GREAT BARRIER REEF: INVASIVE STARFISH DETECTOR

Nick Wagner and David Unger

Institute of Signal Processing and System Theory, University of Stuttgart, Germany

1. Problem Statement

Abstract

The Great Barrier Reef in Australia is known world-wide for its diverse animal and coral species. Recently, the overpopulation of the coral-eating crown-of-thorns starfish threatens the existance of many corals. To allow divers to efficiently remove these star fishes from the corals, we develop an object detection algorithm.

The algorithm is based on the "YOLO"–framework and takes video sequences as input, detects the starfishes and draws bounding boxes around them.

Dataset

- ~23000 images
  from 3 video sequences
- bounding box labels from csv
- binary problem

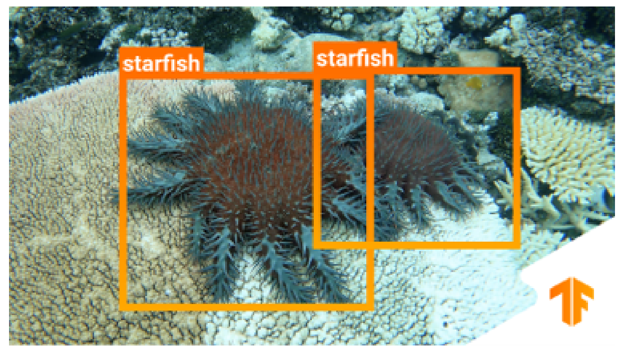

Abbildung 1: test123

2. Architecture

- Text item
- Text item
- Text item

3. Network Output

- Text item
- Text item
- Text item

4. Input Pipeline

Compared to a classification network, a detection network requires a more complicated input pipeline. This mainly comes from the labels which are dependent on the image content and that change, if the image is scaled, rotated, cropped or flipped.Therefore, the input pipeline consists of the following steps:

- CSV loading
- Bounding box text to grid conversion
- Image reading
- Image resizing
- Image augmentation
- Image visualization

5. Loss Function

To ensure that the network learns the bounding boxes as represented by the ground truth labels, using a standard loss function is not sufficient. Therefore, the loss function from the YOLO paper is used and adapted that it fits the described problem setup.

The four parts of our loss function are the following and summed up during training

- Objectness loss: $l_{obj} = \lambda_{obj} \sum_{i=0}^{(S-1)^2} \mathbb{1}_i^{obj} (1 - \hat{p}_i)^2$
- No object loss: $l_{noobj} = \lambda_{noobj} \sum_{i=0}^{(S-1)^2} \mathbb{1}_i^{noobj} (0 - \hat{p}_i)^2$
- BBox center loss: $l_{center} = \lambda_{center} \sum_{i=0}^{(S-1)^2} \mathbb{1}_i^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$
- BBox size loss: $l_{size} = \lambda_{size} \sum_{i=0}^{(S-1)^2} \mathbb{1}_i^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$

The loss term weightings λ are determined in a way, that all loss terms are in a similar range and therefore are optimized in a similar strength.

6. Challenge

Deep Learning Lab 2022, February 8, Stuttgart, Germany