Deep Convolutional Neural Networks for Shark Behavior Analysis

Wenlu Zhang*, Anthony Martinez*, Emily Nicole Meese†, Christopher G. Lowe†, Yu Yang‡, Hen-Geul(Henry) Yeh§
*Department of Computer Engineering and Computer Science, California State University Long Beach, CA, 90840
†Department of Biological Sciences, California State University Long Beach, CA, 90840
‡Department of Chemical Engineering, California State University Long Beach, CA, 90840
§Department of Electrical Engineering, California State University Long Beach, CA, 90840

***Abstract*—An important step in the study of free-ranging animals is to perform automatic identification and estimation of their natural different behaviors. This task is especially challenging for the species in the aquatic environment, for example, California horn sharks (*Heterodontus francisci*). Because they are relatively small, demersal, and active in the nighttime. It is quite impossible to conduct the observations in a continuous direct way. The shark lab at California State University Long Beach (CSULB) conducted laboratory trials to quantify acceleration signatures of horn sharks for different behaviors including resting, swimming, feeding, and nondeterministic movement (NDM). Currently, most of the existing methods have applied machine learning algorithms to estimate sharks' different behaviors. However, there is still a lack of an efficient and effective way to conduct automatic prediction. In recent years, deep convolutional neural networks have shown the great promise in various computational biology, bioinformatics and neuroscience areas such as biological image analysis, gene expression pattern representation, 3D neuron reconstruction, automatic tumor detection, etc. In this work, we propose novel deep learning models to automatically classify four different shark behaviors using overall dynamic body acceleration (ODBA) through laboratory trial data sets. In specific, we design three deep convolutional neural networks (CNNs) to make fast and accurate predictions and classification. We perform thorough experiments, and the experimental results demonstrate that our proposed models overall produce the best performance than the prior traditional machine learning methods.**

***Index Terms*—Activity Recognition, Shark Behavior, Deep Learning, Convolutional Neural Network**

I. Introduction

Identifying and estimating the movement and behavior patterns of free-ranging animal play an important role in biological and environmental sciences [1], [2]. Especially, the study of activities of marine predators is essential to understand their ecological role in rocky reef habitats and thus has drawn considerable attention [3], [4], [5]. This task is particularly challenging for sharks, for example, California horn sharks (*Heterodontus francisci*). They have relatively smaller size (TL $< 122cm$, [6]). Also they are demersal and nocturnally active in rocky reef habitats. It is quite difficult to conduct the continuous and direct observation over long periods. Currently, less knowledge has been known about their activity patterns, behaviors and ecology. Therefore, automatically characterizing the behavior of sharks is fundamentally important for understanding its habitat and maximizing individual ecological fitness.

One biologging tool, accelerometer data loggers (ADLs), have been widely used to monitor the sharks. ADLs have been successfully measuring the individual activities in velocity over a long period while incorporating the spatial movement data [7], [8], [9]. In specific, ADL can measure dynamic acceleration for each orthogonal axis: surge, heave, and sway for x, y, z dimensions [10], [11]. When combined with spatial movement data, acceleration data can be used to generate a spatial representation of the animal's energetic costs and identify profitable areas to the animal needed to identify an energetic landscape [12]–[14]. Quantifying high-resolution, three-dimensional (3D) acceleration data can provide especially valuable information, particularly as it can be difficult to determine specific behaviors (e.g., resting, swimming, feeding and NDM) from movement paths alone.

