Turtle Body Parts Segmentation with U-Net on Focus and Dual Proposal Mask R-CNN

UNSW COMP9517 24T3 Group Project

Haokai Zhao	Jonas Macken	Liangji Kong	Ruiyuan Yang	Yaqing He
z5261811	z5208799	z5453970	z5538752	z5514294

I. Introduction

Sea turtle image segmentation is a valuable computer vision task in wildlife research, aiding in identifying individual turtles by segmenting body parts like the head, flippers, and carapace. This task supports population tracking, health monitoring, and migration studies, which are essential for conservation efforts. Traditionally, manual segmentation is labor-intensive and time-consuming. Recent advancements in computer vision enable the automation of this process, significantly reducing effort and enhancing data analysis efficiency. By leveraging large datasets like SeaTurtleID2022 [1], researchers can efficiently handle extensive image collections, improving the scalability of long-term studies.

With the development of deep learning methods, there are many high-performance models that serve various image segmentation tasks, including instance segmentation, semantic segmentation, and panoptic segmentation. Many of them were pre-trained on large-scale datasets and demonstrated strong one-shot or zero-shot performance on unseen datasets.

However, when applying pre-trained models to turtle body parts segmentation, there are 2 key challenges that can drag the performance down. First, most of the models were designed and trained on everyday picture datasets, such as ImageNet [9] and COCO [12]. These datasets may have labels for the whole turtle image but not the turtle body parts (flippers, head, and carapace), which will result in insufficient performance when adapting pre-trained models to segment turtle body parts as opposed to the whole turtle. Second, unlike in a segmentation task for individual objects, where objects can appear anywhere in the scene, body parts for the same turtle should always be connected, and their constituent pixels in the image should always form a subset of the corresponding turtle. Neither proposal-based models, such as Mask R-CNN, nor proposal-free segmentation models, such as U-Net, have a specific design for utilizing the hierarchical structure of the whole turtle and its body parts.

In this project, we developed Dual-Proposal Mask R-CNN (DPMR) and U-Net on Focus (UFO), for improving Mask R-CNN and U-Net, respectively, for this task. Both methods were motivated by the assumption that whole turtle segmentation is easier than body parts segmentation, as well as by the specific dataset characteristic of a hierarchical structure between the whole turtle and its body parts.

For DPMR, we trained the Mask R-CNN model to segment not only the body parts but also the whole turtle. We used the predicted bounding box for the whole turtle during the prediction stage to filter the predicted segmentation mask for body parts. Any predicted body parts will be dropped if their intersected area with the predicted whole turtle bounding box is below a threshold. In this way, ideally, we can improve the model performance by filtering out those false positive segmentations (the pixels predicted to be body parts that are actually not).

For UFO, we trained a native U-Net to predict the class label (background, carapace, flippers, and head) for each pixel. At the prediction stage, we first use the trained U-Net to predict the segmentation mask, then up-size the original image before cropping it to focus only on the turtle, which is equivalent to a zoom-in operation with more pixels on the turtle. Then we use the same model to predict the segmentation mask on the turtle-focused image. In this way, we can improve the performance by zooming in on the images with small turtles.

Our main developments and experiments can be summarized as the following:

- We conducted experiments and performed data analysis to identify the limitations of existing instance segmentation and semantic segmentation models, in particular, Mask R-CNN and U-Net.
- We developed DPMR, an improved version of Mask R-CNN for turtle body parts segmentation. We demonstrated that DPMR can reduce false positive predictions in certain cases, resulting in a higher mean Intersection over Union (mIoU) for body parts in those specific cases.
- We developed UFO, an improved version of U-Net for turtle body parts segmentation, which improved the average mean Intersection over Union (mIoU) of U-Net by a large margin, from 0.673 to 0.737 (+9%).

II. Literature Review

In the early stages of computer vision, traditional image segmentation methods such as edge detection, region-growing, and graph-based approaches were widely used. Edge detection methods, like the Canny edge detector [4], aimed to

identify object boundaries by detecting points with high-intensity gradients. This approach proved useful for simple segmentation tasks where object contours are well-defined. On the other hand, region-growing [2] and graph-based methods, including Normalized Cuts [16], focused on grouping pixels or partitioning graphs to separate objects from the background. Although these methods are computationally efficient, they often struggle with complex real-world images where object boundaries are less distinct, making them less suitable for segmenting intricate shapes and varied poses, as seen in sea turtle images.

As computer vision progressed, traditional segmentation methods faced limitations in handling complex images with varied shapes and unclear boundaries. This led to the adoption of machine learning methods, which offered advantages through data-driven learning and adaptability to diverse image features. Techniques like Support Vector Machines (SVM) [7] and k-Nearest Neighbors (k-NN) [8] demonstrated potential in image classification by effectively distinguishing between object categories based on feature vectors. Additionally, random forests [3] were applied to image tasks, leveraging ensemble learning to improve robustness. Despite these advancements, segmentation tasks require more precise boundary delineation, which traditional machine-learning methods generally lack. Consequently, machine learning alone did not fully meet the requirements for complex tasks like sea turtle segmentation, highlighting the need for more advanced approaches.

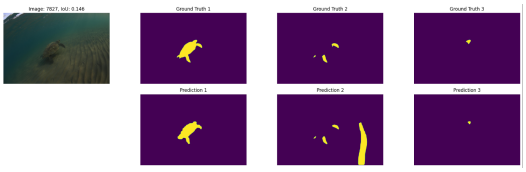
In recent years, deep learning has revolutionized the field of image segmentation through models like Fully Convolutional Networks (FCN) [13], U-Net [15], Mask R-CNN [10], Deeplabv3 [5], YOLO [14], and Mask2Former [6]. FCN [13] was one of the first models to introduce the concept of pixel-wise segmentation, marking a significant advancement in the field. Building on this, U-Net [15] was specifically designed for biomedical image segmentation and has shown effectiveness in handling tasks that require detailed boundary delineation, such as sea turtle body part segmentation. Additionally, Mask R-CNN [10] extended object detection frameworks to support segmentation by adding a mask branch, enabling precise segmentation of complex shapes. Models like Deeplabv3, which employs atrous convolution to capture multi-scale context, YOLO for real-time object detection, and the Masked-Attention Mask Transformer for universal segmentation tasks, further expand the range of deep learning approaches available for segmentation tasks.

