1 Problem statement:

In Maryland's aquatic ecosystems, the presence of invasive fish poses a significant threat to the native biodiversity and balanced ecosystems. Invasive fish species like the *blue catfish* and the *northern snakehead* are thriving in the Maryland ecosystems due to their aggressive behavior and adaptability. As a result, they outcompete native species like the *largemouth bass* and *bluegill*, leading to decreased populations of ecologically and economically important species.

While many fishers have been encouraged to target these invasive species when fishing, manual fish detection (such as googling pictures) can be a bothersome task and is prone to human error. Misidentification is a common issue, particularly among less experienced fishers, due to the similarities of features between native and invasive fish (i.e. invasive catfish vs local catfish).

This challenge grows increasingly as the prevalence of the invasive species grows across Maryland's waterways, such as the Chesapeake Bay, where the blue catfish nearly make up 75 percent of the biomass in some areas. Without efficient and reliable tools for fish identification, the ability of fishers to target these invasive species and control their population is hindered. This delay worsens the ecological damage caused by these invasive fish, leading to more disrupted food chains and out-of-balance ecosystems.

2 Approach

To best create a model that is most applicable in the real world, we had to find data that was representative of how recreational fishers would take pictures of their fish. We chose to collect our data from publicly available social media accounts. This meant most of our pictures were of people holding fish, and the fish was often sharing the frame with, or even secondary to, the person holding the fish. This meant we had to first use a model to preprocess the data by segmenting out the fish. We used a model based on the YOLO v8 object detection architecture to segment the data, and SOMETHING ELSE to extract the fish. The model is trained on identifying fish within images. On a higher level, the segmented first create a mask, then we can crop out the fish from the picture based on the mask.

Our classification model was built off ResNet architecture. In order to train and test it, we used cross validation so that we would have a test to training data ratio of 0.15. We used k-fold validation on the training data to improve our model, and used the test data to validate our model. In order to improve robustness, after splitting the data, we augmented the training data with a ratio of 4 to 1. We achieved this by rotating and adding random noise to each of the training images.

We started with a pretrained ResNet model, called resnet18 that comes pytorch. This comes with batch normalization, which helps with overfitting.

We also used the Adam optimizer, which is the state-of-the-art and has been shown to converge quickly when adjusting learning rate for each parameter. We modified this model to better suit our needs, which included changing the loss to cross entropy loss and changing the number of classes to 4. Cross entropy loss has been shown to be the best loss function for image classification tasks.

3 Results

After some experimentation, we discovered the following setup to be optimal: Epochs=10, learning rate=0.0001, regularization constant=10$^-$4, k=5, batch size is 32.
Our model could consistently achieve a test accuracy of 80 percent, depending on how things were randomly split between train, validation, and test sets. Following figures show model performance at different learning rates, as well as selected results.

