6-4-2020

Autonomous Eye Tracking in Octopus bimaculoides

Mark Andrew Taylor
Dartmouth College

Dartmouth Computer Science Technical Report TR2020-883: Autonomous Eye Tracking in *Octopus bimaculoides*

by

Mark Andrew Taylor

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

Bachelor of Arts in Computer Science

at the

DARTMOUTH COLLEGE

June 2020

Author . . . .
Department of Computer Science
June 4, 2020

Certified by . . . .
Yaroslav O. Halchenko
Research Associate Professor
Thesis Supervisor

Accepted by . . . .
Thomas H. Cormen
Undergraduate Program Director

Dartmouth Computer Science Technical Report TR2020-883:

Autonomous Eye Tracking in *Octopus bimaculoides*

by

Mark Andrew Taylor

Submitted to the Department of Computer Science
on June 4, 2020, in partial fulfillment of the
requirements for the degree of
Bachelor of Arts in Computer Science

Abstract

The importance of the position of cephalopods, and particularly octopuses, as the most intelligent group of invertebrates is becoming increasingly appreciated by the neuroscience research community. Cephalopods are the most distantly related species to humans that possesses advanced cognitive abilities; as their intelligence evolved independently from vertebrates, comparative analyses reveal trends in the evolution of nervous systems and the foundations of intelligence itself.

Vision is an especially important area of cephalopod cognition to research because cephalopods are predominantly visual creatures, like humans, and the rapid transduction of visual signals allows the inner-workings of octopus cognition to be revealed in real time. While octopuses can be conditioned to indicate what they see through responses to conditioned visual stimuli, no system as of yet provides a non-invasive means of determining what an octopus is looking at without training.

This thesis introduces an automated methodological framework to predict the direction of an octopuses gaze for use in visual cognition research. The system utilizes deep learning models to track the eyes of octopuses, then predicts where an octopus is looking based off of the orientation of their eyes and known anatomical traits that constrain where their vision could be directed. Data could not be collected this spring to train a model and test the tool in the experimental setting the system utilizes, however analyses conducted on data not intended for this project suggest the approach is feasible for estimating an octopus' gaze and offer insights into how to do so most effectively.

Thesis Supervisor: Yaroslav O. Halchenko
Title: Research Associate Professor

Acknowledgments

First and foremost, I would like to express my thanks towards Dr. Yaroslav Halchenko for taking me under your wing and advising me through this project. Your support has enabled me to integrate my passions of Neuroscience and Computer Science and your philosophy of collaboration in science and programming is incredibly inspiring.

I would then like to express my deepest gratitude towards Dr. Peter Tse for leading the charge in creating Dartmouth's Octopus lab and mentoring me for the past two years. Without your support I do not know where I would be today. You've taught me what it means to be a scientist.

Thank you to Arnold Song and Douglas Hill for your support with the project and the wealth of information you've been able to provide. Arnold, your assistance made many of the technical portions of the project possible and its been a pleasure working with you.

I am indebted to Marvin Maechler for your continuous mentor-ship and frequent help in matters both big and small. Working on your project is what got me started on this road and you've helped me more than I deserve over the past two years. I only hope I can pay forward all you've done for me.

Finally, thank you to the entire Octopus lab for the academic environment you have facilitated and your assistance throughout my time in the lab. It has been an absolute pleasure working together and I look forward to continuing our research next year.

Contents

Chapter 1

Introduction

This thesis establishes an autonomous system for research data collection and analysis for the study of vision, as well as cognition at large, in octopuses and other cephalopods. The approach is motivated by a desire to repeat attentional tracking studies that have been conducted in humans with octopuses for comparative analysis, to aid existing research projects by reducing experimental demands on researchers and animal subjects alike, and to make the exploration of new topics in octopus vision research feasible by non-invasively estimating the direction of their gaze. Beyond the diverse research applications this non-invasive thesis presents, the methodology underlying the tool itself provides insights into the foundations of the octopus visual system that are currently not well understood.

The project is intended to be used in future studies in Dartmouth's octopus cognition lab, but it is open to the cephalopod research community at large and its use is encouraged. Differing labs will likely have to collect their own data for the deep learning component of the project used to extract eye locations from video data, but the rest of the software should be more or less usable out of the box. The tool was developed for use in octopus research, however could be utilized in research projects with other cepahlopods like cuttlefish just as easily due to the similarities between their anatomy and visual systems [5].

This thesis also provides a review of the existing octopus visual system literature that is pertinent to the methodology behind the project, as well as the potential research possibilities it introduces. Areas of unexplored research that would be useful towards this project and visual research at large are highlighted. The system estimates where an octopus' gaze is directed by making assumptions about the parameters of octopus' visual fields, particularly their orientation in relation to each other and the rest of an octopus' body; these unknown's represent opportunity rather than a hindrance as use of the system will estimate and rely upon probable parameters, which can then be validated in future anatomical studies.

1.1 Motivation

1.1.1 Purpose of studying cephalopods

Most intelligent species as we know them are vertebrates. These species include mammals other than humans such as chimpanzees and dolphins, as well as groups less related to us such as avians. This has led to the perception that advanced intelligence only exists in vertebrates, however recent focus has been placed on the complex behavior and nervous systems of invertebrate species such as spiders, bees, and notably cephalobods [3]. Intelligence in these species is remarkable because of how distantly related they are to vertebrates. While there are many differences between the nervous systems of various mammals and even greater differences between mammalian and avian nervous systems, these differences are largely deviations on the same basic nervous system plan [4]. The last common ancestor invertebrates and vertebrates share was a simple bilateral worm named Urbilaterian; because its nervous system was so rudimentary, complex behavior and a sophisticated nervous system evolved in invertebrates like cephalopods independently from vertebrates [4]. As convergent evolution of the nervous systems of mammals and avains has been analyzed, identifying parallels such as functionally similar circuitry evolving from distinct areas, the study of convergent evolution between vertebrates and invertebrates is beginning to be addressed, with parallels such as short-term memory, long-term memory, and synaptic potentiation already recognized [3]. Insights from these sorts of comparative analyses are not only important for understanding the necessary components behind cognitive tasks like memory, but also trends in how these systems evolve.

1.1.2 Importance of vision research

Vision is one of the most accessible means of studying octopus cognition and a lot can be revealed through visual experiments. Getting an octopus to follow the protocol of any given experiment is often difficult and involves a lot of training; developing an autonomous, reliable eye tracking system for octopuses would be advantageous in conducting many of our current experiments, easing the burden on researchers and octopuses alike, as well as opening the door to exploring new questions regarding how octopuses utilize vision.

1.1.3 Eye tracking and gaze estimation

There are numerous software packages available for eye-tracking in humans as well as software for a few other species, however no software exists to track the vision of octopuses. The goal of this project is to develop a user-friendly, open-source software package allowing researchers to easily track the eye-movements of octopuses in experiments.

The difference between eye tracking and gaze tracking is important to point out. Eye tracking specifically refers to measuring eye movements, which are typically independent of the rest of the body moving. Gaze tracking is all about estimating where a subject is looking and where they are directing their attention; gaze tracking systems often utilize eye tracking data, along with the orientation of the head and body, to predict where a subject is looking. The most accurate systems for tracking gaze typically involve immobilizing the head and tracking eye movements, as eye movements can be precisely tracked and all other degrees of freedom are removed [10].

There are several difficulties associated with developing a precise octopus eye tracker. Traditional eye tracking approaches in animals involve invasive surgical implants into the eye [10]. More modern systems are less invasive and use video-based tracking methods. The most accurate of these systems, which is traditionally used in human eye tracking, measure eye orientation via corneal reflection of an infrared light relative to the center of the pupil [7]. These systems require the subject to remain entirely stationary so the system can be calibrated and only the eyes are able to move. Though this is not an issue in human eye tracking, animals must either be restrained physically or trained to remain stationary. Non-human primates and other vertebrates are often physically restrained using head posts, however this is nearly impossible in octopuses as they lack skeletons and cannot be forced to remain stationary without harming them. Non-human primates have been trained to remain still and utilize a small viewing area during studies; this approach shows promise for precise octopus eye tracking and is considered in the discussion section, however its use-cases are limited and the approach would not allow eye tracking in a freely moving octopus as this project seeks to accomplish [10].

