Examining the impact of image enhancement on machine learning for underwater imagery

In the scope of the course of FAML

Authored by David Almeida

Porto, January 2024

1 | Introduction and Context

Main goal is to improve notions relevant to underwater image classification, leveraging the Enhanced Underwater Vision Perception (EUVP) dataset in an unprecedented way.

- From the University of Minnesota, this dataset offers a unique window into diverse underwater environments

- Perfectly suitable for examining how image quality affects classification accuracy

- How does the quality enhancement of different sets of data (training or testing) impact the overall classification?
    - Why is this dataset so suitable to examine this research question?

2 | Dataset Description and Labelling

The EUVP dataset compromises a collection of underwater images of marine fauna and flora with varied visibility conditions

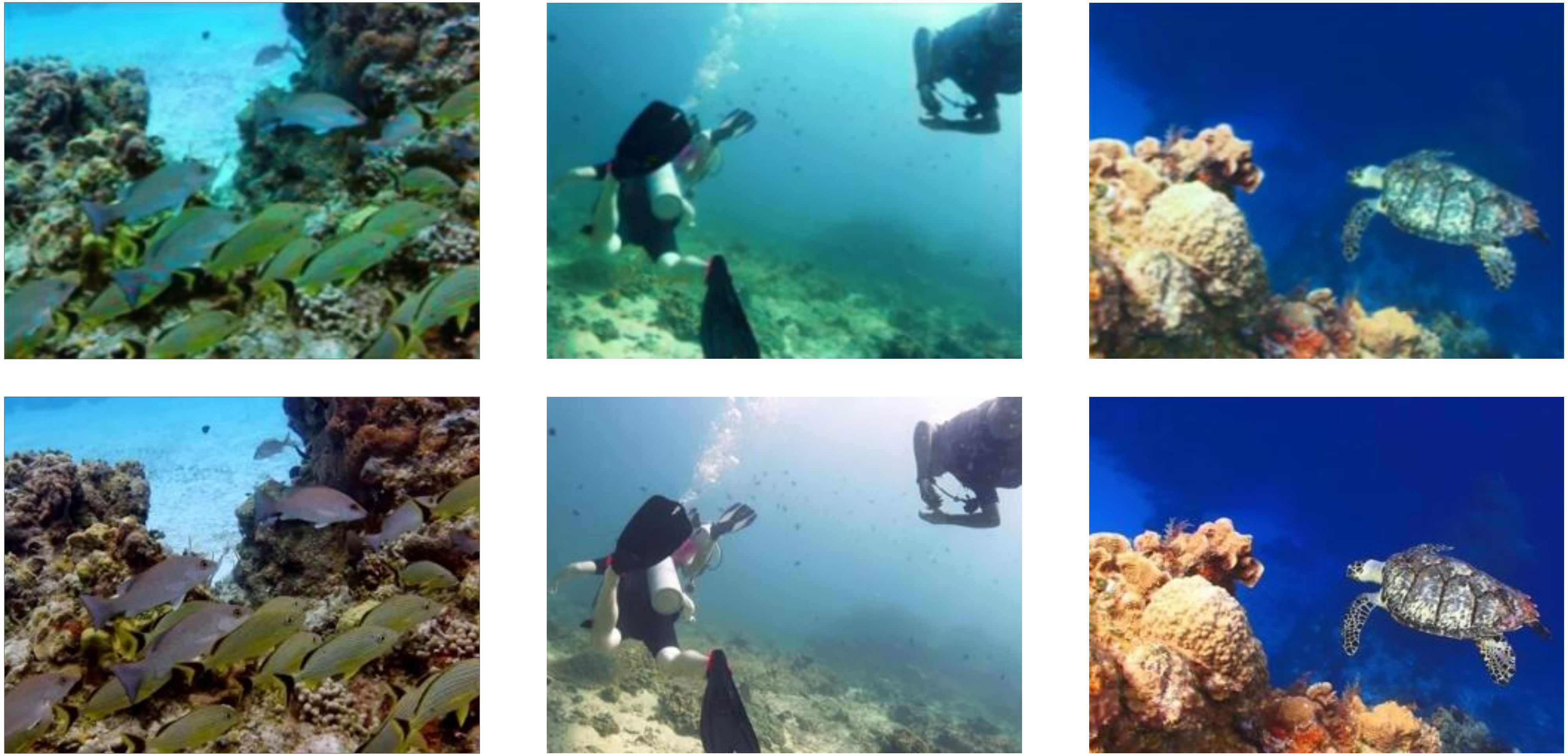
- The *Underwater Scenes* section contains 4370 **paired images**:
  - 2185 **original** images captured with different camera types.
  - 2185 images with **enhanced** image quality (enhanced version of original images)

