Comparative Analysis of Computer Vision Techniques for Sea Turtle Anatomy Segmentation

Chon Kanchand
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
z5274457@ad.unsw.edu.au

Vincent Huang
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
z5457479@ad.unsw.edu.au

Chunhao Lu
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
z5353746@ad.unsw.edu.au

James Lu
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
z5417290@ad.unsw.edu.au

Yunfei Hu
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
z5125817@ad.unsw.edu.au

***Abstract*—This paper provides an objective comparison of 5 deep learning image segmentation models on image segmentation task for sea turtles' body parts identification. The image segmentation models included in this study are Fully Convolutional Network (FCN) [2], Deeplabv3+ [3], PP-LiteSeg[9], Mask R-CNN [4], and U-Net [5]. We trained the models on Kaggle's SeaTurtleID2022 [1] and evaluated the model using IoU (Intersection over Union) metric with both the overall mean IoU (mIoU) and per class IoU (head, flipper, carapace, and background) assessed. Our Mask R-CNN model suffers from unreasonably slow training due to its object detection origin compared to other models and was only trained on 10% of the dataset for 10 epochs while other models were trained on 70% of the dataset for 20 epochs. The Mask R-CNN therefore cannot be objectively compared to other models in this paper. After assessing the experimental results, the best performing model is PP-LiteSeg, achieving a mean IoU of 0.8214. U-Net achieved a mean IoU of 0.8161, FCN achieved a mean IoU of 0.7069, Deeplabv3+ achieved a mean IoU of 0.6129 and Mask R-CNN achieved a mean IoU of 0.502. These quantitative results alongside qualitative analysis resulted in recommending the PP-LiteSeg model method. However, despite being the best model for this study, The PP-LiteSeg model still has space for future improvements such as increasing training epochs, inclusion of additional data augmentation techniques, multiscale feature extraction, or training on a larger dataset.**

I. Introduction

In recent years, computer vision techniques have become crucial for ecological studies, especially with the advent of modern image segmentation techniques, it has enabled researchers to identify, track and monitor distinct animals within their natural habitats. This study focuses on image segmentation of anatomical features of sea turtles, specifically separating the subject into its head, flippers, and carapace. The use of precise image segmentation assists in automating the tasks of identification, behavior analysis and population monitoring, enabling researchers to avoid labor intensive tasks of manually analyzing photographed specimens. However, with factors of background complexity, differing lighting conditions, and varying sea turtle poses, achieving high segmentation accuracy requires the examination of multiple computer vision methods to identify the most optimal solution.

The dataset used for this study, SeaTurtleID2022 from Kaggle, contains 8729 photos of 438 unique sea turtles over 13 years in 1221 encounters. The suitability of this data set for image segmentation tasks is detailed by the dataset author [1]. In this study, we aim to leverage five deep learning models—Fully Convolutional Network (FCN) [2], Deeplabv3+ [3], PP-LiteSeg[9], Mask R-CNN [4], U-Net [5]—to tackle the image segmentation task on this dataset. By altering these models to specifically operate on the SeaTurtleID2022 dataset, we aim to comparatively assess their performance, strengths, and limitations, deducing their suitability for this specific dataset, with detailed analysis of the model results to determine their respective strengths and shortcomings. This study aims to identify methods that efficiently and reliably segment the head, flippers, and carapace of sea turtles, aiding in deducing the most suitable models for not only this dataset, but other similar datasets. Additionally, by identifying the most effective models, and the processes required to make the model functional with the SeaTurtleID2022 dataset, future improvements to the model to improve segmentation accuracy and performance can be deduced.

II. Literature Review

Various deep learning models have been developed to tackle segmentation tasks. This study utilizes Mask R-CNN, Fully Convolutional Network (FCN), Deeplabv3+, PP-LiteSeg, and U-Net, comparing and assessing their capabilities for image segmentation.