As the aforementioned approaches are not well-suited to tracking a free moving octopus in its tank, this tool attempts to leverage the unique anatomy of the octopus visual system, which itself limits the degrees of freedom required for tracking gaze, as well as their stereotypical behavior exhibited when attending to stimuli, to predict if they are attending to a stimulus.

1.2 Background

Like other cephalopods, octopuses rely heavily on visual information and large portions of their nervous systems are dedicated to the processing of visual information [5]. They possess complex camera-like eyes that are remarkably similar to vertebrate eyes which give them much greater visual acuity than other marine species. This convergence in visual evolution begs the question of what other similarities exist between the visual systems of octopuses and humans? Unfortunately, a lot of information is lacking for cephalopods and octopuses in particular, but several researched topics provide useful information for the project.

Though the use of this tool could be expanded to other species of octopuses and even other orders of cephalopods, this thesis focuses on *Octopus bimaculoides* as it is the sole species Dartmouth uses for octopus research. While there is little information on the ocular system of *Octopus bimaculoides* specifically, research on *Octopus vulgaris* is used as the species are similar and cuttlefish information is referenced where none exists for octopuses.

1.2.1 Eye anatomy

The large eyes of octopuses and the proportionally large portions of their brains that are dedicated to the processing of visual information grant them vastly superior visual acuity compared to other species in their environment, such as fish. They are capable of rotating their eyes horizontally in either direction as well as bulging them in and out [5]. Their pupils are generally oval shaped, but widen to a circular shape in low-light conditions and decrease in diameter in bright-light. Contrasted to the fovea humans and other vertebrates possess, a concentration of photoreceptors at the center of the retina that is responsible for the greatest acuity in the center of the visual field, octopuses have a central band of photoreceptors across their retinas presumably granting high acuity vision horizontally across the center of their visual fields [9].

Many aquatic invertebrates possess statocysts, small organs used to maintain balance and orientation with respect to gravity. Cephalopods utilize these organs to constantly keep their eyes oriented horizontally, regardless of the orientation of the rest of their bodies. Through this mechanism, octopuses are able to ground their interpretation of visual information on the basis that their eyes are oriented horizontally; this claim is supported by the inability of octopuses to discriminate stimuli by orientation if their eyes are not horizontally aligned [5]. Octopus eyes are positioned laterally and their eyes can face in entirely opposite directions; they are even able to move them independently. The high visual acuity of octopuses reflects the unique status of cephalopods as molluscan predators, as they rely heavily on vision for capturing prey [8].

1.2.2 Visual field

While there are no studies investigating the visual field of any octopus species, assumptions can be made based off of their physiology that converges with distant species for similar ecological purposes. The anatomy of the ocular system of an animal corresponds to the needs imposed by the animals environment [9]. In vertebrates, predators tend to have eyes oriented in the same direction and a centrally located density of photoreceptors or a vertical stripe of increased photoreceptor density; these attributes enable binocular vision and are well suited to ambushing prey. Prey on the other hand tend to have laterally placed eyes and horizontally oriented photoreceptor densities, resulting in a very wide field of view and the ability to detect predators from all directions. Octopuses are typically both predators and prey in their habitats. Their horizontal "foveas" make them well-suited to detect and focus on predators and prey that may occur on the horizon. [9].

Though the left and right monocular visual fields of octopuses very likely overlap to some degree, their use of binocular vision is undoubtedly minimal; this claimed is backed up by their behavior's consistency with reliance on monocular vision.

1.2.3 Lateralized eye use

Consistent with their lateral eye placement, octopuses prefer monocular vision to binocular when attending to objects of interest. In *Octopus vulgaris*, octopuses demonstrate preferential use for focusing with either the left or the right eye (with neither side preference significantly more prevalent at the population level) [2].

1.2.4 Summary

Octopuses have laterally placed eyes that enable them to have an extremely wide field of view. Unlike the concentrated human fovea, octopuses have a stripe of increased photoreceptor density that grants high acuity vision across the majority of their visual field. Octopuses ensure their pupils remain oriented with the horizon utilizing their statocysts so their "foveas" are always aligned across the center of their visual fields.

Finally, octopuses nearly always attend a stimulus with a single eye and individuals show a preference for either the left or the right eye.

1.3 Hypothesis

Octopuses lateral eye placement, horizontal "foveas", and horizon-aligning statocysts ensure the animal is constantly ready to spot a predator or prey in pretty much any direction. Yet, octopuses consistently use monocular vision and preferentially orient a particular eye towards objects of interest. **As octopuses orient one eye in particular towards a stimulus they are focusing on, can the projection of their visual axis be used to reliably predict the orientation of their gaze and the direction of their attention?** This thesis provides a software framework centered around this hypothesis to learn more about the basic parameters of the octopus visual system and enable reliable predictions about the orientation of an octopuses gaze for practical use in research.

Chapter 2

Approach

For the purposes of creating a gaze tracking system, it is important to outline the necessary components. First the setup of experiments utilizing the tracker must be considered. Our approach considers three necessary steps, and discusses them in detail before going on to their implementation. The aim is to create as general a tracker as possible to allow for a diversity of experiments, but constraints need be defined for experiments to adhere to ensure the system is compatible. Next a means of labeling the locations of the octopuses eyes need to be produced. While a diversity of methods could be used, for the tracker to be as autonomous as possible a deep learning framework is used to achieve human-level labeling without requiring human labor and the requirements to do so are considered. Lastly, post-processing must occur on analyzed data to ultimately determine where an octopus is attending. This involves visualizing what the octopus can see as well as predicting where they are looking.

2.1 Experimental setup

The intention of the eye tracker is to enable various experiments with diverse stimuli. To make sure the system is generalized, care must be taken to ensure two attributes are constant: tank configuration and the filming setup.

2.1.1 Tank Configuration

To enable experiments involving varying degrees of octopus behavior expression and to ensure the octopus feels comfortable within the environment of the experiment, the octopus should have as much freedom as possible to move around. That said, a camera can only film so much space so the octopus must be constrained to some amount of space. In order to enable freedom while ensuring the octopus never leaves the scope of the tracker, we constrain the octopus to a Plexiglas box within a larger experimental tank. The octopus is able to move freely within the box and exhibit behavior, but the camera will be setup such that the octopus is never out of frame. The reason for filming the box within the tank, as opposed to the tank itself, is to ensure the camera is submerged underwater so no image distortion may occur.

2.1.2 Filming

Filming is a crucial step because the input to the tracking system is video data. As a trained model will be used to label novel video data, it is important to ensure data is collected such that it looks like data the model was trained on.

As mentioned in the previous section, footage should be filmed from underwater to avoid distortion. It is also ideal to have the camera positioned as consistently as possible. To meet the aforementioned goals, cameras should ideally be positioned on a frame that goes over the tank in the same location.

The last consideration is what sort of footage should be captured. While any waterproof camera works, we recommend the use of infrared cameras if possible. Octopuses prefer dark conditions and darker lighting ensures the displayed stimulus is more salient, but data cannot be labeled if the octopus is too dark for the camera to see. Octopuses cannot see infrared light, so illuminating them with infrared lights does not disturb them and ensures the octopus will be visible.

2.2 Labeling

While a human could label the location of the eyes in each frame of data, this would require a lot of labor and could lead to inconsistent labeling. A deep learning model trained to label the location of an octopuses eyes in video frames can perform just as well as a human and if anything would perform more consistently than a human or several different humans.

2.2.1 Model

This project is agnostic towards the model used to label data, however we utilize the model provided by the Python package DeepLabCut. The software package is designed for animal researchers interested in autonomously labeling video footage using Deep Neural Networks. DeepLabCut uses the ResNet feature detectors of the human pose estimation algorithm DeeperCut [6]. These models are initialized with weights from a network pretrained on ImageNet; with the pretrained weights, they are able to robustly detect image features. Using transfer learning and a relatively small number of images to classify, the network can be retrained to identify user-defined features of interest for a species.