- Enhancement was performed based on perceptual quality - focusing on colour, contrast, sharpness, and overall visibility

- Dataset’s variety considers the complexities of underwater imagery, such as varying light conditions, suspended particles, and movement blur

2 | Dataset Description and Labelling

The EUVP dataset compromises a collection of underwater images of marine fauna and flora with varied visibility conditions



2 | Dataset Description and Labelling

Establishing an efficient and accurate process for labelling the images in the dataset was key, focusing on creating a consistent and scalable labelling system.

- Initially defined more **categories**, but these converged into just two:
  - **Fish**
  - **None**

- Simplification was strategically applied to combat an impracticable class imbalance

- A total of **1306 images were labelled** (from the enhanced section) with *Roboflow*:
  - **910** belong to class **Fish (70%)**
  - **396** belong to class **None (30%)**

3 | Data Preprocessing and Augmentation

Augmentation increases the dataset size and variety, helping the model generalize better to new, unseen data. Moreover, it may utilized to tackle class imbalance.

- Out of the 1306 images, the dataset was divided from the start as such:
  - **947** for **training**
  - **150** for **validation**
  - **209** for **testing**

- Original images with 320 x 240 pixels were resized to 128 x 128

- **Augmentation** was instrumental in **combating class imbalance**, by generating:
  - **1 new augmented version** of every image labelled as **Fish**
  - **4 new augmented versions** of every image labelled as **None**

- **Augmented dataset** ended up containing a total of **2612 images**, with:
  - **1346 (52%)** belonging to the class **Fish**
  - **1266 (48%)** belonging to the class **None**

3 | Data Preprocessing and Augmentation

Elastic transformations were employed for additional realism in the augmented images, as these mimic underwater distortions.

Data augmentation strategies for each image included a combination of the following:

- Rotations (between -20º and +20º)
- Horizontal/Vertical Flips
- Image shearing (distortion along an axis)
- **Elastic transformations** (displacement vectors based on random offsets)

Original

Resized

Augmented version

4 | Model Architecture

This project employs a Convolutional Neural Network (CNN), constructed to extract and process features from underwater images, a crucial step in identifying the presence of fish.

- The CNN comprises **multiple layers:**
  - **Convolutional** Layers with 32, 64, and 128 filters to capture image details, each followed by ReLU activation for non-linear processing.
  - **MaxPooling** Layers were implemented to reduce dimensionality, enhancing the model's focus on essential features.
  - **Dropout** Layers for generalization with rates between that randomly deactivate neurons, preventing over-dependence on specific features
  - The network ends with a **flattening layer**, transforming 2D feature maps into a 1D vector, followed by dense layers (incl. a 512-neuron layer) for final classification.
- The network employs a **sigmoid activation** function in its final layer, ideal for binary outcomes - determining 'Fish' or 'None'.
- The architecture integrates **L2 regularization** in convolutional layers, reducing the risk of overfitting by penalizing larger weight values.

5 | Regularization and Overfitting Prevention

A diverse set of techniques collectively contribute to a more stable and reliable model, crucial for handling the complexity of underwater imagery.

- In this model, regularization plays a pivotal role in enhancing its generalization ability, ensuring it performs well not just on training data but also on unseen images.

