Lamprey Recognition using LampNet

First Author
Institution1
Institution1 address
firstauthor@i1.org

Second Author
Institution2
First line of institution2 address
secondauthor@i2.org

Abstract

Sea lampreys have become a major pest in the North American Great Lakes. They are considered as an invasive species that prey on many species of commercial value such as lake trouts, fishes and perches. To tackle this problem, we introduce LampNet, a Convolutional Neural Network(CNN) based classifier to detect lampreys in videos. We evaluate the efficacy of the LampNet on the dataset consisting of around 10,629 images with three classes (lamprey, fish and background). Our evaluations show that LampNet can identify and detect lampreys with an F1-score of 0.97 and an overall ROC-AUC score of 0.995. Further, it also shows better performance than previous work involving AlexNet on similar dataset. This framework could be extended to replace the human operating the fish elevator to prevent the lamprey from entering the water bodies.

1. Introduction

Sea Lamprey is an invasive species native to rivers in Northern America. This aquatic invasive species were first recorded in Lakes Ontario [10] and but it took less than a decade for them to gain access to all Great lakes, where they quickly set to work predating on the lakes' commercially important fishes, including trout, whitefish, perch, and sturgeon. The unchecked proliferation of these aquatic parasites had profound negative impacts on the ecosystem such as the extinction of many commercially-important fishes, loss of native diversity, and devastating socio-economic effects [6, 7, 9] including the decimation of commercial fisheries and severe population imbalances within the Great Lakes.

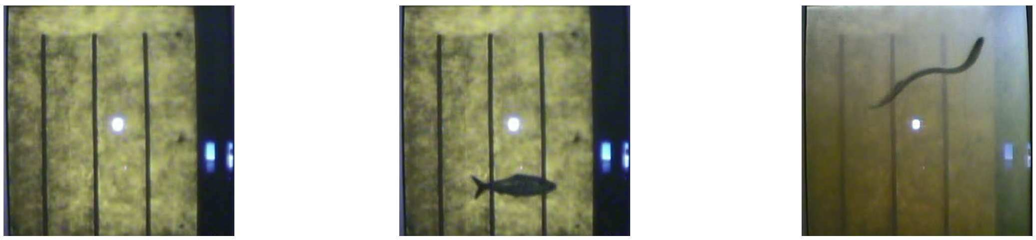
In an intensive effort to control the proliferation of sea lampreys, the Sea Lamprey Control Program was initiated 50 years ago to significantly bring down negative impacts of lampreys in the aquatic ecosystem [15]. Field biologists set up barriers and traps (called fish elevators) in the streams to prevent the lamprey's upstream movements. Currently, the fish elevator operated by a human (gatekeeper) is used to prevent the lamprey from entering the river. The fish elevator is essentially a passage for fish to enter the upstream of a river. Whenever the gatekeeper spots a lamprey, he/she closes the gate otherwise the gate is kept open. New techniques to control sea lampreys are always under development. Since sea lampreys use odors, or pheromones, to communicate, scientists have replicated these odors to increase the efficacy of current control methods. Though these approaches are effective none of them address the root cause of the problem ie., preventing the lampreys from entering into the water bodies. This project aims to replace the human-in-the-loop with an automated gate-keeping system that shuts the gates whenever a lamprey is detected thereby preventing the pest from intruding into water bodies.

The problem statement essentially boils down to a classification task of predicting whether the segments of the video contain lampreys or not. The current solution to the problem requires a human to supervise the movement of lamprey and fishes to prevent them. This not only makes the process slower but is tiresome to the gatekeeper and results in human error. Technological advancements in the field of computer vision can help address this problem, however, it is still unexplored due to limited dataset, and the fact that it is of interest to a small community. We got access to some videos of movement of lamprey and fish through a manually operated gate. From the concept learned in the course, we think we can use the CNNs which have demonstrated better accuracy in image classification tasks to remove human-in-the-loop and reduce human error. In this paper, we demonstrate *LampNet*, a simple 3-layered CNN that can outperform other prior attempts (see Background/Related Work section). We achieved an F1-score of 0.97 on the test set, which makes this deployable in real-world, to replace the gatekeeper by automated doors driven by the real-time predictions of lampreys by *LampNet*.

