Pattern Analysis and Applications (2024) 27:4
https://doi.org/10.1007/s10044-024-01227-6

THEORETICAL ADVANCES

How to track and segment fish without human annotations: a self-supervised deep learning approach

Alzayat Saleh[1] · Marcus Sheaves[1] · Dean Jerry[1,2] · Mostafa Rahimi Azghadi[1,2]

Received: 10 April 2023 / Accepted: 4 December 2023 / Published online: 23 February 2024

Abstract

Tracking fish movements and sizes of fish is crucial to understanding their ecology and behaviour. Knowing where fish migrate, how they interact with their environment, and how their size affects their behaviour can help ecologists develop more effective conservation and management strategies to protect fish populations and their habitats. Deep learning is a promising tool to analyse fish ecology from underwater videos. However, training deep neural networks (DNNs) for fish tracking and segmentation requires high-quality labels, which are expensive to obtain. We propose an alternative unsupervised approach that relies on spatial and temporal variations in video data to generate noisy pseudo-ground-truth labels. We train a multi-task DNN using these pseudo-labels. Our framework consists of three stages: (1) an optical flow model generates the pseudo-labels using spatial and temporal consistency between frames, (2) a self-supervised model refines the pseudo-labels incrementally, and (3) a segmentation network uses the refined labels for training. Consequently, we perform extensive experiments to validate our method on three public underwater video datasets and demonstrate its effectiveness for video annotation and segmentation. We also evaluate its robustness to different imaging conditions and discuss its limitations.

Keywords Computer vision · Convolutional neural networks · Image and video processing · Underwater videos · Machine learning · Deep learning

1 Introduction

The automatic tracking and segmentation of individual fish have emerged as pivotal tools in the field of ecological behavioural analysis, with a broad spectrum of applications. This is evidenced by numerous studies in the domain [1–6]. The ability to understand and predict animal motion in their natural habitats could yield significant benefits across various research and industry domains [7–11]. However, the inherent complexity of animal movement in the wild presents a great challenge. Factors contributing to this complexity include intermittent visibility of animals in videos and the presence of multiple animals within a single video frame, both of which complicate tracking and segmentation tasks. Addressing these challenges often necessitates the deployment of advanced computational methods.

A large number of studies have attempted to tackle these challenges [12–18]. These studies predominantly rely on pixel-level annotations to train or enhance their deep neural networks (DNNs). However, obtaining these annotations is both costly and time-consuming, particularly for fish segmentation in the wild. Most current automated methods operate under the assumption that training data are typically paired with ground truth derived from videos containing a large number of fish [13, 19–22]. Despite the high cost associated with obtaining ground truth, it is necessary to acquire a substantial number of video sequences. This is due to the difficulty in achieving accurate results using only a limited number of sample videos.

This study was motivated by the importance of the challenges faced when trying to annotate and segment animals in videos in the wild. Unlike in controlled conditions, where animals are easily distinguishable from the background, fish are difficult to distinguish in realistic videos [23–25], even with domain knowledge. This is due to the large variations

✉ Mostafa Rahimi Azghadi
mostafa.rahimiazghadi@jcu.edu.au

1 College of Science and Engineering, James Cook University, Townsville, QLD, Australia

2 ARC Research Hub for Supercharging Tropical Aquaculture through Genetic Solutions, James Cook University, Townsville, QLD, Australia

in the appearance of the animals, lighting conditions, and background.

Our approach aims to develop an unsupervised method for fish tracking and segmentation without the need for human annotations, by leveraging spatial and temporal variations in video data using known techniques of background subtraction and optical flow, as shown in Fig. 1. Specifically, we propose to generate pseudo-labels based on unlabelled video data. The use of pseudo-labels can benefit various learning-based algorithms since it can significantly reduce the labelling cost. The key to the proposed method is to take advantage of the intrinsic temporal consistency between consecutive frames to improve the generated labels by refining them with a self-supervised model. We propose training a Deep Neural Network (DNN) to segment individual fish based on the generated pseudo-labels. As long as the pseudo-labels are generated in a way that they have similar structure and appearance to real ones, the model can learn to understand the underlying structure from the pseudo-labels. In general, the more realistic the pseudo-labels, the better the segmentation accuracy. We include a short video of our model prediction in https://youtu.be/Z5G7YBoL3eM and https://youtu.be/8LOKsVSiY9U.

The main contributions of this paper are listed as follows:

- Propose to use pixel-level pseudo-labels generated by an optical flow model and background subtraction to learn the segmentation and tracking of individual fish automatically without manual interaction.
- Demonstrate that using self-supervised refinement, we can further improve the accuracy of the pseudo-labels for fish tracking and segmentation.
- Evaluate our method on three public datasets with different image quality.

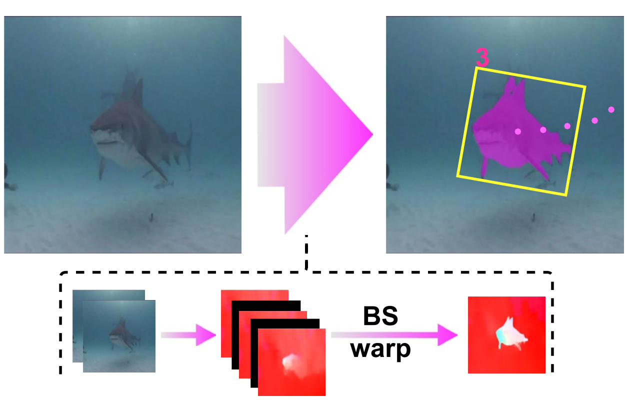

Fig. 1 Combining background subtraction and optical flow demonstrate how both levels work in concert to preserve object boundaries and temporal coherence throughout the video. Please refer to Sect. 3 for details

- Discuss the limitations of the current model and our future research directions.

The rest of the paper is organised as follows. Section 2 covers related works and provides background information on the novel aspects of our work. Our model's framework is described in detail in Sect. 3. Section 4 presents our method for training and evaluating our model. The experimental setup and results are presented in Sect. 5, while detailed discussions of our results are presented in Sects. 6. Finally, Sect. 7 concludes our paper.

2 Related work

The field of video object segmentation and animal tracking has witnessed substantial advancements in recent years. A noteworthy contribution is the unsupervised video object segmentation model, UVOSAM, proposed by Zhang et al. [6]. This model, which operates without the need for masks, has introduced new possibilities in the field. In the realm of animal tracking, Dutta et al. [11] have developed a deep learning workflow that holds particular relevance for ecological studies. Complementing this, Javed et al. [12] have provided a comprehensive survey of visual object tracking techniques, significantly enhancing our understanding of the current landscape.

Further enriching the field, Cao et al. [22] proposed a method of dense spatio-temporal position encoding to improve tracking accuracy. Proencca et al. [25] introduced TRADE, a method that utilises 3D trajectory and ground depth estimates for UAVs. In terms of segmentation, Jahanbakht et al. [26] explored distributed deep learning and energy-efficient image processing for fish segmentation. Zhang et al. [27] developed MSGNet, which uses multiple sources of information to improve the precision of fish segmentation. This approach has been influential in shaping our own methodology. In the following subsection, we will provide a brief review of the research domains that are most relevant to our work.

Video object segmentation is a task that is used to locate and segment each target object [28–30]. The target object to be segmented can be either a class of interest in the videos or moving objects of interest. Object segmentation is generally categorised into two categories: segmentation with instance-level semantics and segmentation without instance-level semantics, which is the main objective of this paper. Therefore, this study focusses on generating labels without human intervention. Some segmentation methods for moving objects have been developed by using background subtraction techniques [31–33]. Several of these approaches are based on the assumption that the scene is locally constant [34, 35]. This means that the background in one frame is assumed to be similar to the background in the next frame or only a few pixels away. In order to use this assumption, they estimate the local background and threshold it according to the similarity threshold to identify foreground regions. However, this method is known to be sensitive to illumination changes and may even lose all detail within the image due to overestimating the local background. Another approach to segmentation uses the detection of optical flow to define motion boundaries [36–39].

Optical flow predicts the relative motion of objects in two consecutive frames of a video [38, 39]. It gives a dense correspondence between frames, but at the cost of being limited to rigid objects, and computation entangled. Additionally, optical flow can only work within scenes where the movement of the camera is significantly lower than the movement of the object [36, 37, 40]. This can be seen as a limitation, as the background subtraction method can be used in a wider range of applications. However, the key element of optical flow is that it can also be used for background subtraction [41, 42]. By tracking the movement of the pixels between frames, we can determine the background. If a pixel that is part of the background does not match the static background within a given threshold, then that pixel is determined to be an instance of an object.

Another segmentation approach is based on the detection of visual motion. It is based on the fact that moving objects in the scene induce consistent changes to the flow of pixels in a region [37, 40]. However, due to substantial displacements or occlusions, their calculated optical flow may contain considerable inaccuracies [43–45]. In our method, we address these issues and enhance both estimated optical flow and object segmentation, simultaneously.

Video object tracking is the task of assigning a consistent label for each individual object in the scene as it moves [46, 47]. This tracking is generally divided into multiple steps, including detection of the object of interest, tracking of the moving object in the scene, and then associating labels between frames. The tracking task, therefore, consists of identifying the bounding box of the object over several video frames and, at the same time, updating the location of the object in the image [48, 49]. This can be done based on a similarity metric between different frames [50, 51]. The idea of such a metric is to find the closest objects in the frame with an overlapping bounding box. This can be performed at either the pixel level or at the region level. The major drawback of this method is the computation time [52, 53] that is needed to compute all the similarities between all the different frames. On the other hand, if the computational resources are available, this method has been proven to be useful when tracking fast-moving objects and when the objects are not occluded in the frame [54, 55]. In our method, we produce the rotating 2D object bounding box from each instance mask of the object over several video frames.