- With **Dropout** rates between 0.25 and 0.32, the model **randomly ignores a subset of neurons** during training. This randomness helps the network to become more robust, **avoiding over-reliance on particular features**.

- Each convolutional layer includes **L2 regularization** (set at 0.002), which **penalizes large weights**. This encourages the model to **find simpler patterns**, reducing the likelihood of memorizing the training data (overfitting).

5 | Regularization and Overfitting Prevention

A diverse set of techniques collectively contribute to a more stable and reliable model, crucial for handling the complexity of underwater imagery..

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(image_width, image_height, 3), kernel_regularizer=l2(0.002)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu', kernel_regularizer=l2(0.002)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu', kernel_regularizer=l2(0.002)),
    MaxPooling2D(2, 2),
    Dropout(0.25),

    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.32),
    Dense(1, activation='sigmoid')
])
```

6 | Training Process

During the training process, the biggest areas of concern were the computational limitations and the potential of model overfitting/underfitting.

- The CNN model underwent extensive training, iterating through **50 epochs**, using a batch size of **48 images**, attempting to strike a balance between computational limitations and learning effectiveness.

- The **ReduceLROn Plateau** callback dynamically **adjusts the learning rate**. If no improvement is seen in **validation loss after 5 epochs**, the learning rate is halved, enabling finer adjustments in the model's learning process.
  - **Binary cross-entropy** function was utilized as the **loss function**

- To **prevent overfitting**, **Early Stopping** was implemented with a **patience of 12 epochs**. This ensures training halts when no improvement in validation loss is observed, optimizing the model's performance without unnecessary computation.