Sea turtle segmentation presents unique challenges, particularly in segmenting detailed body parts such as the head, flippers, and carapace. Given these requirements, we selected Mask R-CNN and U-Net as our foundational models due to their strengths in handling complex segmentation tasks. Mask R-CNN, with its mask prediction branch, enables pixel-level segmentation, making it effective for capturing the intricate shapes of sea turtles' body parts. Similarly, U-Net's encoder-decoder architecture is ideal for tasks requiring detailed boundary delineation.

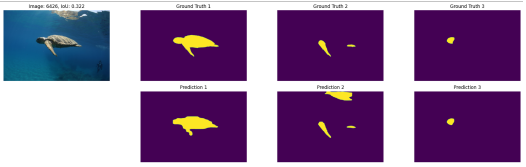
III. Dual Proposal Mask R-CNN (DPMR)

Initial experiments with Mask R-CNN were conducted to segment three types of instances as defined in the original dataset: whole turtle, flippers, and head. During the evaluation, we observed false positives where the model incorrectly classified background elements, such as sandy ground and reflections distant from the turtle body, as flippers (Fig. 1). We attribute these false positives to Mask R-CNN's architecture design, which classifies each proposed bounding box independently without considering relationships between potential instances. While this approach is effective for unstructured scenes like general object detection in everyday photos, it fails to incorporate the inherent anatomical prior knowledge of turtle morphology: specifically, that body parts must be spatially contained within the whole turtle instance.

Based on the hierarchical relationship between the whole turtle and its body parts, and the observation that the whole turtle segmentation demonstrates superior performance compared to individual body part detection, we developed Dual Proposal Mask R-CNN to enhance body part segmentation accuracy. The key innovation of our approach is training the model to simultaneously predict two types of instances: body parts (carapace, flippers, and head) and the whole turtle, effectively creating a dual proposal system for both sub-instances and super-instances. During inference, DPMR implements a two-stage prediction process. First, the model simultaneously predicts both whole-turtle instances (super-instances) and body part instances (sub-instances). Subsequently, body part predictions undergo filtering based on their spatial relationship with the detected whole-turtle instances. Specifically, if a predicted body part instance lacks sufficient overlap with any detected whole-turtle instance, it is excluded from the final predictions. We expect that this spatial constraint mechanism can suppress false positive detections of body parts, leading to improved mean Intersection over Union (mIoU) scores.



(a) Prediction for Image 7827 (average mIoU 0.146)



(b) Prediction for Image 6426 (average mIoU 0.322)

Fig. 1: Cases of finetuned Mask R-CNN for whole turtle, flippers, and head segmentation. Top-left: test image. First row: ground truth mask for whole turtle, flippers, head. Second row: prediction mask for whole turtle, flippers, and head

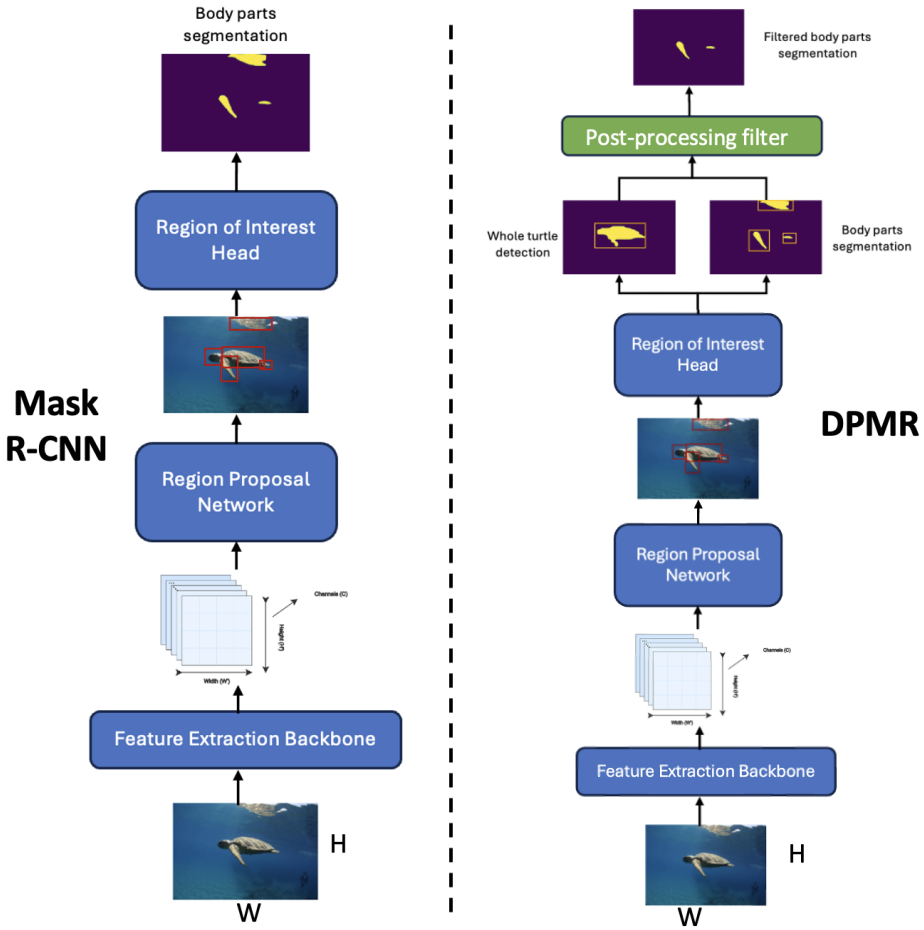


Fig. 2: Overview of Mask R-CNN and Dual Proposal Mask R-CNN

Our DPMR consists of four key modules: a feature extraction backbone (Sec. III-A), a regional proposal network (Sec. III-B), an ROI head for segmentation (Sec. III-C), and a post-prediction filter (Sec. III-D). Fig. 2 illustrates the overall architecture and highlights its architectural differences from the standard Mask R-CNN (Fig. 2).

