ORIGINAL PAPER

Overcoming annotation bottlenecks in underwater fish segmentation: a robust self-supervised learning approach

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

Abstract

Accurate fish segmentation in underwater videos is challenging due to low visibility, variable lighting, and dynamic backgrounds, making fully-supervised methods that require manual annotation impractical for many applications. This paper introduces a novel self-supervised learning approach for fish segmentation using Deep Learning. Our model, trained without manual annotation, learns robust and generalizable representations by aligning features across augmented views and enforcing spatial-temporal consistency. We demonstrate its effectiveness on three challenging underwater video datasets: DeepFish, Seagrass, and YouTube-VOS, surpassing existing self-supervised methods and achieving segmentation accuracy comparable to fully-supervised methods without the need for costly annotations. Trained on DeepFish, our model exhibits strong generalization, achieving high segmentation accuracy on the unseen Seagrass and YouTube-VOS datasets. Furthermore, our model is computationally efficient due to its parallel processing and efficient anchor sampling technique, making it suitable for real-time applications and potential deployment on edge devices. We present quantitative results using Jaccard Index and Dice coefficient, as well as qualitative comparisons, showcasing the accuracy, robustness, and efficiency of our approach for advancing underwater video analysis.

1 Introduction

The health and sustainability of aquatic ecosystems and aquaculture operations depend on effective monitoring and management of fish populations. Fish segmentation, the process of automatically identifying and outlining fish in images or videos, is crucial for estimating fish abundance, tracking movement patterns, analyzing behavior, and assessing fish stock health. However, accurately segmenting fish in underwater environments is challenging due to low visibility, variable lighting, dynamic backgrounds, and the diverse appearances and movements of fish.

Traditional computer vision approaches struggle with these complexities, and fully supervised deep learning methods, while effective, require extensive manually annotated datasets. Creating such datasets is time-consuming, expensive, and prone to biases, limiting scalability and generalizability. For instance, segmenting a single fish in an image can take approximately two minutes, leading to significant annotation costs for even moderately sized datasets. Unsupervised and self-supervised learning methods have emerged as promising alternatives, eliminating the need for costly annotations by leveraging inherent structures and relationships within unlabeled data [1].

Contrastive learning, a prominent self-supervised technique, has shown success in visual representation learning by distinguishing between positive (similar) and negative (dissimilar) pairs of augmented views [2]. Similarly, correspondence learning leverages temporal coherence and spatial relationships in video data to propagate information and segment objects effectively over time [3, 4]. Vision Transformers (ViTs) have also gained traction in computer vision tasks, capturing long-range dependencies and global context

✉ Alzayat Saleh
alzayat.saleh@my.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
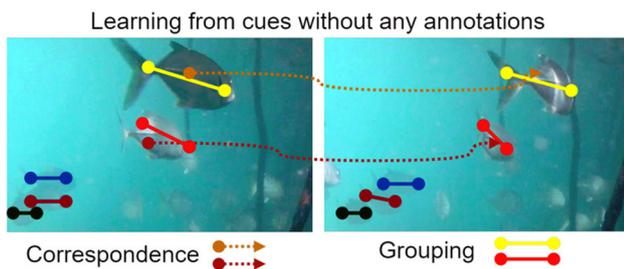
Learning from cues without any annotations

Fig. 1 The natural visual artefact dynamics provide important cues about the composition of scenes and how they change

through self-attention mechanisms [5]. These advancements provide a strong foundation for addressing the challenges of underwater fish segmentation.

To address these challenges, this paper introduces a novel self-supervised learning approach for robust and efficient fish segmentation in underwater videos. Our method leverages Transformer networks, which excel at capturing long-range dependencies and global context in computer vision tasks. Unlike supervised methods, our approach eliminates the need for manual annotation by learning powerful representations directly from unlabeled videos. This is achieved through feature alignment across augmented views and enforcing spatial-temporal consistency, enabling the model to focus on underlying fish features and maintain accuracy in dynamic scenes, see Fig. 1.

Our approach offers several key advantages:

- **Eliminates annotation bottleneck**: Removes the need for manual annotation, reducing time and cost for data preparation.
- **Enhanced generalizability**: Learns robust representations from diverse unlabeled data, improving performance on unseen datasets.
- **Computational efficiency**: Employs an efficient architecture suitable for real-time applications and resource-constrained devices.

We demonstrate the effectiveness of our approach on three challenging datasets: DeepFish [6], Seagrass [7], and YouTube-VOS [8]. Trained solely on DeepFish, our model generalizes well to Seagrass and YouTube-VOS, outperforming existing self-supervised methods and rivaling fully-supervised approaches without costly annotations.

2 Method

This section outlines the architecture and training procedure of our proposed model, which comprises two branches: a main branch and a regularizing branch. Both branches share the same architecture and weights to ensure consistent fea-

ture representation learning. The model processes unlabeled video sequences as input, as illustrated in Fig. 2.