Mask R-CNN, introduced by He et al., extends the Faster R-CNN framework to perform instance segmentation by adding a branch for predicting segmentation masks [4]. This additional mask branch operates parallel to the bounding box recognition, allowing Mask R-CNN to not only detect objects but also generate high-quality segmentation masks. The model uses a region proposal network (RPN) to identify regions of interest, followed by a two-stage process that first classifies and refines bounding boxes and then produces a pixel-level mask for each instance [8]. Mask R-CNN's ability to handle instance segmentation—differentiating multiple occurrences of the same class—makes it particularly valuable for applications requiring fine-grained differentiation, such as in crowd scenes or wildlife monitoring.

The Fully Convolutional Network (FCN) is one of the pioneering deep learning architectures designed specifically for semantic segmentation [2]. Introduced by Long et al., FCN replaces traditional fully connected layers with convolutional layers, enabling pixel-wise prediction across entire images in a single forward pass. This architecture transforms classification networks into fully convolutional ones, allowing them to output segmentation maps instead of single class labels. However, due to its limited ability to capture minute details and contextual information, FCN often struggles with objects of varying scales and intricate boundaries, leading researchers to explore architectures that can overcome these limitations [5].

Deeplabv3+ is an advanced segmentation model that was further developed based on Deeplabv3, integrating an encoder-decoder structure and atrous convolutions to improve segmentation accuracy and detail [3]. Atrous convolution (or dilated convolution) allows the model to capture multi-scale contextual information without losing resolution, which is particularly beneficial for complex scenes with objects of different scales. The encoder-decoder structure of Deeplabv3+ refines the segmentation by recovering lost spatial information, enhancing the boundaries of segmented objects. Chen et al. demonstrated the effectiveness of Deeplabv3+ in semantic segmentation tasks across various datasets, showing significant improvements over traditional convolutional approaches in terms of accuracy and boundary sharpness [6]. The model's architecture, optimized for high-level semantic information and detail preservation, makes it highly suitable for complex tasks like medical and satellite image segmentation [7].

PP-LiteSeg is a lightweight model for real-time semantic segmentation tasks, which was proposed by PaddlePaddle, by using efficient backbone networks and lightweight decoder modules, it can run on limited resources devices and to provide high segmentation accuracy, [9].

The U-Net architecture, proposed by Ronneberger et al., is a fully convolutional network specifically designed for image segmentation [5]. U-Net incorporates a unique encoder-decoder structure with skip connections, which help retain spatial information by connecting high-resolution feature maps from the encoder to the decoder. This structure significantly enhances the model's ability to capture fine details, making it highly effective for segmenting small and intricate structures. The skip connections in U-Net address the spatial context limitations of FCN, enabling it to produce sharper object boundaries and perform well in tasks requiring high-resolution predictions.

III. METHODS

A. *Mask R-CNN*

Built upon the architecture of Faster R-CNN, an object detection model, detailed in Fig. 1, Mask R-CNN introduced instance segmentation in addition to bounding boxes and class labels to the output [4]. In this model, the input image is fed into the convolutional network backbone – we have chosen ResNet-50-FPN network as the backbone – to extract feature maps. The feature maps are also passed through Region Proposal Network (RPN), to generate Regions of Interest (RoI), where the RoIAlign() function is used to extract fixed-size feature maps for each region of interest (RoI) while preserving spatial alignment. The aligned feature maps for each RoI are then split into two branches, classification and bounding box regression branch for object classification and detection, and mask prediction branch –the task we are assessing in this study [4]. We have chosen Mask R-CNN as one of the methods to assess since it has proven to be high accuracy and robust with several types of images. RoIAlign() also results in highly accurate placement of segmentation masks, which would result in high IoU (Intersection over Union) result.

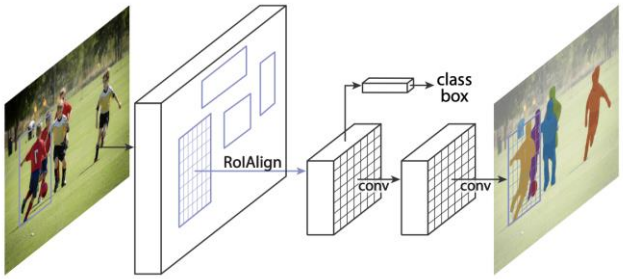


Fig. 1. Mask R-CNN Architecture

B. *Fully Convolutional Network (FCN)*

In this project, we have selected the Fully Convolutional Network (FCN) model for semantic segmentation tasks. This choice is motivated because it is pivotal in computer vision, and it is simple yet efficient in converting classification to dense prediction [2].

