Senior Project: Lionfish Detection System

Carmelo Furlan and Andrew Boniface

Advisor: Andrew Danowitz

CPE Senior Project

6/15/18

Abstract:

Deep neural networks have proven to be an effective method in classification of images. The ability to recognize objects has opened the door for many new systems which use image classification to solve challenging problems where conventional image classification would be inadequate. We trained a large, deep convolutional neural network to identify lionfish from other species that might be found in the same habitats. Google's Inception framework served as a powerful platform for our fish recognition system. By using transfer learning, we were able to obtain exceptional results for the classification of different species of fish. The convolutional neural network was then moved to a Raspberry Pi system enclosed in a water proof case that allowed for the convolutional neural network to be run on underwater images taken by the system.

Problem Statement:

Lionfish are one of the fastest growing invasive species in the Caribbean and parts of the Gulf of Mexico with very little effective ways of limiting their population [8]. Lionfish are native to the Indo-Pacific Ocean region but have been introduced to the southeast coast of the United States, the Caribbean, and parts of the Gulf of Mexico. Since the lionfish are not native to these areas they have few natural predators. Lionfish are carnivorous fish that feed on others and are currently wreaking havoc on the areas where they have been introduced. Without intervention from humans these lionfish may eliminate native species from their environments and can lead to extinction of certain fish species. Lionfish are not only decimating native species and environments, they are also causing a significant financial impact on the regions they were introduced. Some of the fish the lionfish prey on are valuable to the local fishing industries which are seeing all-time lows since the introduction of the invasive lionfish. Lionfish are not easily caught through traditional fishing methods due to the fact they have venomous spines throughout their back that can be dangerous for fisherman. National Oceanic and Atmospheric Administration researchers have claimed that invasive lionfish populations will continue to grow and decimate native species and cannot be eliminated using conventional methods. One promising new method is identifying and targeting only the lionfish in traps to later be removed from the area. These new traps will use computer vision and neural networks to identify the when the lionfish have entered the trap and then close the trap around the lionfish capturing it for removal.

System Architecture:

The fish identifying system is comprised of a Raspberry Pi 3, Jackery Portable Charger, and Raspberry Pi Camera V2. The system is all contained within a waterproof GSI Outdoors Lexan Gear Box to allow the system to operate while submerged underwater. The Raspberry Pi contains the pretrained neural network and receives images from the Pi Camera while both are powered by the Jackery Portable Charger. The images taken by the Pi Camera are processed by the neural network being run on the Raspberry Pi. If the neural network is unsure of the image the image is then saved on a 4GB flash drive. This flash drive will then save the images the to be later classified by a human and then added to the training data when the neural network is retrained in order to constantly improve the neural network for the system.

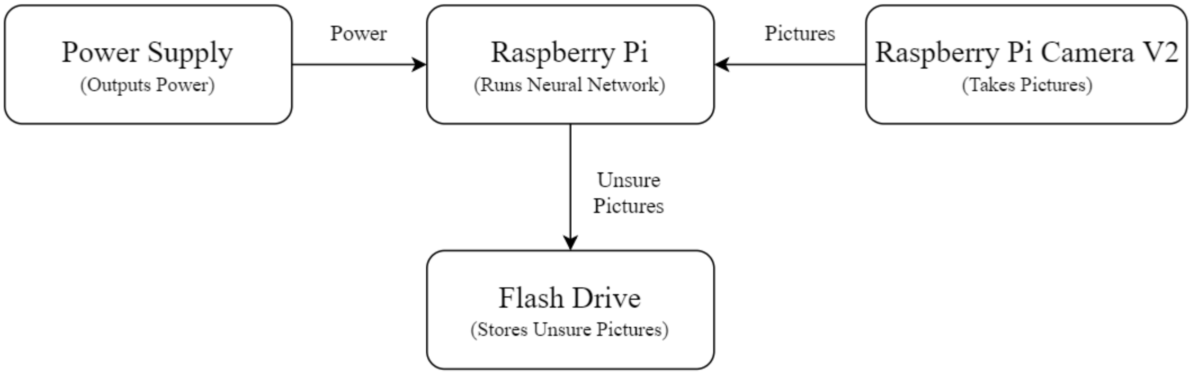

Figure 1: Shows the system block diagram.