The next section talks about the prior attempts in using a neural network model for lamprey prediction and their limitations. Section 3 talks about our approach to the prob-

(a) Background (b) Fish (c) Lamprey

Figure 1: Example of frames extracted from the video



(a) Background (b) Fish (c) Lamprey

Figure 2: Cropped images

(a) (b) (c)

Figure 3: Example of frames containing both fish and lamprey

lem starting from data preprocessing to developing a CNN model. Section 4 details all the experiments performed and a detailed analysis of the results obtain. It also shows a comparative analysis between *LampNet* and other models. Finally, Section 5 gives an overall conclusion of this study and proposes some future directions.

2. Background/Related Work

Detecting fishes using computer vision is an active area of research to marine biologists, and has applications in various marine businesses. [16] , [17] and [18] used CNNs to detect fishes. Other studies like [19] and [20] were motivated towards localization of fishes, and used YOLO [21] and Faster-RCNN [22] respectively. All these studies use different species of fishes. However, our problem statement differs from these attempts, as we are more concerned with the accurate detection of lamprey rather than fishes. We couldn't find any published literature on lamprey detection, the primary reason being there is no open lamprey dataset available to perform this study.

However, we found two project reports [1] and [2] who were the students of COMPSCI 682 previously offered. They worked on the lamprey detection problem after getting the video data from Prof. Erik Miller. Our study is different from [2] which compared the performance of AlexNet using pre-trained weights with AlexNet trained from scratch on the classification task. The best-observed accuracy was 0.9643 with a precision and recall of 0.8619 and 0.8377 respectively. [1] used only 440 images for training and tested on 240 images using model architecture used in course assignments, which achieved a precision of 0.90 and recall of 0.82. Since both these performances are not sufficient to replace humans, we aimed at improving on this accuracy by using *LampNet*, which is customized CNN, trained from scratch, and is tailor-made for lamprey detection. We further show that using a lot more data coupled with preprocessing techniques increases the generalization capacity of *LampNet*, which was not well explored in [1] and [2].

3. Approach

3.1. Data Preprocessing

The availability of the dataset is the first step to make any neural network model work. For this project, we received the data from Prof. Erik Miller, which contained 4 different videos of movement of fishes and lampreys through a fish elevator. A gatekeeper manually closed the gates whenever he/she sees a lamprey. With this data, building any supervised learning model was difficult as there were no labels provided. We first extracted different frames from the video at the rate of 10 frames per second. We then manually created labels for each of the frames copying each frame into one of the three folders in 'Lamprey', 'Fish', and 'Background'. All the frames that contained the only lamprey were copied to the 'Lamprey' folder (shown in Figure 1c). Frames that contained fishes were copied to the 'Fish' folder (shown in Figure 1b) and any frames which didn't have any of these were copied background' folder (shown in Figure 1a). For the frames that contained both lamprey and fish at the same time were ignored for this part of the study(shown in Figure 3). The manual labeling was done for all the 4 videos that we revived, and it was the most time-consuming task concerning this project. All the extracted images were of the shape 480x360 pixels, as shown in Figure 1. Since the frames contained date and time of video getting recorded, as well as other uninformative surroundings, we cropped the images to size 128 x 128 as shown in Figure 2. Using this data, we created the training, validation, and test splits for our experiments.

3.2. Baseline Model

Using extracted images just from one of the videos, we developed a baseline model which was also reported for our milestone report. Images of 128 x 128 pixels were given as input. Detailed network architecture is shown in Table 1. We achieved an F1- score of **0.915**. We further improved on the performance of the model by working on the network architecture, result of which is *LampNet* which is described more in the next section.

Layer	Output Shape	Parameters
Conv2D	(None, 128, 128, 32)	896
Conv2D	(None, 128, 128, 64)	18496
MaxPooling2D	(None, 64, 64, 64)	0
Flatten	(None, 262144)	0
Dense	(None, 3))	786435

Table 1: Architecture of Baseline model

3.3. LampNet Architecture

We improved on the baseline model by deploying various strategies. All the inputs to the network were standardized by subtracting the mean of the data and dividing by the standard deviation of the data. The network architecture for LampNet is shown in Table 2, that achieved a F1-score of 0.97 and test-accuracy of 0.9714. A more detailed analysis of the results of *LampNet* is provided in the next section.