2.2.2 Data

The data used to train the model is perhaps the most important component of the labeling step. The model can only accurately label novel data if the training data set is representative of any data that may be encountered. Most or all of the ways the octopus can configure themselves should be present in the training data, so data should be collected with octopuses moving around and positioning themselves in various ways. Any possible lighting conditions for experimental data should be used to produce the training set. Because the tracker is intended for use in *Octopus bimaculoides*, data from multiple individuals of the species should be gathered so the model isn't specific to an individual octopus.

In addition to the above considerations, it is vital to acquire data where it is known where the octopus is directing their attention. To do so, footage needs to be gathered where a stimulus is presented to the octopus in a location that will be identifiable from the video footage. Octopuses consistently direct their attention towards crabs so would work great for training an eye tracker, although any stimulus an octopus reliably attends to works just as well.

2.2.3 GPU

The model can be trained on any computer, but is a computationally expensive process. A GPU should be utilized to train the model in a reasonable amount of time. In ideal circumstances a cluster should be utilized to reduce training time even further.

2.3 Post-Processing

Once video footage is acquired and octopus eye are labeled with a model, it must be analyzed to determine if the octopus is attending to a stimulus. This involves measuring where the octopus is directing their vision then making a judgment on whether or not the octopus is looking at the stimulus.

2.3.1 Defining error

In line with our hypothesis statement that the projection of an octopuses visual axis is representative of where they are looking, we calculate error as the angle difference between the vector of the visual axis and the vector from the attending eye and the stimulus. Octopuses monocularly attend to objects and their eyes are facing in opposite directions, so we need only consider the visual axis of the closer of the two eyes. Though there is a slight binocular overlap in front of the octopus, it would be difficult to keep both eyes focused on an object at once and behavioral experiments demonstrate octopuses do not tend to track objects using this region [5].

2.3.2 Classifying attention

The error angle is a measure of how directly the octopus is looking at an object. The lesser the error, the more directly an octopus is attending to a stimulus. This can be used to determine where the octopus is attending in several ways.

If it is known when an octopus is attending to a stimulus or not, a threshold can be calculated using the error angle. Whatever the greatest error calculated is such that the octopus is still attending to the stimulus is the maximum error possible; if the error is below the threshold then the octopus is attending to the stimulus, while if below then the octopus is not attending. This error can also be interpreted as a confidence score: the smaller the angle, the more likely it is the octopus is attending the object.

A single measurement leaves no room for error and does not take into account movements by the octopus that may be unrelated to a stimulus being tracked. Segmenting measurements into groups allows for data from multiple time points to be consolidated. Segments where it is known if a stimulus is attended can be used to train a classifier to tell if an octopus is attending a stimulus in new data segments.

If there are competing stimuli the octopus may be attending, the previous methods can be slightly altered. The error angles for the two stimuli can be calculated and the octopus is more likely to be attending the stimulus with the smaller error. The errors to the two stimuli, or segments of error angles for each, can be used to train a classifier to determine which stimulus an octopus is attending.

Chapter 3

Implementation

This section describes the latest implementation of the eye tracking system. Though all of the component elements are present, there is a critical shortcoming in the current instantiation of the project. The training data underlying the project was to be collected in the spring of 2020, but due to COVID-19 closures no data could be collected.

Currently, a video data set from an unrelated project is being utilized to test the system and develop the post-production component. For the purposes of developing a general purpose eye tracker the data is far from ideal, and the reasons it is ill-suited for the project are discussed below, but it is sufficient to formulate the codebase for the project so an intentional data set in the future can be utilized. All code for the project can be found at the following repository: octo_eye_tracking.

3.1 Experimental setup

The experimental set up is designed to provide detailed footage of octopuses. Though data could not be collected with the set-up, the Dartmouth octopus lab intends to utilize it in the future so we describe it here.

A frame is placed above the experimental tank to position the camera. As undistorted footage is ideal, the camera is attached to a rod mounted on the frame so it is below the surface of the water. Lights are also mounted to the frame to provide

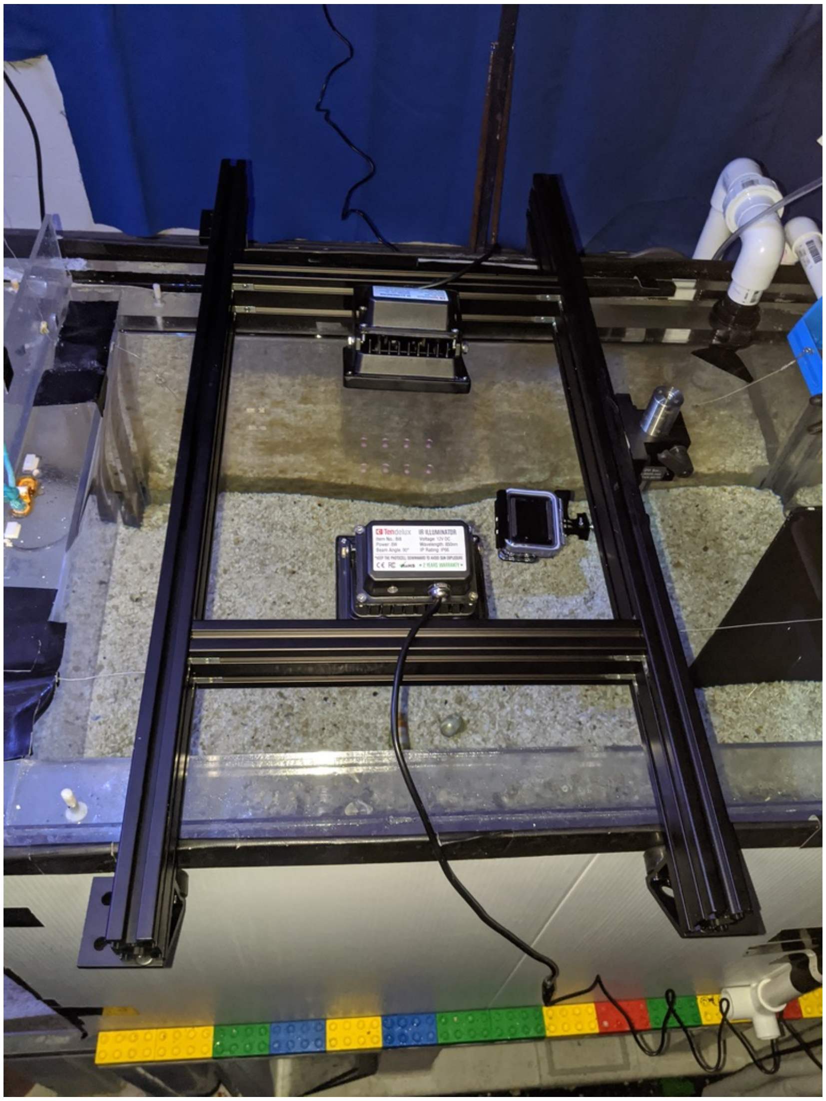


Figure 3-1: The IR camera and lighting setup.

well illuminated footage.

A Go Pro is used for the camera because it provide high-resolution data while performing reliably in a high-salinity aquatic environment. Though a standard Go-Pro is adequate, we utilize a Go-Pro equipped with an infrared lens so the octopus can be highly illuminated with infrared light without disturbing it (infrared light is outside of the octopus visual spectrum).

A projector is used to project stimuli onto one of the outside walls of the tank.

Within the tank, a Plexiglass box is positioned in front of the stimulus wall and underneath the camera frame. The camera is close enough to the box to get detailed footage of the octopus, while being far enough such that the octopus is never out of frame wherever it goes within the box.

3.2 DeepLabCut

The following section details the utilization of the python package DeepLabCut to train a deep learning model for autonomous eye labeling.

3.2.1 Training data

As a dedicated training data set could not be collected for the project, footage from another experiment was used used to design the program around and assess its feasibility for future eye tracking studies.

The experiment utilizes a drifting grating moving across the stimulus screen, physically divided in two with a plastic barrier. Octopuses are released from a box and conditioned to touch the screen on the side the grating is drifting towards to receive a reward. This data set includes 55 recordings taken from above the tank during the training period for one octopus.