System Integration:

Hardware Integration

To keep the electrical components of the system (Raspberry Pi, Power Source, Camera, Flash Drive) from getting damaged by the salt water, a waterproof container was used to hold all the equipment. The Raspberry Pi was chosen for the computer system of this project due to its small size and computational ability to run software as complex as a convolutional neural network. The Raspberry Pi Camera was chosen to be used as the camera for the system due to its ability to easily interact with the Raspberry Pi. Not only was the Pi Camera easier to implement with the Raspberry Pi but the Pi Camera has the ability to take higher quality of images than conventional webcams. If the neural network is unsure of the species of fish in the image taken by the Pi Camera the Raspberry Pi will save that image to the flash drive attached to the system. This will allow a human to later categorize the picture to the correct species of fish and then add that image to the training set. This will allow the neural network to retrain on images the system is unsure about to help improve the performance of the system for future uses.

Neural Network Integration

Based on the previous work done in the field of image recognition and analysis it was clear that a very large amount of data needed to be analyzed and a large convolutional neural network would give the best performance. However, the amount of data that we have access to is limited along with the computational power needed to properly train a neural network of such size. From the literature review it was shown that the best performing image recognition neural networks were trained on millions of images using multiple GPUs specifically built for convolutional neural networks. Due to the structure of convolution neural networks a very large neural network can have parts of the network retrained in order to identify and classify new labels that were not in the original training in a process called transfer learning [5]. For our system we used the concepts of transfer learning to retrain Google's Inception Framework [7] (that has previously been trained on millions of images) to create a system that could identify images of several species of fish including lionfish.

Google's Inception-Resnet-v2 network is formed by a stem convolutional neural network that is directly followed by several inception blocks. The output from the last inception block is then fed into a fully connected layer which preforms the image classification. Figure 1 below shows the complete architecture of the Inception neural network. Each inception block is comprised of several convolutional layers that are connected in both series and parallel. Figure 2 below shows an example of the architecture used in these inception blocks. The Inception neural network uses rectified linear units for activation functions for the individual neurons. Rectified linear units are used because their non-saturating shape allows the network to learn faster than other types of activation functions.

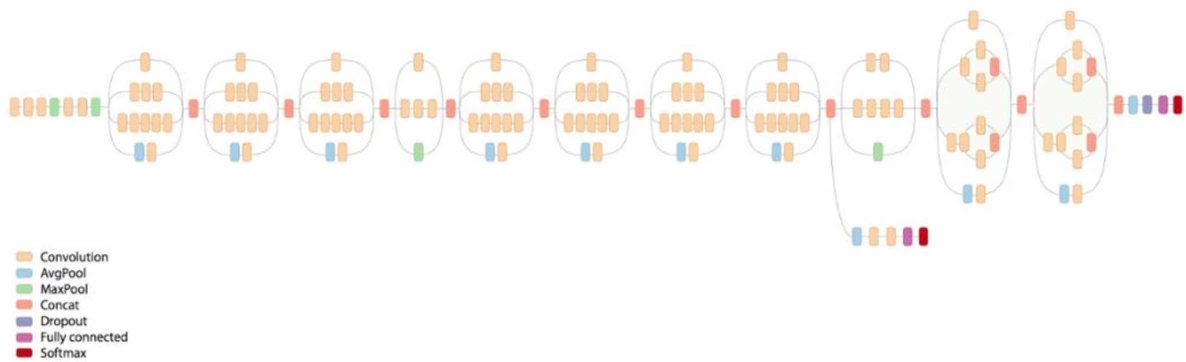

Figure 2: Shows the architecture of the Inception neural network.