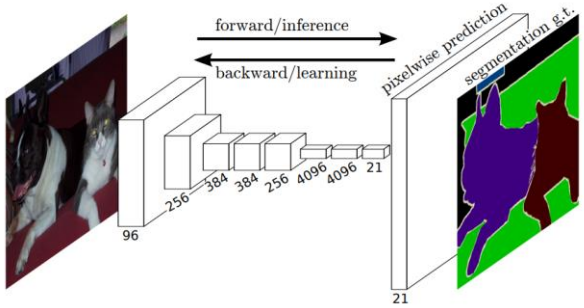


Fig. 2. FCN Architecture

As detailed in Fig. 2, to convert classification to dense prediction, the FCN uses convolutional layers instead of fully connected layers, which makes flexible input size to the network possible and could generate maps of corresponding dimensions. By upsampling the low-resolution feature map back to the original image size through deconvolution, the FCN generates a full-resolution segmentation map. At the same time, to enhance the capability for precise position, high-level semantic information at various levels is combined with low-level spatial details with jump connections.

C. *DeepLabv3+*

For sea turtle body part segmentation, we implemented DeepLabv3+, a state-of-the-art semantic segmentation architecture. This choice was motivated by DeepLabv3+'s unique ability to encode multi-scale contextual information by probing the incoming features with filters and pooling operations at multiple rates and multiple effective fields-of-view, while simultaneously capturing sharper object boundaries by gradually recovering the spatial information [3]. This combined capability is particularly crucial for our task, as different turtle parts (head, flippers, and carapace) exhibit varying scales and require precise boundary delineation. The architecture employs ResNet50 pretrained on

ImageNet as the backbone network for robust feature extraction, where the low-level features are preserved for later refinement of segmentation boundaries.

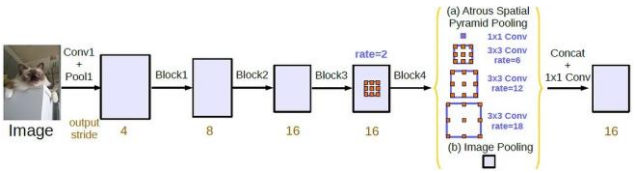


Fig. 3. DeepLabv3+ Architecture

A key component of our implementation is the Atrous Spatial Pyramid Pooling (ASPP) module. As Chen et al. demonstrated in their DeepLabv3+ architecture in Fig 3, the ASPP module applies several parallel atrous convolutions with different rates to capture the contextual information at multiple scales [3]. We implement this using parallel atrous convolutions with rates of 6, 12, and 18, which proves essential for handling the scale variations between turtle parts. The decoder module then fuses the high-level semantic features from ASPP with low-level features through skip connections, enabling precise boundary delineation. To enhance model performance, we developed a combined loss function that incorporates cross-entropy loss for pixel-wise classification, Dice loss for handling class imbalance, and Focal loss for focusing on challenging examples during training.

D. PP-LiteSeg

The selection of PP-LiteSeg is driven by its balance between performance and accuracy, by using the encode-decoder architecture and a combination of three modules, detailed in Fig. 4: Flexible and Lightweight Decoder (FLD), Unified Attention Fusion Module (UAFM) and Simple Pyramid Pooling Module (SPPM), it successfully achieved the goal [9].