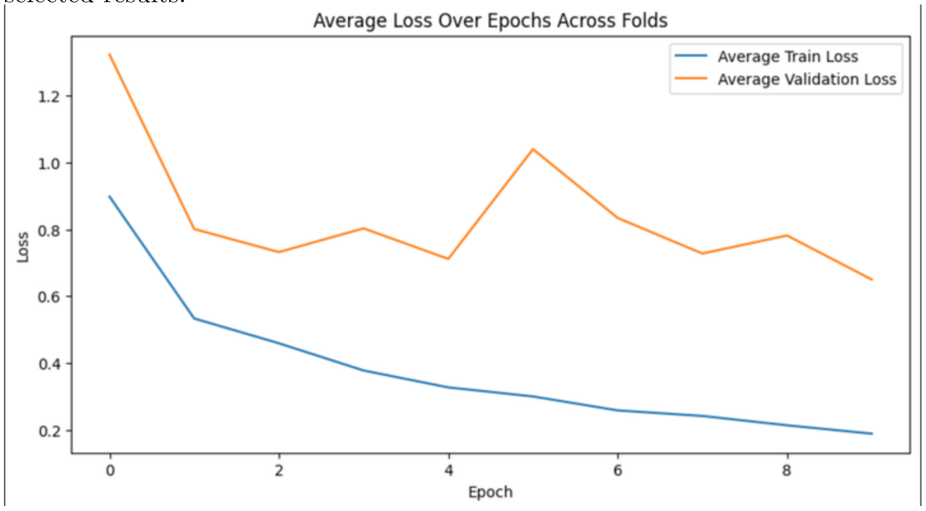

Figure 1: Average Loss at Learning Rate = 0.0005

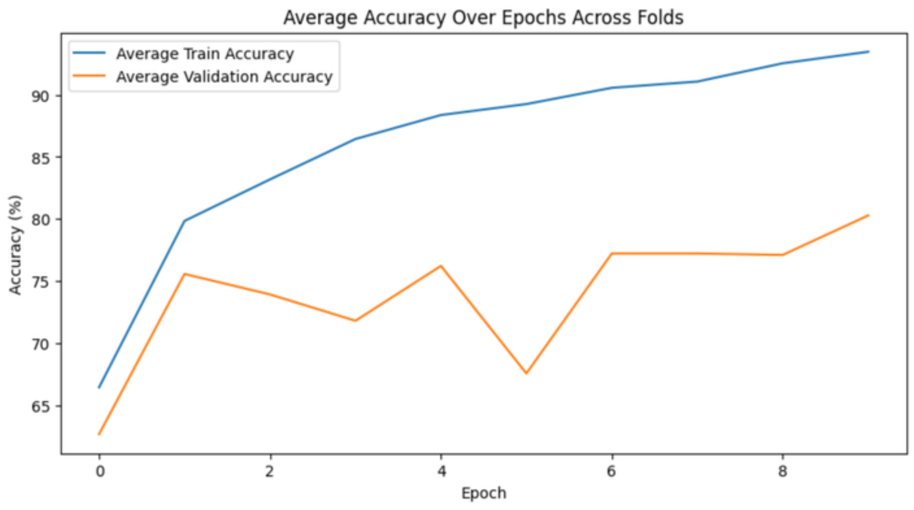

Figure 2: Average Accuracy at Learning Rate = 0.0005
The wide disparity between training and validation loss indicated overfitting.