7 | Model Evaluation and Metrics

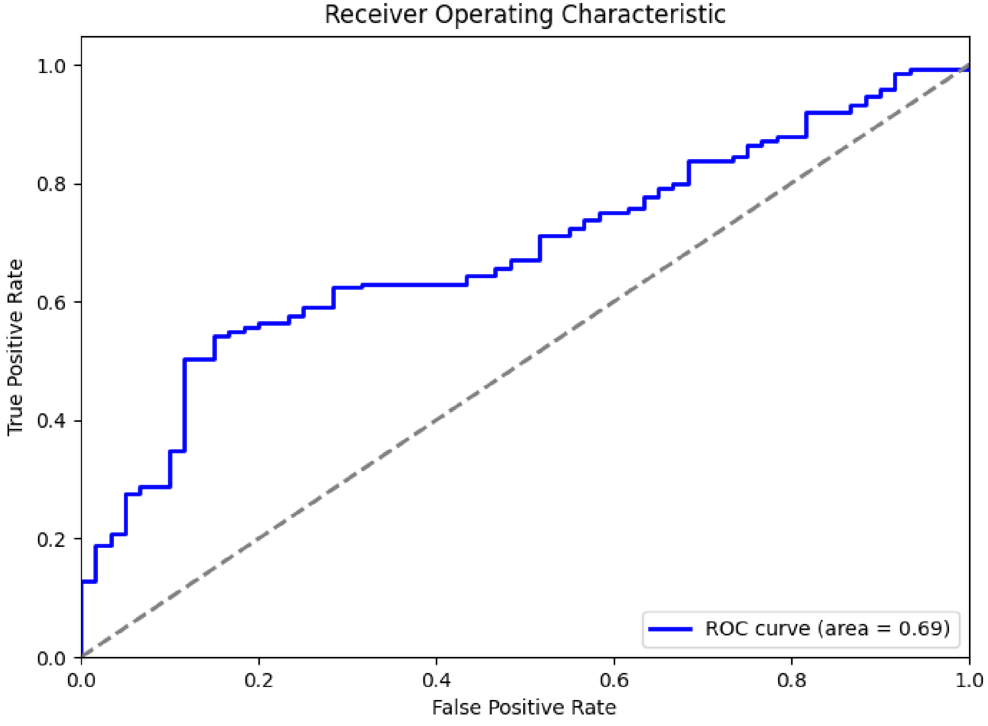
Considering the class imbalance in the dataset, a variety of metrics were chosen to comprehensively evaluate the model's performance.

- **Accuracy:** This metric provides a basic understanding of overall performance but can be misleading in imbalanced datasets.
- **F1 Score:** A more reliable metric in cases of class imbalance, combining precision and recall to provide a balanced view of model performance.
- **ROC-AUC Score:** Useful for understanding the model's ability to distinguish between classes, particularly important in imbalanced datasets.
- **Confusion Matrix:** Offers detailed insight into the types of errors made by the model, such as misclassifications between classes.

8 | Results and Analysis

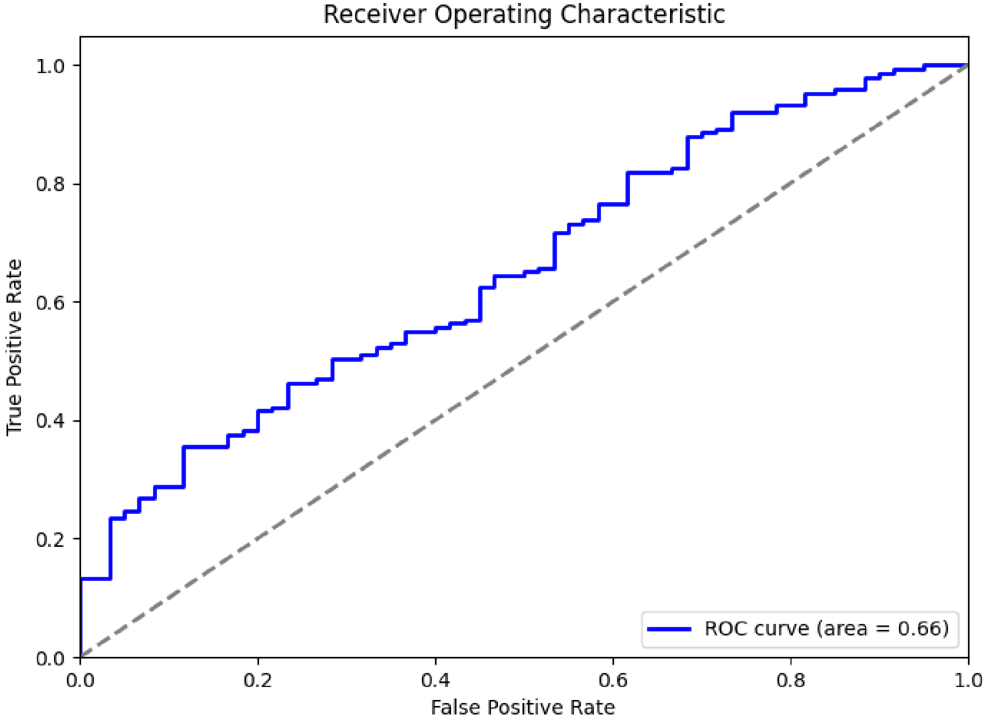
The model was trained/tested with the original/enhanced data, giving rise to four distinct configurations which were evaluated.