In contrast to our work, Yang et al. [56] uses a Siamese network with an anchor-free tracker for general object tracking, simplifying the tracking algorithm by avoiding the anchor box design that predicts the tracking target with a pair of corners (top-left and bottom-right corners). While both our work and SiamCorners [56] utilise deep learning techniques for object tracking, there are key differences. Our approach specifically addresses the challenges of fish segmentation and tracking in underwater videos, leveraging optical flow and background subtraction to generate pseudo-ground truth labels. In contrast, SiamCorners simplifies the tracking algorithm by avoiding anchor box design but does not specifically address the unique challenges of fish segmentation and tracking in underwater videos. We believe these distinctions highlight the novelty and significance of our work in this specific domain.

Supervised And Unsupervised Learning. Supervised learning has been used to build object detection [57–59], video object segmentation [29, 30] and video object tracking [47, 48]. These methods require extensive human annotation and therefore are not suitable for video annotation in the wild. To reduce the labelling costs of data, unsupervised learning has emerged as a powerful technique for the learning of video data. In the traditional image domain, unsupervised methods are expected to outperform their supervised counterparts [13, 14, 16, 60, 61] due to their potential to train data without labels.

The idea behind many of the unsupervised DNN models is to learn a feature representation from unlabelled data [62–64]. Then, a DNN model can be applied to the learned feature representation to produce the output. For example, in the domain of video segmentation [15, 65–67], DNNs have been used to learn a representation from the difference between a pair of unlabelled videos [68–70] and from warped frames [71].

In this work, we focus on unsupervised learning. Our proposed method will generate labels referred to as pseudo-labels to train a multi-task supervised DNN for video object segmentation and video object tracking.

3 Framework

The overall framework can be divided into three stages as shown in Fig. 2. The first stage is to generate pseudo-labels using background subtraction and optical flow for both videos and still images. The second stage is to train a self-supervised model to refine the pseudo-labels using their spatial structure. In the last stage, the refinement of the video and still-image versions are applied jointly to train the segmentation network and to predict the final label. The segmentation

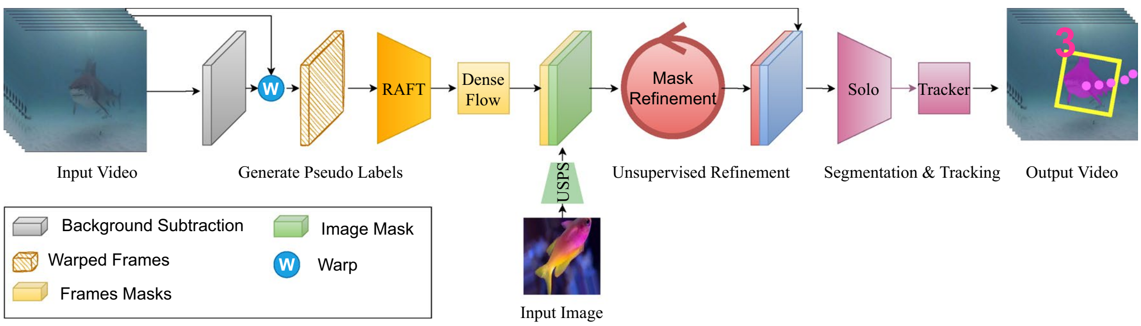

Fig. 2 Our proposed framework consists of three main components: generate pseudo-labels, unsupervised pseudo-labels refinement, and segmentation network. The proposed segmentation model trains with the generated pseudo-labels, which are refined with self-supervised training. Please refer to Sect. 3 for details

network's training behaviour closely matches supervised training because we employ improved pseudo-labels. As a result, the network's training process is more reliable than that of current unsupervised learning techniques [68–70, 72]. In the following subsections, we describe the details of these three components and the corresponding loss functions.

3.1 Background subtraction

As a first step to generating pseudo-labels, background subtraction is performed on the video frames. A clean background image is estimated for every video sequence by computing the median of the first 10 frames of the video sequence along the first axis. This is to average out any distracting elements that come in front of the clean background. Then, each video frame is subtracted from the clean background to create the mask sequence. After subtracting, all foreground pixels take on a value of 1, and pixels belonging to any background region have 0 values using adaptive Gaussian thresholding [73].

Adaptive Gaussian thresholding is used instead of one global value as a threshold because it sets a pixel's threshold based on a local region surrounding it. As a consequence, we obtain various thresholds for various areas inside the same image, which produces better results for images with varying illumination.

This background subtraction step is crucial in eliminating any stationary elements or shadows from the video sequences that might disturb the next step, optical flow.

3.2 Optical flow

The next step in pseudo-label generation is to calculate the optical flow using recurrent all-pairs field transforms (RAFT) [74]. However, optical flow is frequently inaccurate at object boundaries, so we want our segmentation to be accurate exactly at these borders. Therefore, we consider video segmentation from background subtraction and optical flow estimation simultaneously. Using pixel level and temporal information sources, the segmentation algorithm is improved by removing artefacts induced by background subtraction and optical flow. We demonstrate how both levels work in concert to preserve object boundaries and temporal coherence throughout the video. The key is that we need to remove motion blurs while preserving the motion of the fish boundaries.

To achieve the pseudo-labels, we first deconstruct a pair of video frames, x_t and x_{t+1}, and estimate a mask m_t and m_{t+1} with the background subtraction method as described in Sect. 3.1. Segmented masks m_t, m_{t+1} are used to synthesise frames $\hat{x}_t$ and $\hat{x_{t+1}}$ by warping x_t and x_{t+1} with m_t, m_{t+1}, respectively. The optical flow [74] takes two frames $\hat{x}_t$ and $\hat{x_{t+1}}$, and produces a motion vector $\hat{v}$ between them. This motion vector is used to compute the magnitude and angle of the motion. Specifically, pixels with a motion vector $\hat{v}$ outside m_t (and m_{t+1}) are assigned the value of the background, and pixels with a motion vector $\hat{v}$ inside m_t (and m_{t+1}) are reassigned the object. We denote the reassigned images as $\hat{x_t^*}$ and $\hat{x_{t+1}^*}$ and use them as input for our segmentation step, as shown in the top panel of Fig. 2 (Fig. 3).

We show the optical flow results for the three video datasets with and without background subtraction of frames x_t and x_{t+1} in Figs. 4, 5, and 6. A mask m_{t+1} that better distinguishes the background from the foreground from the optical flow step is then refined with our proposed unsupervised refinement method in the next section. A sample optical flow comparison video before and after background subtraction is available at https://youtu.be/8LOKsVSiY9U.

3.3 Unsupervised refinement

The second stage in our method is cumulative pseudo-label refinement through unsupervised historical moving averages (MVA) [77] using DeepLabv3 [78] network for semantic segmentation and Conditional Random Fields

Fig. 3 Sample image from each of the four utilised datasets. From left: Seagrass [23], DeepFish [24], YouTube-VOS [75], and Mediterranean Fish Species [76]