A. Feature Extraction Backbone

The feature extraction backbone in our DPMR maintains the same architecture as the standard Mask R-CNN for extracting deep feature representations from input images. We employed ResNet-50, pre-trained on ImageNet, as our feature extractor. The backbone processes an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, where H and W denote the height and width of the input image, respectively, and 3 represents the RGB channels. The backbone outputs a feature map $\mathbf{F} \in \mathbb{R}^{H' \times W' \times C}$, where H' and W' represent the spatial dimensions after feature extraction, and C denotes the number of output channels, encoding multi-scale semantic information. The feature map $\mathbf{F}$ is subsequently fed into the Region Proposal Network (RPN) to facilitate object detection and segmentation in later stages.

B. Regional Proposal Network (RPN)

The Region Proposal Network (RPN) in our DPMR adopts the same architecture as the original RPN in Mask R-CNN, generating candidate bounding boxes (region proposals) for potential object instances. The RPN architecture comprises a shared convolutional backbone followed by two parallel branches: an objectness classification branch and a bounding box regression branch. The classification branch performs binary classification to determine whether each proposed region contains an object (foreground) or background, while the regression branch optimizes the spatial coordinates of the bounding boxes. A key distinction in DPMR is that the RPN generates proposals across four categories: whole turtle (super-instance) and its constituent parts- carapace, head, and flippers (sub-instances). This dual proposal mechanism explicitly models the hierarchical relationship between the complete turtle and its anatomical components.

The Region Proposal Network (RPN) takes the feature map $\mathbf{F}$ from the feature extraction backbone as input and generates two outputs: a set of bounding boxes $\mathcal{B} = \{B_i\}_{i=1}^{N}$, and their corresponding objectness scores $\{s_i\}_{i=1}^{N}$. Each bounding box $B_i = (x_i, y_i, w_i, h_i)$ is parameterized by its center coordinates (x_i, y_i) and dimensions w_i and h_i representing width

and height, respectively. The objectness scores $s_i \in [0, 1]$ quantifies the network's confidence that the i-th bounding box contains a foreground object.

C. ROI Head

The ROI Head processes the region proposals $\mathcal{B}$ from the RPN through two parallel streams: the whole turtle stream and the body part stream. Each stream follows the standard Mask R-CNN architecture, comprising a classification layer, a bounding box regression layer, and a mask prediction branch. The whole turtle stream performs binary classification (whole turtle vs. background) and refines the bounding boxes for the complete turtle instance. The body part stream processes individual anatomical components (head, flippers, and carapace), performing classification, bounding box refinement, and instance segmentation for each body part.

The ROI Head takes two inputs: the region proposals $\mathcal{B} = \{B_i\}_{i=1}^{N}$ and the feature map $\mathbf{F}$. It produces three outputs: 1. the refined bounding boxes $\hat{\mathcal{B}} = \{\hat{B}_i\}_{i=1}^{N}$, where each $\hat{B}_i = (\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$ represents the optimized coordinates and dimensions of the i-th bounding box with center $(\hat{x}_i, \hat{y}_i)$ and dimensions $\hat{w}_i$ $\hat{h}_i$; 2. class scores $\hat{\mathbf{C}} = \{\hat{c}_i\}_{i=1}^{N}$, where $\hat{c}_i$ denotes the predicted class probability for the i-th proposal, indicating the likelihood of each class (e.g., whole turtle, head, flippers, or carapace); 3. instance segmentation masks $\hat{\mathbf{M}} = \{\hat{M}_i\}_{i=1}^{N}$, where each $\hat{M}_i \in \mathbb{R}^{m \times m}$ represents the predicted binary mask for the i-th proposal at resolution $m \times m$.

D. Post-processing Filter

The post-prediction filtering stage aims to enforce consistency between the predicted whole-turtle instances and their constituent body parts by leveraging the inherent hierarchical structure of the data. Following the ROI Head's generation of bounding boxes $\hat{\mathcal{B}}$ and instance masks $\hat{\mathbf{M}}$, we implemented a spatial filtering mechanism that validates body part predictions based on their overlap with whole-turtle instances.

For each predicted body part instance $\hat{B}_{\text{part}}$ (head, flipper, or carapace) and predicted whole turtle instance $\hat{B}_{\text{turtle}}$, we compute the intersection area $\text{Area}(\hat{B}_{\text{part}} \cap \hat{B}_{\text{turtle}})$. We then define the overlap ratio r as:

$$r = \frac{\text{Area}(\hat{B}_{\text{part}} \cap \hat{B}_{\text{turtle}})}{\text{Area}(\hat{B}_{\text{part}})}.$$

If the overlap ratio r falls below a predefined threshold τ, the corresponding body part instance $\hat{B}_{\text{part}}$ is discarded from the final predictions. This filtering mechanism enforces the anatomical prior that body parts must be spatially contained within the whole turtle instance, thereby effectively suppressing false positive detections arising from visually similar background elements.

E. Overall Pipeline

The complete pipeline of our DPMR is formally detailed in (Algorithm. 1). The process consists of four sequential stages:

1) Feature extraction: A ResNet-50 backbone processes the input image to generate high-level feature representations;
2) Proposal generation: The RPN produces candidate bounding boxes for both super-instances (whole turtle) and sub-instances (carapace, head, flippers);
3) Instance refinement: The dual-stream ROI head processes these proposals to:
   - Refine the bounding box coordinates
   - Predict instance classifications
   - Generate segmentation masks
4) Hierarchical validation: The post-processing filtering mechanism enforces anatomical constraints by validating spatial relationships between predicted body parts and whole-turtle instances, thereby suppressing false positives and enhancing segmentation accuracy.