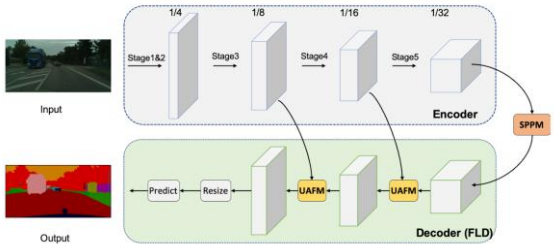


Fig. 4. PP-LiteSeg Architecture

In semantic segmentation, unlike traditional encoder-decoder, the FLD has a special advantage that it could increase the spatial dimensions while decreasing the number of channels in decoding. This advantage improves efficiency and lets the FLD have more room to adjust its complexity, which is the key to balance accuracy and performance [10]. Also, by using spatial and channel attention mechanisms to enhance feature representations, the UAFM improves accuracy in a minimal computational cost. By using SSPM, the PP-LiteSeg improves segmentation accuracy with contextual aggregation for real-time networks with minimal computational cost [9].

E. U-Net

The core of the U-Net architecture is derived from the fully convolutional network (FCN)[5], and it is optimized by adopting a unique U-shaped structure shown in Fig. 5.

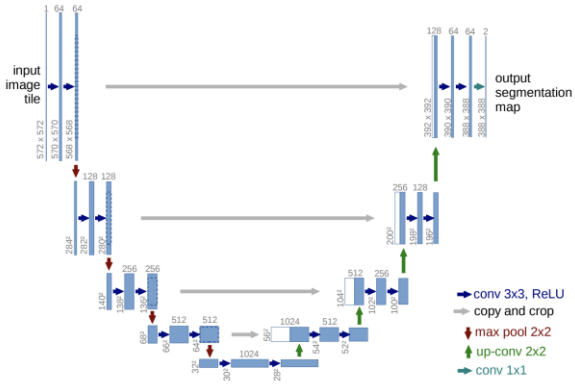


Fig. 5. U-Net Architecture

Its structure enables it to effectively segment images with blurred edges. A symmetric encoder and decoder form its unique U-shaped structure, with the encoding performed in the downsampling path to extract features through convolutional layers and pooling layers. It uses 2x2 max pooling operation and then uses two convolution operations with a convolution kernel of 3*3 with each convolution being followed by a rectified linear unit (ReLU) [11]. During downsampling it reduces the spatial dimension and gradually captures high-level semantic information. Then it is decoded in the upsampling path to restore spatial information through deconvolution layers or upsampling operations, and while upsampling, the skip connections are used to directly connect the encoder's feature map to the corresponding layer of the decoder to achieve fine-grained pixel-level reconstruction [11]. The skip connection mentioned above passes low-level features, such as edges and textures, directly to the decoder to avoid information loss – helping to capture more details and effectively segment images with blurred edges [11]. We chose this model for image segmentation in this assessment as it combines the multi-scale information of the encoder and decoder and can effectively retain the segmentation characteristics of edge and texture features at the pixel level, all while requiring a relatively smaller data set for effective implementation of image segmentation.

IV. Experimental Results

A. Mask R-CNN Results

For the implementation of Mask R-CNN segmentation model, we prepared a COCO dataset formatted by matching images with their annotated masks and bounding boxes via the provided “annotations.json”. We excluded data without valid annotations. The data loader utilizes configurable batch size, set as 4, with shuffling enabled for the training set.

The maskrcnn_resnet50_fpn_v2 model from PyTorch was utilized, allowing us to leverage the pretrained weights to suit our model. The model uses ResNet50 with Feature Pyramid Network as the backbone. SGD optimizer is used with 0.9 momentum and 0.0005 weight decay to enhance generalization performance.

Despite only training the model on a 10% subset of the SeaTurtleID2022 data set, for 10 epochs at a learning rate of 0.005, the model still required an exorbitant amount of time to train due to its object detection origin. From a visual inspection of the results, the qualitative product of applying the model on test images results in decent segmentation mask, as seen in Fig. 6. However, it is required to note that the displayed results only detail the successful segmentation

masks with similar classes and is not an accurate indicator of the performance of the model evidenced by the calculated average IoU values.