Currently, most of the previous work have used machine learning methods to identify and estimate the behaviors of free-ranging animals. Those algorithms usually fall into two categories, supervised learning and unsupervised learning. In the supervised learning, for example, decision tree, random forest, k-nearest neighbor, logistic regression have been explored. In the unsupervised learning, some of the major clustering algorithms have already been implemented. However, these methods either have low accuracy for behavior prediction on validation and test data sets, or are problem-dependent classifiers only providing local features using less training parameters. In [15], the authors proposed artificial neural network (ANN) to classify the behaviors of wild sharks. Particularly, they only used one hidden layer containing 100 neurons to construct fully-connected neural networks. However, they failed to provide more detailed training procedure. And they are quite "shallow", because they only applied two levels of feature extractions. The total number of trainable parameters are quite small. It cannot be considered as deep learning models. In this work, we propose to use deep convolutional neural networks (CNNs) to improve the estimation performance of existing methods on shark behavior analysis. CNN is one of the most popular deep learning algorithms. CNN is a class of methods that can learn hierarchical features through nonlinear mapping. This model is fully trainable and contains

978-1-7281-5650-7/19/$31.00 ©2019 IEEE

TABLE I
STATISTICS OF THE DATA SETS USED IN THIS WORK.THE TABLE BELOW DEMONSTRATES THE TOTAL NUMBER OF DATA POINTS FOR EACH CLASS AND NUMBER OF TOTAL DATA POINTS FOR EACH TRIAL.

Trial	No.1	No.2	No.3	No.4	No.5	No.6	No.7
Feeding	5,700	350	200	1,375	875	2,900	2,100
Swimming	6,200	61,475	5,100	81,525	7,975	19,750	2,100
Resting	157,750	565,580	379,850	10,250	6,150	27,975	2,100
NDM	51,950	7,400	15,700	1,750	1,775	27,025	2,100
Total #	221,600	634,805	400,850	94,900	16,775	77,650	8,400

multiple stacked layers to compute high-level representations between inputs and outputs. CNN has set many records on various vision tasks such as image classification, recognition, object detection, digit recognition, inference and semantic segmentation [16]–[18]. In specific, we build three CNN classifiers that can predict the shark behavior during certain short time point through ODBA data which was calculated by summing the absolute value of the dynamic axes.

II. MATERIAL AND METHODS

A. Horn Sharks Data Preparation

California horn sharks are chosen as a model species to represent a demersal, resting shark that are capable of eating hard-shelled invertebrates [19]–[21]. Laboratory trials were conducted to quantify acceleration signatures of California horn sharks for different behaviors including resting, swimming, nondeterministic movement (NDM) and feeding (prey capture and handling). Sharks were collected from Santa Catalina Island, and transported back to the Shark Lab in CSULB. Each trial was conducted with a single shark in a $340l$ tank with two GoPro video cameras, one mounted on the bottom of the tank, and one above the tank. For each trial, a shark was fitted with an ADL (Cefas G6a+ or Technosmart AxyDepth) via a custom tag package. Sharks were tagged approximately 6 hrs before the start of the trial to allow the shark to acclimate. To characterize motion patterns associated with prey manipulation and handling, instrumented sharks were then fed three different prey types including a live purple urchin ($< 4cm$ test diameter), a frozen and thawed squid (whole, with pen), and a frozen and thawed shrimp. Prey was buried or placed away from the shark in order to characterize natural feeding behaviors expected in the field. A trial concluded when the shark had consumed all trial prey. Trials were approximately 2 hrs long, and a total of 7 trials were done on 4 different individual sharks. Table I shows the statistics of the data sets for the total 7 trials and total number of data points for each behavior during 2 hrs of every trial.

B. Overview of Deep Learning Methods

Deep learning is a subset of the machine learning algorithms derived from neural networks. CNN, one particular architecture of deep learning, tries to simulate the visual signal processing in the vision system of human brain. CNNs usually incorporates combinations of convolutional layers, pooling layers to learn a hierarchy of features using nonlinear mapping. CNNs are revolutionizing artificial intelligence, machine learning and computer vision fields now. CNNs have made superior successes in natural image recognition, object detection, image caption, visual question answering [22], and natural language processing. In addition, CNNs have been used in computation biology, medical image analysis, neuroscience [23], [24]. In this paper, we propose to use CNNs on ODBA dataset collecting from laboratory trials for automatically learn temporal features for time-series classification. Through the proposed CNNs, we obtain the prediction of horn sharks behaviors every short certain time points (2 seconds) in an efficient and accurate way, which significantly outperforms the performance of prior traditional machine learning methods.