Eighty percent of the videos were used for a training set of 44 videos, while twenty percent were reserved for a testing set of 11 videos. The videos were loaded into a DeepLabCut project and 448 total frames were extracted for human labeling. The left and right eye were labeled in each frame. As DeepLabCut utilizes transfer learning and initial weights are loaded from a robustly pretrained network, DeepLabCut models need relatively few labeled frames for training. Though accurate networks can be trained with fewer than 50 frames, as resolution lowers more frames are needed for training. Around 500 images are required to achieve below 3 pixel error with 192x192-pixel frames while 100 frames are able to achieve fewer than 5 pixel accuracy on 800x800-pixel frames, although 500 frames is able to achieve human-level precision of 2.7 pixels [6]. The training set contains 448 640x436-pixel frames, slightly below

Figure 3-2: Footage from the data set we utilized for analysis.

but within range of the DeepLabCut benchmark. Higher resolution footage may be downsized, as increases in resolution vastly increase processing time and resources, but the 640x436-pixel frames are small enough to be manageable.

3.2.2 Discovery

Discovery, a 3000+ core Linux cluster, was utilized to train the model and autonomously label novel frames.

Frames are labeled locally before being copied over to the Discovery cluster. A Singularity container is utilized to run DeepLabCut on the cluster without needing to fully install the software on the system. The container can be used to configure the project with a shell on the cluster, but training should be initiated via a PBS script to properly schedule a job. The repository includes an example of a submission Discovery submission script for reference. 12 hours of walltime and one GPU node were sufficient to run 200,000 training iterations, enough for loss to plateau without over-fitting.

Analyzed frames and labeled videos were also generated on Discovery; this task is more feasible for a CPU than training, but would still take a considerable amount of time. Once again a PBS script is included demonstrating job submission on Discovery; four and a half hours of walltime were enough to analyze all training videos and create labeled training videos, while two hours were plenty for labeling the testing videos.

3.3 Post-processing

The final component of the project is to utilize labeled data to determine where an octopus is attending. This implementation of the tracker is used to determine which of two stimuli an octopus is attending, although the software can be used to determine if an octopus is attending a single stimulus or more than two stimuli. All functionality results from four python files, data_load.py, analyze.py, visualize.py, and classify.py, with a Jupyter Notebook included demonstrating their use.

3.3.1 Loading data

The contents of data_load.py contains everything needed to load data for the post-processing stage. Labels generated with DeepLabCut are typically saved in the Hierarchical Data Format (HDF), although CSV files can also be produced (both are support by the package). Users specify a directory containing data and the contents are loaded into a python dictionary, where each key represents a different video. In addition to the x and y positions of the two eyes, several other attributes are loaded by default. For ease of downstream analysis the eyes are packaged into pairs by eye, as well as the line between the two eyes. Each frame of the video is also loaded for later visualization. Once all data is loaded, the dictionary can optionally be saved as a pickle file to speed up future loading.

3.3.2 Analysis

The x and y positions of the two eyes are used to find the line between the eyes. The x and y differences are calculated between the two points to find the visual axis vector, then divided by the absolute distance between the eyes to normalize the vector. The further eye is subtracted from the closer eye, as the vector extends from the closer eye. The same method is then used to calculate the vector between the closer eye and the stimulus.

With the vector of the visual axis and the vector from the eye to the stimulus, taking the angle difference using $atan2$ yields the error angle θ. Theta values are stored in the dictionary for the video for visualization and classification.

3.3.3 Visualization

Visualization includes several functions for illustrating the octopus' attention, as well as some for plotting data and classification results. The location of the eyes, the line between them, the projection of the visual axis of the closer eye, and the line from the closer eye to the stimulus can all be overlayed onto video frames. These frames can be viewed one at a time or stitched together into a video. Any of the components can be optionally disabled, including the video background.

3.3.4 Classification

Classification is the final step in determining where an octopus is attending.

The most straight-forward means of determining if an octopus is attending to an object is to define a threshold error; if theta is greater than the threshold then the octopus is not attending to the stimulus, if less than then the octopus is attending the stimulus.

Segments of data can be grouped together and classified for more robust analysis. Adjacent frames can be packaged together into a group of features, which can be fed into the rows of a matrix X. The labels for each row are then appended to the label vector y. Machine learning can then be used to train a model to classify the label

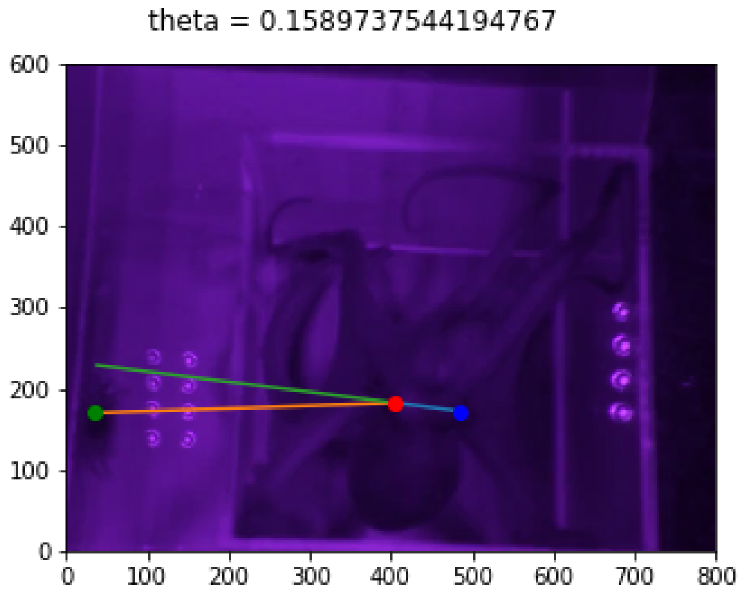


Figure 3-3: This example utilizes the IR cameras we would have used if we could have collected data this spring.

from y based on rows of features from X. The error to other stimuli, or multiple error values, can be combined into groups of features for classification.

For our current implementation, we demonstrate classification utilizing the final method. Although there is not a single stimulus presented to the octopus (rather a constant grating across a distant wall) we consider two potential locations for the octopus' gaze, the left half of the screen and the right, and analyze them like stimuli as they are relevant locations for the octopus to orient their attention towards.

Details are covered in the results section, but we attempt to classify which side the stimulus is presented on utilizing errors between the visual axis and the left and right sides of the screen. As the data is from the conditioning phase of an experiment, the octopus has not learned the paradigm yet and often mistakenly goes to the incorrect side; we consider the effect of these erroneous trials on our ability to classify.

With the aforementioned criteria, we run an assortment of machine learning algorithms and evaluate their success in the next section. This feature comes built into the classification phase, so users can easily determine how accurately they can classify and which classifier is best suited.

Chapter 4

Results

This section details the results that were obtained from analysis on the data set described in **3.2.1**.

4.1 Model performance

The model was trained on the Discovery cluster until loss plateaued after 200,000 iterations, which is aligned with the suggested number of iterations in the DeepLabCut documentation[6]. It took about 12 hours of GPU walltime for the network to train, with novel videos being labeled in a manner of minutes.

The performance of the trained network was evaluated by computing the mean average Euclidean error between the manually labeled points and those labeled by the network. Average errors were calculated separately for the points used to train the network and a subset of the labeled points that were reserved for testing the network's performance. For the training points, the average error was 1.83 pixels. For the testing points, the average error was 2.79 pixels.