Our DPMR model uses the same loss functions as standard Mask R-CNN: the RPN loss (classification and bounding box regression), the ROI Head loss (object classification and box refinement), and the mask loss (binary cross-entropy for segmentation). These components are summed to form the final loss for end-to-end training.

Algorithm 1 Dual Proposal Mask R-CNN Pipeline

Input: Image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, threshold τ
Output: Filtered masks $\hat{\mathbf{M}}^{\text{filtered}}$

Extract feature map:

$$\mathbf{F} \leftarrow \text{Backbone}(\mathbf{I})$$

Generate region proposals:

$$\mathcal{B} \leftarrow \text{RPN}(\mathbf{F})$$

Refine proposals, classify, and generate masks for each proposed instance:

$$(\hat{\mathcal{B}}, \hat{\mathbf{C}}, \hat{\mathbf{M}}) \leftarrow \text{ROI Head}(\mathcal{B}, \mathbf{F})$$

$\hat{\mathbf{M}}^{\text{filtered}} \leftarrow \varnothing$
for each body part $(\hat{B}_{\text{part}}, \hat{M}_{\text{part}})$ **do**
  for each whole turtle $\hat{B}_{\text{turtle}}$ **do**
    Compute overlap ratio:

$$r \leftarrow \frac{\text{Area}(\hat{B}_{\text{part}} \cap \hat{B}_{\text{turtle}})}{\text{Area}(\hat{B}_{\text{part}})}$$

    if $r > \tau$ **then**
      $\hat{\mathbf{M}}^{\text{filtered}} \leftarrow \hat{\mathbf{M}}^{\text{filtered}} \cup \{\hat{M}_{\text{part}}\}$
    end if
  end for
end for

return $\hat{\mathbf{M}}^{\text{filtered}}$

IV. U-Net on Focus (UFO)

Initial experiments with U-Net were conducted to segment three types of objects: carapace, flippers, and head. We found that U-Net generally performs poorly on images with small turtles (i.e., images where the turtle is far away and therefore takes up a small number of pixels), mixing up the carapace, flippers, and head. However, the model was still effective in distinguishing the background from the entire turtle even for these zoomed-out images, as shown in Fig. 3.

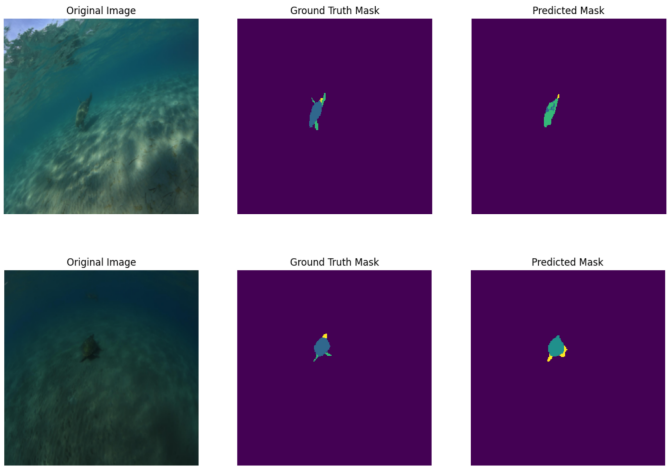


Fig. 3: Examples of U-Net failing to distinguish between body parts on zoomed-out images. Left: Test image. Middle: Ground truth mask. Right: Prediction mask

The limited success on zoomed-out images is understandable; the original images are large and need to be downsized before training for computational reasons. So, the body parts for small turtles will take up very few pixels and will therefore be very difficult to distinguish from one another after the initial downsizing process. To tackle this problem, we propose U-Net on Focus (UFO), a unique architecture that makes use of two inference stages and custom processing steps to ensure turtles are large enough (i.e., they consist of enough pixels) before predictions are made.

Our UFO consists of two key modules: a U-Net architecture (Sec. IV-A), and a cropping block (Sec. IV-B). (Fig. 4) illustrates the overall architecture and highlights its architectural differences from the standard U-Net implementation.

A. U-Net Architecture

The U-Net architecture is a convolutional neural network architecture consisting of a symmetric encoder-decoder structure with skip connections to capture both high-level and fine-grained features. We train a U-Net model on pairs of images and segmentation masks (arrays of the same dimension as the images with each pixel being assigned to a single category). Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, it first gets resized to a smaller image $\mathbf{I}' \in \mathbb{R}^{H' \times W' \times 3}$, where H' and W' denote the height and width of the resized image, respectively, and 3 represents the RGB channels. The resize is necessary for computation feasibility as the original image is too large. The resized image $\mathbf{I}'$ is then fed into the U-Net model. It outputs a segmentation mask $\mathbf{M} \in \mathbb{R}^{H' \times W' \times 3}$, with each pixel assigned a single classification value, corresponding to either the background or the carapace, flippers, or head of the turtle.

B. Cropping Block

The cropping block receives the segmentation mask that has been output by the U-Net model in the first stage of inference and scaled up to $H \times W$. The locations of the uppermost, lowermost, leftmost, and rightmost pixels whose label is *not* background (i.e., is labeled as one of carapace, flippers, or head) are recorded as (x_t, y_t), (x_b, y_b), (x_l, y_l), and (x_r, y_r), respectively. Since the U-Net accurately predicts the turtle as a whole, despite failing to distinguish body parts, these coordinates can be used to reliably give us the corners of a bounding box that minimally captures the whole turtle. Note that if an image has more than one turtle, this method creates a minimal bounding box that includes all turtles in the image.