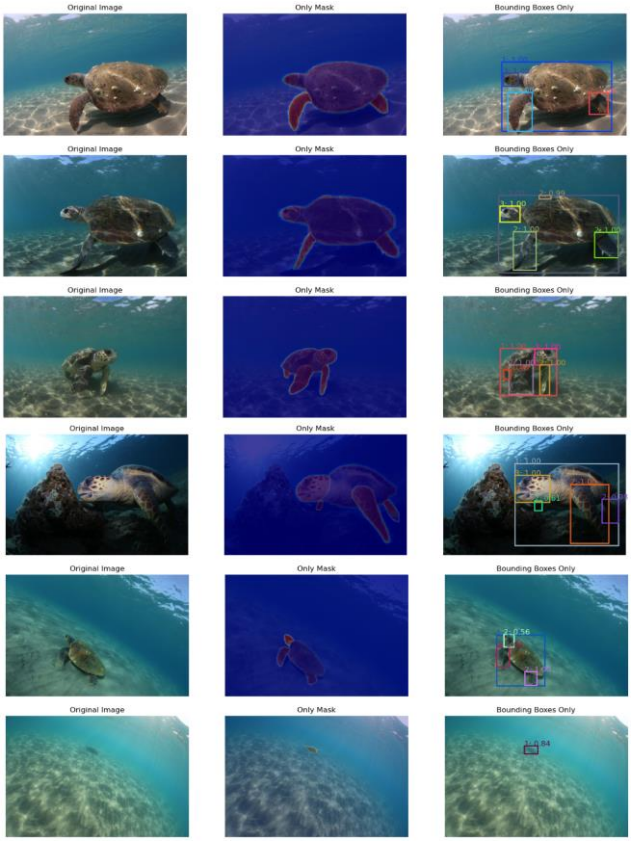


Fig. 6. Successful Mask-RCNN Segmentation Masks

Losses and IoU are tracked for evaluation after the training process. The evaluation metrics include the mean Intersection over Union (IoU) across all classes—Background, Head, Carapace, and Flipper—as well as individual IoU scores for each class to assess both overall and class-specific performance. After 10 epochs, the model achieved a mean IoU of 0.502.

B. Data Set Preparation for FCN, DeepLabv3+, PP-LiteSeg and U-Net Models

Prior to applying the computer vision techniques for image segmentation, the SeaTurtleID2022 data set has been converted into a COCO formatted dataset by matching images with their annotations via the provided "annotations.json", and we exclude those data without valid annotations. The dataset is split into training (70%), validation (15%), and test (15%) sets using the dataset's provided open-set metadata CSV. This ensures the model is evaluated entirely on unseen images, resulting in a more realistic performance evaluation than closed-set or random splitting. Also, any invalid images incorrectly referenced in the metadata CSV file have been identified and removed. To enhance model generalization, data augmentation techniques such as flips, rotations, and color adjustments from the Albumentations library are applied to the training data, while validation and test data undergo minimal transformations to preserve evaluation integrity.

C. Fully Convolutional Network (FCN) Results

For training the FCN, we only update the decoder and attention layers so we could optimize the efficiency of computation, by using a CombinedLoss() function with an alpha parameter, different loss components could be balanced, and the optimization employs AdamW with weight decay to prevent overfitting and a cosine annealing scheduler with warm restarts for dynamic learning rate adjustment. Besides, the loss and Intersection over Union (IoU) metrics are tracked for both training and validation sets during training as seen in Fig. 7, where the graphs are used for evaluation of the model.

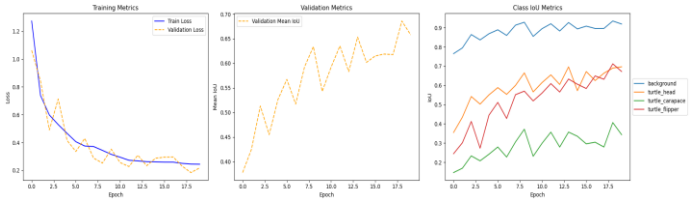
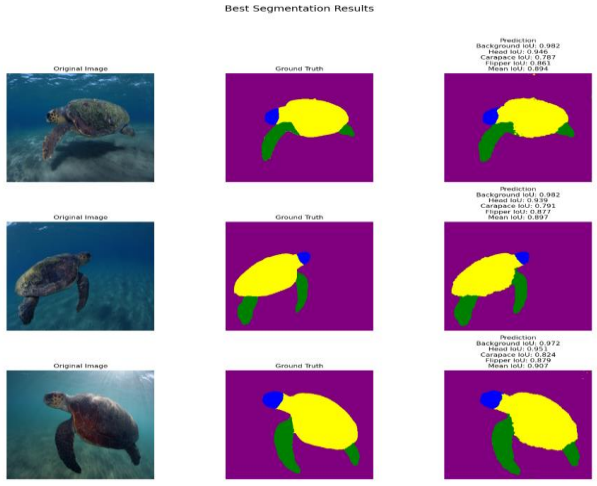