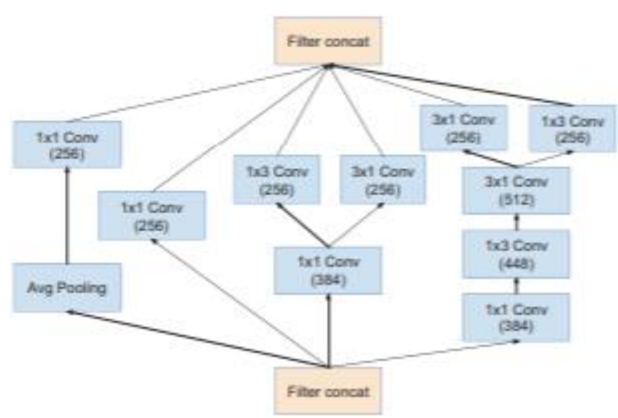

Figure 3: Shows an example of the architecture of an inception block.

In the Inception neural network, the first couple of layers preform basic edge detection and shape detection for the images. The weights of these layers will therefore not change when these layers are used for different image classifications. In fact, even the higher Inception blocks which learn abstract qualities of images do not need to change in order to recognize new objects. Using the idea of transfer learning we could retrain on the fully connected final layer to use the abstractions made by the previous layers to recognize several species of fish. We collected 1000 images of different species of fish and used those images to retrain the last layer of the Inception network. For retraining 900 of the original 1000 images were used with 500 training epochs with 80% used for training and 20% of the images set aside for testing. The remaining 100 images were set aside and used for final performance validation of the neural network.

The neural network uses a backpropagation algorithm to retrain the fully-connected layer to the new data. This system works by first computing the “bottlenecks” from the previous layers (ones before the fully-connected layer). The bottlenecks are a term that refer to the outputs of the previous layers for each image in the training sets. These bottlenecks are used so the network does not have to compute the values over the full neural network for every training iteration. The neural network then uses these bottleneck values as the input data and uses back-propagation to train the fully-connected layer to identify the images.

Computer Simulations/ Experimental Results of Neural Network:

The training of the neural network was performed using 500 training steps. Figure 4 shows how how the training and validation accuracy are being affected by each of the training steps.

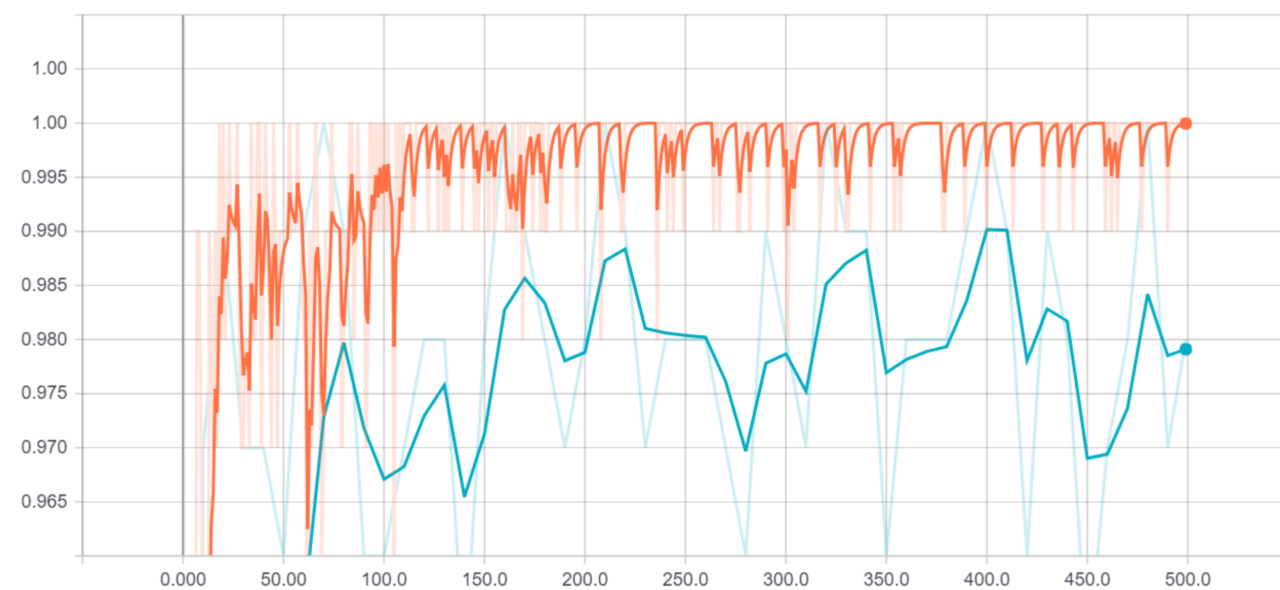

Figure 4: Shows the training accuracy in orange and the validation accuracy in blue throughout the training steps.

