A Two-Feature Underwater Swimmer Detection System

Alden Kane[1], Dr. Adam Czajka[1]
[1]University of Notre Dame

ABSTRACT

DDS (Drowning Detection System) is a two-feature underwater swimmer detection system. It detects and localizes swimmers in a frame-by-frame manner, using both color and motion detection from an underwater video feed to do so. Using the logical AND of the extracted contours from the color and motion detection features, a reliable and robust underwater swimmer detection system, with an Intersection over Union of 0.81 ± 0.12, has been constructed.

INTRODUCTION

Drowning accounts for approximately 3500 deaths annually in the USA. Lifeguarding is a known method of decreasing its incidence, however, many pools are left unguarded (e.g. hotels, gyms, apartments) and human error remains present in drowning detection.

DDS is a computer vision algorithm that operates on optical camera underwater video feeds to provide swimmer detection. It employs two features, color and motion, to do so.

DDS presents an opportunity to automate lifeguarding, or implement redundant systems for lifeguarding. Swimmers that remain underwater for too long, or that remain still underwater, present as likely subjects for drowning. Software-based alerts would be able to alert a human lifeguard, such that he or she can save the troubled swimmer and alert emergency responders.

GitHub Repository w/ Source Code

YouTube Video Demonstration

METHODOLOGY

Data Collection:

- Collected approximately 1.5 hours (23.75 GB) of underwater video feed from both Rolf's Aquatic Center and Rockne Memorial Pool at the University of Notre Dame
- Collected on GoPro Hero7 Black (1920x1080 at 30 FPS)
- Segmented videos every 30 seconds
- Divided into 50% Train/25% Validation/25% Test

Color-Based Feature Extraction:

- Used Hue, Saturation, Value (HSV) values of swimmers from Train Set to set a range of upper and lower HSV values for swimmers
- Binarize color image based on these bounds, with morphological operations applied

Motion-Based Feature Extraction:

- Convert color image to grayscale and apply Gaussian blur to eliminate high-frequency noise
- Perform absolute difference operation between the first and current frame of the video
- Globally threshold image to create binary image

Classifier:

- Binary classifier (i.e. Swimmer(s), Not a Swimmer)
- Take the logical AND of the binary images resulting from color-based and motion-based feature extraction
- Search for contours using cv2.findContours() function, then ignore contours < minObjectSize

ACCURACY

- Accuracy for DDS was defined by the calculation of Intersection over Union (IoU) for random samples from the training and validation sets
- For three training, validation, and test videos, 30 frames were random selected and written to a file
- Ground truth bounding boxes, defined as the smallest box to encompass the whole swimmer, were drawn manually in Preview