C. VGGNet

VGGNet [25] is the extension work of Alexnet [16]. VGG proposed a deeper network including 16 and 19 layers, which is much deeper than Alexnet. In addition, VGG implemented smaller kernels. The authors only used 3×3 filters replacing the large kernels in Alexnet. The main advantage of VGG is to build deeper and more efficient network with less trainable parameters. Previous work on the [26] has demonstrated the efficiency of applying convolutional neural networks to time-series data instead of using hand-crafted features. In this work, we propose three CNN models that automatically learn temporal features for time-series classification. One dimensional (1D) convolutions are used on this univariate data. The first proposed network we name as SharkVGG 1 has 9 layers and is built based on VGGNet. Unlike VGGNet, our model uses 1D convolutions of kernel size 5. The first section of the network consists of 2 stacked convolution layers followed by a max-pooling layer. After each max-pooling layer, the number of filters is doubled. Each convolution is also passed to a rectified linear unit (ReLU) [27] and batch normalization [28]. As shown in figure 1, this section is repeated twice. After the second max-pooling layer, a third set of stacked convolution layers follow with its output passed sequentially to 3 fully connected layers (FCs) of sizes 1×1000, 1×100, and 1×4. Finally the output is passed to a softmax layer.

The second proposed model, SharkVGG 2, is a variation of the first model. In this architecture, the convolution section of SharkVGG 1 is constructed with the different kernel sizes including 3,5, and 7. The output of the last convolution is then concatenated and fed into the FC layers of SharkVGG 1 as shown in Figure 2.

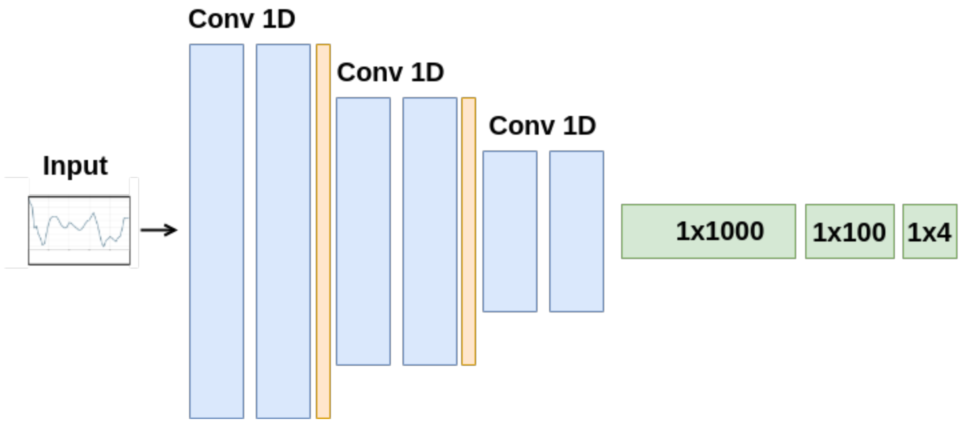

Fig. 1. The architecture of SharkVGG 1 is based on VGG containing 9 layers deep. The input to the network is 25Hz data sampled in 2 second windows. The first section of the network consist of 2 stacked convolution layers followed by a max-pooling layer. After each max-pooling layer, the number of filters is doubled. Each convolution layer is also passed into a rectified linear unit (ReLU) and batch normalization. After the second max-pooling layer, a third set of stacked convolution layers follow with its output passed sequentially to 3 fully connected layers (FCs) of sizes 1×1000, 1×100, and 1×4. Finally, the output is passed to a softmax layer.