4.2 Classifying gaze direction

We attempt to classify which side of the screen a stimulus is presented to an octopus on using a few different methods to group the data. For each method, the accuracy of

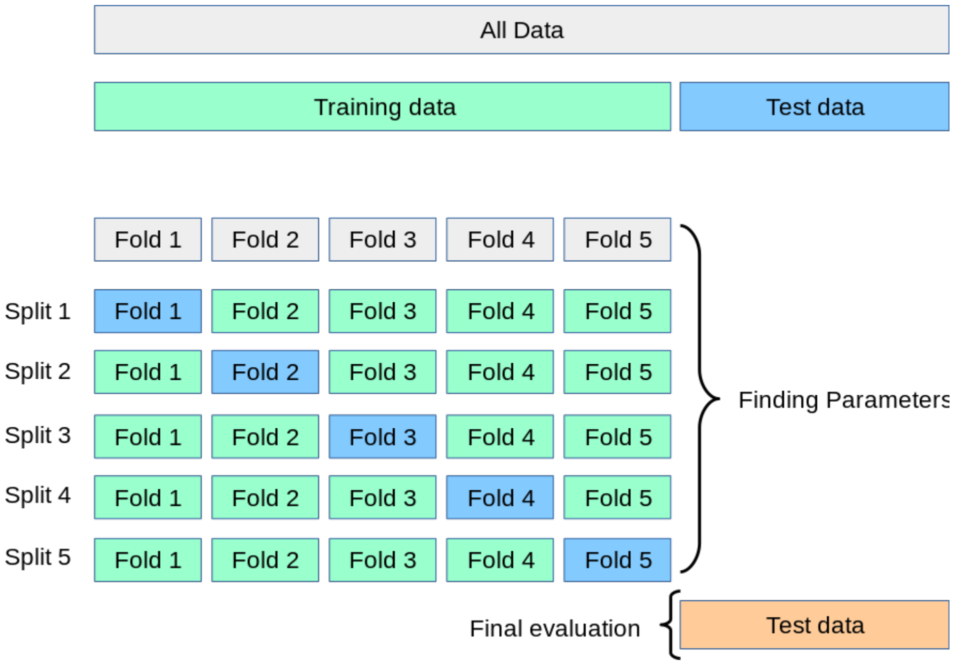


Figure 4-1: K-fold Cross-Validation: we use K-fold cross validation with 10 folds when assessing the success of classifiers to avoid over-fitting. Data is split into 10 folds. For each fold, a classifier is trained on the other 9 folds and the score with which it classifies the withheld fold is computed. We average these scores to ensure a classifier and its parameters are a good fit for the data, rather than finding a classifier that only performs well on the test set (this visual aid was obtained from scikit-learn's documentation [1]).

several different classifiers on the training set is assessed using 10-fold cross validation (this process is visualized in Figure 4-1). The highest scoring classifier is then used to predict the labels on data from the testing set, and an accuracy score is computed from the predicted and ground-truth labels. All classifiers are obtained from the scikit-learn package.

As the stimulus moves across the entire screen (which takes up one of the tanks walls), there is no single point to calculate an error angle from. We consider two different approaches to finding a point of interest with which we can calculate angle errors to use for classification. First, we use the absolute error angle between the visual axis vector and the vector from the closer eye to the center of the screen the stimulus is presented on. As opposed to just using the magnitude, the sign tells us in which direction the visual axis is unaligned. Second, we treat the center of the left

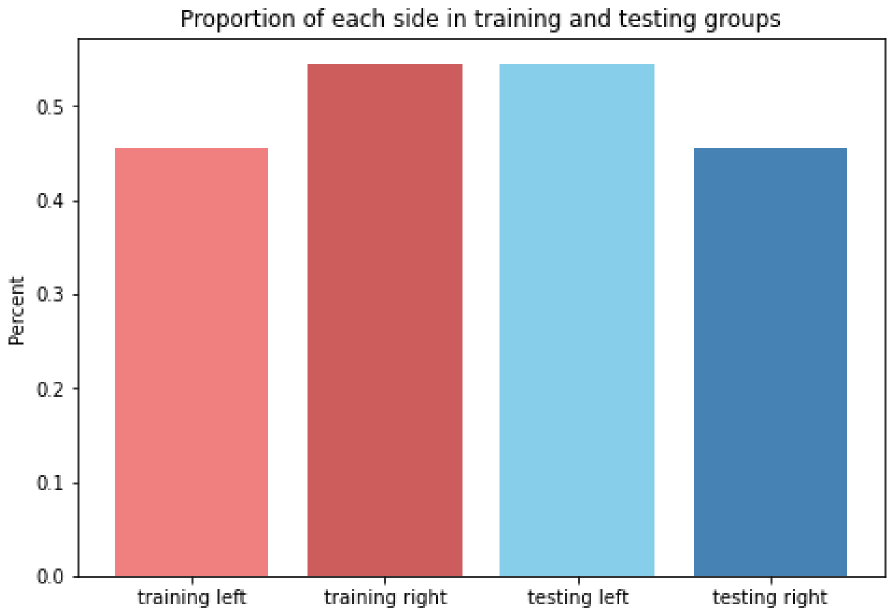


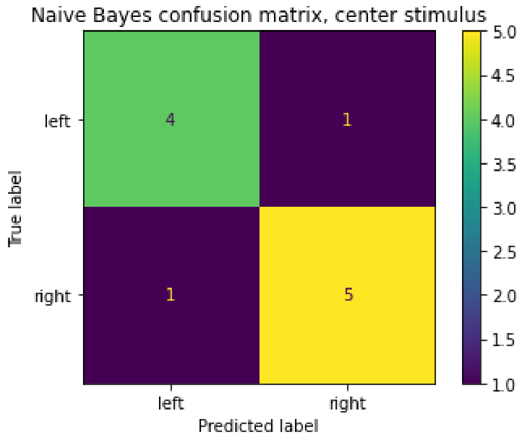
Figure 4-2: The proportion of left trials and right trials in the training and testing sets.

half of the screen and the center of the right half of the screen like two stimuli. We calculate the error angle to both stimuli, and use both angles when classifying which direction the stimulus is presented.

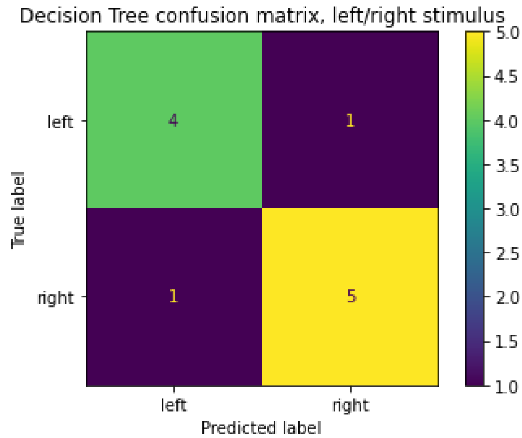
4.2.1 Initial n frames of each trial

We begin by classifying which side the stimulus is presented on using the error angles computed for the first n=12 frames for each trial (n was computed by finding which number of features yields the highest cross validation score). Though increasing the number of frames does not yield better classification scores, it is important to constrict the number for additional reasons. The octopus begins trials in the same place, but can move throughout the tank as a trial goes on. Using only a small amount of frames ensures the octopus is in approximately the same location each time when calculating the angles. The 44 training and 11 testing trials mean 44 rows for training and 11 for testing.

First we utilize the absolute error angles to the center of the screen and train a model on each of our ten classifiers. In this instance Naive Bayes has the highest



(a) Confusion matrix from classifying the first n error angles from the octopus to the center of the stimulus screen for each trial.



(b) Confusion matrix from classifying the first n error angles from the octopus to both the left and right halves of the stimulus screen for each trial.

cross validation score of 0.665 (+/- 0.203). The testing accuracy with this classifier is 0.545.

	Nearest Neighbors	Gaussian Process	Neural Net	Naive Bayes
cross val mean	0.570000	0.615000	0.580000	0.665000
cross val std	0.211187	0.173277	0.161555	0.168893
test accuracy	0.818182	0.545455	0.818182	0.818182

Next we utilize the error angles to the left half and right half of the screen for the first 12 frames and train a model on each of our ten classifiers. In this instance Decision Tree has the highest cross validation score of 0.67 (+/- 0.174). The testing accuracy with this classifier is 0.727.