2.1 Model architecture

- **Input**: The model receives a batch of unlabeled video sequences.
- **View augmentation**: Two distinct views of each input frame batch are generated. The main branch processes the original frames, while the regularizing branch operates on augmented versions (random cropping and flipping) to promote invariance to transformations.
- **Feature encoder (CoaT Transformer)**: The CoaT Transformer [9] is used as the feature encoder backbone. It captures both local and global image features through its co-scale mechanism, enabling multi-scale representations for improved segmentation accuracy.
- **Feature embedding (MLP)**: The CoaT encoder output is passed through a multilayer perceptron (MLP) to produce compact feature embeddings. The MLP consists of two `Conv2d` layers with Layer Normalization and ReLU activation, reducing the feature dimensionality from 512 to 128.

2.2 Co-scale conv-attention

The CoaT Transformer consists of two key submodules: the **Conv-Attentional Image Transformer (CAIT)** and the **Co-Scale Feature Attention Network (CFAN)**. These submodules work together to capture both local and global features, enabling the model to handle the complexities of underwater fish segmentation.

- **Conv-attentional image transformer (CAIT)**: The CAIT module combines the strengths of convolutional operations and self-attention mechanisms. It uses a **spatial transformer network** to process the input image and generate a **co-scale feature pyramid**. This pyramid captures features at multiple resolutions, allowing the model to focus on both fine-grained details (e.g., fish scales, fins) and broader contextual information (e.g., fish shape, background). The CAIT also employs **convolutional relative position embeddings** in its factorized attention mechanism, which helps the model understand spatial relationships between different parts of the image. This is particularly useful in underwater scenes, where fish may appear at various scales and orientations.
- **Co-scale feature attention network (CFAN)**: The CFAN module dynamically selects informative regions of the image, enabling the model to focus on relevant areas while ignoring less important ones. It introduces **fine-to-coarse**, **coarse-to-fine**, and **cross-scale information** into
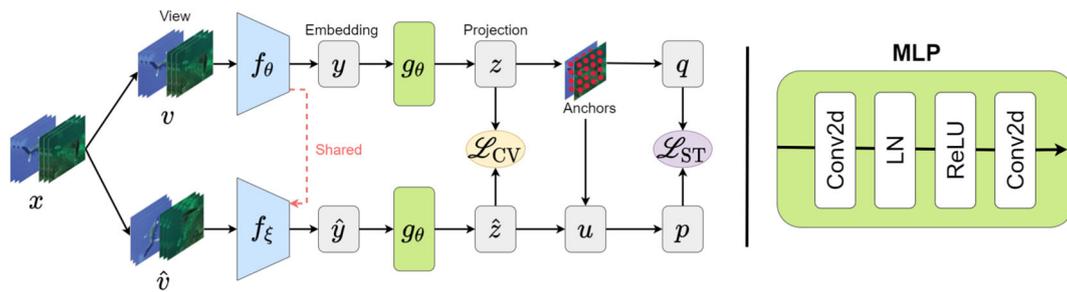
Fig. 2 Our proposed framework consists of a single feature extractor that processes video sequences. Given a batch of unlabeled video sequences x, two batches of different views v and $\hat{v}$ are produced and are then encoded into embeddings y and $\hat{y}$ through the main branch f_θ and the second regularising branch f_ξ, respectively. The embeddings are fed to a multilayer perceptron (MLP) g_θ to produce the projections z and $\hat{z}$ to compute the cross-view consistency loss $\mathcal{L}_{CV}$. The self-training loss $\mathcal{L}_{ST}$ learns space-time embeddings between the anchors q and pseudo labels p (arg max of u, affinities of $\hat{z}$ $w.r.t.$ anchors.). The two branches are identical in architecture with shared weights

Fig. 3 Schematic graph of the serial block in CoaT transformer [9]. Input feature maps are first down-sampled by a patch embedding layer and then flatten the reduced feature maps into a sequence of image tokens. Multiple conv-attention and feed-forward layers process the tokenized features, along with a class token (a vector to achieve image classification)
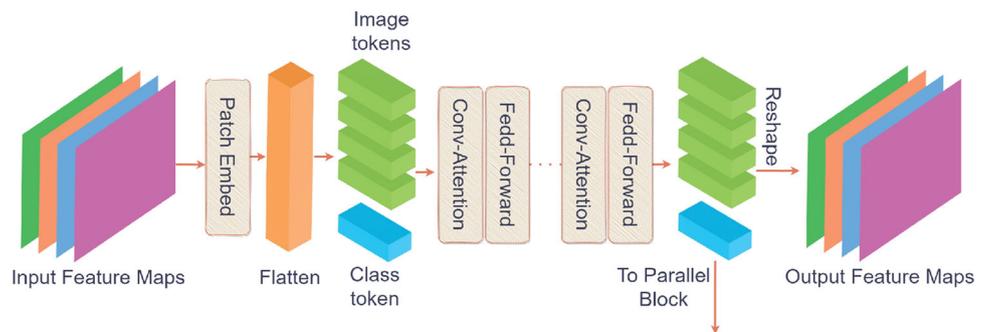
the image transformer through a combination of serial and parallel blocks, see Fig. 3. These blocks process the image at multiple resolutions, ensuring that the model can capture both high-level semantic information and low-level texture details. For example, given an input image $I \in \mathbb{R}^{H \times W \times C}$, the serial block downsamples the image features into four resolutions: $F_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C_1}$, $F_2 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_2}$, $F_3 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_3}$, $F_4 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_4}$. This multi-scale approach ensures that the model can effectively segment fish in diverse underwater environments, even when they are partially occluded or appear in dynamic backgrounds.