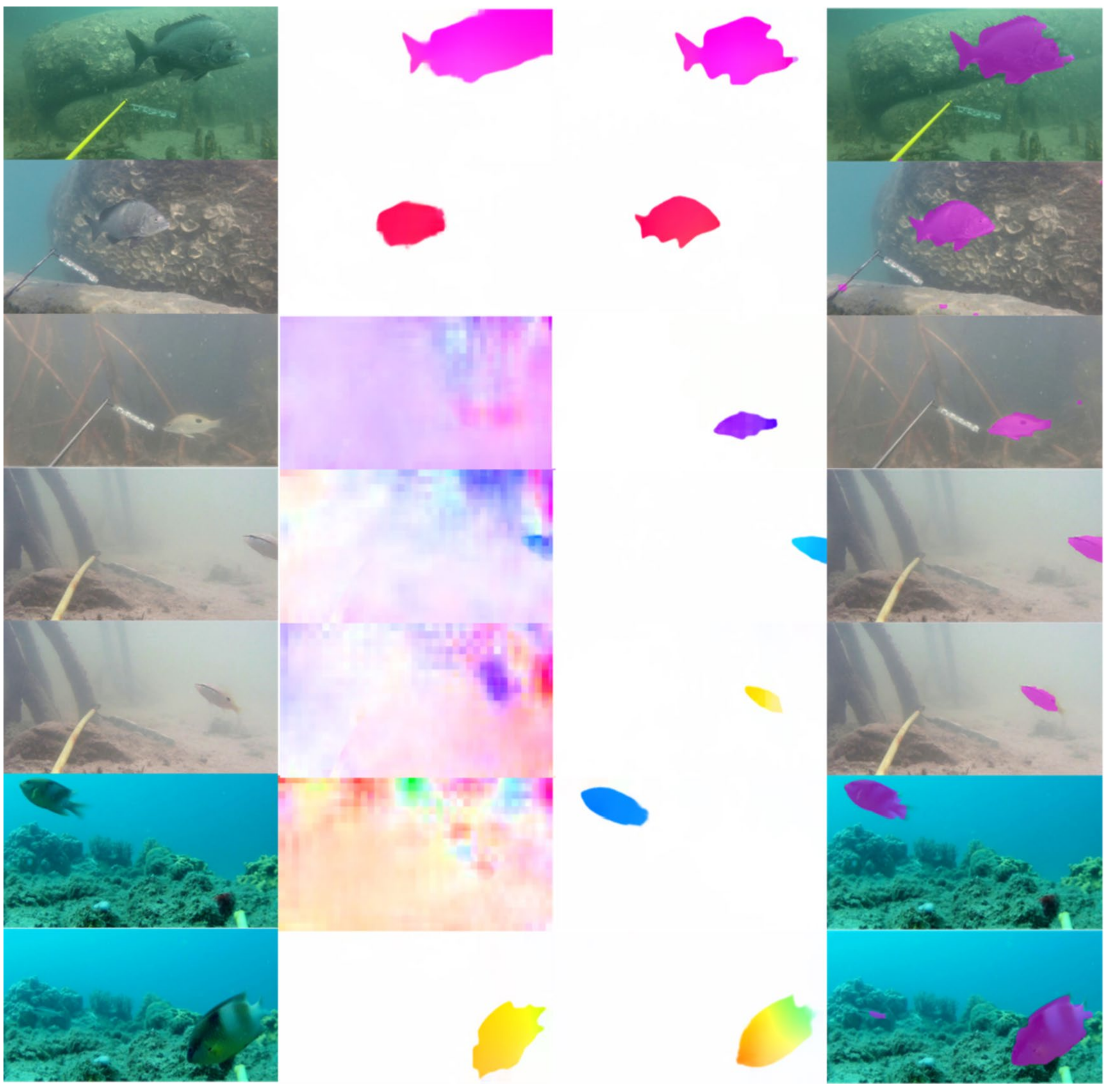

Fig. 4 Sample optical flow results for Seagrass [23]. From left, the original image, optical flow without background subtraction, optical flow with background subtraction, mask overlay

(CRF) [79] by minimising the F-score until the MVA predictions reach a stable state. The CRF can "sharpen" initial location predictions to make them more accurate and consistent with edges and parts of the source image that have a constant colour.

Given the pseudo-labels of the previous step, we train the network for 50 epochs. The number of epochs is low to avoid a significant over-fitting of the network to the noisy pseudo-labels. Then, the network is reinitialised with trained weight to predict a new set of pseudo-labels to train on again.

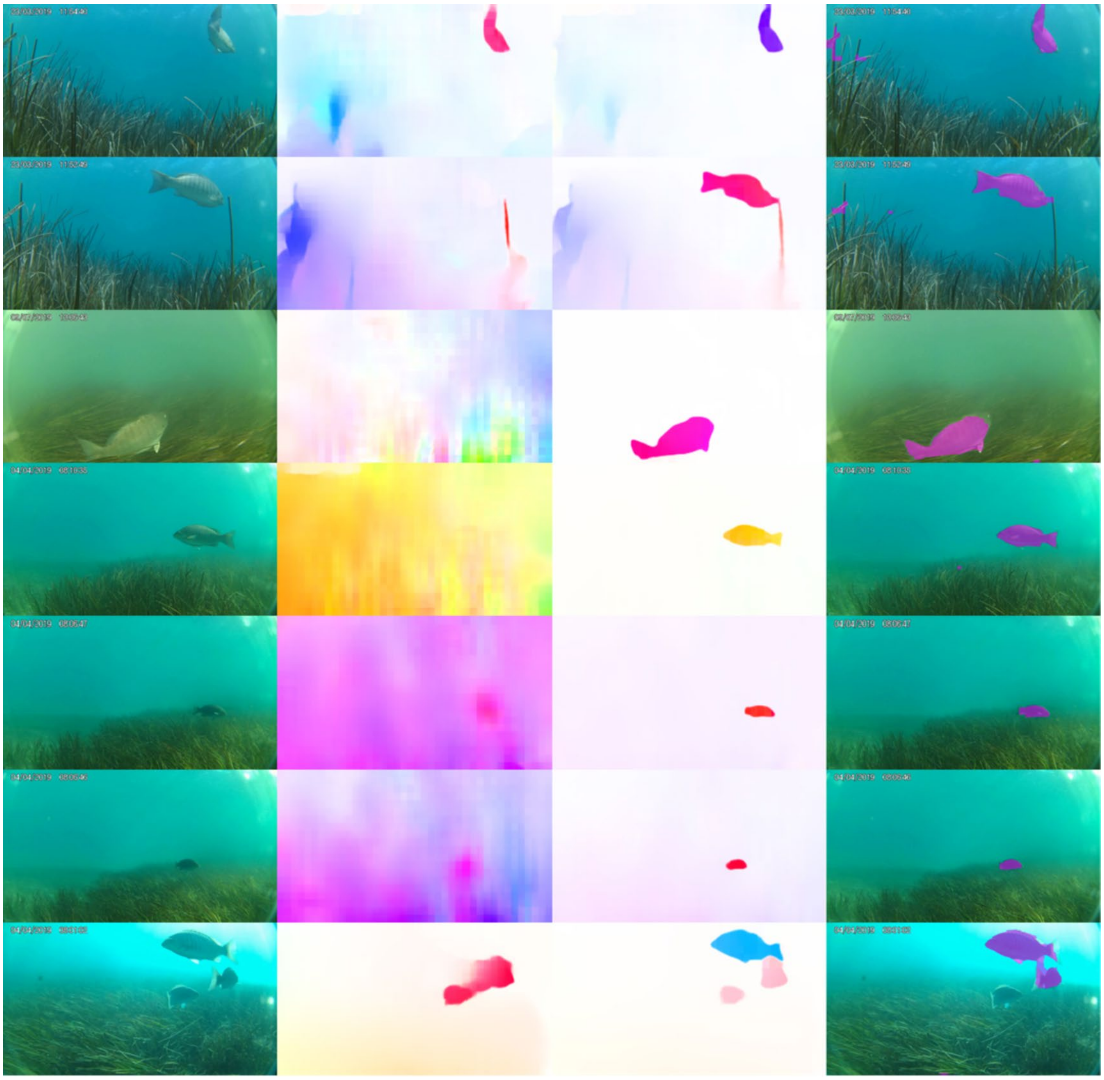

Fig. 5 Sample optical flow results for DeepFish [24]. From left, the original image, optical flow without background subtraction, optical flow with background subtraction, mask overlay

Let D be the set of training examples and M be the network model. By $M(x, p)$, we denote the mask prediction of model M on the pixel p of the image $x \in D$. During this stage, a historical moving average (MVA) from the last training stage is composed as follows:

$$\mathrm{MVA}(x, p, k) = (1-\alpha) * CRF(M(x, p)) + \alpha * \mathrm{MVA}(x, p, k-1),$$

where $M(x, p)$ is the network mask prediction, k is the epoch number, α is a positive real factor, and CRF is the conditional random fields (CRF) [79].

We use $L_\beta = 1 - F_\beta$ as an *image-level loss* function w.r.t. each training example x. F-score (F_β) is the harmonic mean of precision and recall of the prediction output of pixel p on image x w.r.t. the pseudo-labels, which use a positive real factor β as follows:

$$F_\beta = \left(1 + \beta^2\right) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \text{ precision} + \text{recall}}.$$

The network is retrained until the MVA reaches a stable state, as shown in the middle panel of Fig. 2. By doing so, the quality of pseudo-labels is improved over time.

3.4 Segmenting objects by locations

Our last stage is training a supervised segmentation model using the refined pseudo-labels from the previous stage. The supervised model is based on segmenting objects by locations (SoloV2) [80]. SoloV2 is an updated version of Solo [81], a previous method for instance segmentation. The idea is to dynamically segment objects by location.

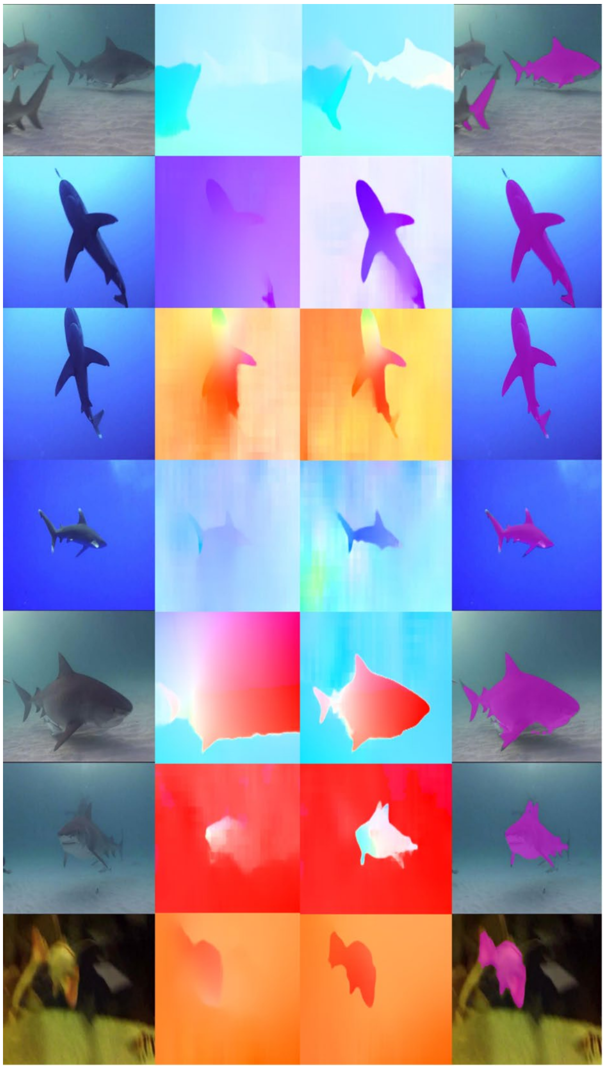

Fig. 6 Sample optical flow results for YouTube-VOS [75]. From left, the original image, optical flow without background subtraction, optical flow with background subtraction, mask overlay

Given an image as input, the network generates the object mask, then the object mask generation is decoupled into a mask kernel prediction and mask feature learning. Furthermore, matrix non-maximum suppression (MNMS) is applied to reduce inference overhead. Specifically, SoloV2 is composed of two modules: (1) dynamic instance segmentation and (2) matrix non-maximum suppression (MNMS). The dynamic instance segmentation scheme dynamically segments objects by location by learning the mask kernels and mask features separately. The mask kernels are predicted dynamically by the fully convolutional network (FCN) [82] when classifying the pixels into different location categories, then constructing a unified mask feature representation for instance-aware segmentation. The non-maximum suppression process is achieved by performing NMS with a parallel matrix operation in one shot to reduce inference overhead and suppress duplicate predictions. Compared to the widely adopted multi-class NMS [83], where the sequential and recursive operations result in non-negligible latency, the parallel non-maximum suppression with matrix operation can achieve similar performance with much lower latency. The parallel processing strategy performs MNMS inference on-the-fly and enables processing at a high frame rate (*34 frames per second*). For more details, we refer the reader to [80].