Next, some margin will be added to each edge of the generated bounding box to ensure entire turtles are captured. Applying some margin ratio m, the margin size is calculated as follows:

$$\text{margin}_h = m \cdot (y_b - y_t), \quad \text{margin}_w = m \cdot (x_r - x_l) \quad (1)$$

The total height and width of the bounding box are then calculated:

$$\text{h}_0 = 2{\cdot}\text{margin}_h + (y_b - y_t), \quad \text{w}_0 = 2{\cdot}\text{margin}_w + (x_r - x_l) \quad (2)$$

Finally, the original $H \times W$ image is cropped at the bounding box coordinates and passed to the next step in the pipeline, where it will undergo a second stage of inference.

C. Overall Pipeline

The complete pipeline of our UFO process consists of four sequential stages:

1) First-stage inference: Original image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ is downsized to $H' \times W'$ and passed through a trained U-Net model, which outputs a segmentation mask of size $H' \times W'$. This segmentation mask is then up-sized using bilinear interpolation to size $H \times W$.
2) Cropping block: The $(H \times W)$ segmentation mask is passed through the cropping block to obtain a cropped image of the turtle(s) of size $h_0 \times w_0$ from the original image.
3) Second-stage inference: The cropped $(h_0 \times w_0)$ image is resized back to $H' \times W'$ and passed through the same U-Net model again for the second inference stage. After obtaining the segmentation mask for this second inference stage, we resize the mask to dimensions of $h_0 \times w_0$. This ensures that the mask is now the size of the cropped turtle from the output of the first stage.

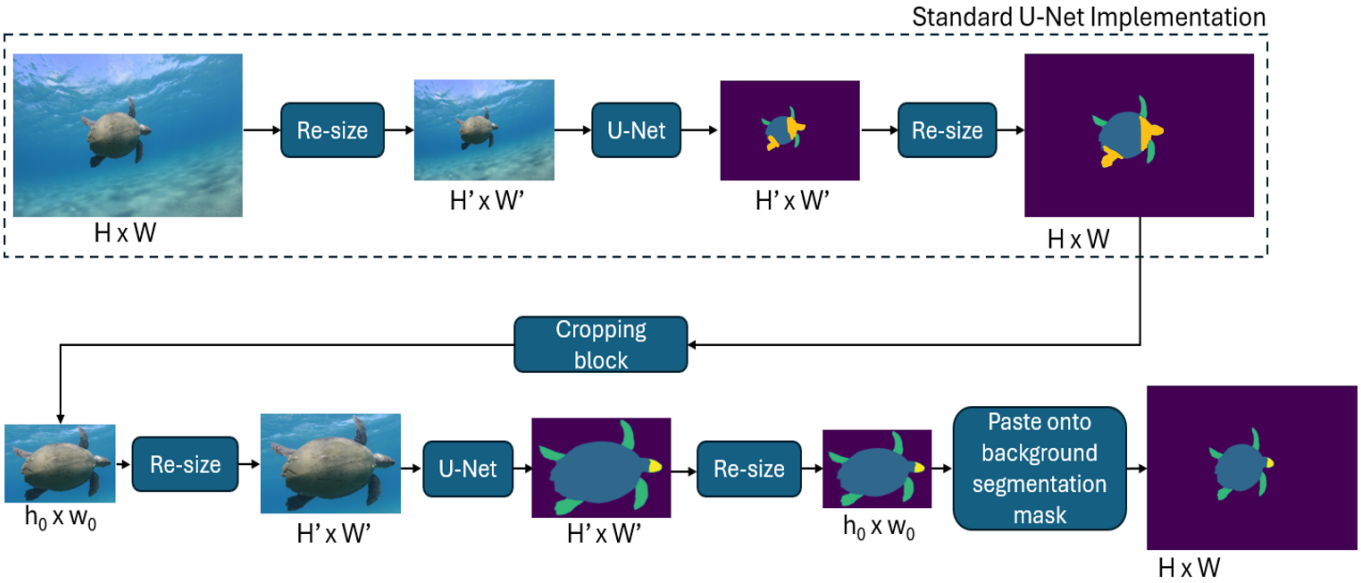


Fig. 4: Overview of UFO architecture

4) Final mask construction: We create a new segmentation mask with dimensions $H \times W$ and with all values set to 0 (background label). We then paste our $h_0 \times w_0$ "zoomed" turtle mask onto the newly constructed segmentation mask at location $(x_l - \text{margin}_w, y_t - \text{margin}_h)$. This final predicted mask is then evaluated against the ground truth mask.

For training, we used Cross Entropy Loss to classify each pixel to one of the background, carapace, flippers and head.

V. Experiment Settings

A. Datasets

We used the SeaTurtleID2022 [1] to develop and evaluate our models. This dataset contains 8729 photographs of sea turtles, and each image comes with annotated segmentation masks for the body parts. It is worth noting that the annotations in the dataset are for the following categories: {*head, flippers, whole turtle*}; whereas our task deals with a slightly different set of categories: {*head, flippers, carapace*}, where *carapace* refers to all parts of the turtle that is not a head or flipper. We have therefore processed the ground truth masks according to our task, by subtracting flippers and heads from turtle ground truth masks to get masks for the carapace.

Creators of the SeaTurtleID2022 dataset have also proposed a realistic and ecologically motivated split: the *time-aware open-set split*, which is readily available with the dataset. In this split, training, validation and testing sets are from different spans of years, and the test and validation sets contain images of turtles newly introduced to the population (i.e., 'unknown' to the training set). The creators have explained in [1] that time-unaware splits (e.g., random splits) of the dataset may lead to significant overestimation bias. Based on this recommendation, we have decided to develop and evaluate our approaches using the *time-aware open-set split*.