By combining these submodules, the CoaT Transformer captures **long-range dependencies** and **global context** through its self-attention mechanisms, while also preserving fine-grained local details through convolutional operations. This makes it particularly well-suited for underwater fish segmentation, where the model must handle variable lighting, low visibility, and complex backgrounds.

2.3 Multilayer perceptron (MLP)

The MLP processes the feature encoder output to generate feature embeddings, reducing the dimensionality from 512 to 128. It consists of two `Conv2d` layers, with the first layer followed by Layer Normalization and ReLU activation. The PyTorch-like implementation of the MLP is shown in Algorithm 1.

2.4 Regularizing branch

The regularizing branch processes augmented versions of the input frames to avoid degenerate solutions, such as encoding positional cues into feature representations [4]. This ensures robust and generalizable feature learning.

Algorithm 1: Multilayer Perceptron (MLP) Forward Pass

Input: Input feature tensor $x \in \mathbb{R}^{B \times C \times H \times W}$, where B is the batch size, C is the number of input channels, and H, W are the height and width of the feature map.

Output: Output feature embeddings $z \in \mathbb{R}^{B \times C' \times H \times W}$, where C' is the number of output channels.

1 **for** *Each layer in MLP* **do**
2   Apply Conv2d layer with kernel size 1×1 and stride 1;
     Apply Layer Normalization to the output;
     Apply ReLU activation function;
     Update feature tensor x with the output of the current layer;
3 **return** *Processed feature embeddings z*

2.5 Anchor sampling

To improve training efficiency and reduce computational overhead, we define a spatially invariant grid of size $n \times n$ on the feature tensor from the main branch z, obtaining n^2 z-dimensional feature embeddings. One sample per grid cell is selected as an anchor k, ensuring spatial distinctness and full coverage of the feature embeddings. These anchors are shared with the regularizing branch.

Instead of computing pairwise distances between all feature vectors in the batch, we calculate the cosine similarities between the embeddings of the anchors k and the current features z using:

$$q_{i,j} = \frac{\exp(z_i \cdot k_j / \tau)}{\sum_l \exp(z_i \cdot k_l / \tau)}, \tag{1}$$

where $\tau \in \mathbb{R}^+$ is a temperature hyperparameter, z represents features from the main branch, k represents the anchors, l indexes batch samples, and i, j index the vector dimension.

For the regularizing branch features, only the predominant anchors are selected. The cosine similarities are calculated as:

$$p_i = \arg\max_{j \in \mathcal{N}(i)} \frac{\exp(\hat{z}_i \cdot k_j / \tau)}{\sum_l \exp(\hat{z}_i \cdot k_l / \tau)}, \tag{2}$$

where $\mathcal{N}(i)$ is the index set of anchors from the same video clip as the feature vector $\hat{z}_i$, and $\hat{z}$ represents features from the regularizing branch.

Equation 1 leverages features from multiple videos in the training batch, while Eq. 2 focuses on features from the same video sequence. This dual approach enables the framework to learn both intra-video and inter-video feature embeddings, preserving fine-grained correspondence associations and instance-level feature discrimination.

2.6 Loss function

The training objective of this framework is to learn representations as similarity across views (Fig. 4). This is achieved by calculating pairwise affinities between features from the model's two branches and minimizing the distance of features extracted from temporally close frames to the anchors. The overall loss combines cross-view consistency and space-time self-training losses.

Cross-view consistency
The regularizing branch processes augmented versions of the original video frames, and the model should produce consistent segmentations regardless of random flipping or scaling.
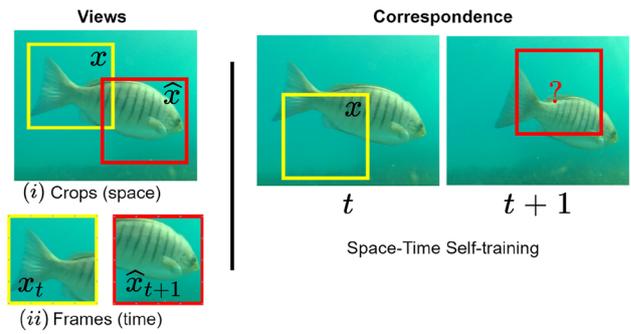

Fig. 4 Representation learning as similarity across views by discriminating features *(i)* spatially within individual frames and *(ii)* temporally, to represent each frame in a video sequence in terms of the same feature set

The cross-view consistency loss enforces this property:

$$\mathcal{L}_{CV} = -\sum_{i \in \mathcal{R}} \log \frac{\exp(z_i \cdot \hat{z}_i / \tau)}{\sum_{l \neq i} \exp(z_i \cdot \hat{z}_l / \tau)}, \tag{3}$$

where:

- $\mathcal{R}$ is the index set of features from the reference frames,
- $\tau \in \mathbb{R}^+$ is a temperature hyperparameter,
- z represents features from the main branch,
- $\hat{z}$ represents features from the regularizing branch.