All the experiments were performed on Google Colab with a TPU hardware accelerator unless otherwise specified, with the following hardware specifications: RAM - 35.35 GB, CPU: 1xsingle core hyperthreaded Xeon Processors @2.3Ghz (i.e., 1 core, 2 threads). It took around 2 hours to train *LampNet*for 100 epochs with 10629 images. The next section talks about more details of the experiments performed.

Layer	Output Shape	Parameters
Conv2D	(None, 128, 128, 32)	2432
Activation	(None, 128, 128, 32)	0
MaxPooling2D	(None, 64, 64, 32)	0
Conv2D	(None, 64, 64, 64)	51264
BatchNormalization	(None, 64, 64, 64)	256
Activation	(None, 64, 64, 64)	0
MaxPooling2D	(None, 32, 32, 64)	0
Conv2D	(None, 32, 32, 128)	204928
Activation	(None, 32, 32, 128)	0
MaxPooling2D	(None, 16, 16, 128)	0
Flatten	(None, 32768)	0
Dense	(None, 3)	98307

Table 2: Architecture of *LampNet*

4. Experiment

All the experiments and results that are reported in this section are carried out on a dataset consisting of 10,629 images in total with the following class distribution: 2,477 background (Class 0), 4,064 fish (Class 1) and 4,088 lamprey (Class 2) images. Lesser number of background images were considered because of two reasons: (1) The video frames had more fish and lamprey frames (2) There was not much information to be learned from background images i.e, the background images were similar. The models were trained on 6376 samples and validated on 1595 samples. To prevent over-fitting we do early stopping after 100 epochs. Further, the performance of the models was evaluated on test-set consisting of 2658 samples with the following class distribution: 652 background (Class 0), 983 fish (Class 1) and 1,023 lamprey (Class 2) images.

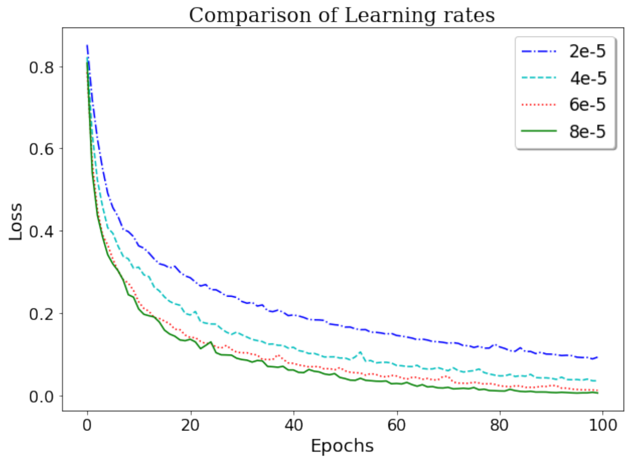


Figure 4: Plot depicting the convergence of different learning rates for *LampNet* architecture where the learning rate of 8e-5 being the optimal one.

Initially, the performance of the baseline model was evaluated on this dataset, which did reasonably well achieving an F1-score of 0.915 given its simple and shallow architecture. After experimenting with a different number of layers, depths, activations, pooling layers, and regularizations, we zeroed in on the final architecture of *LampNet* as shown in Table 2. After fixing the architecture, a complete analysis was carried out for choosing the right set of hyper-parameters. One such analysis was carried out to decide on the learning rate is depicted in Figure 4. The final architecture uses *Lecun Uniform* kernel initializer, batch size of 128, Adam optimizer with a learning rate of 8e-5 and default parameters provided in the original paper [11]. Figures 5, 6 depicts the training of the *LampNet* for 100 epochs.

After the training phase, the model's performance was evaluated on test data and the following metrics are reported: (1) Test loss and accuracy (2) Confusion matrix (3) ROC-AUC curve for each class.

Metric	**Value**
Test loss	0.1142
Test accuracy	0.9714

Table 3: Performance on test-set

As discussed earlier, due to the nature of the problem statement it is more meaningful to evaluate of performance of the model based on precision, recall and ROC-AUC curve rather than accuracy and loss alone. Table 4 reports the confusion matrix that summarizes the number of correct and incorrect predictions for each class.