OPERATIONS PERFORMED

Figure 1. Color Detection Binary Image

Figure 2. Color Detection Binary Image after Morphological Operations

Figure 3. Baseline First Frame for Motion Detection

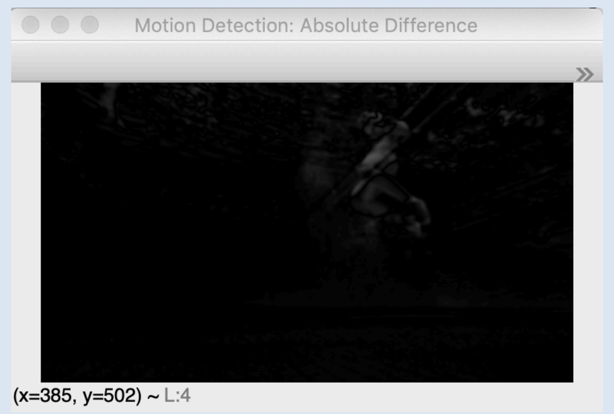

Figure 4. Absolute Difference Between Current Frame and First Frame

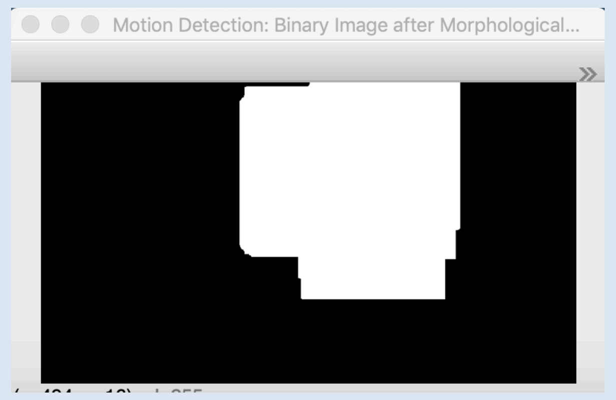

Figure 5. Motion Detection Binary Image with Morphological Operations

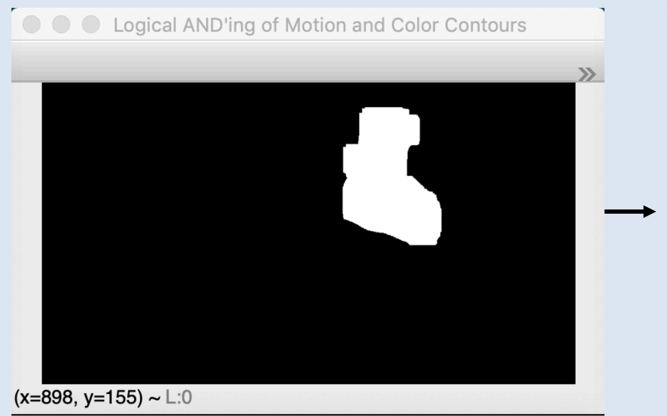

Figure 6. Logical AND of Figures (2) and (5)

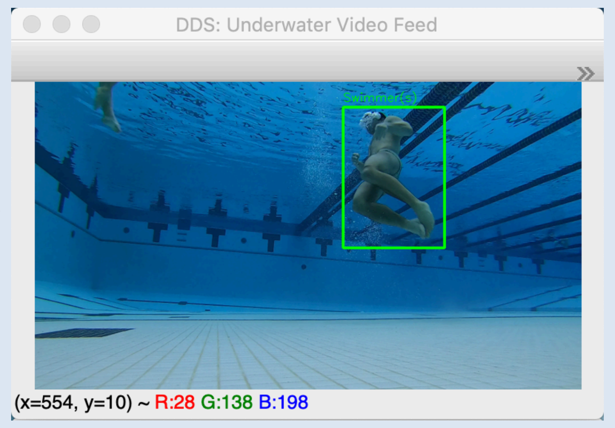

Figure 7. Boxed Swimmer with Contours, Bounding Boxes Drawn from Figure (6)

ACCURACY (CONT.)

Table 1. IoUs for Test, Validation, Train Sets

Data Set	Video 1 IoU	Video 2 IoU	Video 3 IoU	Overall IoU
Test	0.501 ± 0.196	0.466 ± 0.077	0.599 ± 0.158	0.525 ± 0.161
Validation	0.846 ± 0.118	0.781 ± 0.108	0.782 ± 0.151	0.797 ± 0.134
Train	0.813 ± 0.128	0.840 ± 0.110	0.812 ± 0.089	0.825 ± 0.111

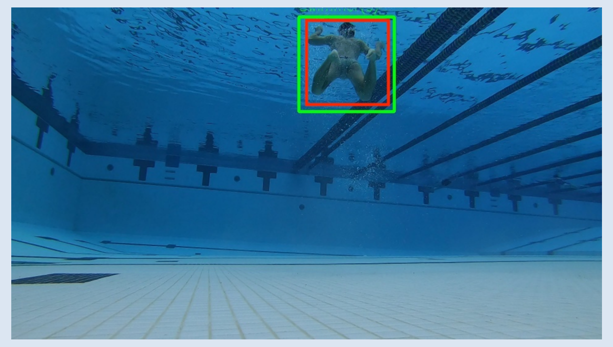

Figure 8. IoU Close to Mean, IoU = 0.796. Ground Truth Bounding Box = Red, DDS Bounding Box = Green. Photo from Validation Set.

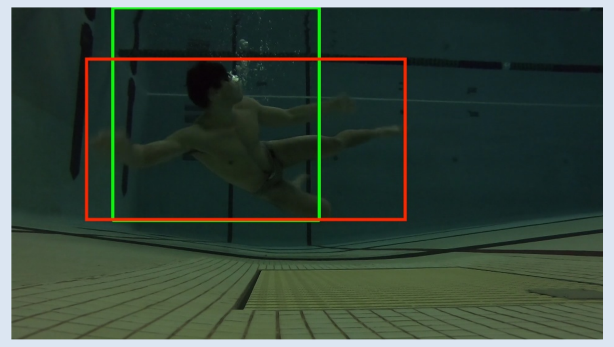

Figure 10. IoU Close to Mean, IoU = 0.542. Photo from Test Set

CONCLUSION

The two-feature vision algorithm employed here provides good classification of swimmers in well-lit underwater environments. Its ability to generalize is affected by local pool conditions, as seen in the decreased Test Set accuracy (set sourced from different pool than Train, Validation sets). Adding additional features to this system will likely improve performance and accuracy.

NEXT STEPS

System accuracy could be improved by:

- Training a neural network of underwater swimmers, then implementing majority-voting
- Adding behavioral models for drowning and timers to log underwater swimmer time

ACKNOWLEDGEMENTS

- Dr. Adam Czajka for his direction
- Adrian Rosebrock (PyImageSearch) for his useful CV tutorials