Underwater Object Detection through HOG-SVM and Deep Learning

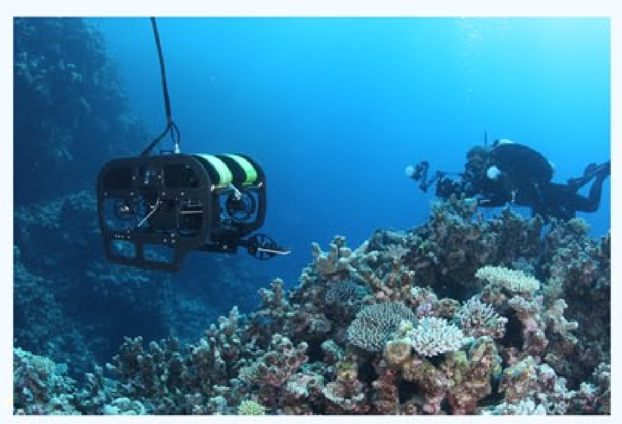
THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

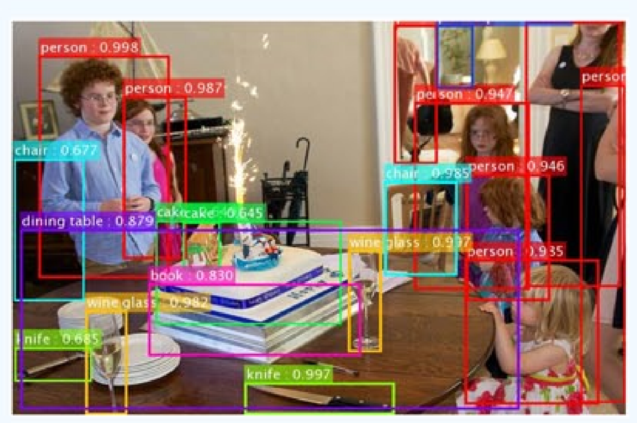
UY, Mikaela Angelina Chan

Advised by Prof. Chi-Keung Tang

OVERVIEW

In the recent years, development in the robotics industry has been growing fast. Much research has been devoted to many of its applications, including aerial drones that resulted in many commercially used drone applications. However, limited works have been done in the field of underwater robotics, especially to the development of Autonomous Underwater Vehicles (AUV). This study aims to contribute to the research of AUV's by developing a real-time underwater diver video detection system using object detection algorithms.





Object detection is one of the fundamental problems in computer vision. This refers to identifying specific objects of interest on a given image, and enclosing them in tight bounding boxes.Conventional methods primarily use a feature-based, sliding-window approach, which means sliding a fixed sized window on an input image and extracting features from each that are then fed into a classifier. More recent and state-of-the-art approaches for object detection use deep learning architectures through convolutional neural networks.

OBJECTIVE

A Real-Time Underwater Diver Video Detection System

APPROACH

1. Conventional — **HOG-SVM**
2. State-of-the-art — **YOLO**

TRAINING DATASET

HOG-SVM

TOO SMALL!

Original Dataset
127 positive (diver) images — 349 negative (non-diver) images

Data Augmentation

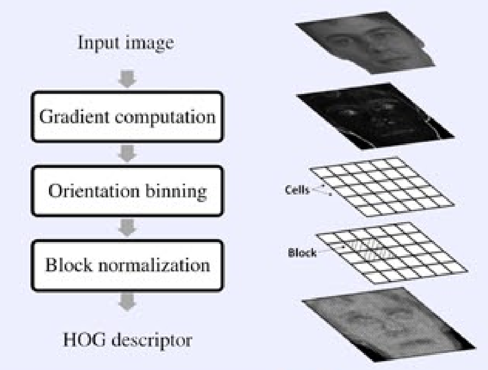
Augmented Dataset
381 positive images — 554 negative images