Enhanced Train and Test Data



Optimal Threshold: 0.767

Enhanced Train Data and Original Test Data

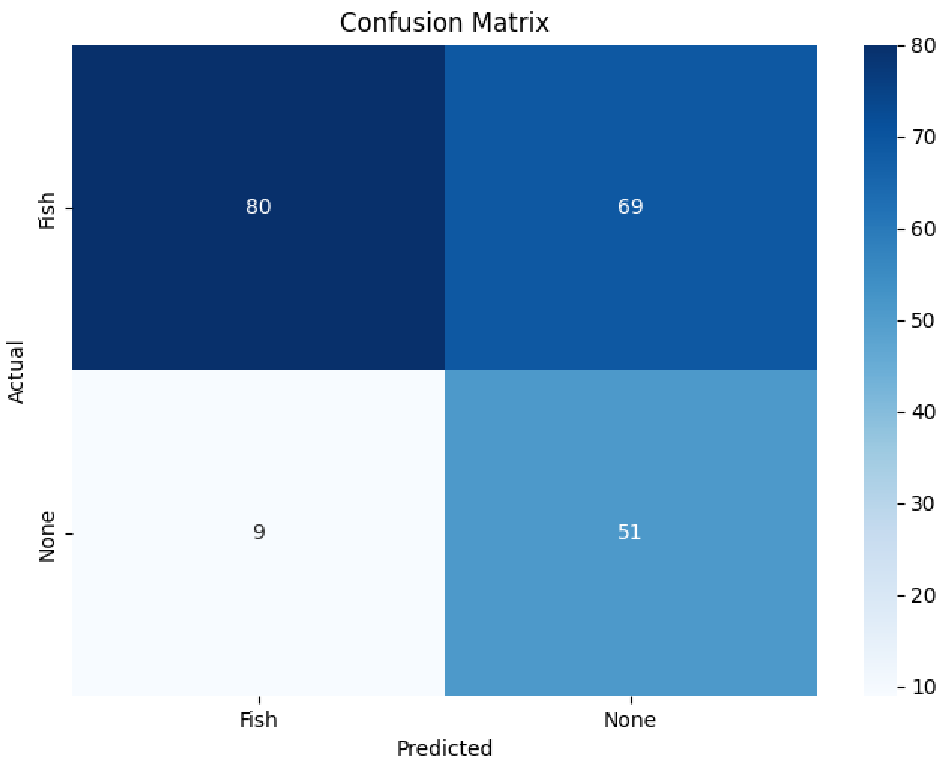


Optimal Threshold: 0.711

8 | Results and Analysis

The model was trained/tested with the original/enhanced data, giving rise to four distinct configurations which were evaluated.

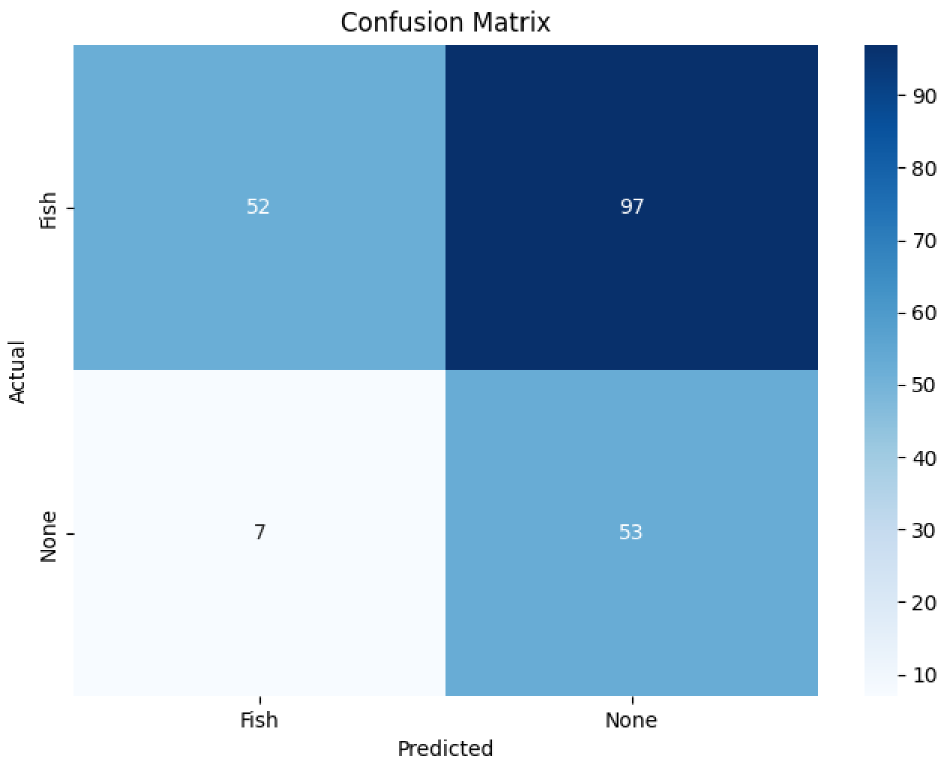
Enhanced Train and Test Data



F1 Score: 0.67
Accuracy: 62.7%

Precision: 89.9%
Recall: 53.7%

Enhanced Train Data and Original Test Data



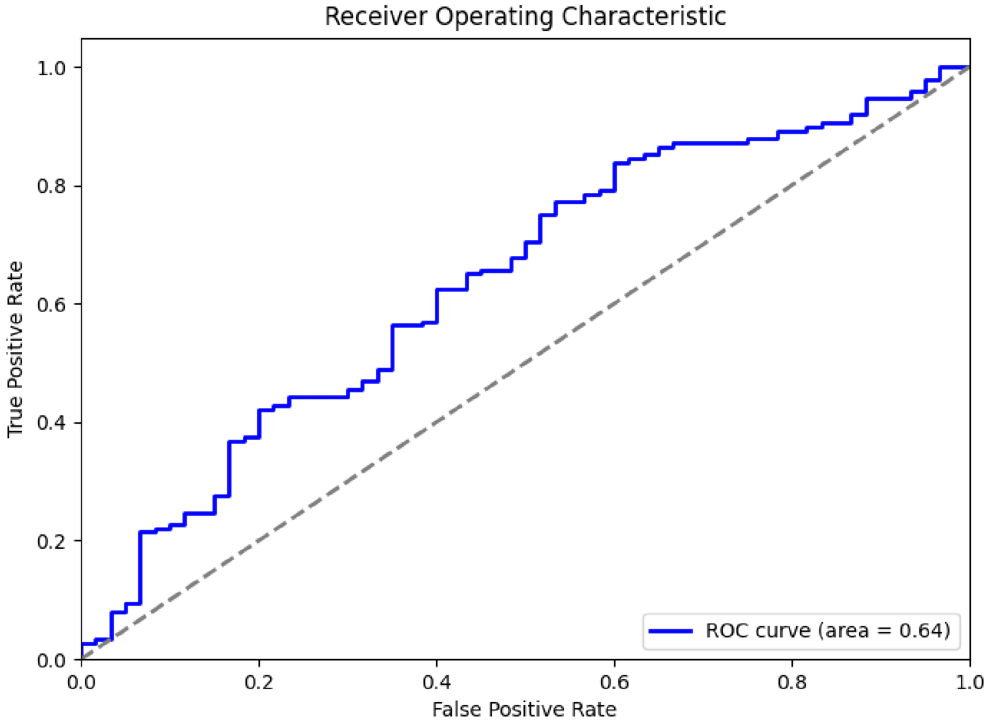
F1 Score: 0.5
Accuracy: 50.2%

Precision: 88.1%
Recall: 34.9%

8 | Results and Analysis

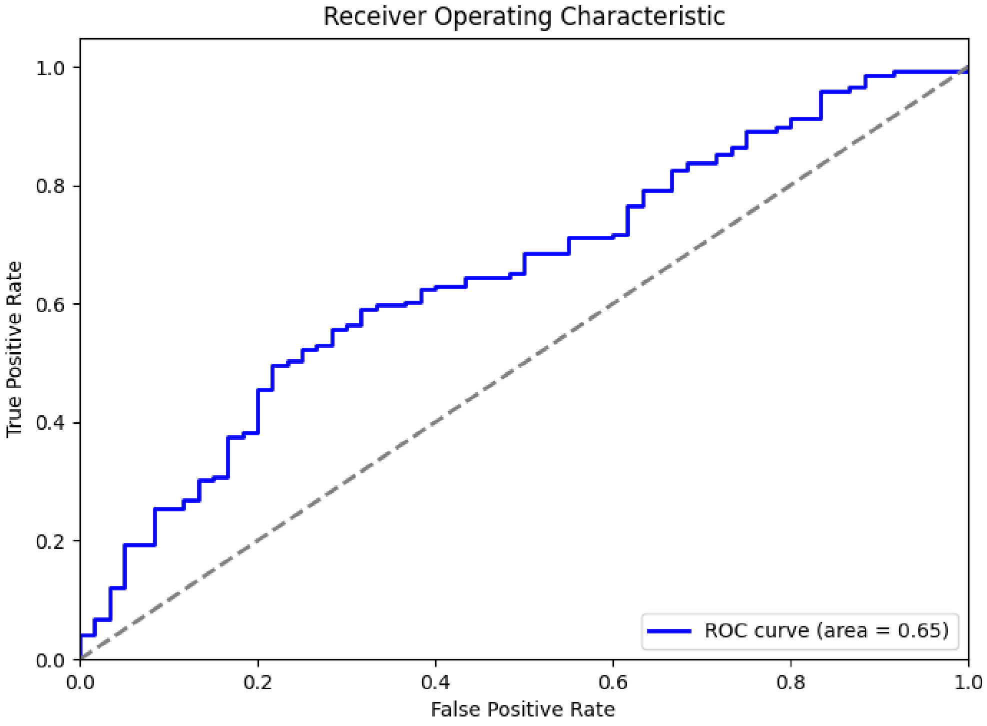
The model was trained/tested with the original/enhanced data, giving rise to four distinct configurations which were evaluated.

Original Train Data and Enhanced Test Data



Optimal Threshold: 0.348

Original Train and Test Data

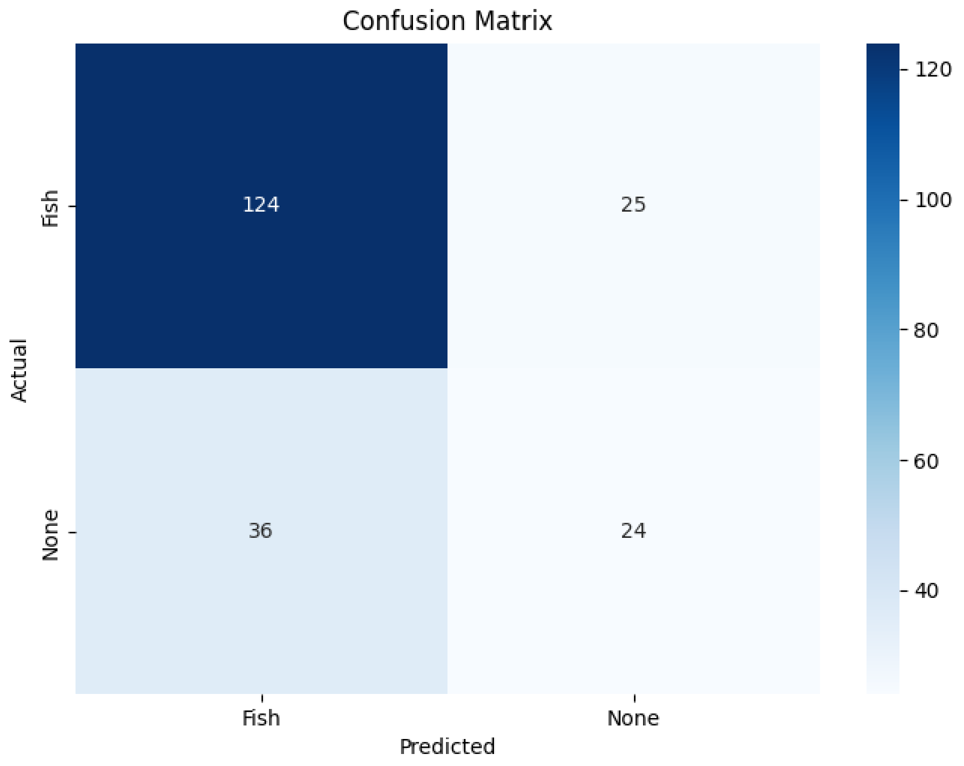


Optimal Threshold: 0.801

8 | Results and Analysis