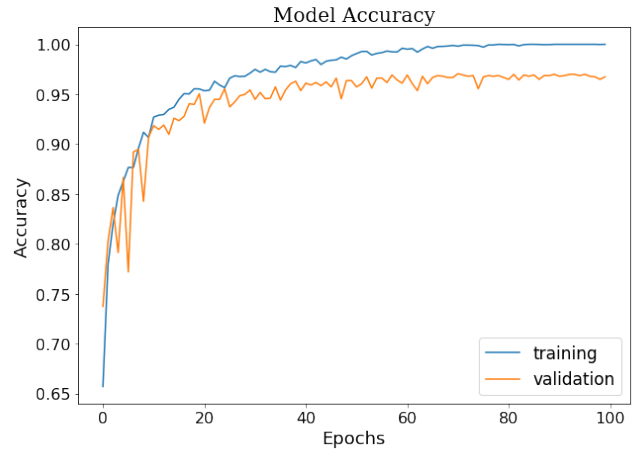


Figure 5: Plot depicting training vs. validation accuracy

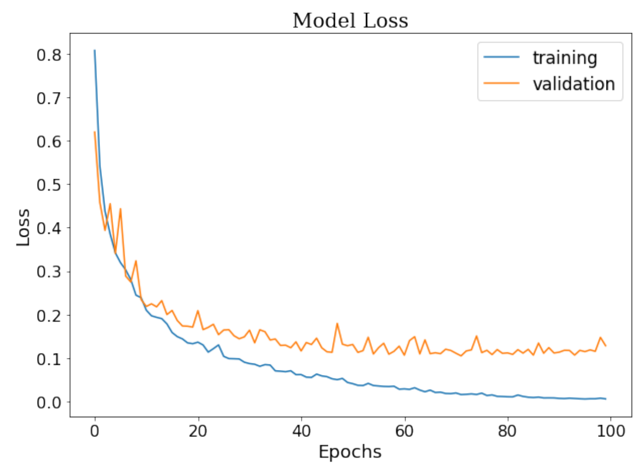


Figure 6: Plot depicting training vs. validation loss

	Class 0	Class 1	Class 2
Class 0	638	5	9
Class 1	34	944	5
Class 2	15	8	1000

Table 4: Confusion matrix for test-set where class 0, 1, 2 depict background, fish and lamprey respectively.

	Precision	**Recall**	**F1-score**	**Support**
Class 0	0.93	0.98	0.95	652
Class 1	0.99	0.96	0.97	983
Class 2	0.99	0.98	0.98	1023
Weighted Avg	**0.97**	**0.97**	**0.97**	2658

Table 5: Classification report for test-set where class 0, 1, 2 depict background, fish and lamprey respectively.

From the results reported in Table 5, it is evident that there is a significant improvement in F1-score for *LampNet* compared to the baseline model. Moreover, it also outperforms previous work by [1], [2]. The model misclassified 15 images of lampreys as background and 8 images as fish. Figure 9, 8 depict the scenarios where model misclassifies the lampreys. This seems to be reasonable mistake because the lampreys are not completely visible in the frame and such scenarios are even difficult for the naked eye to spot the lampreys. The trade-off that demands more attention here is the true positive rate and false-positive rate for lamprey class which can be better argued with the ROC-AUC curve as shown in Figure 7.

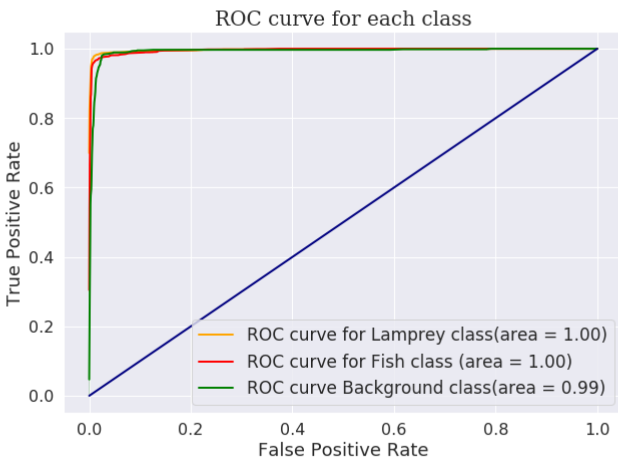


Figure 7: ROC-AUC curve for each class

Figure 7 shows the trade-off between true positive rate and false-positive rate for each of the classes. Since it is more important to not let any lampreys go unnoticed, the ideal model should have a higher true positive rate compared to the false-positive rate. This is because it is reasonable to block a few fishes from entering the water body rather than allowing a few lampreys to escape. The *LampNet* model achieves overall ROC-AUC score of 0.995.