The size of each set in splits that we use is summarized in Table. I

Set	Images	Head	Flipper	Carapace
Training	5303	5354	12768	5182
Validation	1118	1129	2720	1086
Testing	2308	2318	5599	2258

TABLE I: Number of samples and number of instances in each split set

B. Model Settings

We employed Detectron2 [17] for implementing the baseline Mask R-CNN models and used segmentation-models-pytorch [11] for the U-Net baselines. All baseline methods, along with our proposed model, were integrated into the Detectron2 framework for unified training, evaluation, and inference. To ensure a fair comparison, all models were trained with a batch size of 16 for 20,000 iterations. During training, model performance was validated on the validation set every 1,000 iterations. The checkpoint with the highest validation performance was selected for final evaluation on the test set. Standard data augmentations, including random scaling and flipping, were applied across all training processes as per Detectron2's default settings.

All of our training and evaluation are completed on a server with 32Gi Memory, 2 CPU cores, and an NVIDIA TITAN X (Pascal) (12Gi CUDA Memory). The training time for Mask R-CNN, DPMR, U-Net, UFO is 8.4, 9.6, 2.0 and 2.0 hours, respectively.

For Mask R-CNN and DPMR, we used the pre-trained R50-C4 Mask R-CNN model from Detectron2 and fine-tuned it on the turtle dataset. The output class count was set to 3 for the baseline Mask R-CNN (carapace, flippers, and head) and to 4 for DPMR (whole turtle, carapace, flippers,

and head). All other hyperparameters were kept as per the default configuration. We explored the intersection acceptance threshold $\tau \in \{0.2, 0.4, 0.6, 0.8\}$ for DPMR to gain a deeper understanding of its behavior.

For U-Net and UFO, we used a pre-trained ResNet-34 as the backbone and fine-tuned it on the turtle dataset. The output class count was set to 4, corresponding to background, carapace, flippers, and head. All other hyperparameters followed the default settings in segmentation-models-pytorch [11]. As UFO does not require additional training, we utilized the trained baseline U-Net model for a consistent comparison and faster development. We experimented with different crop margin ratios $m \in \{0.1, 0.4, 0.7, 1.0, 2.0, 5.0, 10.0\}$ for UFO to further analyze the model's behavior.

VI. Experimental Results

The overall per-body-part mIoU and average mIoU for Mask R-CNN, DPMR, U-Net, and UFO are reported in (Sec. VI-A). Results with various IoU filtering thresholds and case analyses for DPMR are presented and discussed in (Sec. VI-B). Results with different crop margin ratios and case analyses for UFO are presented and discussed in (Sec. VI-C).

A. Overall Results

Tab. II shows the test-set mIoU for Mask R-CNN and DPMR. Tab. III shows the test-set mIoU for U-Net and UFO. There are several key observations:

- Among the four tested methods, Mask R-CNN achieved the highest mIoU on average and for each category.
- DPMR achieved a similar mIoU to Mask R-CNN, but did not show the expected improvements.
- UFO outperformed U-Net across all body parts and on average, demonstrating its effectiveness.
- Comparing Mask R-CNN and U-Net, Mask R-CNN significantly outperformed U-Net.

Further discussion of the overall results is provided in (Sec. VII).

Model	Carapace	Flipper	Head	Average
Mask R-CNN	**0.865**	**0.848**	**0.814**	**0.842**
DPMR	0.862	0.825	0.806	0.831

TABLE II: Comparison of mIoU Results for Mask R-CNN and DPMR

Model	Carapace	Flipper	Head	Average
U-Net	0.818	0.608	0.594	0.673
UFO	**0.848**	**0.689**	**0.672**	**0.737**

TABLE III: Comparison of mIoU Results for U-Net and UFO

B. Results of Dual Proposal Mask R-CNN (DPMR)

In our experiments, we assessed the performance of Mask R-CNN and compared it to the results of Dual Proposal Mask R-CNN (DPMR) with different intersection acceptance threshold as described in (Sec. III-D).

IoU Distribution and Mean IoU Analysis:

As shown in Fig. 5 and Fig. 6, the mean IoU increases with higher thresholds. While lower thresholds allow for more high-IoU values, they also introduce a larger number of low-IoU values, which ultimately lowers the overall mean IoU. This effect is particularly notable for the Flipper and Head regions, where a lower threshold results in a more pronounced presence of low-IoU predictions.

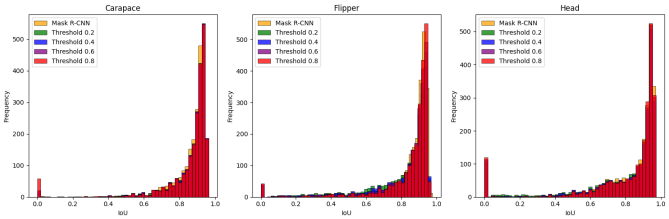


Fig. 5: IoU Distribution across thresholds for Mask R-CNN and DPMR for Turtle, Flipper, and Head.

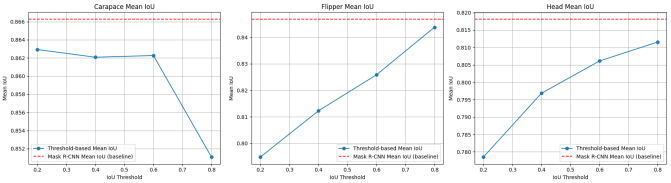


Fig. 6: Carapace, Flipper, and Head Mean IoU at increasing thresholds.

For the Carapace, which initially has a relatively high IoU, increasing the threshold excessively can lead to the exclusion of some true positive detections, as shown by the sharp drop in mean IoU at a threshold of 0.8. This observation suggests that setting an optimal threshold is crucial to balancing high IoU values without compromising true positive retention. Based on this analysis, we determine that a threshold of 0.6 offers the best trade-off, as it maximizes mean IoU without a significant loss of true positives. To further illustrate the selected threshold's effect, Fig. 7 provides a comparison of the IoU distributions for Mask R-CNN and DPMR06, highlighting the improvements in high IoU frequencies across different body parts of the turtle.