	Gaussian Process	Decision Tree	Naive Bayes	QDA
cross val mean	0.620000	0.755000	0.665000	0.645000
cross val std	0.164621	0.199311	0.202546	0.173853
test accuracy	0.818182	0.818182	0.818182	0.727273

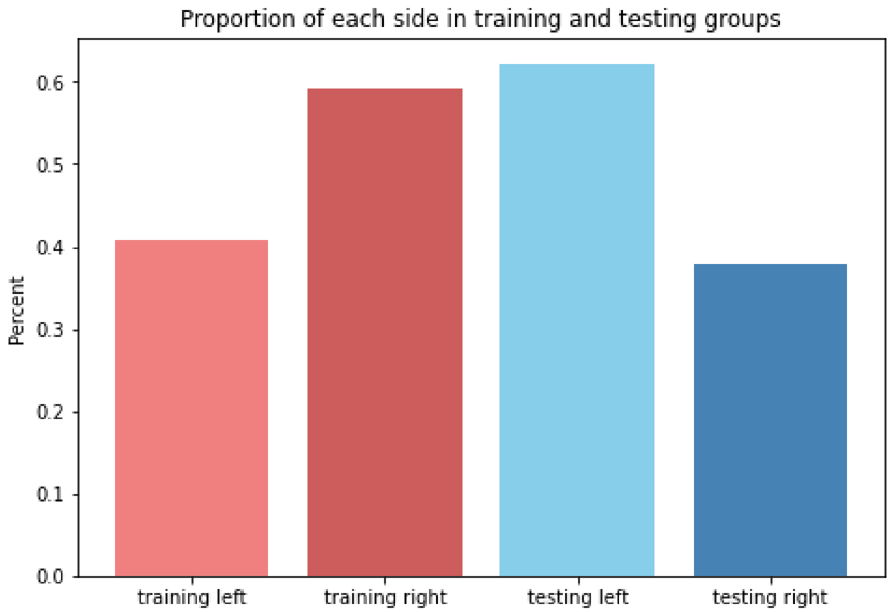


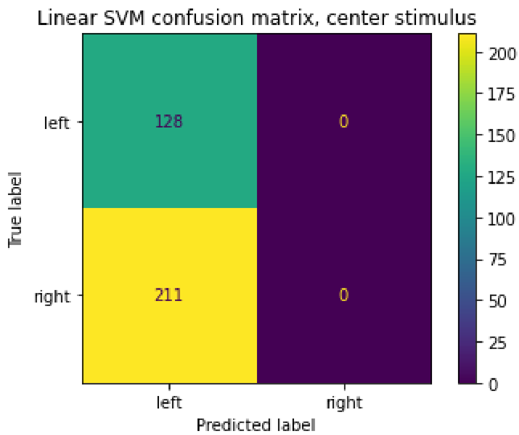
Figure 4-4: The proportion of left trials and right trials in the training and testing sets using all frames 300 pixels or more away from the stimulus.

The hyper-parameters for the displayed classifiers are as follows: 5 neighbors are used for K-Nearest Neighbors, the scikit learn default RBF kernel is used for Gaussian Process, an alpha of 1 and 1000 max iterations are used for the Multi-Layer Perceptron (Neural Net), no parameters are specified for Naive Bayes, a max depth of 5 is used for the decision tree, and none are specified for Quadratic Discriminant Analysis.

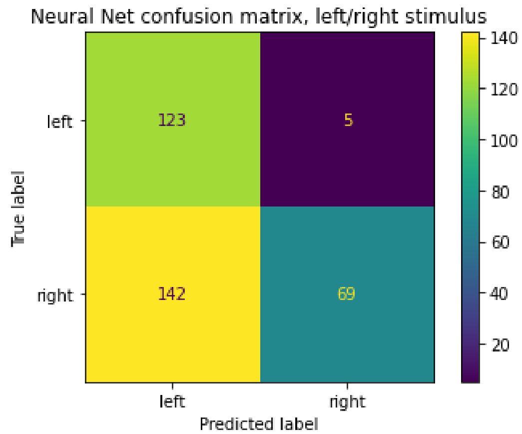
4.2.2 Individual frames

Next we classify which side the stimulus is presented on using the error angle for each individual frame. To ensure the octopus' orientation towards the stimulus is not confounded by its proximity to the side of the screen it touches, only frames 300 or more pixels away from the screen are utilized. There are 1972 frames greater than 300 pixels away in the training trials and 339 frames in the testing trials, yielding 1972 rows for training and 339 for testing.

First we utilize the absolute error angles to the center of the screen and train a model on each of our ten classifiers. In this instance Linear SVM has the highest cross validation score of 0.592 (+/- 0.032). The testing accuracy with this classifier



(a) Confusion matrix from classifying all error angles from the octopus to the center of the stimulus screen when the octopus is at least 300 pixels away.



(b) Confusion matrix from classifying all error angles from the octopus to both the left and right halves of the stimulus screen when the octopus is at least 300 pixels away.

is 0.525.

	Linear SVM	Neural Net	Naive Bayes	QDA
cross val mean	0.592293	0.587722	0.590783	0.590783
cross val std	0.002084	0.034862	0.032326	0.032326
test accuracy	0.377581	0.525074	0.525074	0.525074

Next we utilize the error angles to the left half and right half of the screen for each frames and train a model on each of our ten classifiers. In this instance Neural Net has the highest cross validation score of 0.61 (+/- 0.034). The testing accuracy with this classifier is 0.501.

	Linear SVM	Gaussian Process	Neural Net	QDA
cross val mean	0.597880	0.592852	0.607517	0.604986
cross val std	0.028388	0.093955	0.037524	0.033642
test accuracy	0.501475	0.587021	0.566372	0.501475

The hyper-parameters for the displayed classifiers are as follows: a C of 0.025 are used for Linear SVC, an alpha of 1 and 1000 max iterations are used for the Multi-Layer Perceptron (Neural Net), no parameters are specified for Naive Bayes, none

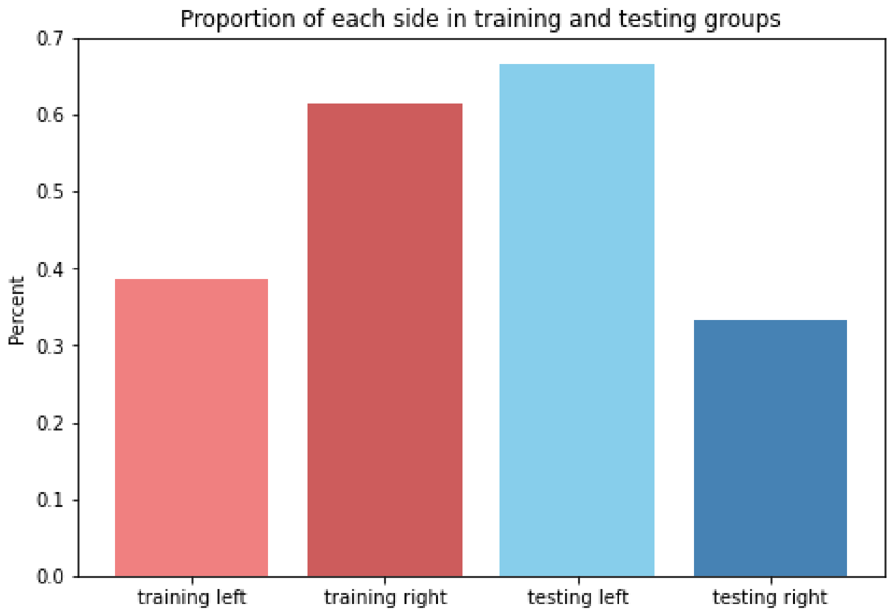


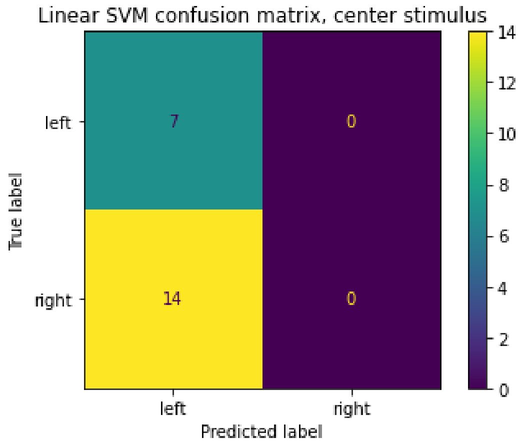
Figure 4-6: The proportion of left trials and right trials in the training and testing sets using segments of 12 frames 300 pixels or more away from the stimulus.

are specified for Quadratic Discriminant Analysis, and the scikit learn default RBF kernel is used for Gaussian Process.