Model	Precision	Recall	F1-score	Test Acc
Model 1	0.8619	0.8377	0.8496	0.9643
LampNet	**0.9700**	**0.9700**	**0.9700**	**0.9714**
Baseline	0.9100	0.9200	0.9150	0.9183
Model 2	0.8200	0.9050	0.8604	0.9501

Table 6: Comparitive study of precision, recall, F1-score and test-set accuracy of different models. *Model 1* refers to the best model from [2] and *Model 2* refers to the best model from [1].

Table 6 compares the performance of the our model with best models from [1] and [2]. Aaron *et al.* [1] adopted three-layer convolutional network from assignment 2 and while Samuel *et al.* [2] used AlexNet [3] with pre-trained wights. It can be observed that *LampNet* surpasses the performances of all the models in all the evaluation metrics that are considered in the problem statement.

5. Conclusion and Future work

Lamprey detection is an unexplored task rife with numerous challenges. In this paper, we propose a convolution neural network named *LampNet* for detection of lampreys which achieved an F1-score of 0.97 and test-set accuracy of 97.14%. The experiments conducted on this dataset show that it performs better than models proposed in [1], [2] and pre-trained *AlexNet*. Figures 8 and 9 show the scenarios where *LampNet* fails to identify lampreys. In Figure 8, it is tough for even humans to detect the lampreys because a very small part of lamprey is visible. Figure 9a is an example of human error while labelling the dataset. The frame is labelled as lamprey even though there is no lamprey present. Figures 9b and 9c were classified as fishes because the lamprey positions are deceiving. In 9b, the projecting parts of lamprey are mistaken to be fins of the fish. In 9c, the lamprey tail along with the grill closely resembles the tail of the fish. These subtleties can be learned by further training the model on more of such samples. On a broader scope, this application could be extended to replace the gatekeeper employed in the fish elevator. With the permission of Prof. Erik Miller, we would like to release this annotated dataset for lamprey detection, so that other fellow researchers can develop efficient models for lamprey detection.

In this work, we have ignored scenarios where there are both fish and lamprey are present in the same image. An immediate next step would be to use such these images, classify them as lamprey and train *LampNet* to predict lamprey both in presence and absence of other fishes, so that no lampreys are allowed to enter the water body.

References

[1] Aaron Traylor, *Convolutional Neural Networks for Lamprey Detection*, COMPSCI 697L Fall 2016 Project.

[2] Samuel Orloff, COMPSCI 697L Fall 2016 Project.

[3] Krizhevsky, A., Sutskever, I. and Hinton, G.E., *Imagenet classification with deep convolutional neural networks*, In Advances in neural information processing systems (pp. 1097-1105).

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

[5] Timothy D. Gingera, Todd B. Steeves, David A. Boguski, Steven Whyard, Weiming Li, Margaret F. Docker *Detection*