This loss distinguishes the cosine similarity between corresponding features from non-corresponding pairs.
Space-time self-training
Pseudo labels are generated from the predominant anchor index for each feature in the regularizing branch. The space-time self-training loss minimizes the distance between features from the original view q (Eq. 1) and pseudo labels p (Eq. 2):

$$\mathcal{L}_{ST} = -\sum_{i \notin \mathcal{R}} \log \mathcal{T}(q_i, p_i), \tag{4}$$

where:

- $\mathcal{R}$ is the index set of features from the reference frames,
- $\mathcal{T}(\cdot)$ is a random similarity transform aligning q and p after cropping and flipping.

This loss encourages increased cosine similarity of features to anchors and decreased similarity between anchors.
Final loss
The total loss combines the cross-view consistency and space-time self-training losses:

$$\mathcal{L} = \mathcal{L}_{CV} + \lambda \mathcal{L}_{ST}, \tag{5}$$

where λ is a hyperparameter weighting the contribution of $\mathcal{L}_{ST}$.

2.7 Label propagation

Label propagation is used to predict semantic labels for all video frames using only the ground truth from the first frame as input. This is essentially a classification task where each pixel in every frame of a video is assigned a label, given that only the first frame has ground truth labels available. Following previous work, this study employs representation as a similarity function for k-Nearest Neighbour (KNN) prediction. Algorithm 2 shows the label propagation procedure used in this work.

The mask m_t for the current timestep t is predicted using *context* embeddings and masks from previous frames. The embedding for frame t is obtained from the `CoaT Transformer` output. Next, the cosine similarity of embedding e_t with respect to all embeddings in context $\mathcal{E}$ is calculated. This approach is commonly used in correlation layers of optical flow networks. Following this, local attention is computed in a single operation using `kNN-Softmax`. Finally, the oldest entries in the mask $\mathcal{M}$ and embedding contexts $\mathcal{E}$ are updated by replacing them with m_t and e_t. This process is repeated for all remaining frames in the video clip. Bilinear interpolation is used to resize the final object masks back to their original resolution.

Algorithm 2: Label Propagation

Input: Embeddings $\mathcal{E}$ and mask $\mathcal{M}$ from the first frame.
Output: Mask m_t prediction for timestep t.
1 **for** t *and frame in (frames)* **do**
2 　| 　Computing embeddings e_t at timestep t; Computing local spatial correlation between $\mathcal{E}$ and e_t; Computing softmax between K-Nearest Neighbors; Mask m_t prediction for timestep t; updating $\mathcal{E}$ and $\mathcal{M}$;

---

3 Experiments

This section outlines the training and evaluation methodology for the self-supervised learning model for underwater video segmentation. Quantitative and qualitative results demonstrate the model's ability to generalize across diverse underwater habitats.

3.1 Datasets

Experiments were conducted using three publicly available datasets: DeepFish [6], Seagrass [7], and YouTube-VOS [8].

- **DeepFish**: Contains 40k video frames captured in Full HD (1920 x 1080 pixels) across 20 habitats in tropical Australia.
- **Seagrass**: Includes 4280 video frames with 9429 annotations of *Girella tricuspidata* in Australian estuaries, recorded using submerged action cameras.
- **YouTube-VOS**: Comprises 4453 YouTube video clips with pixel-level annotations for every 5th frame. Only 130 "fish" videos (4349 frames) were used for this study.

The feature extractor was trained on DeepFish and evaluated on Seagrass and YouTube-VOS.

3.2 Data augmentation

Training data was augmented using similarity transformations (random cropping and flipping), as illustrated in Fig. 4. These augmentations were applied to create variations for the regularizing branch, enabling pseudo-label generation. Experiments with additional augmentations (e.g., shearing, rotations, RGB-Shift, color jittering) showed no significant accuracy improvements and were computationally expensive. Thus, only random flips and cropping were used.

3.3 Implementation details

3.3.1 Model training

The Transformer-based feature encoder (Sect. 2.2) served as the backbone network. For comparison, a ResNet-18 baseline was used, with strides removed from the `res3` and `res4` blocks. Both models were randomly initialized and trained on 256×256 pixel inputs, achieved by scaling the shortest side to 256 and extracting random crops. Each forward pass used $B \times T = 2 \times 5 = 10$ frames (2 video sets, 5 frames each).