Fig. 7. Training Metrics, Validation Metrics and Class IoU Metrics for FCN

The evaluation metrics include the mean IoU across all classes (Background, Head, Carapace, and Flipper) and per-class IoU scores to evaluate overall and class-specific performance. After training, the model that has the best performance will be evaluated on the test set to obtain unbiased performance estimates, and visual analyses are conducted by comparing model predictions with ground truth on selected six samples, including three the best and three worst cases shown in Fig. 8, to qualitatively assess segmentation accuracy and identify areas for improvement.



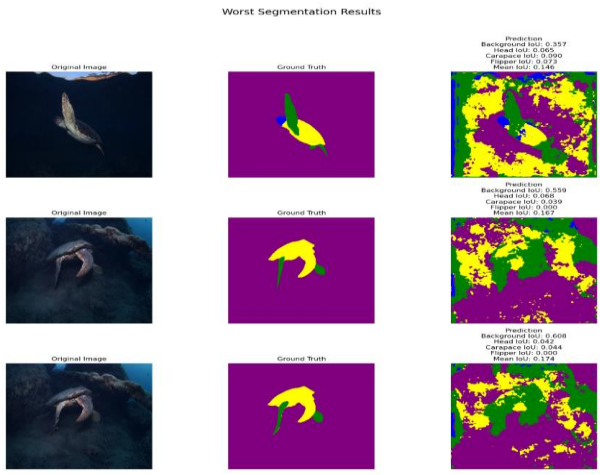


Fig. 8. Three Best and Three Worst Model Predictions for FCN

The FCN demonstrates effective learning during training, with a steady decrease in training loss and an increase in test Mean IoU, which reached 0.7069 after 20 epochs, indicating improved segmentation performance on unseen data.

D. DeepLabv3+ Results

The implementation involves dedicated pre-processing steps to enhance efficiency and performance of the model. The TurtleSegDataset class implements mask pre-computation and caching mechanisms, significantly reducing training time by eliminating redundant mask generation during iterations. The data loading pipeline utilizes PyTorch's Data Loader function with pinned memory and appropriate worker numbers, optimizing GPU memory transfers and CPU utilization. We implemented a sophisticated data augmentation pipeline using the Albumentations library, which provides efficient augmentations. The augmentation strategy combines spatial transformations (random resized crop, rotation) with appearance modifications (brightness, contrast), and elastic deformations, working together to improve model robustness and prevent overfitting.

The training process employs several optimization techniques to improve model training performance. Our combined loss function integrates Cross-entropy, Dice, and Focal losses with an alpha parameter of 0.5, creating a balanced optimization objective. Each component serves a specific purpose: Cross-entropy provides stable gradient signals, Dice loss handles class imbalance effectively, and Focal loss focuses on hard examples. The AdamW optimizer (learning rate: 1e-4, weight decay: 1e-4) works in concert with a ReduceLROnPlateau scheduler, which dynamically adjusts the learning rate based on validation performance, implementing a robust training strategy that adapts to plateaus in model performance.

For performance evaluation, we developed a comprehensive suite of testing tools. The evaluate_model() function calculates both per-class and mean IoU metrics, while our visualization functions (visualize_best_predictions and visualize_worst_predictions) provide insights into model behavior across different scenarios. These tools work together to offer both quantitative metrics and qualitative analysis, with graphs of the model's training performance and IoU throughout training detailed in Fig. 9. During training, we employed an early stopping mechanism with a patience of 7 epochs, working in sync with our learning rate scheduler to prevent overfitting while ensuring optimal model convergence.

