Jamison Meindl

Spring 2021 UROP Summary/User Manual

**Goals:**

The goal of this research was to develop optical recognition software to count herring and other species of fish in fish ladders in the Massachusetts region. This was done in order to improve the manual techniques currently used, which are highly inefficient, time-consuming, and not incredibly accurate.

**Timeline:**

This research began a few years ago in 2017, when another group of UROP students researched using machine learning methods on images of herring to count and identify them. I began this research over IAP 2021 by looking at video feed of cameras in various Massachusetts waterways. This research continued into Spring 2021, where I continued my research and combined the methods I had been studying into a more viable product for counting river herring in fish ladders.

**Methods:**

A combination of optical flow and background removal is used in this analysis. These techniques are useful in extracting information from video, especially when movement is involved. To begin, OpenCV background removal is used to isolate moving parts of the image, such as the fish or variations in the water motion. A mask of these areas of the image is applied, so that the image is black and white. Morphological transformations are then applied to an image with the background removed, so that random isolated points are not included and the part representing the fish is expanded and connected. Contours describing regions where the mask is present are found and boxed. These bounding boxes are then analyzed for their size and location.

Separately, but on the same video feed, optical flow is used to find key points in the image and analyze their movement over time to determine fish directionality. In particular, Lucas-Canade optical flow is used, which looks at the movement of a few selected points. Points that are within the above-described bounding boxes are included in directionality analysis. This is done by averaging the movement of key points within a bounding box to determine a probable direction of the fish.

Through these two methods, a counter is implemented such that fish are tracked across the screen and added to the counter if they are moving right to left. There are various parameters that can be used modify the tracker for different input parameters, such as the number of frames it takes into account before a fish hits the center and whether we include fish where it does not find directionality data.

**Use:**

Currently, the program works by running through a folder of video files and keeping a continuous count of the total number of fish. This can also be run on individual files and is relatively fast compared to the amount of video time. This is currently done in a Python environment, but a user-interface could be implemented.

**Results:**

The program was run on the IRWA 2017 herring data, in which manual counting produced 299 herring and 476 total fish. The following shows the yearly totals the program achieved vs. the hand counted totals.

|  |  |  |
| --- | --- | --- |
|  | Hand Counted | Program |
| Herring | 299 | 312 |
| All Fish | 476 | 426 |
| All Fish, other category removed | 449 | 420 |

**Potential Add-ons:**

Previously, classification of different species of fish, as well as size and speed of fish had been discussed as a possibility but was not completed in this research. In the future, these ideas could be implemented.



Video 2: Background Removal and Morphological Transformations Applied to Video

Video 1: Video of Boxed Fish Alongside Counter