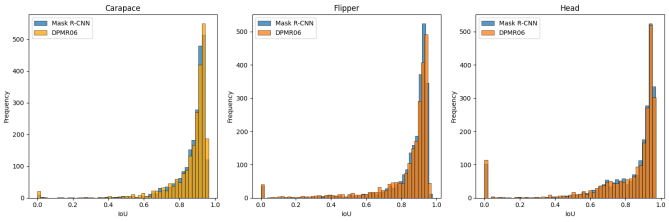


Fig. 7: Comparison of IoU distributions for Mask R-CNN and DPMR06.

False Positive Rate (FPR) Analysis:

The FPR for the Carapace at lower thresholds exhibits minimal variation and aligns closely with the baseline Mask

R-CNN value, as illustrated in Fig. 8. However, a notable increase in FPR occurs at the threshold of 0.8. This surge may be attributed to instances where the turtle's overall size is relatively small within the image; in such cases, an elevated threshold likely filters out correctly detected regions, artificially inflating the FPR for the carapace.

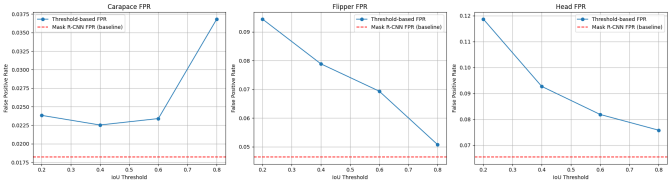


Fig. 8: False Positive Rate (FPR) Analysis for Mask R-CNN baseline and DPMR at various thresholds.

Good Case and Bad Case Analysis for DPMR:

For the good case analysis of DPMR, as illustrated in Fig. 9a and Fig. 9b, DPMR demonstrates notable improvements over Mask R-CNN in accurately segmenting smaller structures, specifically the flipper. In both cases, DPMR effectively reduces false positives that Mask R-CNN mistakenly includes due to background noise and complex underwater lighting conditions.

In Fig. 9a, DPMR captures the flipper's shape accurately, excluding extraneous regions that Mask R-CNN incorrectly labels as part of the flipper. This precision is achieved through DPMR's dual proposal mechanism, which helps focus on true object boundaries rather than surrounding distractions. Similarly, Fig. 9b highlights DPMR's capability to retain fine details, even in challenging lighting where Mask R-CNN struggles. The model preserves the flipper's structure without blending surrounding pixels, demonstrating DPMR's superior boundary adherence.

These examples affirm DPMR's enhanced ability to minimize false positives and maintain structural integrity in smaller object segmentations, emphasizing its robustness in complex visual environments.

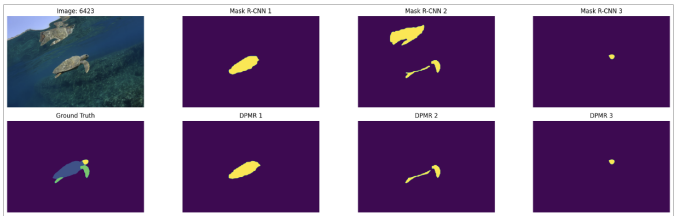
C. Results of U-Net on Focus (UFO)

As shown in Table III, the results of our UFO method showed significant improvements in mIoU values compared with the native U-Net model across all 3 categories. mIoU values for the carapace, flipper, and head category increased by 3%, 13%, and 13%, respectively. UFO gave an average mIoU increase of 9% across all categories.

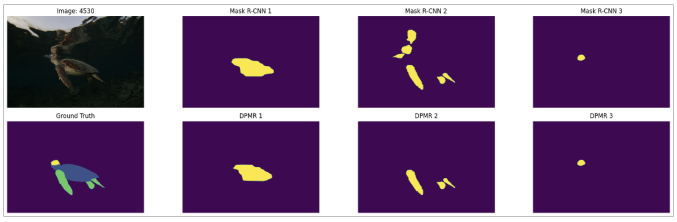
UFO Mean IoU vs Margin Ratio:

In our experiments, we assessed the performance of U-Net on Focus (UFO) using different margin ratios for the bounding box in our cropping block. Note that higher margin ratios correspond to a lesser zooming-in on a turtle in an image.

As shown in Fig. 10, the mIoU of UFO decreases with higher margin ratios, indicating that the more we zoom in on a turtle, the better the performance. We see that as margin ratios increase, hence reducing the zoom factor, mIoU results converge towards those of the original U-Net model.



(a) DPMR Good Case 1: Successfully filtered out the reflection



(b) DPMR Good Case 2: Successfully filtered out the reflection

Fig. 9: Comparison of DPMR and Mask R-CNN on two good case examples. The first column is the image and ground truth mask. From the second column, the first row is the predicted mask from Mask R-CNN, and the second row is the predicted mask from DPMR, for carapace, flippers and head.

IoU Distribution for UFO and U-Net:

The IoU distribution of UFO with the best margin ratio (r_01) has been compared with the IoU distribution of U-Net in Fig. 11. We observe that UFO is performing better than U-Net overall, and that UFO has significantly fewer cases where predictions have missed certain body parts altogether (i.e., UFO plots have a lower count of cases with an IoU of 0).

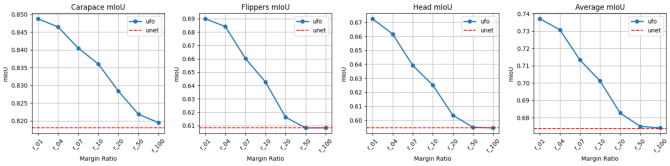


Fig. 10: Carapace, Flipper, Head, and Average Mean IoU for different bounding box margin ratios.