Training used a learning rate of 1×10^{-3} and required approximately 300 epochs. Models were implemented in PyTorch and trained on an NVIDIA GeForce RTX 2080 Ti GPU (11 GB memory) using the Adam optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.999$, $\epsilon = 1.0 \times 10^{-8}$). The same hyperparameters were applied to all models, though optimal configurations may vary by application. The training loop is outlined in Algorithm 3.

3.3.2 Inference

During inference, dense correspondences for video propagation were computed using the learned encoder's representation (Sect. 2.2, Algorithm 3). Segmentation masks for entire video frames were predicted using label propagation (Sect. 2.7). Given the ground-truth mask for the first frame, labels were propagated to subsequent frames without addi-

tional annotations. Labels in the first frame were represented as one-hot vectors, while propagated labels were Softmax distributions.

Algorithm 3: Main Training Loop

Input: Unlabeled video sequences.
Output: Trained weights for the backbone network.
1 **for** *each mini-batch* **do**
2   Extract deep features of the video frames;
3   Regularising branch produces pseudo labels;
4   **for** *each video in the mini-batch* **do**
5   *// Transformer-based encoder*
         Extract feature embeddings (anchors) k;
6   Compute affinity to anchors q (Eq. (1));
7   Compute pseudo labels q (Eq. (2));
8   *// Loss Computation*
         Compute Cross-view consistency $\mathcal{L}_{CV}$ (Eq. (3));
9   Compute Space-time self-training $\mathcal{L}_{ST}$ (Eq. (4));
10   Compute total loss $\mathcal{L}$ (Eq. (5));
11   Back-propagate all the losses in this mini-batch;

3.4 Evaluation metrics

Two commonly used metrics in video segmentation were employed to evaluate the model's performance:

- **Jaccard's index** ($\mathcal{J}$): Measures the overlap between the predicted segmentation mask and the ground truth mask. It is calculated as: $\mathcal{J}(A, B) = \frac{|A \cap B|}{|A \cup B|}$, where A and B represent the predicted and ground truth masks, respectively.
- **Dice coefficient** ($\mathcal{F}$): Measures the overlap between the predicted and ground truth segmentations. It is calculated as: $\mathcal{F} = \frac{2|A \cap B|}{|A| + |B|}$,

This study reports the mean average of $\mathcal{J}\&\mathcal{F}$, as well as the mean and recall of $\mathcal{J}_m$ and $\mathcal{F}_m$, with an IoU threshold of 0.5.

3.5 Compared methods

The proposed method was evaluated against five self-supervised video object segmentation methods and one fully-supervised method:

- **Self-supervised methods**: CRW [4], DenseFlow [10], MAST [11], Colorize [12], CorrFlow [3].
- **Fully-supervised method**: FCN8 [13], which uses true per-pixel class labels for training and handles imbalanced datasets effectively.

To ensure fairness, all self-supervised methods were trained on the DeepFish dataset using the authors' provided code and parameters.

3.6 Quantitative comparisons

The results for the Seagrass and YouTube-VOS datasets are summarized in Tables 1 and 2, respectively. The best result for each metric is highlighted in bold.

Compared to the second-highest performing method, our approach achieves higher $\mathcal{J}\&\mathcal{F}_m$ scores by 4.5% on Seagrass and 3.1% on YouTube-VOS. This improvement is attributed to the self-attention mechanisms, which extract high-level spatial features and model dependencies across temporal steps, aiding in segmenting objects with large motions.

While the fully-supervised FCN8 method achieves higher $\mathcal{J}\&\mathcal{F}_m$ scores, the proposed method does not require annotated frames, making it more practical for large-scale applications.

3.7 Qualitative results

A qualitative comparison was conducted to visually assess the segmentation results and confirm the model's accuracy in identifying objects within images. The proposed model was compared against baseline models on the YouTube-VOS (rows 1 and 4) and Seagrass (rows 2 and 3) datasets, as depicted in Fig. 5. The results demonstrate that the proposed model effectively distinguishes between objects and backgrounds and outperforms the baseline in handling occlusions. Notably, the model excels at segmenting multiple overlapping fish in complex scenes, as shown in the bottom panel of Fig. 5.

The proposed model also exhibits stability and efficacy in accurately locating fish within long videos, even in complex scenes. For example, it successfully segmented up to frame number 80 based solely on the first frame. In contrast, the baseline method struggled with ambiguity between foreground objects and the background or complex transformations in the videos. The proposed method demonstrated a strong ability to differentiate pixels with similar intensities.

Additionally, the model performed effectively on datasets containing very small objects, as seen in Videos 2 and 3 of Fig. 5. To further validate the proposed method, Fig. 6a and b provide comparisons with five state-of-the-art self-supervised models (CRW, DenseFlow, MAST, Colorize, and CorrFlow) on the YouTube-VOS and Seagrass datasets, respectively.

