Underwater Behavior Classification in bottlenose dolphins (Group 11)

Yu-Mu Chen A53311606
Zi-Xiang Xia A59002979
Margaret Kawabata A11088327

Abstract

Classifying animal behaviors on large sets of observational data manually requires intensive human labor along with risks of error even after reaching consensus from inter-rater concordance. One strong candidate that can potentially overcome these limitations is through applications of deep learning on video analysis. Temporal Relation Network (TRN), a deep network module, was designed to analyze frame-by-frame videos that correlates meaningful transformations of the focal object/animal [2]. Thus, the purpose of this project is to investigate the applications of TRN on recognition of different behaviors of bottlenose dolphins Tursiops truncatas, specifically on surfacing behavior.

1. Introduction

Deep learning has long been contributing to animal behavior, especially in rodents like mice used to facilitate research in neuroscience and pharmacology. Despite its endless applications, implementations on cetacean behavioral ethology are rare due to its complications from their underwater habit. These complications can simply be solved in captive settings using underwater cameras, but the traditional method in collecting data has long required cetacean ethologist to do focal follows after strict protocol trainings to detect their behavior. Thus, the motive of this project to ease the need for intensive human labor for observational data collection in cetacean study, specifically the bottlenose dolphins *Tursiops truncatas*. In order to do so, the goal of this project is to detect a specific behavior of dolphins using temporal relation network (TRN) [2], a deep network module designed to analyze frame-by-frame videos that correlate meaningful transformations of the focal animal . The behavior attempted to detect is 'surfacing', which is an indicator for breathing, a commonly studied behavior known to be indicative of social behavior within pods [6, 7]. The input to our algorithm is be a 10 fps color video of underwater footages of seven dolphins housed in Brookfield Zoo, Chicago. TRN was then used to classify the behavior to be either surfacing or not surfacing. The main contributions of this project are summarized in the following:

- We collect our dataset which compose of 600 positive and 567 negative clips of dolphin surfacing.
- We experiment with different CNN backbone models to compare their performance on our collected dataset.
- We divide our dataset into "easy" and "difficult" data. We compare the results from CNN backbones which are trained mostly using easy data and tested only on the difficult data.

2. Related Work

There are a few existing approaches that analyze animal behavior using deep neural networks. For instance, there has been successful Utilization of CNN, convolutional neural network, on three behavioral detection; detection of the location of mice paws placed, detection of the locations of two mice in a shared space, and an estimation of human poses [8]. Furthermore, there is a survey paper that discusses two ways of capturing the animal behavior, one by tracking and the other by estimating pose [9]. However, these approaches mainly focus on analyzing the location and the kinematics of the animals. For our project, we want the model to not only learn the position of the dolphins but also recognize its behavior. A similar problem was attempted in the aims to classify whether a drosophila, a type of fruit fly, is on a substrates in each video frame or not, through the use of simple CNN classifier to process the position of the fly and the image of each frame [10]. Nevertheless, instead of classifying the whole video clip, they only analyze the behavior at each timestamp. Therefore, we solve our problem by using temporal relation network (TRN), which can learn the motion-relation between each video frame and perform classification on the dolphin behaviors.

3. Dataset and Features

In this project,the focal animal as well as its behavior is very specific, and no proper dataset suitable for direct application could be found. Therefore, we decided to collect the dataset from video footages of dolphins in captivity at Brookfield Zoo Chicago, borrowed from Christine Johnson lab at the Cognitive Science department at UCSD. We then edited the footage into numerous 4 to 7 second clips, in which some contain one or more dolphin performing the surfacing behavior.

Surfacing event, the behavior to be detected, is defined

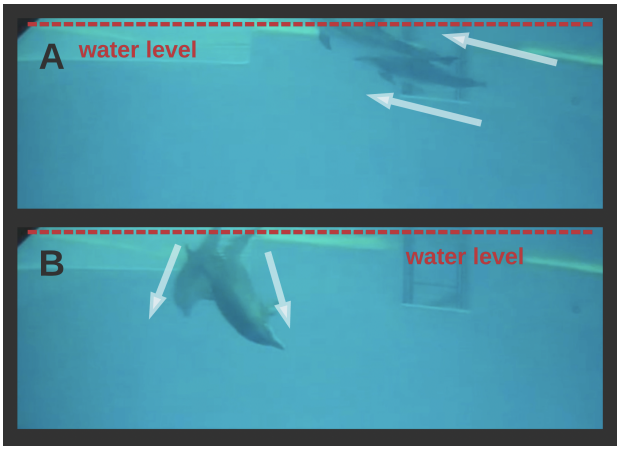

Figure 1. The red dotted lines indicates the water level. A shows the upwards projection of two dolphins indicated by the white arrows. B shows the downwards projection indicated by the two arrows.