3.5 Rotating bounding box

From each instance mask that we predicted from the previous stage, we are able to produce the rotating 2D object bounding box. The minimum bounding rectangle (MBR) technique is used to obtain a rotated bounding box from a binary mask of the object. We used OpenCV [84] to find the minimum area of a rotated rectangle. It takes the binary mask of the object as an input and returns a Box2D structure that contains the following information: (centre (x, y), (width, height), angle of rotation). The output of this step is used to track the objects as discussed in the following section.

3.6 Online tracking

We used simple online and real-time tracking (SORT) [85] as an online tracking framework that focuses on frame-to-frame prediction and association. The position and size of the bounding box are used only for both motion estimation and data association. Kalman filter [86] is used to handle motion estimation and the Hungarian method [87] is used for data association.

Motion estimation is used to propagate a target's identity into the next frame. The inter-frame displacements of each object are approximated with a linear constant velocity estimation. The detected bounding box is used to update the target state where the Kalman filter [86] solves the velocity components. The state of each target is estimated as:

$$x = [h, v, s, r, \hat{h}, \hat{v}, \hat{s}]^T,$$

where h and v represent the horizontal and vertical pixel location of the centre of the target, while s and r represent the scale and the aspect ratio of the target's bounding box, respectively. Here, $\hat{h}, \hat{v}, \hat{s}$ are for the source.

Data association is assigning new detections to existing targets. Each target's bounding box is estimated by predicting its new location in the current frame. The intersection-over-union (*IOU*) distance between each detection and each forecasted bounding box from the existing targets is used to calculate the assignment cost matrix. The assignment cost matrix is then resolved using the Hungarian technique [87] to produce the fish trajectory as shown in Fig. 7.

4 Method

This section describes our method in detail. Our method is based on three main components: the pseudo-labels generation, the unsupervised learning method to refine the generated pseudo-labels and the DNN for fish tracking and segmentation. Figure 2 shows the algorithm flow diagram for the fish tracking and segmentation framework.

4.1 Datasets

We performed experiments using four publicly available datasets, i.e. Seagrass [23], DeepFish [24], YouTube-VOS [75], and Mediterranean Fish Species [76]. Figure 3 demonstrates a sample image from each dataset.

Seagrass [23] is comprised of annotated footage of *Girella tricuspidata* in two estuary systems in south-east Queensland, Australia. The raw data were obtained using submerged action cameras (HD 1080p). The dataset includes 4280 video frames and 9429 annotations. Each annotation includes segmentation masks that outline the species as a polygon.

DeepFish [24] consists of a large number of videos collected from 20 different habitats in remote coastal marine environments of tropical Australia. The video clips were captured in full HD resolution (1920×1080 pixels) using a digital camera. In total, the number of video frames taken is about 40k.

YouTube-VOS [75] is a video object segmentation dataset that contains 4453 YouTube video clips and 94 object categories. The videos have pixel-level ground truth annotations for every 5th frame (6fps). For a fair comparison, we extracted only the videos that contained `fish`, which include 130 video clips and 4349 video frames in total.

Mediterranean Fish Species [76] consists of a large number of images collected from 20 different Mediterranean fish species. In total, the number of images is about 40k. The dataset was split into two subfolders, training and test sets. The training set contains 34k and the test set contains 6k images. The image resolution ranges between (220×210 pixels) and (1920×1080 pixels). The original images are stored in an RGB file format in subfolders as a class label.

We train our feature extractor on all of the four datasets and evaluate it on the video datasets only, Seagrass [23], DeepFish [24], and YouTube-VOS [75].

4.2 Pseudo-labelling

To train our supervised model, which is explained in Sect. 3.4, we first generate pseudo-labels for the image dataset, Mediterranean Fish Species [76] and the video datasets, Seagrass [23], DeepFish [24], and YouTube-VOS [75].

4.2.1 Image dataset

Since our image dataset [76] is curated from static images of different fish species, our framework discussed in Sect. 3 was not applicable to this dataset. Therefore, we used DeepUSPS [77] as an unsupervised saliency prediction network for a pseudo-labels generation. DeepUSPS is trained on the unlabelled MSRA-B dataset [88] for predicting salient objects. And it is an unsupervised learning method that produces pseudo-labels with high intra-class variations, which is useful for the training of the supervised model.

However, DeepUSPS is only good in pseudo-prediction for a single object in the image that is not disturbed by additional intricate details, which is not ideal for the more challenging video datasets [23, 24, 75].

Fig. 7 Sample fish trajectory results. Zoom in for a better view. See also a short video of fish trajectory results at https://youtu.be/Z5G7YBoL3eM

4.2.2 Video datasets

Unlike our image dataset, our video datasets contain multiple objects in a single frame as well as across multiple frames. Therefore, we adapted our pseudo-label generation framework discussed in Sect. 3 that is capable of predicting multiple salient objects in the same video clip and handling the case of a cluttered background. This pseudo-label generation framework aims to tackle the issue of single-image datasets by generating more pseudo-labels with intra-class variations in image space.

The pseudo-label generation framework consists of three steps:

1) Obtain salient objects by performing background subtraction using adaptive Gaussian thresholding [73], as explained in Sect. 3.1.
2) Enhance the obtained salient object boundaries from the previous step with optical flow using RAFT [74], as explained in Sect. 3.2.
3) Apply cumulative pseudo-label refining via unsupervised historical moving averages (MVA) [77], as explained in Sect. 3.3.

In this way, we can get pseudo-labels for video datasets, Seagrass [23], DeepFish [24], and YouTube-VOS [75], which are used to train the supervised model.

4.3 Model training

Our models were trained with an input resolution of 256×256 pixels. We scale the lowest side of the video frames to 256 and then extract random crops of size 256×256. We sample two video sets, $B = 2$ (of size $T = 5$ frames); therefore, $B \times T = 2 \times 5 = 10$ frames are used per forward pass.

We found that for this problem set, a learning rate of 1×10^{-3} works the best. It took around 300 epochs for all models to train on this problem. Our networks were trained on a Linux host with a single NVidia GeForce RTX 2080 Ti GPU with 11 GB of memory, using Pytorch framework [89]. We used stochastic gradient descent (SGD) optimiser [90] with an initial learning rate of 0.01, which is then divided by 10 at 27*th* and again at 33*th* epoch. We use light augmentation (resizing, greyscale). Following [80, 91], a scale jitter is used, where the shorter image side is randomly sampled from 640 to 800 pixels.

We applied the same hyperparameter configuration for all of the models. However, the optimum model configuration will depend on the application, hence, these results are not intended to represent a complete search of model configurations.

4.4 Inference

During tracking, we extract frames from the input video, forward each frame through the network, and obtain the fish category score from the classification branch. Initially, to filter out predictions with low confidence, we use a threshold of 0.1 and perform convolution on the mask feature using corresponding predicted mask kernels. Then, after applying a per-pixel sigmoid, we binarise the output of the mask branch at the threshold of 0.5. The final step is the matrix NMS, which fits the output mask with the Min-max box.

Our model operates online without any adaptation to the video sequence. On a single NVidia GeForce RTX 2080 Ti GPU, we measured an average speed of 34 frames per second.

5 Experiments

We report experimental results for our model's trained representation on 50% of the DeepFish, Seagrass, YouTube-VOS datasets and the train set of the Mediterranean Fish Species dataset. We then evaluated it in the other 50% of the first three datasets. We provide quantitative and qualitative results that demonstrate our model's generalisation capabilities to a range of different underwater habitats.

5.1 Results

We summarise our main results on Seagrass [23], DeepFish [24] and YouTube-VOS [75] datasets in Table 1. The quantitative results for all datasets were obtained using the COCO dataset [92] evaluation script. The average precision (AP), the average recall (AR), and intersection over union (IoU) were measured for the predicted bounding boxes and segmentation masks in the output images obtained from the trained SoloV2 [80], as explained in Sect. 3.4 in detail.

The average precision (AP) and average recall (AR) metrics provide a comprehensive view of the model's performance. The values $AP^{.50}$ and $AP^{.75}$ indicate that the model has a high precision rate when the intersection over union (IoU) thresholds are 0.5 and 0.75, respectively. This means that the model is able to accurately predict the bounding boxes and segmentation masks for a majority of the objects in the images. The values AP^{M} and AR^{M} show that the model maintains its precision and recall across a range of IoU thresholds, indicating its robustness to variations in object size and shape. The AP^{L} and AR^{L} values specifically measure the model's performance on large objects. These metrics are particularly important in our case, as they reflect the model's ability to accurately segment and track larger fish species.