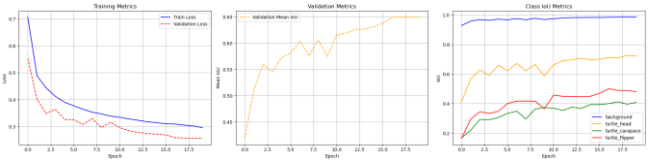


Fig. 9. Training Metrics, Validation Metrics and Class IoU Metrics for DeepLabv3+

A key innovation in our testing framework is the visualize_sample_predictions() function, which generates detailed performance visualizations including color-coded segmentation masks and per-class IoU scores, displayed in Fig. 10. This visual feedback mechanism proved invaluable for understanding model behavior and identifying areas for improvement. The integration of these various components from efficient data handling to comprehensive evaluation tools creates a robust and informative experimental framework that enables thorough analysis of model performance and behavior.

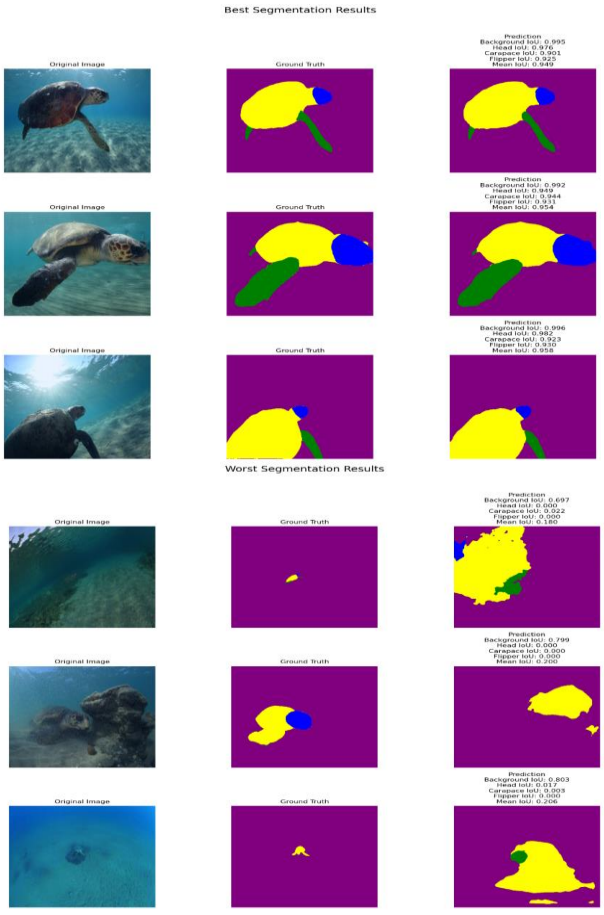


Fig. 10. Three Best and Three Worst Model Predictions for DeepLabv3+

The model achieved a mean IoU of 0.6129 on the test set after 20 epochs of training. Breaking down the per-class performance, the model showed varying effectiveness across different turtle parts: background segmentation achieved the highest IoU of 0.9836, followed by turtle head segmentation at 0.7035, turtle flipper at 0.4170, and turtle carapace at 0.3477.

The training metrics reveal interesting patterns in the model's learning process. The training loss steadily decreased from an initial value of approximately 0.7 to 0.3, while the validation loss stabilized around 0.25, indicating good convergence without significant overfitting. The validation

mean IoU indicated consistent improvement over the training period, suggesting relatively stable learning dynamics.

E. PP-LiteSeg Results

We initialized PP-LiteSeg with three input channels and progressively increasing layer dimensions ([16, 32, 64, 128, 256]) to capture complex spatial features while maintaining a manageable model size. A segmentation-specific loss function is used to enhance the model's ability to distinguish between different classes, like background, head, carapace, and flipper, to achieve a better performance. Also, we use the AdamW optimizer with weight decay to prevent overfitting, and a ReduceLROnPlateau scheduler adjusts the learning rate based on validation loss improvements.