by an upwards project to the top of the screen towards the water level, immediately followed by a downwards projection. A visual aid for this behavior can be seen in Figure 1. We then divide our collected dataset into two categories: positive and negative, where positive clips where at least one surfacing event is present and negative where surfacing even is absent. Overall, we have successfully collected a total of 600 positive clips and 567 negative clips. The clips along with the raw video can be found in this link: Google Drive link.

After all clips have been collected and labeled, we split the them into single frames in order to put them into the model for training.

4. Methods

The most critical difference between our project and the aforementioned related work is that aim to extract information from a video clip while others mostly do so on a single frame. To do so, we utilized a network specifically designed for extracting video information: Temporal Relation Network (TRN).

Figure 2 depicts the overall architecture of the model. The frames of a clip is first sampled into different scales, where a scale consists of different amount of frames. For example, in Figure 2, frames 1 and 9 are sampled in the 2-frame scale as well as frames 5 and 9. Frames 3, 8, and 12 are sampled in the 3-frame scale, and so on. Note that the frames are sampled in order so that the CNN can correctly "analyze" the change over time.

After the frames are sampled, we put them into a CNN that extracts the features of the frames, and fuse them together as the output of the part. For instance, the temporal relation between the 2 frame parts is defined as a composite function as follows:

$$T_2(V) = h_\phi \left(\sum_{i<j} g_\theta(f_i, f_j) \right)$$

Backbone	BNInception	InceptionV3	ResNet101
Accuracy	86.92%	88.82%	88.82%

Table 1. Results of different CNN backbones using ordinary data splititing strategy.

Where g_θ and h_ϕ are multi-layer perceptrons (MLP). Then, after all the frame scales are computed, they are summed together to produce a final output for classification. For this project, we chose the frame scale to be from 2 to 8, which is to say, the parts that we feed into the CNN can contain 2 to 8 frames. Finally, we tweaked the number of classes to 2, representing positive and negative, to fit our problem.

As for the CNN portion of the method, we conducted experiments on three different backbones: BNInception, ResNet101, and InceptionV3. The two Inception structures are wider, utilizing different sizes of kernels in a layer, while ResNet used residual blocks to build a deeper network with more layers. The difference between BNInception and InceptionV3 lies in the usage of 7x7 factorized kernel, and a label smoothing function that prevents the model from predicting with too much confidence.

5. Experiment Result

In the first section, we train and evaluate our model using the ordinary way of splitting training and testing set. In the second part, we do further discussion on our defined *easy and different dataset*. Here, we look at our experiment results. We randomly split our collected data into training, validation, and testing sets, and we experiment with three different backbones to compare their performance. For evaluation matrices, we simply compute the mean accuracy of the predicted results.

The results of the testing set are shown in the table below. We can see that the model can perform great detection from the video clips with at least 86% testing accuracy, and using the Resnet and InceptionV3 have the highest accuracy of 88%.

The InceptionV3 has a higher accuracy because it's essentially an improved version of the BNInception, where they perform batch normalization in the fully connected layer of the auxiliary classifier. As for resnet101, it has a deeper convolutional structure, which helps the model to gather more informative visual features.

Ordinary: We use the ordinary strategy for training and testing a dataset, which is randomly split our collected data into 70%, 20%, 10% for training, validation, and testing sets, respectively. After the data preparation, we experiment with three different CNN backbones to compare their performance. For evaluation matrices, we compute the mean accuracy of the predicted results with the ground truth label. The results are show in Table 1. We can see that the model can perform great detection from the video clips with at least 86% testing accuracy, and using the Resnet and InceptionV3 have the highest accuracy of 88%. We think that the InceptionV3 has a higher accuracy because it is essentially an