Not only can the accuracies be shown for the validation and training data, but the cross-entropy can also be shown for both. The cross-entropy is a loss function which shows how well the learning process is progressing during the retraining of the network. The training is attempting to minimize the loss. Figure 5 shows a graph of the cross-entropy over the training steps.

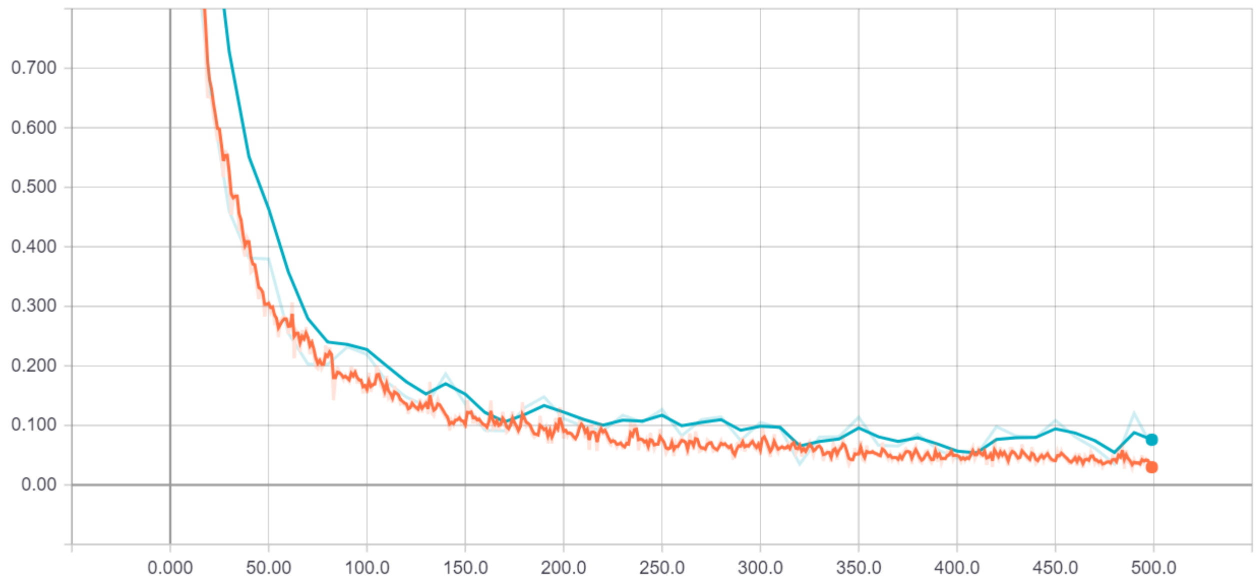

Figure 5: Shows the training cross-entropy in orange and the validation cross-entropy in blue throughout the training steps.

The neural network performed very well when tested with the testing set of data after the retraining of the neural network was completed. This is shown by the results in Table 1 which show that the output of the trained neural network when grouped into two categories of whether the fish was a lionfish or if the fish was not a lionfish.

Type of Results	**Number of Outcomes**
True Positives	36
False Positives	0
True Negatives	64
False Negatives	0
Total	100

Table 1: Shows the results of the testing data.

The testing set consisted of 99 total images with 36 of those images containing lionfish while 63 of the images contained a random assortment of the other types of fish classifications. Although the neural network has 6 possible outputs Table 1 only shows 2 outputs either lionfish or not lionfish this is due to the fact the application of the neural network is to capture only lionfish. The neural network has 6 possible outputs (one for each fish species) and the confusion matrix can be a useful tool in gauging the ability of the neural network. Table 2 below shows the confusion matrix for the neural network.