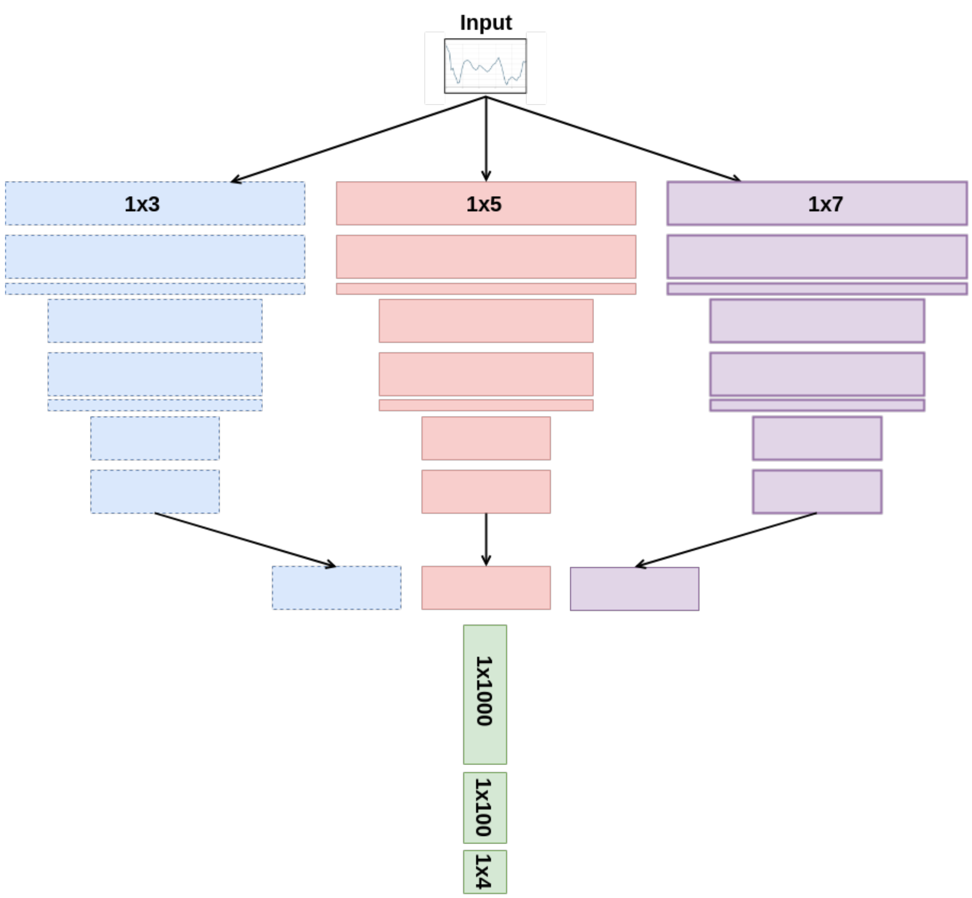

Fig. 2. The architecture of SharkVGG 2. This model has three subnets of the convolutional section of SharkVGG 1, each containing different kernel sizes including 1×3, 1×5 and 1×7 convolutions. The output of each network is then concatenated along their axes. Using the "same" padding, the output size of each CNN will be 1×12. The concatenation operation therefore results in an output size of 1×36. Lastly, the concatenated output is passed into three sequential FC layers.