Table 1 Comparison of *unsupervised* detection and segmentation on Seagrass [23], DeepFish [24] and YouTube-VOS [75] datasets

Dataset	AP^{M}	$AP^{.50}$	$AP^{.75}$	AP^{L}	AR^{M}	AR^{L}
Evaluating detection						
Seagrass [23]	22.1	72.5	13.7	38.2	61.4	61.3
DeepFish [24]	11.7	35.0	05.3	19.3	34.5	57.1
YouTube-VOS [75]	23.6	43.2	18.4	26.9	46.1	57.5
Evaluating segmentation						
Seagrass [23]	12.0	37.6	05.2	20.8	31.2	52.0
DeepFish [24]	31.2	75.0	24.4	43.8	56.6	59.4
YouTube-VOS [75]	15.4	33.0	12.2	19.2	33.8	42.2

The results across different datasets demonstrate that our model is capable of generalising well to unseen videos in other environments. This is a significant achievement, as it suggests that our approach could be applied to a wide range of underwater video data.

To the best of our knowledge, no prior research has reported detection and segmentation evaluation for these datasets. To compare our proposed unsupervised method to a supervised approach, we present the results of SoloV2 [80] in the three data sets in Table 2. This table displays the results of a fully supervised model with the original labels, not our generated pseudo-label.

In both tables, Tables 1 and 2, higher values are better because they indicate that the model's predictions are more accurate. From these tables, we can see that both unsupervised and supervised methods perform well across all three datasets, with some variation in performance depending on the specific dataset and whether detection or segmentation was being evaluated.

For example, in Table 1 (unsupervised method), we can see that the model performs best on the DeepFish dataset in terms of segmentation ($AP^{M} = 31.2$, $AR^{M} = 56.6$), but struggles more with detection on this dataset ($AP^{M} = 11.7$, $AR^{M} = 34.5$).

In contrast, in Table 2 (supervised method), we can see that although performance generally improves across all metrics compared to the unsupervised method, there are still some challenges with certain datasets—for example, detection on the DeepFish dataset ($AP^{M} = 12.2$, $AR^{M} = 41.0$).

Our proposed unsupervised method has yielded close accuracy results to the original supervised SoloV2 [80] in both detection and segmentation experiments, validating the effectiveness of our generative approach. Furthermore, our results suggest that the proposed model is not heavily impacted by different underwater habitats, with almost similar performance for DeepFish [24] and Seagrass [23] datasets. The latter is particularly challenging due to the difficulty of visually detecting the fish. In some cases, the proposed model is not as good as fully supervised approaches. However, the primary objective of this study is the development of an unsupervised method for fish tracking and segmentation. We postulate that our proposed approach offers enhanced stability during training compared to other unsupervised methods without a dedicated pseudo-label generation step. This stability, coupled with the robust performance of our method across diverse datasets, underscores its potential for further refinement and application in this domain.

The qualitative results of our algorithm for the DeepFish [24], Seagrass [23] and YouTube-VOS [75] datasets are illustrated in Figs. 8, 9 and 10, respectively. Additional examples of failure cases are provided in Fig. 11.

Despite the challenges posed by fast movements and complex, crowded backgrounds, which often result in significant distortion, our algorithm produces favourable outcomes for the majority of images. This is particularly noteworthy for non-rigid objects.

For a more dynamic view of our model's predictions, you can watch a short video at this link https://youtu.be/

Table 2 Comparison of *supervised* detection and segmentation on Seagrass [23], DeepFish [24] and YouTube-VOS [75] datasets

Dataset	AP^{M}	$AP^{.50}$	$AP^{.75}$	AP^{L}	AR^{M}	AR^{L}
Evaluating detection						
Seagrass [23]	32.4	82.2	13.0	34.9	68.5	72.4
DeepFish [24]	12.2	41.8	04.3	20.9	41.0	68.0
YouTube-VOS [75]	25.9	56.2	21.6	32.8	54.1	69.9
Evaluating segmentation						
Seagrass [23]	18.0	56.4	07.8	31.2	36.8	68.0
DeepFish [24]	46.8	72.5	36.6	50.7	64.9	72.1
YouTube-VOS [75]	23.1	49.5	18.3	28.8	40.7	53.3

Fig. 8 Sample images from our model results for DeepFish [24]; from left, the original image, the ground truth, the predicted image

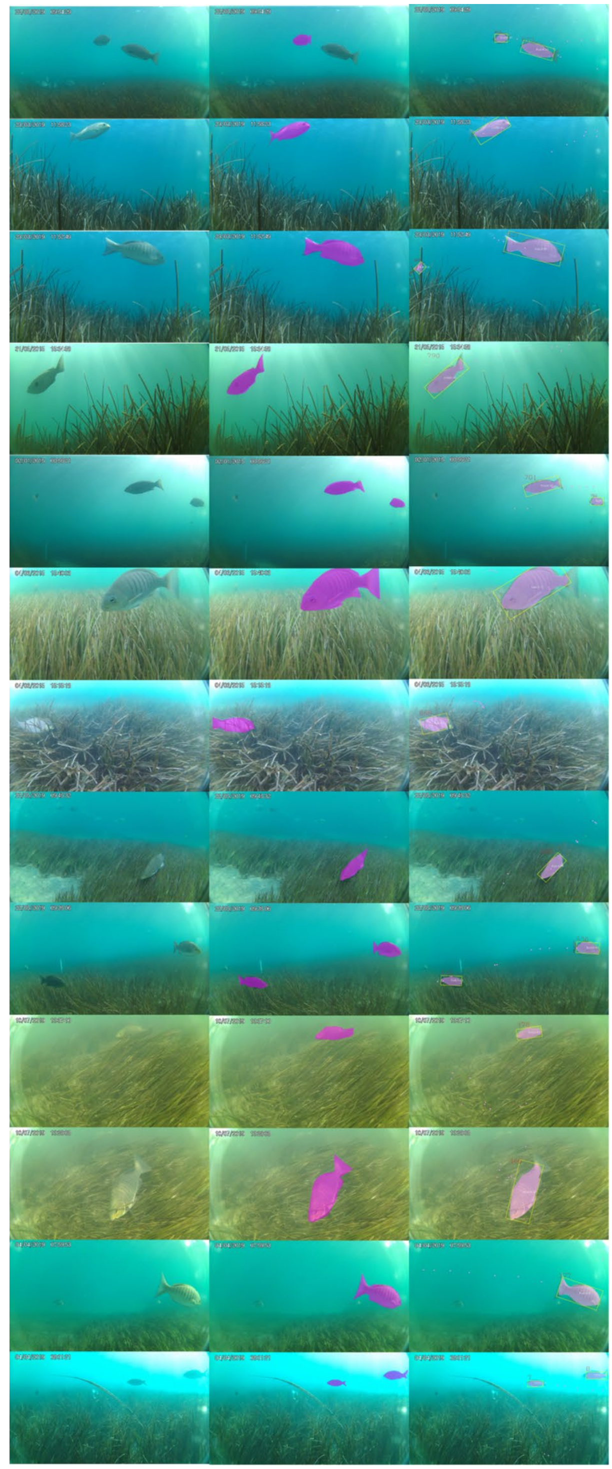

Fig. 9 Sample images from our model results for Seagrass [23]; from left, the original image, the ground truth, the predicted image

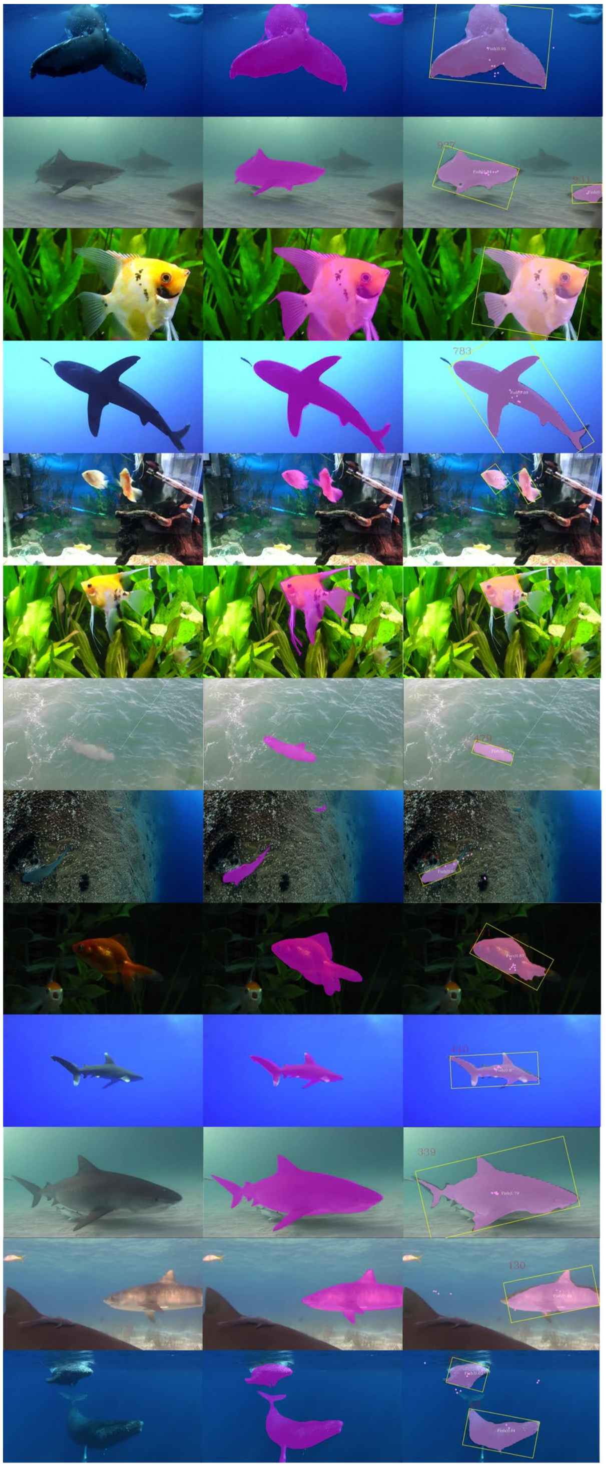

Fig. 10 Sample images from our model results for YouTube-VOS [75]; from left, the original image, the ground truth, the predicted image

Z5G7YBoL3eM. The video showcases the performance of our model in various scenarios, further demonstrating its effectiveness.