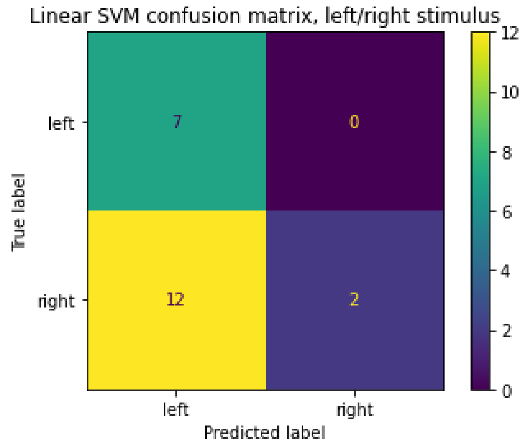
4.2.3 Segmenting frames

Finally, we classify which side the stimulus is presented on based on the error angles from groups of n=12 frames; all groups of frames are utilized until the octopus is within 300 pixels of the screen. Once again, n is obtained from testing a range of values for the best classification score. There are 150 groups of 12 frames from the training trials and 21 groups from the testing trials, yielding 150 rows for training and 21 for testing.

First we utilize the absolute error angles to the center of the screen and train a model on each of our ten classifiers. In this instance Linear SVM has the highest cross validation score of 0.613 (+/- 0.093). The testing accuracy with this classifier is 0.476.



(a) Confusion matrix from classifying segments of 12 error angles from the octopus to the center of the stimulus screen when the octopus is at least 300 pixels away.



(b) Confusion matrix from classifying segments of 12 error angles from the octopus to both the left and right halves of the stimulus screen when the octopus is at least 300 pixels away.

	Linear SVM	Gaussian Process	Neural Net	QDA
cross val mean	0.613333	0.573333	0.573333	0.586667
cross val std	0.026667	0.130639	0.080000	0.093333
test accuracy	0.333333	0.476190	0.523810	0.476190

Next we utilize the error angles to the left half and right half of the screen for each frames and train a model on each of our ten classifiers. In this instance Neural Net has the highest cross validation score of 0.61 (+/- 0.034). The testing accuracy with this classifier is 0.501.

	Nearest Neighbors	Linear SVM	Decision Tree	AdaBoost
cross val mean	0.566667	0.626667	0.560000	0.566667
cross val std	0.108525	0.053333	0.161107	0.143759
test accuracy	0.619048	0.428571	0.476190	0.619048

The hyper-parameters for the displayed classifiers are as follows: a C of 0.025 are used for Linear SVC, the scikit-learn default RBF kernel is used for Gaussian Process, an alpha of 1 and 1000 max iterations are used for the Multi-Layer Perceptron (Neural

Net), no parameters are specified for Naive Bayes, none are specified for Quadratic Discriminant Analysis, 5 neighbors are used for K-Nearest Neighbors, and none are specified for AdaBoost.

4.2.4 Removing failures

As the data is of the octopus being conditioned to go to the correct side and it has not yet learned the paradigm, the octopus often goes to the incorrect side. We tried removing these failure trials and running the same analyses, so the octopus always goes to the side the stimulus is presented on.

Using the first n=12 frames of each correct trial, there are 25 training and 5 testing trials, yielding 25 rows for training and 11 for testing. For the absolute error angle, Naive Bayes has the highest cross validation score of 0.817 (+/- 0.221), which classifies with a testing accuracy of 0.4. For the left and right error angles, Linear SVM has the highest cross validation score of 0.85 (+/- 0.245), which classifies with a testing accuracy of 0.0.

Using individual frames, there are 1179 training and 102 testing frames, yielding 1179 rows for training and 102 frames for testing. For the absolute error angle, Linear SVM has the highest cross validation score of 0.655 (+/- 0.253), which classifies with a testing accuracy of 0.490. For the left and right error angles, Linear SVM has the highest cross validation score of 0.67 (+/- 0.196), which classifies with a testing accuracy of 0.716.

Using segments of frames, there are 90 training and 8 testing frames, yielding 90 rows for training and 8 for testing. For the absolute error angle, Nearest Neighbors has the highest cross validation score of 0.644 (+/- 0.161), which classifies with a testing accuracy of 0.625. For the left and right error angles, Nearest Neighbors has the highest cross validation score of 0.633 (+/- 0.143), which classifies with a testing accuracy of 0.25.

Chapter 5

Discussion

5.1 Project evaluation

This section evaluates the success of this implementation of the project. The setup and deep learning components are successful based on the criteria they can be evaluated on, however it is difficult to evaluate the full system's success with the shortcomings of the training data. Nonetheless, we interpret the results to evaluate what we can and inform future efforts for a more successful implementation.

5.1.1 DeepLabCut

Despite the footage being filmed from a couple feet above the surface of the water, the DeepLabCut model is able to label data with a relatively low error. The average error between the labeled locations of the points used to train the model against their predicted locations is less than the average error for the labeled points reserved for testing, but this finding is to be expected. The error of less than three pixels for the testing data is less than the error DeepLabCut attributes to expected discrepancies in human labeling for their models, and the predicted locations appear to be reasonable estimates as it is sometimes even difficult for a human to label locations consistently [6].

Despite the error of less than three pixels that results from footage recorded out of water being within usable range, the effect the distortions from the surface of the water could have on the predicted visual axis vector should not be underestimated. If ripples on the surface of the water distort the locations of the eyes, particularly if their perceived locations are distorted in opposing directions, the visual axis could be miscalculated by a modest angular difference. Filming footage from underwater, as we suggest in the **Approach** section, would not only eliminate this possibility, but almost certainly decrease the pixel error in eye location labeling.

5.1.2 Classification

None of the classification results obtained are close to being accurate enough for use in a research setting, however useful findings can be drawn from the impact different classification approaches had on accuracy.

Shortcomings from the data

Data used to train a classifier to determine if an octopus is attending a stimulus requires ground truth data where it is known when an octopus is and is not looking at a known stimulus. An ideal stimulus would not only be reliably salient to the octopus, but would also be small enough to be approximated by one point in space; the stimulus in the data meets neither of these criteria. An octopus can be conditioned to pay attention to a random stimulus paired with a reward, but the octopus has not been conditioned to the stimulus in this data. Throughout all 55 trials, the octopus barely goes to the side it is queued towards more than by chance (30/55) and continues going to the incorrect side often by the last day of data collection. Even if the octopus learned the paradigm, the location of the stimulus cannot be approximated by a single point in space. The stimulus drifts across the entirety of the stimulus screen, queuing the octopus to the direction it is moving towards. The octopus could look at any point on the screen to tell which direction it should move in and it is only an assumption they would look in the direction they are heading (though octopuses look in the direction of prey they are tracking, that requires more accurate vision and the tracking of an object moving in unpredictable directions).

The setup of the experiment has several flaws for eye tracking. Footage is recorded from outside of the water, so the locations of the eyes in relation to each other may be misrepresented and there is a greater chance for mislabeling by the model or a human with less detailed footage. Finally, the location of the stimulus in relation to the octopus would ideally be much closer. As the octopus is further from the stimulus screen, fewer degrees of their visual field are needed to encompass the entire screen; this means if the octopus is far from the screen there is a much smaller difference in the angle of the visual axis focused on an object on the left side of the screen and the right side of the screen, than if the octopus were closer to the screen. In our ideal experimental setup, the octopus is constrained to a box that is near the stimulus screen. It is big enough to allow the octopus to move around freely, but small enough such that the octopus is never far from the screen.

Though there was sufficient data to train an accurate DeepLabCut model, there was not enough data to evaluate all the classification approaches. Utilizing the *first n frames* approach the standard deviation for the most accurate cross validation score never got below 0.15, and was consistently around 0.1 for the segmenting approach. The best *single frame* models had standard deviations around 0.03, which were presumably lower because of more training data.

Removing data from trials where the octopus went to the wrong side helped somewhat with cross validation scores, however these did not translate well to the testing set (likely due to discrepancies between the proportions of labels in each group). This treatment may have been helpful with more data but there simply wasn't enough to train with to begin with, much less so after throwing out about half of the trials.