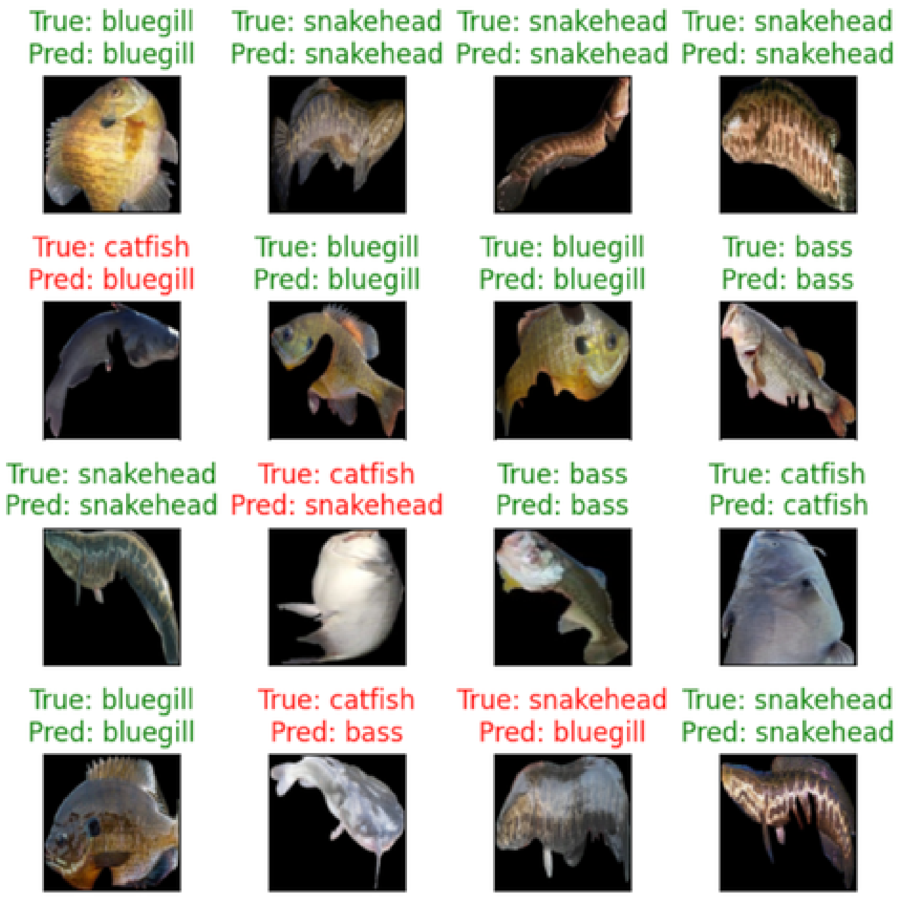

Figure 3: Results of the model when trained using learning rate = 0.0005
Interestingly, the model seemed to misclassify catfish the most.