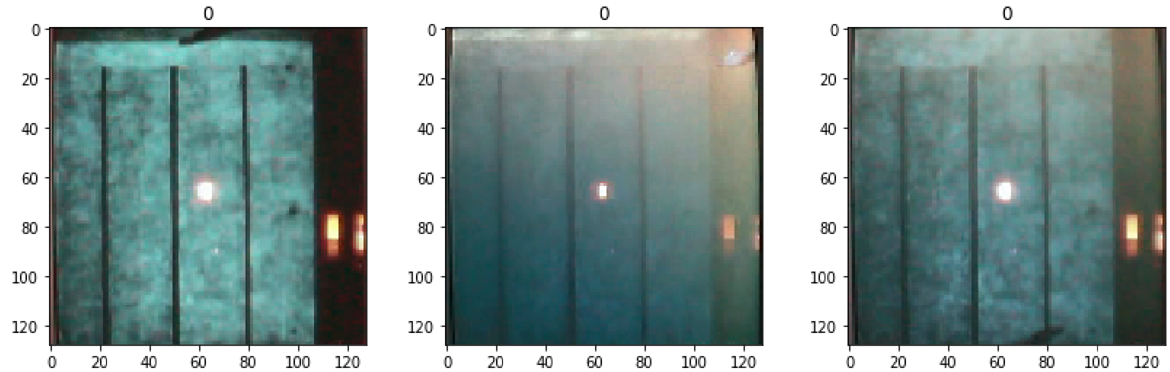
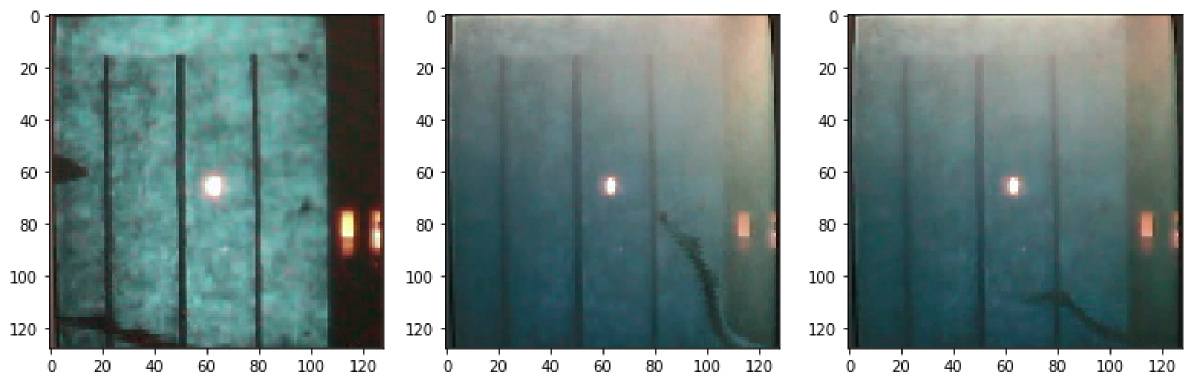


Figure 8: Lamprey images misclassified as background



(a) (b) (c)

Figure 9: Lamprey images misclassified as Fish

and identification of lampreys in Great Lakes streams using environmental DNA, JGLR 42, pp. 649-659.

[6] Reaser, J.K., Meyerson, L.A., Cronk, Q., De Poorter, M., Eldrege, L.G., Green, E., Kairo, M., Latasi, P., Mack, R.N., Mauremootoo, J., O'Dowd, D., Orapa, W., Sastroutomo, S., Saunders, A., Shine, C., Thrainsson, S., Vaiuti, L., *Ecological and socioeconomic impacts of invasive alien species in island ecosystems.*, Environ. Conserv. 34 (2), pp. 98–111.

[7] Pimentel, D., Zuniga, R., Morrison, D., *Update on the environmental and economic costs associated with alien-invasive species in the United States*, Ecol. Econ. 52 (3), pp. 273–288.

[8] L. Chen, S. Peng, and B. Yang., *Predicting alien herb invasion with machine learning models: biogeographical and lifehistory traits both matter*, Biological Invasions, 17(7):2187– 2198, 2015.

[9] Nienhuis, S., Haxon, T.J., Dunkley, T.C., *An empirical analysis of the consequences of zebra mussel invasions on fisheries in inland, freshwater lakes in Southern Ontario*, Manag. Biol. Invasion. 5 (3), pp. 287–302.

[10] Moser M., Almeida P., Kemp P., Sorensen P. (2015), *Lamprey Spawning Migration*, Spawning Migration. In: Docker M. (eds) Lampreys: Biology, Conservation and Control. Fish Fisheries Series, vol 37. Springer, Dordrecht.

[11] Diederik Kingma, Jimmy Ba, (2014), *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980v8 [cs.LG]

[12] Sergey Ioffe, Christian Szegedy, (2015), *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167v3 [cs.LG].

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions*, CoRR, abs/1409.4842, 2014.

[14] J. Nagi et al., *Max-pooling convolutional neural networks for vision-based hand gesture recognition*, 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, 2011, pp. 342-347.

[15] https://oceanservice.noaa.gov/facts/eutrophication.html, 10/05/17

[16] Ding, G., Song, Y., Guo, J., Feng, C., Li, G., He, B. and Yan, T., 2017, September. Fish recognition using convolutional neural network. In OCEANS 2017-Anchorage (pp. 1-4). IEEE.