5.2 Ablation study

We performed an ablation study to demonstrate the proposed approach's effectiveness in generating pseudo-labels. Specifically, we analysed the contribution of the vital component in the proposed method, the optical flow with background subtraction (Sect. 3.2). In addition, we evaluated the segmentation network training with refined pseudo-labels (Sect. 3.4) for different epochs. The results reported in Table 3 are for unsupervised segmentation based on optical flow without background subtraction as a baseline. And the results reported in Table 4 are for the four epoch trials with the same random seeds, see Sect. 4.3 for the details.

The metrics used in the ablation study are as follows:

1) AP^M (Average precision for medium objects): This is the average precision for medium-sized objects. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The higher the AP^M, the better the model is at predicting medium-sized objects correctly.
2) $AP^{.50}$ and $AP^{.75}$: These are the average precision values at different intersection over union (IoU) thresholds. IoU is a measure of overlap between two bounding boxes. $AP^{.50}$ is the average precision when IoU is 0.50, and $AP^{.75}$ is the average precision when IoU is 0.75. Higher values indicate better precision at these IoU thresholds.
3) AP^L (Average precision for large objects): This is the average precision for large-sized objects. Like AP^M, a higher AP^L indicates that the model is better at predicting large-sized objects correctly.
4) AR^M (Average recall for medium objects): This is the average recall for medium-sized objects. Recall is the ratio of correctly predicted positive observations to all actual positives. The higher the AR^M, the better the model is at identifying all actual medium-sized objects.
5) AR^L (Average recall for large objects): This is the average recall for large-sized objects. Like AR^M, a higher AR^L indicates that the model is better at identifying all actual large-sized objects.

In all these metrics, higher values are better because they indicate that the model's predictions are more accurate.

It is apparent from the results that the segmentation accuracy of our proposed method has improved significantly when compared to that of the baseline method. We also note that the accuracy of the models also depends on the number of epochs used in the training. We observe from the results shown in Table 4 that the segmentation accuracy decreases after 100 epochs. The reason for this is the over-fitting of the network to the noisy pseudo-labels. While the training losses for both the baseline and our model decreased, the

Fig. 11 Sample of the failure cases of our model. From the left, the original image, the ground truth mask overlay, and the predicted image. Images show instances where the model struggles with heavy occlusion, variability in fish size and shape, segmentation of foreground items, and influence of training videos. These scenarios highlight the limitations of our current approach and provide directions for future improvements

Table 3 Comparison of *unsupervised* segmentation based on optical flow *without* background subtraction

Dataset	AP^{M}	$AP^{.50}$	$AP^{.75}$	AP^{L}	AR^{M}	AR^{L}
Evaluating segmentation						
Seagrass [23]	05.0	23.8	03.1	14.7	19.7	29.5
DeepFish [24]	15.3	44.8	13.6	33.5	42.7	37.4
YouTube-VOS [75]	07.2	23.8	07.4	11.9	26.1	33.0

Table 4 Comparison of *unsupervised* segmentation for different epochs: 50, 100, 150, 300

Dataset	AP^{M}	$AP^{.50}$	$AP^{.75}$	AP^{L}	AR^{M}	AR^{L}
50 epochs						
Seagrass [23]	12.4	33.6	07.2	20.4	28.4	47.2
DeepFish [24]	32.0	68.6	30.8	34.8	53.6	56.2
YouTube-VOS [75]	15.8	34.0	13.8	19.8	33.8	42.2
100 epochs						
Seagrass [23]	12.0	37.6	05.2	20.8	31.2	52.0
DeepFish [24]	31.2	75.0	24.4	43.8	56.6	59.4
YouTube-VOS [75]	15.4	33.0	12.2	19.2	33.8	42.2
150 epochs						
Seagrass [23]	12.0	36.0	04.8	20.4	30.0	48.8
DeepFish [24]	30.4	69.8	23.2	32.4	54.2	56.8
YouTube-VOS [75]	15.2	34.0	14.0	20.2	32.8	41.0
300 epochs						
Seagrass [23]	10.8	33.6	04.0	18.8	28.0	46.4
DeepFish [24]	29.8	70.0	22.4	31.8	53.0	55.6
YouTube-VOS [75]	15.2	33.8	14.4	23.0	32.0	40.0

segmentation accuracy for our model was still greater than that for the baseline.

5.3 Failure cases

While our model has shown promising results, there are specific scenarios where it fails to perform optimally.

- Occlusion: Our model's performance degrades when several fish are heavily occluded. While it can estimate the fish mask in some parts as long as they are part of the animal body, it struggles when the occlusion is severe, see Fig. 11.
- Variability in fish size and shape: The large variability in the size and shape of fish presents a challenge for our model. It can identify a certain shape of fish, but determining the number of fish in an image remains a difficult task.
- Segmentation of foreground Items: Given a set of unlabelled video collections, our model is only capable of segmenting foreground items and cannot distinguish between distinct object instances or semantic classes. Occasionally, the whole object or parts of the object may not be segmented out.
- Influence of training videos: Our model's performance is highly influenced by the characteristics of training videos, the coverage of object categories, and the motion of both the camera and the objects. This is similar to other data-driven learning techniques.

These failure cases provide valuable insights for future improvements to our model.

6 Discussion

Fish segmentation and tracking are notoriously difficult tasks, especially for small fish in video data where the background, lighting conditions and fish shape can vary significantly. In particular, for real data, the quality of ground truth labels varies from video to video, since it is difficult to annotate the animal's entire path. Therefore, our model aims to generate a pseudo-ground truth by leveraging temporal consistency between frames and improving its quality based on self-supervised learning. The key to our proposed model is to leverage the intrinsic temporal consistency between consecutive frames by using the optical flow and background subtraction method to improve the generated labels. This is especially important when the fish is moving quickly and not in the same location in consecutive frames, as is the case in natural data. Tracking fish in video data is also challenging because their motion is very irregular and small fish may not be visible throughout the entire dataset. The other problem is that segmentation and tracking are time-consuming tasks, especially when dealing with large datasets.

Our model outperforms the baseline method (the optical flow without background subtraction) with higher *AP* values in most of the cases. Our approach can utilise temporal consistency to produce consistent labels. In the case of the DeepFish dataset [24], we observed that our proposed unsupervised model results in higher accuracy compared to the Seagrass dataset [23]. This is mainly due to the more challenging videos in the Seagress data set [23] compared to the DeepFish video data [24]. Furthermore, we show that for different video datasets, our model shows similar accuracy. Therefore, we can expect that the accuracy would be similar when tested under the same conditions but in new underwater video datasets.

In addition, segmentation accuracy does not degrade after training with supervised training, and training converges in only a few epochs, as shown in Table 4. In our experiments, we found that segmentation quality has a significant impact on tracking performance. This is because the quality of the produced object bounding box has a high impact on tracking performance. Even in this case, we still achieved decent results.

We also analysed the robustness of our proposed model with respect to the environmental conditions. We observed degradation of the model's performance when several fish were heavily occluded, like in Fig. 11. However, our proposed model is still able to estimate the fish mask in some parts as long as they are part of the animal body. One of the main challenges in this task is the large variability in the size and shape of fish, as well as the variation in the shape of the fish's body. Although it is possible to identify a certain shape of fish, it is not always possible to determine the number of fish in the image.

Given a set of unlabelled video collections, the main limitation of our study is that it is only capable of segmenting foreground items and cannot distinguish between distinct object instances or semantic classes. Occasionally, the entire object or parts of the object may not be segmented out. The performance of our model is highly influenced by the characteristics of training videos, the coverage of object categories, and the motion of both the camera and the objects, similar to other data-driven learning techniques. Our results are based on a few assumptions. One is that a small subset of semantically similar objects (e.g. all fish) exists in the scene, and these objects are likely to share the same motion feature or to be semantically similar. These assumptions are reasonable if the objects are within a certain size range, they all belong to the same class, and most of them share similar colours, shapes, and sizes. Another limitation of our approach is that we used a relatively large number of videos with a relatively small number of object categories (for instance compared to ImageNet). This allows our model to segment objects of all shapes and colours with only a handful of training examples.

One other limitation of our current framework is that in some cases, it is unable to detect all the objects that appear in the video. In future work, we intend to study how to develop a detection-based model that is able to detect all objects appearing in a given scene. Therefore, in the next step, we should look for a more robust and generic objectness model that is able to generalise across a variety of object categories and a variety of background types. Further work could be conducted on more fine-grained object segmentation, especially with new video datasets.

