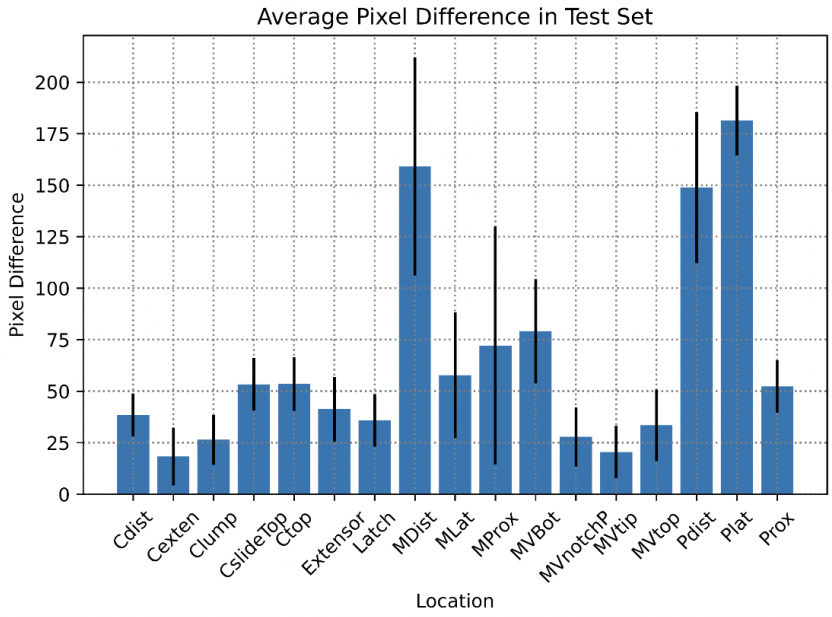


Figure 3: Average Pixel Difference by Location in Test Set

In Figure 3, the pixel distance between DeepLabCut predictions and manual labeled locations was averaged for each tracked point across all twenty-four test videos. Error bars were added to show the standard deviation for each tracked point.

Most locations had an average pixel difference of 50 or less. MDist, Pdist, and Plat had significantly higher average pixel difference than the other tracked points. MDist, MProx, and Pdist had the largest standard deviation.

5 Discussion

According to Figures 2 and 3, the model was most accurate in predicting Cexten and MVtip, and least accurate at predicting MDist, Pdist, and Plat. Predictions were the most variable for MDist, MProx, and PDist, indicating that the model would be most unreliable to use for digitizing those points.

Points in the M-series are the most subjective points to manually label, since lab members are asked to look for a general area on the claw as opposed to a defined morphological location. Therefore, it is expected that the model performed most poorly on most M-series points. Additionally, points in the P-series are where most of the movement occurs during the striking motion, and often appear blurry in video frames. Thus, the low accuracy in predicting most P-series points was to be expected as well. Points in the C- and MV-series were mostly based on the morphology of the claw and had the least amount of difference across manual scorers, which likely contributed to their high predictability in the model.

Moreover, based Figures 1a and 1b, DeepLabCut's predictions are not accurate enough to be on par with human scorers. In order to be viable for lab use, the model should be approximately 10 pixels away from a typical scorer. At its current state, the DeepLabCut model is not satisfactory for replacing all manual labeling in the lab. For some types of points that is is already proficient at predicting, such as the C-series, it may be a feasible method after some additional training and fine-tuning.

At this stage in the project, there are several key limitations. First, background lighting and the speed of the strike was inconsistent between videos, since shrimp were allowed to move and not all shrimp

willing to strike at top speed. Modifications to the video-recording setup and video brightness and contrast during data pre-processing could help to alleviate these issues.

Next, given the scale of the data pre-processing and size of the final dataset, a 10,000-iteration model was not trained in time for analysis. Since we have yet to see the model loss plateau and become irreducible even at 3000+ iterations, it would be worth investigating if very large iterations have a good marginal benefit.

Our third limitation is that there is a lack of ground truth in the model, because human error is involved in manual labeling and there is no way to determine the "true" location of each tracked point. It would be helpful to further increase train and test set sizes in the future to minimize the effect of human error.

Finally, in this project we were primarily focused on studying the accuracy of location prediction. It would be interesting to perform more analysis on the precision of the model as well as the accuracy. In the context of research, consistency is very important so that the model isn't producing results that seem significant but may just be due to high model variability.

References

[1] Jorge, J.F., J.S. Harrison, P.S. Manos, and S.N. Patek. "Biomechanics of ballistic seed dispersal in the witch hazel (Hamamelis)." Integrative and Comparative Biology (2019): E115-E115.

[2] K. He, X. Zhang, S. Ren and J. Sun. "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA (2016): 770-778.

[3] Mathis, A., Mamidanna, P., Cury, K.M. et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning." Nat Neurosci 21 (2018): 1281–1289.

[4] Patek, S.N. and, and R. L. Caldwell. "Extreme impact and cavitation forces of a biological hammer: strike forces of the peacock mantis shrimp Odontodactylus scyllarus." Journal of Experimental Biology 208.19 (2005): 3655-3664.

[5] Patek, S.N. "Lab 05 - Tracking the jump of the trap-jaw ant." How Organisms Move, Bio 490S, 2014.

Acknowledgments

We would like to thank Dr. Roarke Horstmeyer, Amey Chaware, and Kanghyun Kim from BME 548 for their help and support throughout the project. We would also like to thank the Patek lab for providing our dataset, and John Efromson for his advice on using DeepLabCut.