The model was trained/tested with the original/enhanced data, giving rise to four distinct configurations which were evaluated.

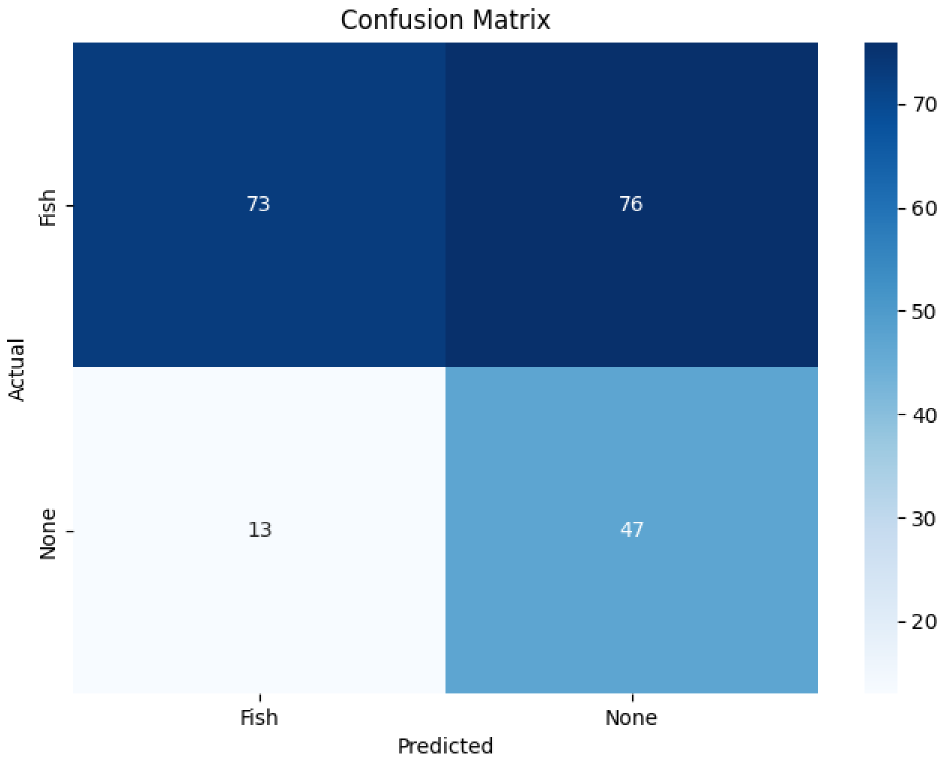
Original Train Data and Enhanced Test Data



F1 Score: 0.80
Accuracy: 70.8%

Precision: 77.5%
Recall: 83.2%

Original Train and Test Data



F1 Score: 0.62
Accuracy: 57.4%

Precision: 84.9%
Recall: 48.9%

9 | Conclusion and Future work

The model was trained/tested with the original/enhanced data, giving rise to four distinct configurations which were evaluated.

- The model's performance across different training and testing configurations was variable, reflecting the complexities of image classification in underwater conditions.

- The optimal thresholds and metric scores such as F1, accuracy, and precision indicate a model that has learned to an extent but also displays notable room for improvement.

- The **disparity in model performance** suggests that image quality has a tangible impact on model efficacy, **aligning with the initial hypothesis**.
    - However, the **performance of the model with original train data and enhanced test data was surprising** (even if understood now, in hindsight)

9 | Conclusion and Future work

The model was trained/tested with the original/enhanced data, giving rise to four distinct configurations which were evaluated.

- **Precision generally exceeded recall**, indicating a model that is cautious in predicting the presence of fish, but at the cost of missing several true positives
  - This could be due to a variety of factors, including possible **class imbalance**, **feature extraction** limitations, or **data quality** issues.

- Future work should focus on addressing these challenges, possibly through further data augmentation, exploring more complex model architectures, or employing advanced techniques like **transfer learning to leverage pre-trained models**.
  - Additionally, investigating the misclassified examples could yield insights for iterative model improvement.