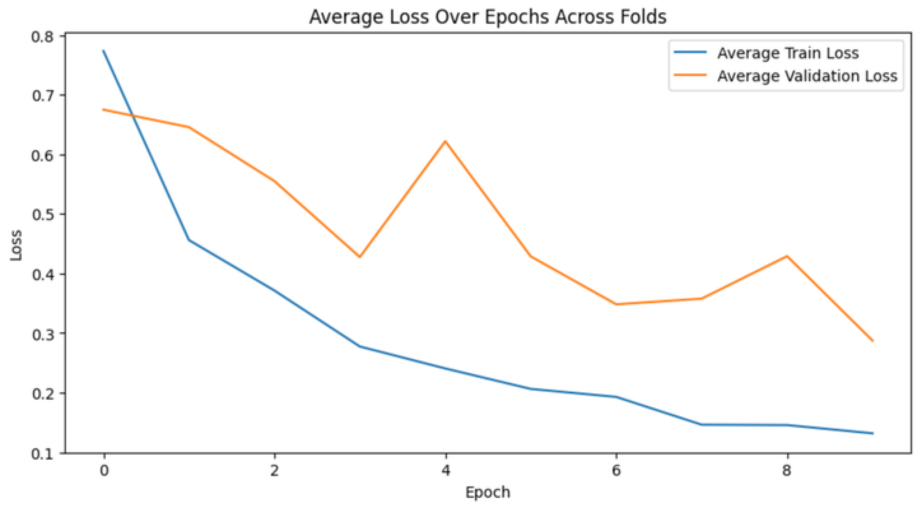

Figure 4: Average Loss at Learning Rate = 0.00025

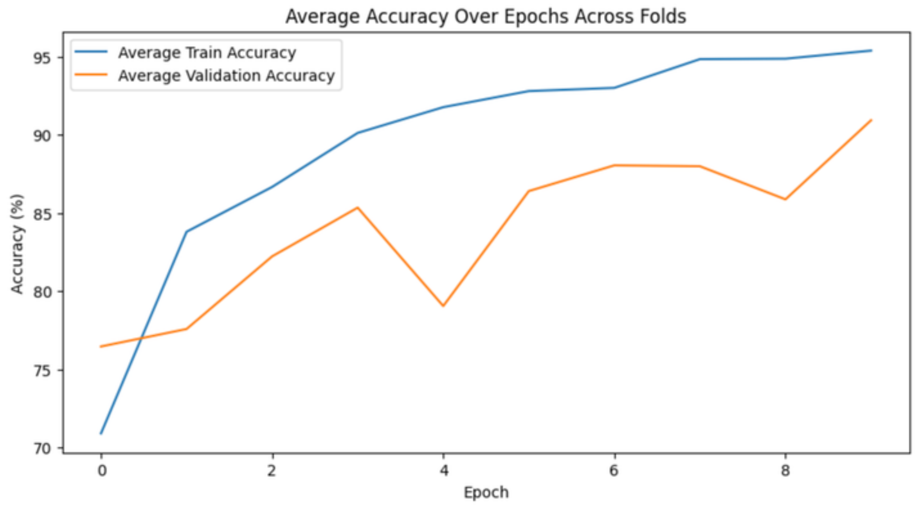

Figure 5: Average Accuracy at Learning Rate = 0.00025

Here, we try a lower learning rate. It seems to be in the right dr

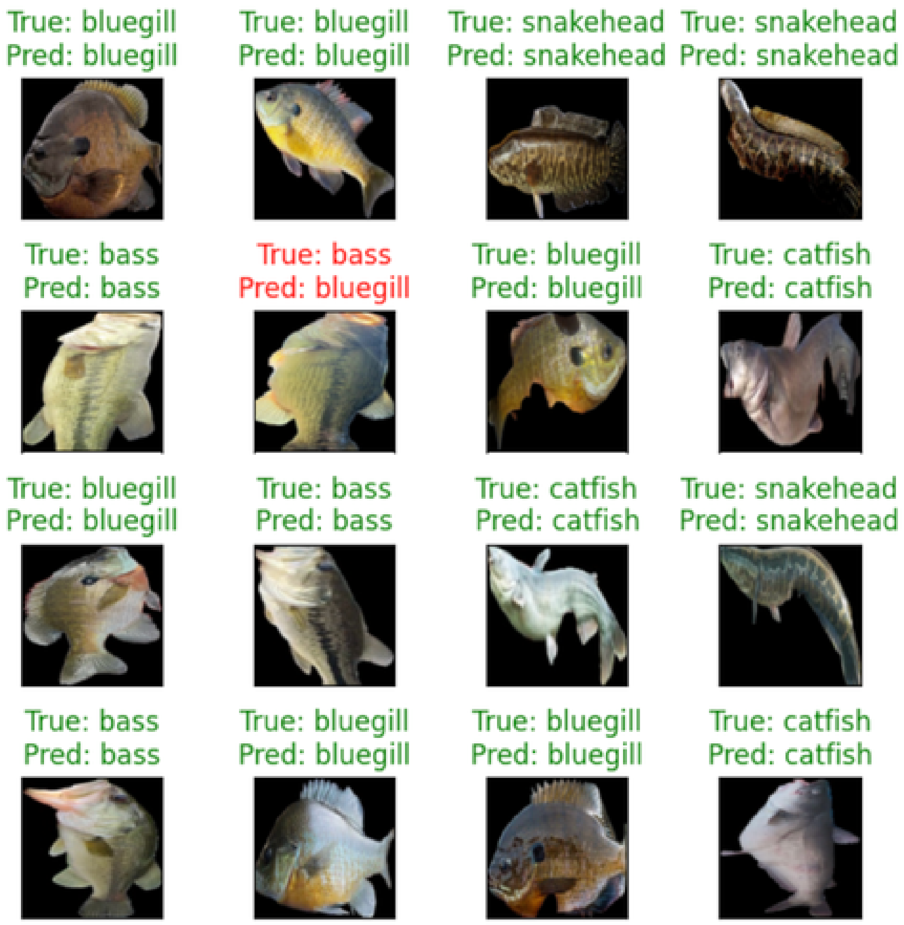

Figure 6: Results of the model when trained using learning rate = 0.00025

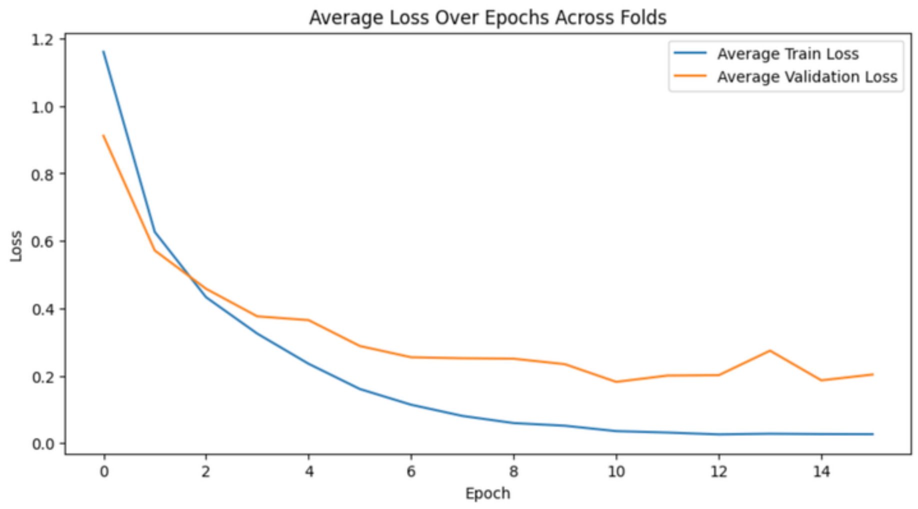

Figure 7: Average Loss at Learning Rate = 0.00001 and Epoch = 16

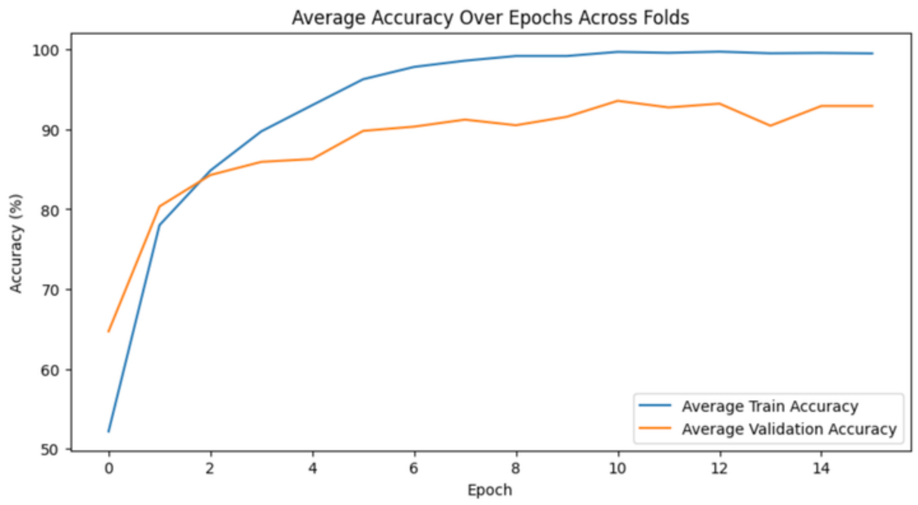

Figure 8: Average Accuracy at Learning Rate = 0.0001 and Epoch = 16

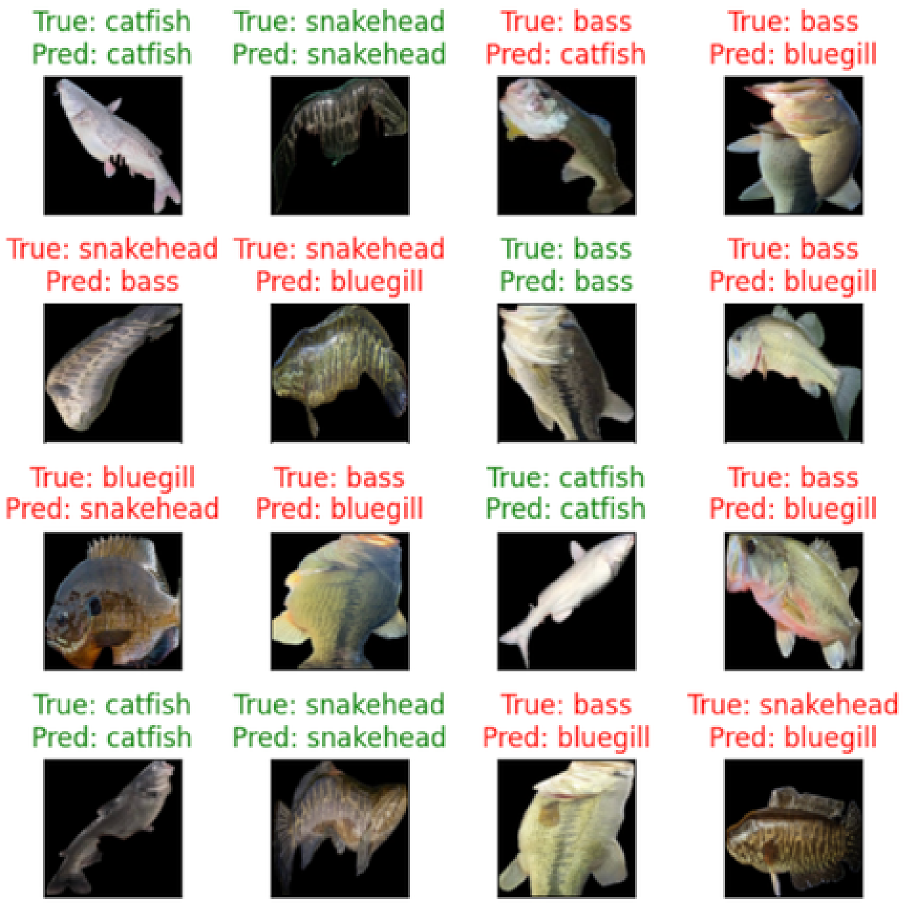

Figure 9: Results of the model when trained using learning rate = 0.00001 and Epoch = 16

4 Difficulties

Our biggest challenge was overfitting. We had suspiciously high training and validation accuracy, but our testing accuracy was way lower. We discovered this was because of multiple reasons. We originally did not use k-folds to train our model, nor did we augment. This was very bad because since we did not use k-fold, each group of training and testing data was the same, so the model learned patterns unique to that specific split. Also, there were only 100 images per class, which was not enough to create high enough variance. To fix this, we implemented k-folds and augmenting data, which we sent time