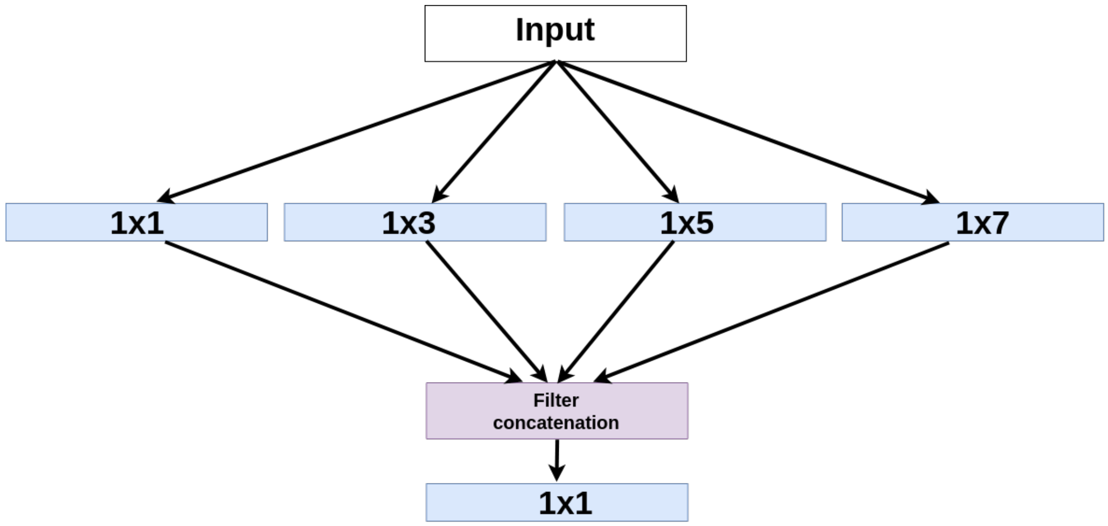

Fig. 3. Inception Module

TABLE II
CLASS SAMPLES

Class	Training	Validation	Testing
Feeding	10,000	2,000	17
Swimming	10,000	2,000	159
Resting	10,000	2,000	123
NDM	10,000	2,000	35
Total	40,000	8,000	334

Each sample is a 2-second window.

D. Inception Learning

The design of smaller kernels can build the much deeper convolutional neural networks, the larger kernels are still needed to capture local receptive filed of large input regions. Inception networks [29]–[31] try to overcome this difficulty by using multiple kernels of different size in parallel. In each inception module, convolutions with small filter size are first inserted to reduce the dimensionality such as the number of featured maps, then convolutions with large filter size are inserted to capture larger local receptive fields. The outputs of different convolutions are aggregated and pass down to the next stage. Such design leads to several advantages. First, it allows the network to process the hidden information at various scales simultaneously. Second, even if the number of units increases significantly in a stage, the computational complexity remains under control at later stages.

We implement the inception module into our third proposed model named as SharkInception in Figure 4. Each blue block in this proposed model consists of three inception modules. In each inception module of Figure 3, the input from the previous block is passed to four 1D convolutions of sizes 1×1, 1×3, 1×5, 1×7. The four different convolutions attempt to capture local features at different scales while also increasing the number on non-linearities. Each convolution is followed by batch normalization and ReLU. The output of each convolution is then concatenated by channel axis. Finally, the concatenated filters are reduced by a factor of 4 using a 1×1 convolution.

III. Experimental Results

A. Experimental Setup

Since each trial and activity class fail to contain the proportional amounts of observations, the data for each class is quite unbalanced. Data augmentation was performed by randomly sampling 2-second windows of time-series data. Also all the seven trails came from four individual sharks. Generated quantities are listed in Table II. Before sample generation, the following trial data sets were selected for training, validation, and testing:

- Training: 1, 2, 3, 4, 7
- Validation: 6
- Testing: 5

Once data sets are separated for their respective purpose, each class activity from the different data sets was aggregated and kept in sequence. Subsequently, each class is treated as one continuous time-series observation. Therefore, training samples can be taken by uniformly sampling one data point from the duration of the class. Once the data point is selected, a 2-second window is taken with the randomly selected point as its center. For the testing data set, random sampling is not used. Instead, a sliding window approach is used where 2-second samples are taken without overlapping data points. All the three models are trained on a workstation with GPU: four NVIDIA Geforce GTX 1080 TI 11 GB GDDR5X, CPU: Intel Core i7-7820X 8 core, and operating system: CentOS 7.