Fish Species	Lionfish	Turtle	Red Snapper	Grouper	Clownfish	Royal Blue Tang	Lobster
Lionfish	36	0	0	0	0	0	0
Turtle	0	21	0	0	0	0	0
Red Snapper	0	0	7	0	0	0	0
Grouper	0	0	0	4	0	0	0
Clownfish	0	0	0	0	19	0	0
Royal Blue Tang	0	0	0	0	0	11	0
Lobster	0	0	0	0	0	0	8

Table 2: Shows confusion matrix for the neural network.

As demonstrated by the two tables the system not only was able to identify with 100% accuracy when a picture contained a lionfish, but it was also able to identify the other species of fish with a 100% accuracy.

Experimental Results of System:

To test the performance of the entire system the neural network was retrained on two different groups of images each composed of around 50 photos. One of the groups of images was of a blue toy shark the other was of a grey toy shark. All images in the photos were taken with varying backgrounds and at varying angels in an ability simulate real world conditions.

Figure 6: Shows two example images for the training group of images.

Once the neural network was retrained to classify the two toy sharks the network was moved to the Raspberry Pi and the entire system was then enclosed in the waterproof case. The system was then submerged in water and a series of test was performed and the results were included to see if the system can identify the two different toy sharks.

Type of Results	**Number of Outcomes**
Identified Grey Shark when Grey Shark	15
Identified Grey Shark when Blue Shark	0
Identified Blue Shark when Blue Shark	15

Identified Blue Shark when Grey Shark	0
Total	30

Table 3: Shows the results of the testing of the toy sharks on the entire system.

The testing set consisted of 30 total images with 15 of those images containing a grey toy shark while 15 of the images contained a blue toy shark. The system was able to identify with 100% accuracy which toy shark was present in the image.

Conclusion:

The results of our final system were great and showed that a neural network can be used to identify and classify different species of fish. The testing of the neural network along with the testing of the entire system both were able to produce results with 100% accuracy. Even though the testing of the neural network was not an extremely large set the ability to identify the different species of fish with 100 percent accuracy along with the type of toy in the underwater images with 100 percent accuracy show that a system using a neural network to identify and remove lionfish is a viable option for the safe and effective removal of the invasive creatures.

Future Work:

The first way the project can be improved is by adding more fish species to the classification system. This can be done simply by researching the area to find more fish species that inhabit the same area the system will be placed and then retraining the system with the new fish species added to the training data. This would ensure that when the system is used in real world conditions the system has a better chance of identifying any fish that might be seen on camera. This would make the system perform better since as the system currently is set up to guess which species the fish in the photo most resembles. This would mean that if the fish was not in any of the fish species used to train the system the system will say the fish belongs to the species that looks most like the fish in the image which can result in a number of false positives. Another way the project can be improved is by capturing and saving the photos the system takes while in use and then adding those images to the training data and retraining the neural network. This will improve the performance of the system since the pictures used to train the system were taken from the internet and may not have been taken by the same camera used in the system. By using pictures that the system has taken itself and adding those to the training data will help the system train on data that most resembles the data that the system will see when in use.

References:

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Annual Conference on Neural Information Processing Systems*, *25*. 2012. Retrieved from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. Going Deeper with Convolutions. 2015. Retrieved from https://arxiv.org/pdf/1409.4842.pdf

[3] LeCun, Y., Kavukcuoglu, K., Farabet, C. Convolutional Networks and Applications in Vision. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, 2010, pp. 253-256. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5537907&isnumber=5536941

[4] Hinton, G., Krizhevsky, I., Salakhutdnov, R., Srivastava, N., Sutskever, I. Improving Neural Networks by Preventing Coadaptation of Feature Detectors. 2012. Retrieved from https://arxiv.org/pdf/1207.0580.pdf

[5] Yoshua, B. Deep Learning of Representations for Unsupervised and Transfer Learning. *JMLR: Workshop and Conference Proceedings.* 2012. Retrieved from http://proceedings.mlr.press/v27/bengio12a/bengio12a.pdf

[6] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the Inception Architecture for Computer Vision. 2015. Retrieved from https://arxiv.org/pdf/1512.00567.pdf

[7] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence.* 2017. Retrieved from http://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14806/14311

[8] NOAA. (n.d.). Impacts of Invasive Lionfish. Retrieved from https://www.fisheries.noaa.gov/feature-story/impacts-invasive-lionfish