YOLO

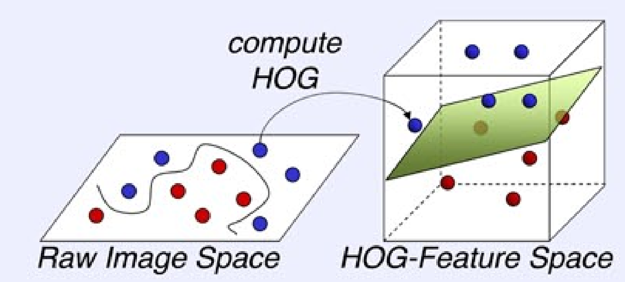
- 300 diver images
- Size: 480 x 480 pixels
- Comprising of divers with various orientations

HOG-SVM

Feature extraction via Histogram of Oriented Gradients (HOG)



Classification via Linear Support Vector Machine (SVM)



PROCESS

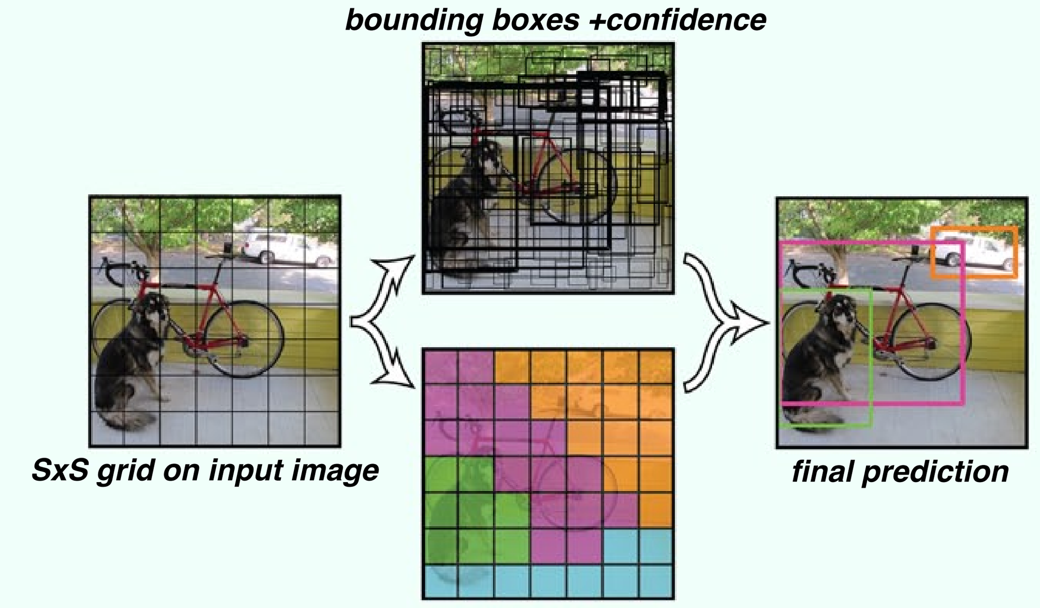
Input dataset (sized 64x128) → HOG feature extraction → SVM Classifier Optimization → HOGDescriptor *detect()* function Optimization → HOGDescriptor *detectMultiscale()* function Optimization

FINDINGS

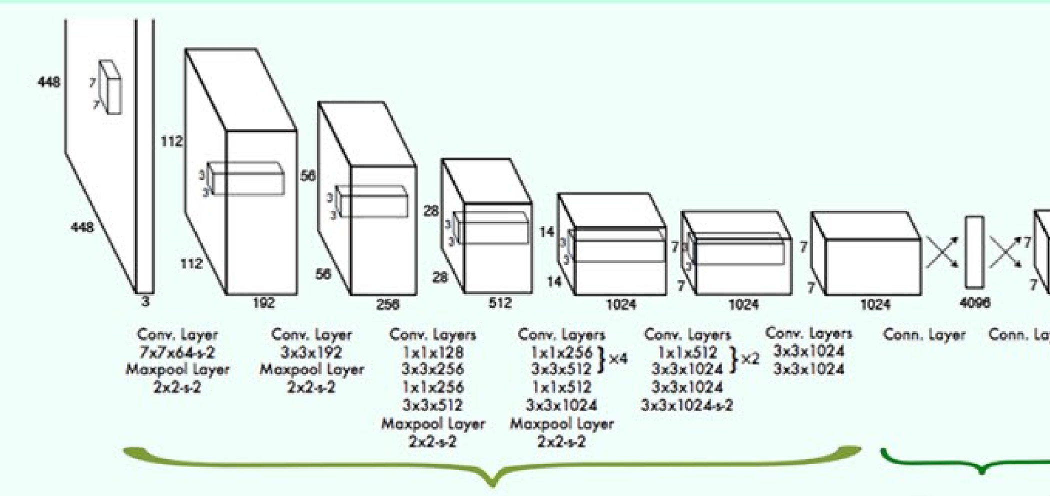
Original Dataset (BAD) → Augmented Dataset (BETTER)