7 Conclusion

In this study, we introduced an innovative unsupervised methodology for the segmentation and tracking of fish in uncontrolled video environments. Our approach leverages a pseudo-label generation method that combines optical flow with background subtraction, followed by an unsupervised refinement network. This method has proven to yield accurate segmentation results when used to train a supervised deep neural network (DNN) for segmentation. Furthermore, our approach has shown its efficacy in tracking applications.

Our methodology was rigorously tested on three challenging datasets, with the results indicating its robustness and adaptability across different scenarios. This suggests that our approach could serve as a valuable tool for researchers and conservationists working with video data in aquatic environments.

Future research directions include extending our methodology to encompass other aquatic species. This extension, however, would necessitate further adaptations to account for the unique movement patterns and physical characteristics of these species. Another promising avenue for future research is the application of our model to autonomous driving systems for tracking-by-detection. Although this application would present additional challenges, such as dealing with faster-moving objects and more complex backgrounds, we believe the core principles of our approach remain applicable.

In conclusion, our study contributes to a significant advancement in the field of video processing for fish behaviour analysis. The proposed methodology not only enhances our ability to study fish behaviour, but also has potential implications for conservation efforts by providing more accurate data on fish populations and movements. Despite the limitations and challenges, we believe that our work lays a solid foundation for future research in this area.

Acknowledgements This research is supported by the Australian Research Training Program (RTP) Scholarship and Food Agility HDR Top-Up Scholarship. D. Jerry and M. Rahimi Azghadi acknowledge the Australian Research Council through their Industrial Transformation Research Hub program.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. No funding was received to assist with the preparation of this manuscript. The authors have no relevant financial or non-financial interests to disclose.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

Data availability Data sets generated and analysed during the current study are publicly available.

References

1. LopezMarcano S, Jinks E, Buelow CA, Brown CJ, Wang D, Kusy B, Ditria E, Connolly RM (2021) Automatic detection of fish and tracking of movement for ecology. Ecol Evol 11(12):8254–8263. https://doi.org/10.1002/ece3.7656
2. Zou L, Zhao M, Cao F, Zan S, Cheng X, Liu X (2021) Fish tracking based on feature fusion and scale adaptation in a real-world underwater environment. Mar Technol Soc J 55(2):45–53. https://doi.org/10.4031/MTSJ.55.2.12
3. Gatti P, Fisher JAD, Cyr F, Galbraith PS, Robert D, Le Bris A (2021) A review and tests of validation and sensitivity of geolocation models for marine fish tracking. Fish Fish 22(5):1041–1066. https://doi.org/10.1111/faf.12568
4. Wageeh Y, Mohamed HE-D, Fadl A, Anas O, ElMasry N, Nabil A, Atia A (2021) YOLO fish detection with Euclidean tracking in fish farms. J Ambient Intell Hum Comput 12(1):5–12. https://doi.org/10.1007/s12652-020-02847-6
5. Saleh A, Sheaves M, Rahimi AM (2022) Computer vision and deep learning for fish classification in underwater habitats: a survey. Fish Fish. https://doi.org/10.1111/faf.12666
6. Zhang Z, Wei Z, Zhang S, Dai Z, Zhu S (2023) Uvosam: a mask-free paradigm for unsupervised video object segmentation via segment anything model. arXiv preprint arXiv:2305.12659
7. Guida VG, Valentine PC, Gallea LB (2013) Semidiurnal temperature changes caused by tidal front movements in the warm season in seabed habitats on the Georges Bank Northern Margin and their ecological implications. PLoS ONE 8(2):e55273. https://doi.org/10.1371/journal.pone.0055273
8. Sundin J, Morgan R, Finnøen MH, Dey A, Sarkar K, Jutfelt F (2019) On the Observation of Wild Zebrafish (Danio rerio) in India. Zebrafish 16(6):546–553. https://doi.org/10.1089/zeb.2019.1778. (**12 . [Online]. Available:**)
9. Olsen EM, Heupel MR, Simpfendorfer CA, Moland E (2012) Harvest selection on Atlantic cod behavioral traits: implications for spatial management. Ecol Evol 2(7):1549–1562. https://doi.org/10.1002/ece3.244. (**7 . [Online]. Available:**)
10. Wang NXR, Cullis-Suzuki S, Branzan Albu A (2015) Automated analysis of wild fish behavior in a natural habitat. In: Proceedings of the 2nd international workshop on environmental multimedia retrieval, New York, NY, USA. ACM, vol. 6, pp 21–26. https://doi.org/10.1145/2764873.2764875
11. Dutta A, Perez-Campanero N, Taylor GK, Zisserman A, Newport C (2023) A robust and flexible deep-learning workflow for animal tracking. bioRxiv, pp 2023-04
12. Javed S, Danelljan M, Khan FS, Khan MH, Felsberg M, Matas J (2022) Visual object tracking with discriminative filters and siamese networks: a survey and outlook. IEEE Trans Pattern Anal Mach Intell 45(5):6552–6574
13. Saleh A, Laradji IH, Konovalov DA, Bradley M, Vazquez D, Sheaves M (2020) A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. Sci Rep 10(1):14671
14. Konovalov DA, Saleh A, Efremova DB, Domingos JA, Jerry DR (2019) Automatic weight estimation of harvested fish from images. In: 2019 Digital image computing: techniques and applications, DICTA 2019. Institute of Electrical and Electronics Engineers Inc., 12
15. Laradji IH, Saleh A, Rodriguez P, Nowrouzezahrai D, Azghadi MR, Vazquez D (2021) Weakly supervised underwater fish segmentation using affinity LCFCN. Sci Rep 11(1):17379
16. Konovalov DA, Saleh A, Domingos JA, White RD, Jerry DR (2018) Estimating mass of harvested Asian Seabass Lates calcarifer from Images. World J Eng Technol 6(03):15
17. Konovalov DA, Saleh A, Bradley M, Sankupellay M, Marini S, Sheaves M (2019) Underwater fish detection with weak multi-domain supervision. In: 2019 International joint conference on neural networks (IJCNN), vol. 2019-July. IEEE, 7, pp 1–8. https://ieeexplore.ieee.org/document/8851907/
18. Jahanbakht M, Rahimi Azghadi M, Waltham NJ (2023) Semi-supervised and weakly-supervised deep neural networks and dataset for fish detection in turbid underwater videos. Ecol Inf 78:102303
19. Wang SH, Zhao J, Liu X, Qian Z-M, Liu Y, Chen YQ (2017) 3D tracking swimming fish school with learned kinematic model using LSTM network. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 3, pp 1068–1072. http://ieeexplore.ieee.org/document/7952320/
20. Villon S, Mouillot D, Chaumont M, Darling ES, Subsol G, Claverie T, Villéger S (2018) A deep learning method for accurate and fast identification of coral reef fishes in underwater images. Ecol Inf
21. Li Z, Li W, Li F, Yuan M (2021) A review of computer vision technologies for fish tracking. IEEE, 10. arXiv: http://arxiv.org/abs/2110.02551
22. Cao J, Wu H, Kitani K (2022) Track targets by dense spatio-temporal position encoding. arXiv preprint arXiv:2210.09455
23. Ditria EM, Connolly RM, Jinks EL, Lopez-Marcano S (2021) Annotated video footage for automated identification and counting of fish in unconstrained seagrass habitats. Front Mar Sci 8:3. https://doi.org/10.3389/fmars.2021.629485/full