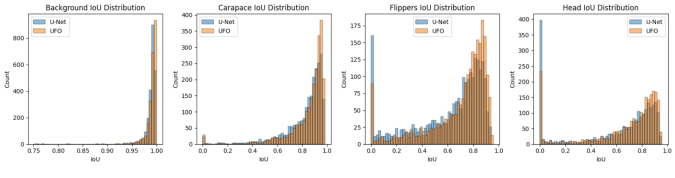
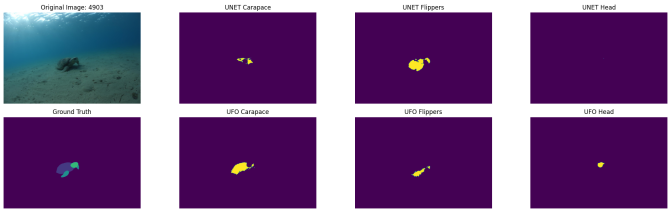


Fig. 11: IoU Distribution for U-Net and UFO for Background, Carapace, Flipper, and Head categories.

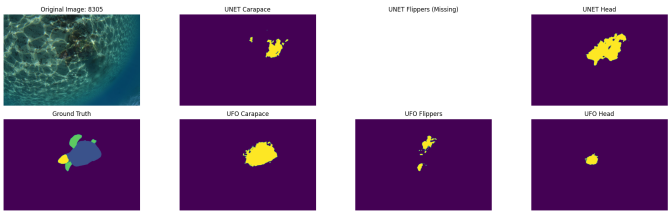
Good Case Analysis for UFO:

UFO was able to label individual turtle body parts much more successfully than U-Net, indicating that the UFO architecture achieved exactly what it intended to. As illustrated in Fig. 12a and Fig. 12b, U-Net struggles to differentiate between

the carapace, head, and flippers, often getting them mixed up, blending them together, or sometimes even missing them altogether. UFO, on the other hand, gives much more accurate segmentation predictions for these parts.



(a) UFO Good Case 1: More accurate prediction on small turtle



(b) UFO Good Case 2: More accurate prediction on small turtle (U-Net predicts no flippers)

Fig. 12: Comparison of UFO and U-Net on two good case examples. The first column is the image and ground truth mask. From the second column, the first row is the predicted mask from U-Net, and the second row is the predicted mask from UFO, for carapace, flippers and head.

VII. Discussion

A. DPMR vs. Mask R-CNN

Compared to the base Mask R-CNN, our DPMR model did not achieve the expected improvement in performance; instead, the mIoU metric showed a slight decline of 1% (Table II). One possible explanation is that the addition of an extra category (whole turtle) during training may have negatively impacted the model's ability to accurately segment the individual body parts. We leave the validation for this statement to future work.

Despite the slight drop in performance, we have also observed that as the threshold for the post-prediction filter increased, the False Positive Rates (FPR) for the flippers and heads decreased correspondingly (Fig. 8). This demonstrates the overall effectiveness of our post-processing strategy, which draws upon the spatial hierarchy between masks that is unique to body parts segmentation tasks and serves the purpose of eliminating false positives.

B. UFO vs. U-Net

Our UFO model sees a significant improvement in performance over the standard U-Net, achieving an average mIoU that is 9% higher (Table III).

This enhanced performance is mainly attributed to the UFO's two-stage inference design, which incorporates a turtle-focused cropping process. By applying cropping with the predicted first-stage inference results and re-inference on the turtle-focused images, the model can produce more focused pixel representations of the turtle's body parts, enabling it to capture finer detailed features that are often overlooked in the original U-Net model. This strategy ensures that the model not only adapts to variations in turtle size across images but also improves its capability to make more precise predictions for smaller body parts. Consequently, UFO's dual-stage design enhances both the accuracy and efficiency of segmentation, making it a more robust solution for detailed body part recognition in sea turtles.

C. Mask R-CNN vs. U-Net

The comparison between Mask R-CNN (Table II) and U-Net (Table III) reveals that Mask R-CNN demonstrates superior performance in handling complex scenes, particularly due to its dual-branch architecture that distinctly separates object detection and pixel-wise segmentation. This structural distinction allows Mask R-CNN to first localize the object and then focus on refining the segmentation within the identified region, which is especially advantageous in environments with background noise and varying lighting conditions, as frequently encountered in underwater images of sea turtles. U-Net, by contrast, utilizes a one-stage structure that performs segmentation directly on the image without object localization, making it more susceptible to misclassifications in noisy scenes. Therefore, the dual-branch approach of Mask R-CNN offers a more robust framework for detailed and accurate segmentation of specific body parts against complex backgrounds, enhancing its applicability for refined tasks such as sea turtle body part segmentation.

VIII. Conclusion

We have introduced two segmentation models that are specifically tailored for the task of segmenting sea turtle body parts. Our proposed models, the Dual Proposal Mask R-CNN (DPMR) and the U-Net on Focus (UFO) address key limitations of existing semantic segmentation methods for sea turtles. The DPMR Model leverages spatial hierarchy between the whole turtle and its body parts, effectively addressing bad cases and avoiding misclassifying areas outside the turtle. One area for potential improvement is to train DPMR separately for whole-turtle and body-part detection, allowing it better to capture the detailed features of each target without redundancy.

On the other hand, UFO involves two-step inference in which the final segmentation is performed on the turtle with the background cropped off, enhancing the model's overall segmentation performance, especially for photographs with small turtles. These tailored approaches not only improve segmentation accuracy over the baseline models but also demonstrate the importance of incorporating domain-specific knowledge into model design.

To enhance future research in the segmentation of small parts of objects, particularly within specific domains like wildlife conservation, we recommend extending the focused approach used in our UFO model to other, more advanced architectures, such as Deeplabv3 [5]. This integration could

leverage the model's ability to isolate relevant features and reduce background noise, especially in conservation-focused tasks. In summary, our research emphasizes the critical role of tailored model designs in advancing monitoring efforts in wildlife conservation.

REFERENCES

[1] Lukáš Adam, Vojtěch Čermák, Kostas Papafitsoros, and Lukas Picek. Seaturtleid2022: A long-span dataset for reliable sea turtle re-identification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7146–7156, 2024.

[2] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647, 1994.

[3] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[5] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, June 2022.

[7] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

[8] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[11] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[14] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[16] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[17] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.