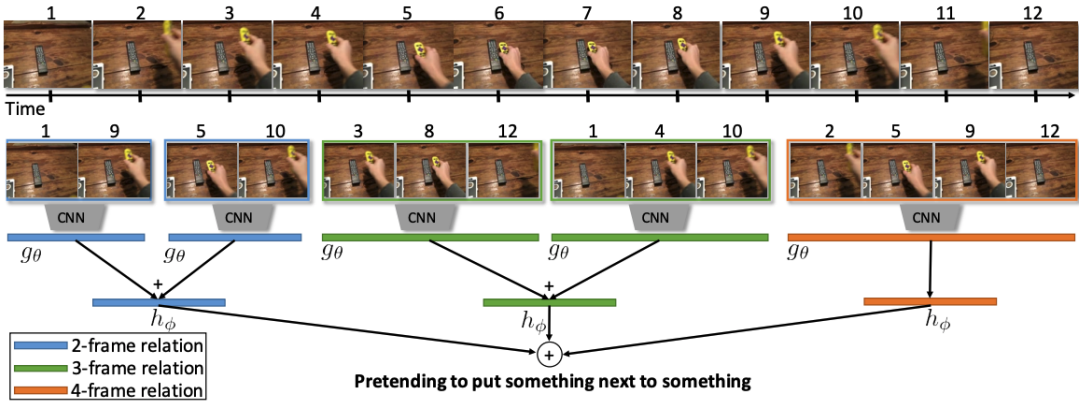

Figure 2. The architecture of Temporal Relation Networks. Here we use multilayer perceptrons (MLP) as h_ϕ and g_θ respectively to fuse features of different ordered frames.

Easy dataset: animals are close to the camera allowing clearer view of their behavior.

Positives: Surfacing that have a clear upwards and a downwards projection

Negatives: Bottom swimming where the dolphins are no where near the surface.

Difficult dataset: animals are further away from to the camera and behaviors are blurry AND OR ultiple dolphins are in view.

Positives: Although blurry, the upwards and downwards projejection can be seen. False negatives account for this too, where the upwards projection or downwards projection are slightly off frame.

Negatives: Blurry vision of bottom swimmers and false positives, where dolphins are projecting up to linger (staying at the same location for an extended period of time) near the surface, resembling a surfacing.

Figure 3. The definition for our easy and difficult dataset.

improved version of the BNInception, where they perform batch normalization in the fully connected layer of the auxiliary classifier. As for resnet101, it has a deeper convolutional structure, which helps the model to gather more informative visual features.

Easy and Difficult dataset: During our data collection, we discover that there are a few ambiguous or unclear clips. Therefore, we conduct a further experiment on our dataset. We separate our data into 2 different categories: "easy" and "difficult". In general, we classify them by sorting the relative difficulty in detecting the surfacing, and the definition is listed in Figure 3.

After we divide the data based on the definition above,

ResNet101		InceptionV3	
Validation	Testing	Validation	Testing
90.31%	72.38%	91.13%	80.46%

Table 2. Results of different CNN backbones using easy data and difficult dataset. The testing accuracy comes from difficult data only, and the validation accuracy comes from easy data and the rest of difficult data from testing.

we want to see if the model can learn the motion flow from the easy data to detect the behavior in difficult data. Therefore, we use the same amount of testing set from the experiment above, but we only use the clips within the difficult dataset. The training and validation set is composed of the easy and the remaining difficult dataset. We leverage the backbones which have a higher accuracy from the earlier experiment. The results are shown in Table 2. Not surprisingly, the models have a significant accuracy drop on difficult data only, and higher accuracy when there is easy data involved. However, the InceptionV3 has 80% accuracy, which means that the model can still learn from the easy data if there is a better choice of visual extraction model.

6. Conclusion

In this project, we collect a dataset for dolphin behaviors, which composes of positive clips for dolphins surfacing and negative clips for dolphins that are not. And we leverage the deep learning method to tackle dolphin behavior classification. The models show prominent results to detect if there is surfacing behavior within the clips. Furthermore, we discuss our collected dataset and divide them into easy and difficult data. The result shows that the model can still learn the motion pattern if we choose the right CNN backbone for feature extraction. In the future, adding time sequence information would be another great approach for action recognition.

7. Member Contribution

Zi-Xiang: Set up learning environment, construct basic model, and help with report writing. Margaret: Data collection, help with report writing. Yu-Mu: Data preprocessing, report writing, and help with data collection.