[17] Rathi, D., Jain, S. and Indu, S., 2017, December. Underwater fish species classification using convolutional neural network and deep learning. In 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR) (pp. 1-6). IEEE. Ren, S., He, K., Girshick, R. and Sun, J., 2015.

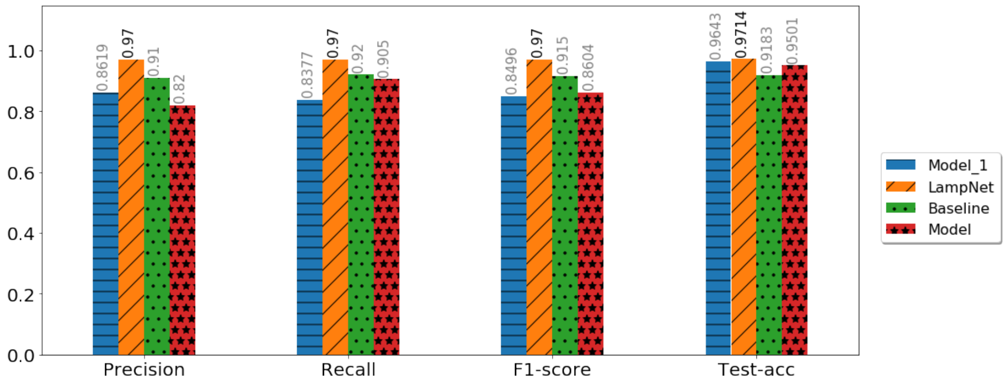


Figure 10: Bar plot comparing *LampNet* with different models

Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

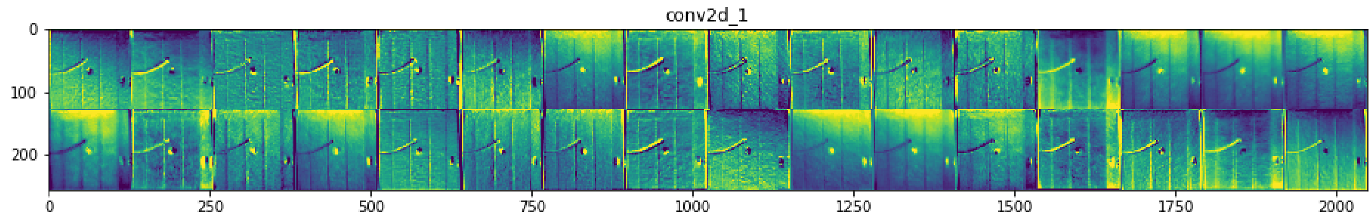
[18] Qin, H., Li, X., Liang, J., Peng, Y. and Zhang, C., 2016. DeepFish: Accurate underwater live fish recognition with a deep architecture. Neurocomputing, 187, pp.49-58.

[19] Sung, M., Yu, S.C. and Girdhar, Y., 2017, June. Vision based real-time fish detection using convolutional neural network. In OCEANS 2017-Aberdeen (pp. 1-6). IEEE.

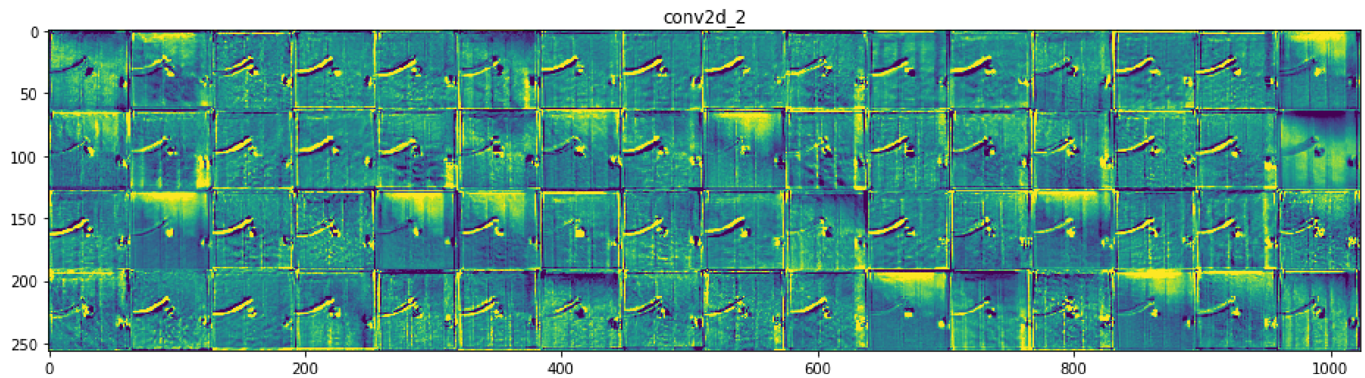
[20] Li, X., Shang, M., Qin, H. and Chen, L., 2015, October. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In OCEANS 2015-MTS/IEEE Washington (pp. 1-5). IEEE.