24. Saleh A, Laradji IH, Konovalov DA, Bradley M, Vazquez D, Sheaves M (2020) A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. Sci Rep 10(1):14671
25. Proença PF, Spieler P, Hewitt RA, Delaune J (2023) Trade: object tracking with 3D trajectory and ground depth estimates for UAVs. In: 2023 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3325–3331
26. Jahanbakht M, Xiang W, Waltham NJ, Videos MR (2022) Distributed deep learning and energy-efficient real-time image processing at the edge for fish segmentation in underwater. IEEE Access 10:117796–117807
27. Zhang P, Yu H, Li H, Zhang X, Wei S, Tu W, Yang Z, Wu J, Lin Y Msgnet: multi-source guidance network for fish segmentation in underwater videos. Front Mar Sci 10:1256594
28. Yao R, Lin G, Xia S, Zhao J, Zhou Y (2020) Video object segmentation and tracking. ACM Trans Intell Syst Technol 11(4):1–47. https://doi.org/10.1145/3391743
29. Khoreva A, Benenson R, Ilg E, Brox T, Schiele B (2019) Lucid data dreaming for video object segmentation. Int J Comput Vis 127(9):1175–1197. https://doi.org/10.1007/s11263-019-01164-6. (**9 . [Online]. Available:**)
30. Maninis K-K, Caelles S, Chen Y, Pont-Tuset J, Leal-Taixe L, Cremers D, Van Gool L (2019) Video object segmentation without temporal information. IEEE Trans Pattern Anal Mach Intell 41(6):1515–1530
31. Bouwmans T, Javed S, Sultana M, Jung SK (2019) Deep neural network concepts for background subtraction: a systematic review and comparative evaluation. Neural Netw 117:8–66
32. Kalsotra R, Arora S (2019) A comprehensive survey of video datasets for background subtraction. IEEE Access 7:59143–59171
33. Garcia-Garcia B, Bouwmans T, Rosales Silva AJ (2020) Background subtraction in real applications: challenges, current models and future directions. Comput Sci Rev 35:100202
34. Pan H, Zhu G, Peng C, Xiao Q (2021) Background subtraction for night videos. PeerJ Comput Sci 7:e592
35. Maddalena L, Petrosino A (2018) Background subtraction for moving object detection in RGBD data: a survey. J Imaging 4(5):71
36. Lu S, Luo Z, Gao F, Liu M, Chang K, Piao C (2021) A fast and robust lane detection method based on semantic segmentation and optical flow estimation. Sensors 21(2):400
37. Anthwal S, Ganotra D (2019) An overview of optical flow-based approaches for motion segmentation. Imaging Sci J 67(5):284–294. https://doi.org/10.1080/13682199.2019.1641316. (**7 . [Online]. Available:**)
38. Cheng J, Tsai Y-H, Wang S, Yang M-H (2017)SegFlow: joint learning for video object segmentation and optical flow. In: 2017 IEEE international conference on computer vision (ICCV), vol. 2017-October. IEEE, 10, pp 686–695. http://ieeexplore.ieee.org/document/8237343/
39. Ding M, Wang Z, Zhou B, Shi J, Lu Z Luo P (2020) Every frame counts: joint learning of video segmentation and optical flow. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07, pp 10713–10720
40. Garcia-Dopico A, Pedraza JL, Nieto M, Pérez A, Rodríguez S, Osendi L (2014) Locating moving objects in car-driving sequences. EURASIP J Image Video Process 1:24,12. https://doi.org/10.1186/1687-5281-2014-24. (. **[Online]. Available:**)
41. Chraa Mesbahi S, Mahraz MA, Riffi J, Tairi H (2018) Head gesture recognition using optical flow based background subtraction. Lecture Notes Netw Syst 37:200–211. https://doi.org/10.1007/978-3-319-74500-8_18. (**[Online]. Available:**)
42. Kushwaha A, Khare A, Prakash O, Khare M (2020) Dense optical flow based background subtraction technique for object segmentation in moving camera environment. IET Image Process 14(14):3393–3404. https://doi.org/10.1049/iet-ipr.2019.0960. (**12 [Online]. Available:**)
43. Sun D, Liu C, Pfister H (2014) Local layering for joint motion estimation and occlusion detection. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
44. Chen Z, Jin H, Lin Z, Cohen S, Wu Y (2013) Large displacement optical flow from nearest neighbor fields. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
45. Brox T, Malik J (2011) Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Trans Pattern Anal Mach Intell 33(3):500–513
46. Guan H, Xue XY, An ZY (2016) Advances on application of deep learning for video object tracking
47. Ciaparrone G, Luque Sánchez F, Tabik S, Troiano L, Tagliaferri R, Herrera F (2020) Deep learning in video multi-object tracking: a survey. Neurocomputing 381:61–88
48. Gomez-Nieto R, Ruiz-Munoz JF, Beron J, Franco CAA, Benitez-Restrepo HD, Bovik AC (2022) Quality aware features for performance prediction and time reduction in video object tracking. IEEE Access 10:13290–13310
49. Qiu J, Wang L, Hu YH, Wang Y (2020) Two motion models for improving video object tracking performance. Comput Vis Image Understand 195:102951
50. Kang X, Song B, Sun F (2019) A deep similarity metric method based on incomplete data for traffic anomaly detection in IoT. Appl Sci 9(1):135
51. Dadgar A, Baleghi Y, Ezoji M (2021) Improved object matching in multi-objects tracking based on zernike moments and combination of multiple similarity metrics. Int J Eng 34(6):6
52. Bag S, Kumar SK, Tiwari MK (2019) An efficient recommendation generation using relevant Jaccard similarity. Inf Sci 483:53–64
53. Zhu B, Jiang Y, Gu M, Deng Y (2021) A GPU acceleration framework for motif and discord based pattern mining. IEEE Trans Parallel Distrib Syst 32(8):1987–2004
54. Zhu J, Wang Z, Wang S, Chen S (2020) Moving object detection based on background compensation and deep learning. Symmetry 12(12):1965
55. Chapel M-N, Bouwmans T (2020) Moving objects detection with a moving camera: a comprehensive review. Comput Sci Rev 38:100310
56. Yang K, He Z, Pei W, Zhou Z, Li X, Yuan D, Zhang H (2022) Siamcorners: Siamese corner networks for visual tracking. IEEE Trans Multimedia 24:1956–1967
57. Zhu H, Wei H, Li B, Yuan X, Kehtarnavaz N (2020) A review of video object detection: datasets, metrics and methods. Appl Sci 10(21):7834
58. Jiao L, Zhang L, Liu F, Yang S, Li L, Feng Z, Qu R (2019) A survey of deep learning-based object detection. IEEE Access 7:128837–128868
59. Zhao Z-Q, Zheng P, Xu S-T, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232
60. Jiang T, Gradus JL, Rosellini AJ (2020) Supervised machine learning: a brief primer. Behav Ther
61. Wang X, Lin X, Dang X (2020) Supervised learning in spiking neural networks: a review of algorithms and evaluations. Neural Netw
62. Zhou Z, Zhang R, Yin D (2020) A strong feature representation for siamese network tracker. Multimedia Tools Appl 79(35–36):25873–25887. https://doi.org/10.1007/s11042-020-09164-2
63. Peng J, Li J, Shang X (2020) A learning-based method for drug-target interaction prediction based on feature representation

learning and deep neural network. BMC Bioinf 21(S13):394, 9. https://doi.org/10.1186/s12859-020-03677-1. (**[Online]. Available:**)
64. Xie Y, Du Z, Li J, Jing M, Chen E, Lu K (2020) Joint metric and feature representation learning for unsupervised domain adaptation. Knowl Based Syst 192:105222
65. Garcia R, Prados R, Quintana J, Tempelaar A, Gracias N, Rosen S, Vågstøl H, Løvall K, (2020) Automatic segmentation of fish using deep learning with application to fish size measurement. ICES J Mar Sci
66. Chang CC, Wang YP, Cheng SC (2021) Fish segmentation in sonar images by mask R-CNN on feature maps of conditional random fields. Sensors
67. Alshdaifat NFF, Talib AZ, Osman MA (2020) Improved deep learning framework for fish segmentation in underwater videos. Eco Inform 59:101121
68. Jabri AA, Owens A, Efros AA (2020) Space-time correspondence as a contrastive random walk. In: Advances in neural information processing systems
69. Araslanov N, Schaub-Meyer S, Roth S (2021) Dense unsupervised learning for video segmentation. IEEE. arXiv: org/abs/2111.06265v1
70. Wang N, Zhou W, Li H (2020) Contrastive transformation for self-supervised correspondence learning. IEEE. arXiv: org/abs/2012.05057v1
71. Liu R, Wu Z, Yu SX, Lin S (2021) The emergence of objectness: learning zero-shot segmentation from videos. Adv Neural Inf Process Syst 16:13137–13152
72. Saleh A, Sheaves M, Jerry D, Azghadi MR (2022) Transformer-based self-supervised fish segmentation in underwater videos. IEEE. http://arxiv.org/abs/2206.05390
73. Golilarz NA, Demirel H, Gao H (2019) Adaptive generalized Gaussian distribution oriented thresholding function for image de-noising. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2019.0100202
74. Teed Z, Deng J (2021) RAFT: recurrent all-pairs field transforms for optical flow (extended abstract). In: Proceedings of the thirtieth international joint conference on artificial intelligence, California: international joint conferences on artificial intelligence organization, 8, pp 4839–4843. https://www.ijcai.org/proceedings/2021/662
75. Xu N, Yang L, Fan Y, Yang J, Yue D, Liang Y, Price B, Cohen S, Huang T (2018) YouTube-VOS: sequence-to-sequence video object segmentation. In: Lecture Notes in Computer Science (including subseries Lecture notes in artificial intelligence and lecture notes in bioinformatics)
76. Georgiou G (2021) Mediterranean fish species. https://www.kaggle.com/datasets/giannisgeorgiou/fish-species
77. Nguyen DT, Dax M, Mummadi CK, Ngo TPN, Nguyen THP, Lou Z, Brox T (2019) DeepUSPS: deep robust unsupervised saliency prediction with self-supervision. In: Advances in neural information processing systems, vol. 32
78. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. PAMI 40(4):834–848
79. Krähenbühl P, Koltun V (2011) Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Advances in neural information processing systems, pp 109–117
80. Wang X, Zhang R, Kong T, Li L, Shen C (2020) SOLOv2: dynamic and fast instance segmentation. In: Advances in neural information processing systems, vol. 2020-December
81. Wang X, Kong T, Shen C, Jiang Y, Li L (2020) SOLO: segmenting objects by locations. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12363:649–665. https://doi.org/10.1007/978-3-030-58523-5_38
82. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651 (**4**)
83. Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In: 18th International conference on pattern recognition (ICPR'06) 3:850–855
84. OpenCv (2014) OpenCV Library. OpenCV Website. https://opencv.org/about.html
85. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In: Proceedings of international conference on image processing, ICIP, vol. 2016-August, pp 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003
86. Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng 82(1):35–45
87. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Res Logist Q 2:1–2
88. Liu T, Yuan Z, Sun J, Wang J, Zheng N, Tang X, Shum H-Y (2011) Learning to detect a salient object. IEEE Trans Pattern Anal Mach Intell 33(2):353–367
89. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) PyTorch: an imperative style, high-performance deep learning library. In: Advances in neural information processing systems
90. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
91. Chen X, Girshick R, He K, Dollar P (2019) TensorMask: a foundation for dense object segmentation. In: 2019 IEEE/CVF international conference on computer vision (ICCV), vol. 2019-October. IEEE, 10, 2061–2069
92. Lin TY, Maire M, Belongie, S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)