3.8 Ablation study

An ablation study was performed to thoroughly examine the impact of the proposed framework. This study involved comparing the baseline models with our method. To ensure a comprehensive analysis, various configurations of video features, and MLP layers were evaluated to determine the

Table 1 Performance comparison on Seagrass [7] dataset between our model and five state-of-the-art models (CRW [4] DenseFlow [10] MAST [11] Colorize [12] CorrFlow [3])

Method	$\mathcal{J}\&\mathcal{F}$(Mean) ↑	$\mathcal{J}$(Mean) ↑	$\mathcal{J}$(Recall) ↑	$\mathcal{F}$(Mean) ↑	$\mathcal{F}$(Recall) ↑
CRW [4]	43.2	38.9	40.4	46.2	50.8
DenseFlow [10]	45.5	40.2	41.0	50.7	54.7
MAST [11]	40.3	37.1	38.7	43.8	48.5
Colorize [12]	34.9	34.5	35.1	40.8	47.9
CorrFlow [3]	39.4	36.8	36.9	42.7	47.2
Ours	**50.0**	**41.5**	**43.3**	**58.1**	**65.4**
Fully-supervised [13]	64.7	52.4	55.7	71.4	79.1

Table 2 Performance comparison on YouTube-VOS [8] dataset between our model and five state-of-the-art models (CRW [4] DenseFlow [10] MAST [11] Colorize [12] CorrFlow [3])

Method	$\mathcal{J}\&\mathcal{F}$(Mean) ↑	$\mathcal{J}$(Mean) ↑	$\mathcal{J}$(Recall) ↑	$\mathcal{F}$(Mean) ↑	$\mathcal{F}$(Recall) ↑
CRW [4]	59.9	58.6	71.5	57.8	68.7
DenseFlow [10]	60.2	60.9	72.7	59.5	**70.0**
MAST [11]	57.4	57.9	68.1	56.9	65.2
Colorize [12]	53.7	54.1	65.9	55.4	64.8
CorrFlow [3]	56.4	55.9	66.7	54.3	64.8
Ours	**63.3**	**63.9**	**74.0**	**62.7**	69.6
Fully-supervised [13]	79.3	78.2	89.3	70.5	83.7

Fig. 5 Qualitative comparison between our model and a baseline [10] model applied on the YouTube-VOS (rows 1 and 4) [8], and Seagrass (rows 2 and 3) [7] datasets. The representation learned by our model effectively distinguishes between objects and background ambiguity and is robust to occlusions
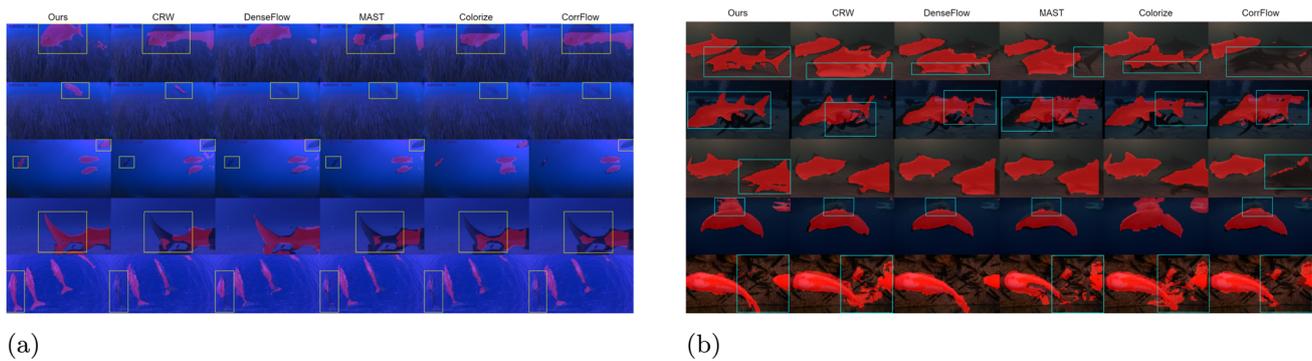
(a)   (b)
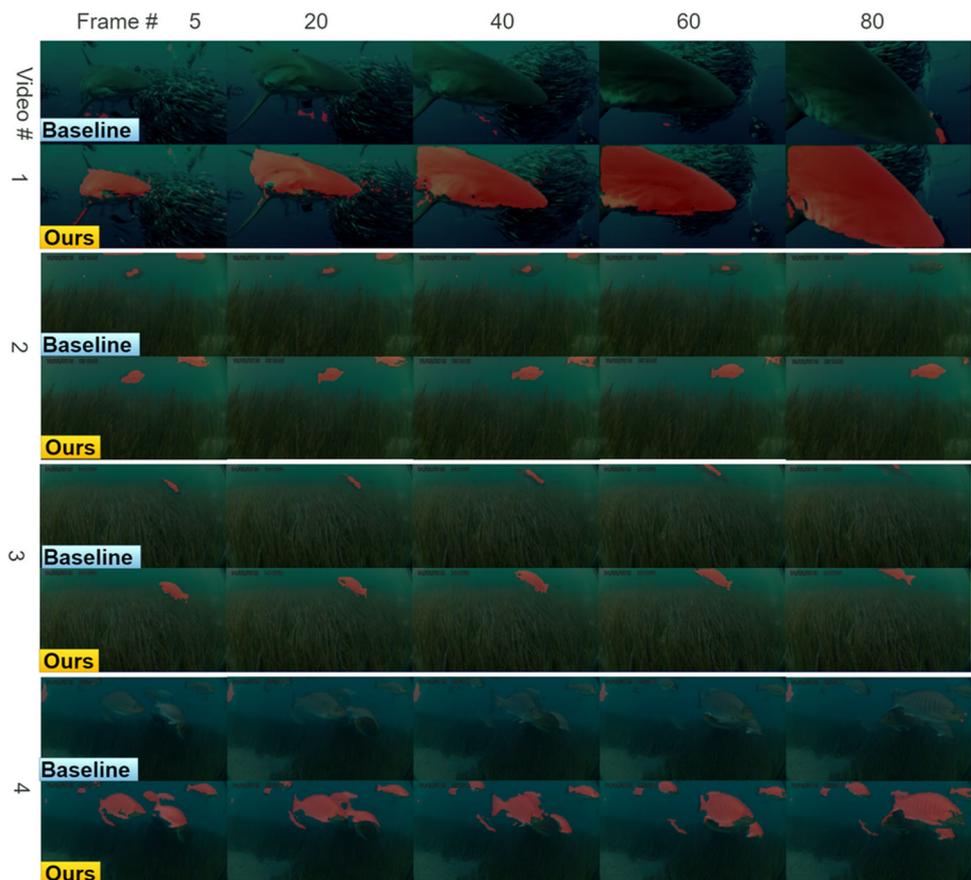
Fig. 6 Qualitative comparison between our model and five state-of-the-art models on the Seagrass and YouTube-VOS datasets. **a** Results from Seagrass and YouTube-VOS datasets. **b** Results from the YouTube-VOS dataset. Note: For better visibility of details, please view the figures online and zoom in