finding the best setup for. After all this, the test accuracy dropped to 50 percent. We noticed after looking at the plot of loss to epoch graph within each fold that our model was getting stuck in local maximas, so we spent some time tweaking the learning rate and got it to the final test accuracy of around 80 percent.

5 Implications

The model provides fishers with an easy-to-use tool to accurately identify the fish they catch in Maryland. This addresses common misidentification issues, especially when it comes to similar species of fish such as two different catfish. This allows the model empower to fishers to correctly target invasive species without relying on manual methods that are prone to error.

This reduction in human error plays a critical role in supporting conservation efforts, as it ensures that native species are protected while invasive species are effectively managed. Accurate identification of invasive fish enables fishers to focus their efforts on removing these harmful species, which outcompete native fish and disrupt aquatic ecosystems.

By combining its practical application for fishers with the potential to generate valuable ecological data, the model bridges individual action and broader conservation goals. This seamless integration of user-friendly functionality and scientific insight strengthens efforts to protect Maryland's aquatic ecosystems on multiple fronts.

6 Extending

One obvious way to extend this work is by increasing the number of fish the model can classify. We could also train different models for different regions, or have a large universal fish species classifier. If we could also somehow collect data on the time and location of each image, we could create much more precise and high-resolution information on the migration of different fish species.
There was also some promise when we increased the number of epochs, but we didn't have enough computing power to increase number of epochs within a reasonable amount of time.

References

[1] Maryland Department of Natural Resources. (2024). *Aquatic invasive species*. Maryland.gov. `https://dnr.maryland.gov/waters/bay/Pages/Aquatic-Invasive-Species.aspx`

[2] Maryland Department of Natural Resources. (2019, December 9). *Save the Bay: Eat invasive!*. Maryland.gov. `https://news.maryland.gov/dnr/2019/12/09/save-the-bay-eat-invasive/`

[3] Maryland Fishing Regulations. (2024). *Invasive species information*. eRegulations. `https://www.eregulations.com/maryland/fishing/invasive-species`

[4] Fishial. (n.d.). *Fish identification model using machine learning* [Machine learning model]. GitHub. `https://github.com/fishial/fish-identification/blob/main/README.md`