[21] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

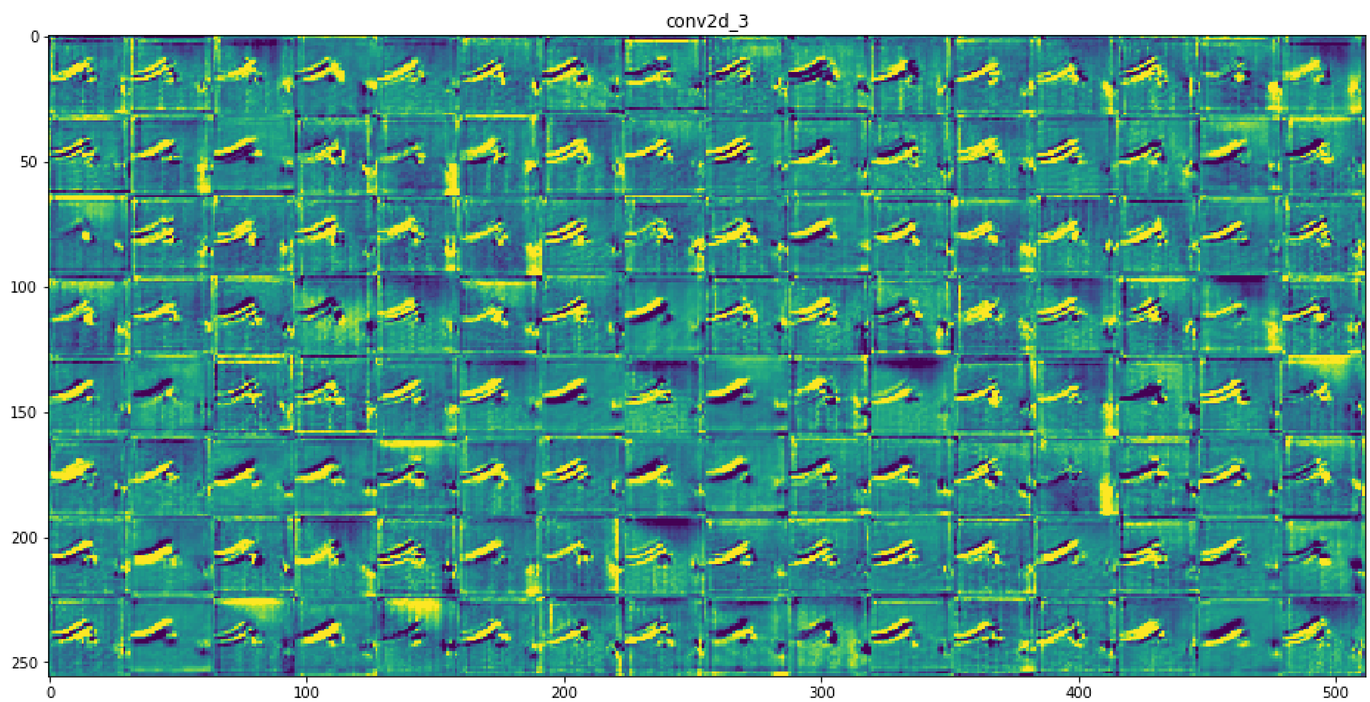
[22] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).



(a) 1st convolutional layer feature maps



(b) 2nd convolutional layer feature maps



(c) 3rd convolutional layer feature maps

Figure 11: Observing feature maps of different convolution layers after passing a lamprey image