Table 3 Ablation study for other models on Seagrass [7] and YouTube-VOS [8] datasets

Method	$\mathcal{J}\&\mathcal{F}_{\text{mean}}$	
	Seagrass [7]	YouTube-VOS [8]
Baseline [10]	45.5	60.2
FFC [14]	46.2	60.6
Segmenter [15]	41.5	52.7
PoolFormer [16]	42.8	54.9
XCiT [17]	43.7	56.8
Ours	**50.0**	**63.3**

The bold values highlight the best-performing results for each evaluation metric across different methods

combination yielding the best performance. Only the results obtained from the most optimal configurations are reported.

Table 3 presents the segmentation accuracy of four different models based on the $\mathcal{J}\&\mathcal{F}_m$ metric, along with the baseline model and the proposed models. In Table 3, the second row corresponds to the Fast Fourier Convolution (FFC) model. The third row represents the Transformer for Semantic Segmentation, while the fourth row presents the MetaFormer-based architecture, PoolFormer. The fifth row represents the Cross-Covariance Image Transformer (XCiT). Based on the evaluation summarised in Table 3, the proposed model significantly outperforms the baseline models and other meta-architecture methods across both datasets.

3.9 Failure case

During the experiments, it was observed that in a few instances, portions of the background were mistakenly segmented as fish. Two such examples are illustrated in Fig. 7a. It is suggested that incorporating additional modalities, such as depth or motion information, could potentially enhance the attention model's performance in scenarios where the input contains objects visually similar to the background.

This could be particularly beneficial when objects and backgrounds share similar colour or texture characteristics. Another observation was that the model struggled to accurately segment the entire fish body in cases of heavy occlusion from seagrass. Figure 7b showcases instances of this issue.

4 Limitations

While this study introduces a novel self-supervised learning approach for underwater fish segmentation, several limitations must be acknowledged. First, the model was trained on the *DeepFish* dataset, which is specific to tropical Australian coastal environments. Although it generalized well to the Seagrass and YouTube-VOS datasets, its performance in other habitats or on rare fish species remains unvalidated. Second, the model occasionally missegmented background elements resembling fish in color or texture and struggled with heavy occlusions (e.g., seagrass). Performance in extreme conditions like poor lighting, turbidity, or variable water clarity was not extensively tested, which could limit its applicability in such scenarios. Third, errors in the initial ground-truth segmentation could propagate to subsequent frames due to the label propagation approach. Finally, while computationally efficient, the suitability of the model for low-power or mobile edge devices was not evaluated, and further optimisation may be required for deployment in resource-constrained environments.