B. Comparison with other methods

We also use the traditional machine learning algorithm to do the comparison. Random forest (RF) classifier is used through the Scikit-learn library. The number of estimators is set to 1000, minimum samples per split are set to 2. Samples for the RF are generated by calculating statistical features from 2-second windows on the ODBA and the fast Fourier transform

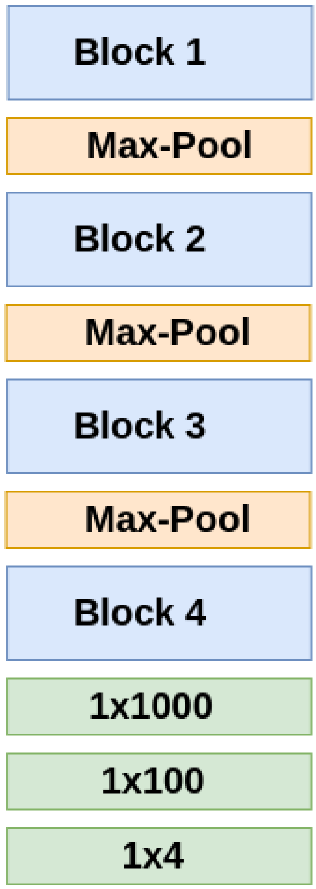

Fig. 4. The architecture of SharkInception. Each blue block consists of three Inception modules, as shown in figure 3. The output of each 1×1 convolution is fed directly into the next module. Each block, except the fourth, is followed by a max-pooling layer. The last block passes its output to three sequential FC layers.

(FFT) data. Calculated features include minimum, maximum, standard deviation, skewness, kurtosis, and mean. The classification performance is measured by training, validation and test accuracy, and precision, recall and F1-score.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

The detailed experimental results is shown in Table III. Our CNN models are significantly better than the shallow methods, random forest. Overall, SharkVGG1 gives the best results for test accuracy, precision recall and F1-score among the four deep convolutional neural networks. Because all the laboratory trails do not have many samples, it will easily fall into overfitting once the network is quite complicated such as inception.

IV. Conclusion and Future Work

In this paper, we aim to improving the classification of shark behavior using deep convolutional neural networks. This goal is achieved by employing several novel deep learning architectures. We compared the performance of our proposed methods with other commonly used traditional machine learning method. Results show that our proposed methods outperform prior methods. Overall, our deep learning models produce more accurate and efficient results on shark behavior classification. In this study, we only implement our models using laboratory trial data sets. In the future, we plan to design recurrent neural networks and attention deep models for real field data set.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] S. J. Cooke, S. G. Hinch, M. Wikelski, R. D. Andrews, L. J. Kuchel, T. G. Wolcott, and P. J. Butler, “Biotelemetry: a mechanistic approach to ecology,” *Trends in ecology & evolution*, vol. 19, no. 6, pp. 334–343, 2004.

[2] J. Metcalfe, S. Wright, C. Tudorache, and R. Wilson, “Recent advances in telemetry for estimating the energy metabolism of wild fishes,” *Journal of Fish Biology*, vol. 88, no. 1, pp. 284–297, 2016.

[3] K. Mann, “Ecology of coastal waters with implication for management blackwell sciences,” *Inc. Massachusetts*, 2000.

[4] Y. Papastamatiou and C. Lowe, “An analytical and hypothesis-driven approach to elasmobranch movement studies,” *Journal of Fish Biology*, vol. 80, no. 5, pp. 1342–1360, 2012.

[5] A. M. Schlaff, M. R. Heupel, and C. A. Simpfendorfer, “Influence of environmental factors on shark and ray movement, behaviour and habitat use: a review,” *Reviews in Fish Biology and Fisheries*, vol. 24, no. 4, pp. 1089–1103, 2014.

[6] L. J. Compagno, “Sharks of the world. an annotated and illustrated catalogue of shark species known to date, vol 2. bullhead, mackerel and carpet sharks (heterodontiformes, lamniformes and orectolobiformes),” *FAO species catalogue for fishery purposes*, vol. 1, pp. viii+–1, 2001.

[7] S. Andrzejaczek, A. C. Gleiss, C. B. Pattiaratchi, and M. G. Meekan, “First insights into the fine-scale movements of the sandbar shark, carcharhinus plumbeus,” *Frontiers in Marine Science*, vol. 5, p. 483, 2018.