As can be seen from the confusion matrices for the individual and segmenting approaches, it is apparent the classifiers mostly learned to predict the same side nearly every time to get the highest accuracy (most likely due to a skew in the distribution of each side in the training set). The group labels should have been accounted for when running K-means cross-validation, and the skew should have been incorporated into the final classifier choice to avoid the poor testing performance.

Potential shortcomings of the approach

There are a few shortcomings to the approach that could be problematic even with an ideal data set. Even with our limited knowledge of the visual field size of octopuses, the location of their two eyes alone cannot be enough to know the orientation of their visual field. While the two eyes are located on either side of the brain and do not move in relation to the brain (unlike less rigid parts of their bodies), the eyes can rotate from side to side and can do so independently [5]. Though they would likely orient themselves so their eye is not maximally rotated in one direction while tracking a stimulus, this remains a confound to be aware of. Fortunately for our purposes octopuses always keep both eyes level horizontally and even have trouble seeing when they are not, so it can safely be assumed their eyes are vertically aligned with each other [5].

Another assumption that is unlikely a certainty is that the line going through the two eyes is aligned with the center of the visual axis of each eye. Photo-receptor densities on the retina do not indicate any horizontally lateralized specialization of acuity and no measurements are present indicating the angular difference between the projection of the visual axis of the two eyes. While this uncertainty must be incorporated into a future gaze estimator, it is more-so an opportunity than an obstacle. Intentionally collected data would have an octopus fixating on a stimulus as the stimulus moves around the octopus' environment. For data where the octopus is attending to the stimulus, the angle difference between the projection of the two eyes (what we use to approximate the visual axis) and the vector from the attending eye to the stimulus can be analyzed (this angle is identical to our calculated error). If the error angle is consistent, the visual axis likely is off-set from our calculated visual axis by this angle (a finding that would be useful for calculating a better error metric and consequently for classifying where an octopus is attending). If the error angle is not consistent, octopuses likely do not need to orient their visual axis with a stimulus to attend to it; while this option seems less ideal for tracking a moving object (as an object on the periphery of the visual field could more easily escape the visual field entirely), it would be consistent with the elevated density of photo-receptors in a stripe across the retina of octopuses.

A final consideration to make in training a classifier is that octopuses will inevitably become uninterested in a stimulus overtime if it is not engaging to them. For example, an octopus will reliably and robustly direct its attention to tracking a moving crab. However, if the crab is on the other side of a Plexiglass box and the octopus will never be able to get the crab, it will lose interest and stop focusing on the crab. Assuming that an octopus is attending a salient stimulus just because it is in their visual field could easily pollute a good data set and make it more difficult to classify where their attention is directed. Care should be taken to ensure a stimulus is both salient and that the octopus is attending to it in an ideal data set.

Insights

The results from this implementation are insufficient for use in a study as a tool, but provide important considerations for developing a research asset with future implementations. The amount of data utilized was adequate for training a deep learning model to track eye locations, and likely even less would be necessary with more focused footage, but was not enough to train classifiers for stimulus attending. Consolidating segments of frames together increased classifier accuracy, but meant less data was available for training and a more unpredictable model. Segmenting should be done in the future when classifying, but many more segments are needed than the couple hundred that were tested. Another insight that may seem obvious is that more points are better than fewer; classifying the angular error calculated for multiple points increased accuracy. For instance, this sentiment means the left and right borders of an object should be used to calculate two errors, as opposed to just the center, when classifying.

5.2 Applications

There are endless research applications for an eye tracking system like we propose. The most straight forward applications are simply verifying if an octopus can track a stimulus. For instance, if an octopus is able to track a moving second order stimulus, it must be able to perceive second order motion. More possibilities emerge when conditioning is introduced. Training an octopus to focus on one of two stimuli based on a learned rule could demonstrate much about their cognition. For example, an octopus could be conditioned to track one of two groups of objects that has more objects in it; once the octopus has learned the rule to track the group with the greater number of objects, the difference between the quantity can be decreased to ascertain their numerical cognitive faculties. A sufficiently accurate gaze predictor could even be used to evaluate the viability of stimuli for use in studies, such as exploring whether or not octopuses track a moving virtual crab as they would track a real moving crab.

5.3 Future work

As mentioned throughout the paper, the lack of data intentionally collected for use in this project makes it difficult to fully assess the efficacy of the proposed system. With the code and the framework in place, it remains to be seen how effectively the system will be able to track the gaze of an octopus with future research.

Another promising, though rather different, approach to tracking octopus gaze would be to track the pupils of octopuses from up close. Human eye trackers utilize this approach, as well as those in most other species (including those with laterally placed eyes [10]). Octopuses would be unlikely to respond well to being forced to remain still, but putting them in an object like a pot with a single peephole for them to look out of may prove feasible, as they naturally like to hide and observe their surroundings while sheltered [2].

Lastly, there is simply a lot of research that needs to be conducted to better understand the visual systems of cephalopods. No data exists for visual field estimates or detailed eye measurements for *Octopus bimaculoides*, and there is little information for any species of octopus. Retinotopy in octopus brains has scarcely been studied, but could reveal a lot about the specialization of their visual perception. Lastly, do octopuses exhibit specialization in each eye for different tasks, such as stalking prey in one eye and running from predators with the other? There is research to indicate this lateralization does exist in cuttlefish [5]. The field of cephalopod vision research is young, but many of these research questions could be answered relatively easily.

Chapter 6

Conclusion

This thesis introduces an autonomous framework for predicting the orientation of an octopus' gaze from video footage. The research utilization of the non-invasive tool could be used to expedite data collection for existing research projects and make the study of hitherto un-researched topics possible, while the methodology behind the tool itself yields insights into the foundation of the octopus visual system.

This instantiation of the project presented utilizes data from the training phase of an unrelated project where necessary ground-truth data on the target of an octopus' attention is undecipherable, but nonetheless demonstrates the feasibility of the approach, describes a data-driven outline for the creation of an intentional data set for successful octopus gaze tracking, and provides all of the necessary software underlying the framework in four well documented python files.

While this resource could certainly be used to elucidate basic parameters of the octopus visual system that have yet to be considered, anatomical studies finding these values through measurements are long overdue. Hopefully this project can spur others to begin researching these questions more and the tool, once validated on an intentional data, can enable more research into cephalopod vision.

Bibliography

[1] scikit-learn: 3.1. cross-validation: evaluating estimator performance.

[2] Ruth A Byrne, Michael J Kuba, and Daniela V Meisel. Lateralized eye use in octopus vulgaris shows antisymmetrical distribution. *Animal Behaviour*, 68(5):1107–1114, 2004.

[3] David B. Edelman and Anil K. Seth. Animal consciousness: a synthetic approach. *Trends in Neurosciences*, 32(9):476 – 484, 2009.

[4] Peter Godfrey-Smith. *Other minds: The octopus, the sea, and the deep origins of consciousness*. Farrar, Straus and Giroux, 2016.

[5] Frederike D. Hanke and Almut Kelber. The eye of the common octopus (octopus vulgaris). *Frontiers in Physiology*, 10:1637, 2020.

[6] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281, 2018.

[7] Amy M. Ryan, Sara M. Freeman, Takeshi Murai, Allison R. Lau, Michelle C. Palumbo, Casey E. Hogrefe, Karen L. Bales, and Melissa D. Bauman. Non-invasive eye tracking methods for new world and old world monkeys. *Frontiers in Behavioral Neuroscience*, 13:39, 2019.

[8] Jeanne M Serb and Douglas J Eernisse. Charting evolution's trajectory: using molluscan eye diversity to understand parallel and convergent evolution. *Evolution: Education and Outreach*, 1(4):439–447, 2008.

[9] Christopher M Talbot and Justin N Marshall. The retinal topography of three species of coleoid cephalopod: significance for perception of polarized light. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1565):724–733, 2011.

[10] Luke P. Tyrrell, Shannon R. Butler, Jessica L. Yorzinski, and Esteban Fernández-Juricic. A novel system for bi-ocular eye-tracking in vertebrates with laterally placed eyes. *Methods in Ecology and Evolution*, 5(10):1070–1077, 2014.