5 Conclusion

This study introduced a self-supervised learning approach for segmenting fish in underwater videos, eliminating the need for manual annotation and enabling efficient data preparation. The model achieved promising results, with segmentation accuracy comparable to fully supervised meth-
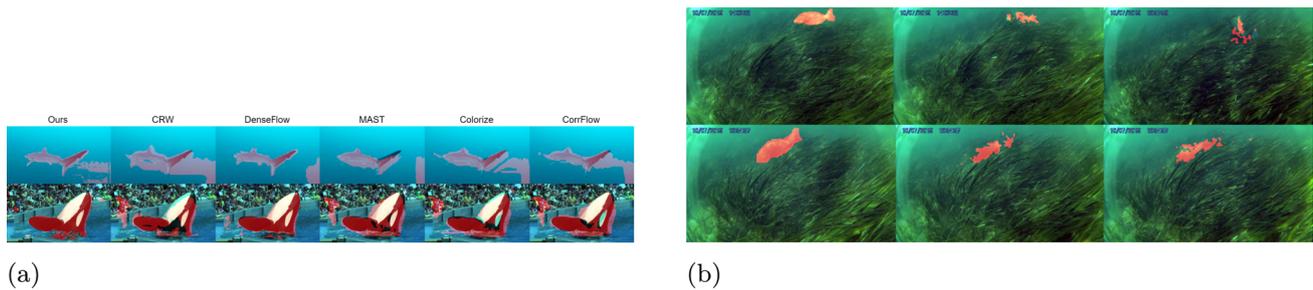
Fig. 7 Failure cases of our model on different datasets. **a** YouTube-VOS dataset failure due to background similarity. **b** Seagrass dataset failure due to heavy occlusion. Note: For better visibility of details, please view the figures online and zoom in

ods in specific scenarios, though supervised methods may outperform it under ideal conditions. While the approach generalized well to the Seagrass and YouTube-VOS datasets after training on DeepFish, further validation in more diverse underwater environments is needed. The model's performance is currently limited in cases of severe occlusion, such as dense seagrass, and its suitability for low-power devices remains unexplored. Overall, this method provides a viable alternative to fully supervised approaches, particularly where manual annotation is impractical. Future work could focus on improving robustness to occlusions and extending the model to tasks like fish species identification and segmentation of other underwater objects.

Author Contributions A.S., Conceptualisation, Data Curation, Data Analysis, Software Development, DL Algorithm Design, Visualization, Writing original draft. M.R.A: Conceptualisation, PhD Supervision, Reviewing/editing the draft. M.S, D.J: Writing-Reviewing and Editing. M.S., D.J., M.R.A.: Supervision.

Data Availability The datasets generated during and analysed during the current study are publicly available.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Saleh, A., Sheaves, M., Rahimi Azghadi, M.: Computer vision and deep learning for fish classification in underwater habitats: a survey. Fish Fish. **23**(4), 977–999 (2022). https://doi.org/10.1111/faf.12666

2. Shao, J., Wen, X., Zhao, B., Xue, X.: Temporal Context Aggregation for Video Retrieval with Contrastive Learning. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 3267–3277 (2021). https://doi.org/10.1109/WACV48630.2021.00331

3. Lai, Z., Xie, W.: Self-supervised video representation learning for correspondence flow. Proceedings of the British Machine Vision Conference (BMVC), 121–112113 (2019). https://doi.org/10.5244/C.33.121

4. Jabri, A.A., Owens, A., Efros, A.A.: Space-time correspondence as a contrastive random walk. Adv. Neural Inf. Process. Syst. **33**, 19545 (2020)

5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR) (2021)

6. Saleh, A., Laradji, I.H., Konovalov, D.A., Bradley, M., Vazquez, D., Sheaves, M.: A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. Sci. Rep. **10**(1), 14671 (2020). https://doi.org/10.1038/s41598-020-71639-x

7. Ditria, E.M., Connolly, R.M., Jinks, E.L., Lopez-Marcano, S.: Annotated video footage for automated identification and counting of fish in unconstrained seagrass habitats. Front. Mar. Sci. **8**, 629485 (2021). https://doi.org/10.3389/fmars.2021.629485

8. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: 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) (2018). https://doi.org/10.1007/978-3-030-01228-1_36

9. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-Scale Conv-Attentional Image Transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 9961–9970 (2021). https://doi.org/10.1109/ICCV48922.2021.00983

10. Araslanov, N., Schaub-Meyer, S., Roth, S.: Dense unsupervised learning for video segmentation. Adv. Neural Inf. Process. Syst. **34**, 25308–25319 (2021)

11. Lai, Z., Lu, E., Xie, W.: MAST: A memory-augmented self-supervised tracker. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2020). https://doi.org/10.1109/CVPR42600.2020.00651

12. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking Emerges by Colorizing Videos. In: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol. 11217 LNCS, pp. 402–419 (2018). https://doi.org/10.1007/978-3-030-01261-8-24. https://link.springer.com/10.1007/978-3-030-01261-8-24

13. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transact. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2017). https://doi.org/10.1109/TPAMI.2016.2572683

14. Chi, L., Jiang, B., Mu, Y.: Fast Fourier Convolution. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 4479–4488. Curran Associates, Inc., (2020)

15. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7262–7272 (2021)

16. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10819–10829 (2022)

17. Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: cross-covariance image transformers. Adv. Neural Inf. Process. Syst. **34**, 20014–20027 (2021)