8. Reply to review

8.1. Group 4

Q1 Since the goal is to detect dolphins' surfacing, the data obtained from zoom is too ideal. In the ocean, the light condition varies, or it is dark at all. Does this method still work under the sea?

We do not know if this will work at the sea, since we could not find enough data for dolphins in the ocean. Therefore, the very best we can do is to test this model under the worst scenario we can find, and that is exactly why we conduct the easy/difficult experiment.

8.2. Group 5

Q1 Does a TRN require a video input or do you have to pre-process the video into frames?

When testing, the model is designed to take the video input and internal operations will split it into frames. However, when training we need to pre-process the videos into image frames before starting the training process.

Q2 Was splitting the dataset into easy and difficult subsets a manual process?

Yes, we decide the difficulty of the clip when editing and name the clip with a naming convention to separate them.

Q3 How did you select what CNN backbone models to use? Why was ResNet101 used instead of, say ResNet50 or ResNet152?

From the benchmark comparison of CNN on ImageNet[1], we can see that for top1 accuracy, ResNet50 is 72.1% ResNet101 is 78.25%, ResNet152 is 78.57%, BNInception is 74.8%, and InceptionV3 is 78.8%. For ResNet, we want to pick a model which has comparable accuracy and a fair amount of parameters so that the training doesn't take too long. Therefore, ResNet101 is a better choice.

Q4 Are you able to utilize different CNN backbones within a TRN model?

Yes, that is exactly what we did in this project.

Q5 How do you decide how many frames an individual CNN has?

TRN has a multi-scale system, for the cycle of each scale the CNN samples the number of frames it needs. In our project the scale is 8, therefore it will sample 2 frames, and then it will sample 3 frames, so on until 8.

Q6 How do you decide which frames to use as input to the TRN? The sample video is around 4 seconds (100-120 frames). Do you use all 100 frames or do you select certain frames within those 100 frames?

First of all, the frame rate of these clips are 10fps so the actual frame count is less than the number suggested. Under this circumstance, all the frames can be utilized.

8.3. Group 37

Q1 Why did you choose to use BNInception, InceptionV3, and ResNet101 for feature extraction?

See the response to Q3 in the previous section.

Q2 What is the data input to the model?

As mentioned in the presentation, clips are separated into frames first. The length of these clips vary from 4 seconds to 7 seconds, with a few longer exceptions. The resolution of these clips are 640×360.

Q3 What are the sample sizes of easy and difficult dataset?

Difficult dataset includes 359 clips, and easy dataset includes 808 clips.

Q4 What is the criteria of positive?

As long as there is a surfacing action, we count it as positive. That is, even when there's only 1 dolphin surfacing among 4 dolphins, it's still positive.

Q5 What is the background of determining the easy and difficult dataset criteria?

We define these criteria manually. When editing the clip, if we think the situation might be difficult for the model to recognize, such as multiple dolphins moving around simultaneously, the dolphins are small in the frame, etc., we add it to difficult set. This is further explained in Figure 3.

Q6 Is the model capable of locating the dolphins or just classifying if surfacing occurs?

The fact that TRN can detect whether surfacing happened inside a clip leads us to believe that the CNN is able to capture the location of the dolphins.

[1] https://paperswithcode.com/sota/image-classification-on-imagenet

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[2] Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal relational reasoning in videos. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 803-818).

[3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[4] 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), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[5] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[6] Sakai, M., Hishii, T., Takeda, S., & Kohshima, S. (2006). Flipper rubbing behaviors in wild bottlenose dolphins (Tursiops aduncus). Marine Mammal Science, 22(4), 966-978.

[7] Connor, R. C., Smolker, R., & Bejder, L. (2006). Synchrony, social behaviour and alliance affiliation in Indian Ocean bottlenose dolphins, Tursiops aduncus. Animal Behaviour, 72(6), 1371-1378.

[8] Arac, A., Zhao, P., Dobkin, B. H., Carmichael, S. T., & Golshani, P. (2019). DeepBehavior: A deep learning toolbox for automated analysis of animal and human behavior imaging data. Frontiers in systems neuroscience, 13, 20.

[9] Mathis, M. W., & Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. Current opinion in neurobiology, 60, 1-11.

[10] Stern, U., He, R., & Yang, C. H. (2015). Analyzing animal behavior via classifying each video frame using convolutional neural networks. Scientific reports, 5(1), 1-13.