But still, high rate of **FALSE POSITIVES**

YOLO Single-shot Detector



YOLO is a **UNIFIED** detector that uses a single neural network to predict: **object class** + **bounding box** + **confidence**



Pretrained model

- 24 CNN layers + 2 fully-connected layer
- Detects 20 object classes
- Used "person" class to detect divers

Problem: undetected & misclassified divers

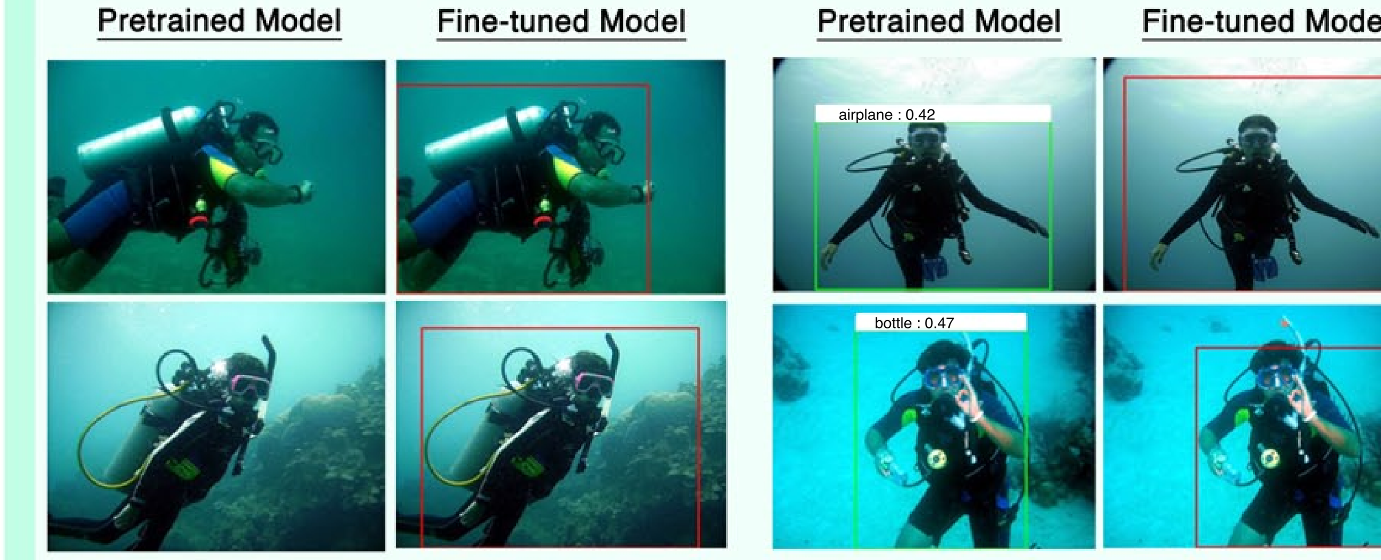
Solution: fine-tuning by **Transfer Learning**

Fine-tuned model

- single-class diver classifier
- fine-tuned weights of last 2 fully-connected layers of pre-trained model

RESULTS

Pretrained Model | Fine-tuned Model | Pretrained Model | Fine-tuned Model



✓ Detect previously undetected divers

✓ Detect previously misclassified divers

SYSTEM

Fine-tuned YOLO model
- Latency = 1 fps
- Called once every 15 frames

\+

Tracking Algorithm
- Used Kernelized Correlation Filters (KCF)
- Latency = 48 fps
- Called 14x every 15 frames

=

Real-Time Video Detection System

Averaged **45 fps**

Satisfies real-time requirement