[8] A. C. Gleiss, D. L. Morgan, J. M. Whitty, J. J. Keleher, S. Fossette, and G. C. Hays, “Are vertical migrations driven by circadian behaviour? decoupling of activity and depth use in a large riverine elasmobranch, the freshwater sawfish (pristis pristis),” *Hydrobiologia*, vol. 787, no. 1, pp. 181–191, 2017.

[9] B. M. Whitmore, C. F. White, A. C. Gleiss, and N. M. Whitney, “A float-release package for recovering data-loggers from wild sharks,” *Journal of experimental marine biology and ecology*, vol. 475, pp. 49–53, 2016.

[10] A. C. Gleiss, J. J. Dale, K. N. Holland, and R. P. Wilson, “Accelerating estimates of activity-specific metabolic rate in fishes: testing the applicability of acceleration data-loggers,” *Journal of Experimental Marine Biology and Ecology*, vol. 385, no. 1-2, pp. 85–91, 2010.

[11] A. C. Gleiss, B. Norman, N. Liebsch, C. Francis, and R. P. Wilson, “A new prospect for tagging large free-swimming sharks with motion-sensitive data-loggers,” *Fisheries Research*, vol. 97, no. 1-2, pp. 11–16, 2009.

[12] J. W. Brownscombe, S. J. Cooke, and A. J. Danylchuk, “Spatiotemporal drivers of energy expenditure in a coastal marine fish,” *Oecologia*, vol. 183, no. 3, pp. 689–699, 2017.

[13] Y. P. Papastamatiou, G. Iosilevskii, V. Leos-Barajas, E. J. Brooks, L. A. Howey, D. D. Chapman, and Y. Y. Watanabe, “Optimal swimming strategies and behavioral plasticity of oceanic whitetip sharks,” *Scientific reports*, vol. 8, no. 1, p. 551, 2018.

TABLE III
EXPERIMENTAL RESULTS

Method	Training Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1-Score
Random Forest	0.9956	NA	0.4820	0.86	0.48	0.54
SharkVGG 1	0.9376	0.6181	0.7635	0.86	0.76	0.80
SharkVGG 2	0.9549	0.6185	0.7515	0.87	0.75	0.79
SharkInception	0.9838	0.6143	0.7216	0.87	0.72	0.76

Precision, Recall, and F1-Score are based on weighted average values.

[14] E. L. Shepard, R. P. Wilson, W. G. Rees, E. Grundy, S. A. Lambertucci, and S. B. Vosper, "Energy landscapes shape animal movement ecology," *The American Naturalist*, vol. 182, no. 3, pp. 298–312, 2013.

[15] L. Brewster, J. Dale, T. Guttridge, S. Gruber, A. Hansell, M. Elliott, I. Cowx, N. Whitney, and A. Gleiss, "Development and application of a machine learning algorithm for classification of elasmobranch behaviour from accelerometry data," *Marine biology*, vol. 165, no. 4, p. 62, 2018.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] W. Strong, "Behavioral ecology of horn sharks, heterodontus francisci, at santa catalina island, california, with emphasis on patterns of space utilization (sharks)." 1990.

[20] M. A. Edmonds, P. J. Motta, and R. E. Hueter, "Food capture kinematics of the suction feeding horn shark, heterodontus francisci," *Environmental Biology of Fishes*, vol. 62, no. 4, pp. 415–427, 2001.

[21] A. P. Summers, R. A. Ketcham, and T. Rowe, "Structure and function of the horn shark (heterodontus francisci) cranium through ontogeny: development of a hard prey specialist," *Journal of Morphology*, vol. 260, no. 1, pp. 1–12, 2004.

[22] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, "Pythia v0. 1: the winning entry to the vqa challenge 2018," *arXiv preprint arXiv:1807.09956*, 2018.

[23] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214–224, 2015.

[24] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," *IEEE transactions on Big Data*, 2016.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 2014, pp. 197–205.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.