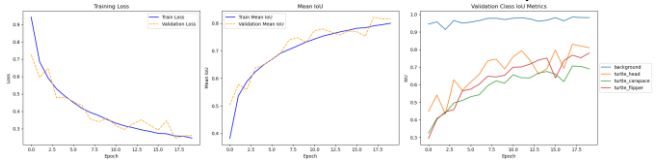


Fig. 11. Training Metrics, Validation Metrics and Class IoU Metrics for PP-LiteSeg

We trained the model with 20 epochs, with losses and IoU metrics tracked and visualized for both training and validation sets, saving the best-performing model for evaluation. The visualized training loss, mean IoU and validation class are displayed in Fig. 11. The IoU metrics are accessed overall and per class, to provide insights into segmentation accuracy and model consistency.

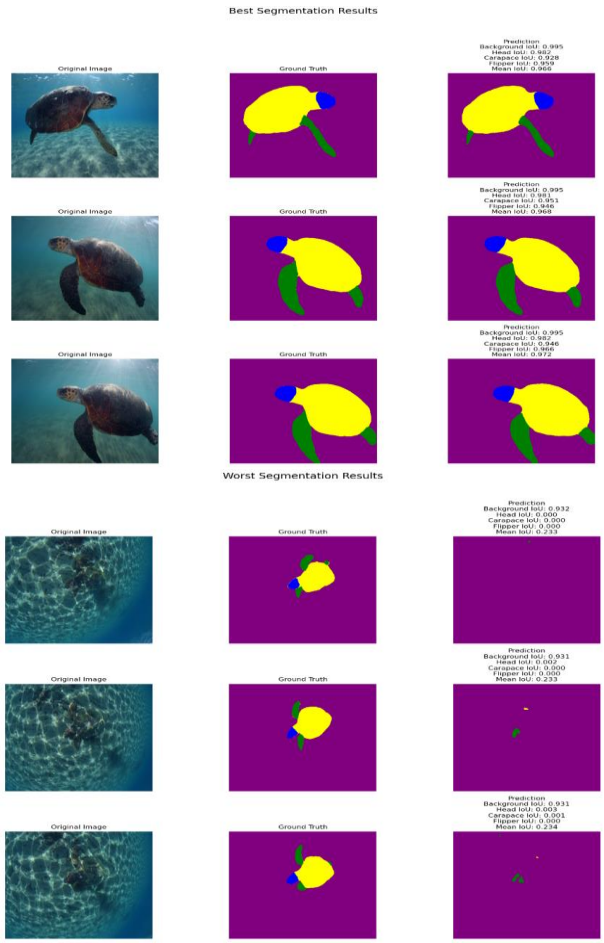


Fig. 12. Three Best and Three Worst Model Predictions for PP-LiteSeg

In evaluation, the model that has the best performance will be evaluated on the test set and calculate the mean IoU and class-specific IoUs, then we display sample predictions alongside ground truth masks, annotating IoU metrics, and highlighting the ground truth on selected six samples, including three the best and three worst cases, seen in Fig. 12, allowing for qualitative analysis of the model.

The PP-LiteSeg model effectively learns the segmentation task, evidenced by steadily decreasing training and validation losses. After 20 epochs, the Mean IoU reaches around 0.8214, indicating good generalization to the test set, and subsequently, has the highest mean IoU out of all the models.

F. U-Net Results

For the initialization of the U-Net model involves defining several key parameters n_channels, n_classes, and bilinear to correctly configure the U-Net method. Firstly, n_channels was defined as three indicating the input consists of three RGB images. Then, n_classes is set to the #define NUM_CLASSES, which in this model was four representing four classes – background, head, carapace, and flippers. The bilinear variable was defined as True, specifying the use of bilinear interpolation during upsampling.

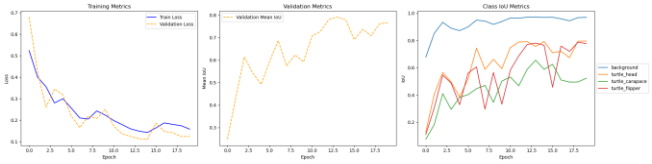


Fig. 13. Training Metrics, Validation Metrics and Class IoU Metrics for U-Net

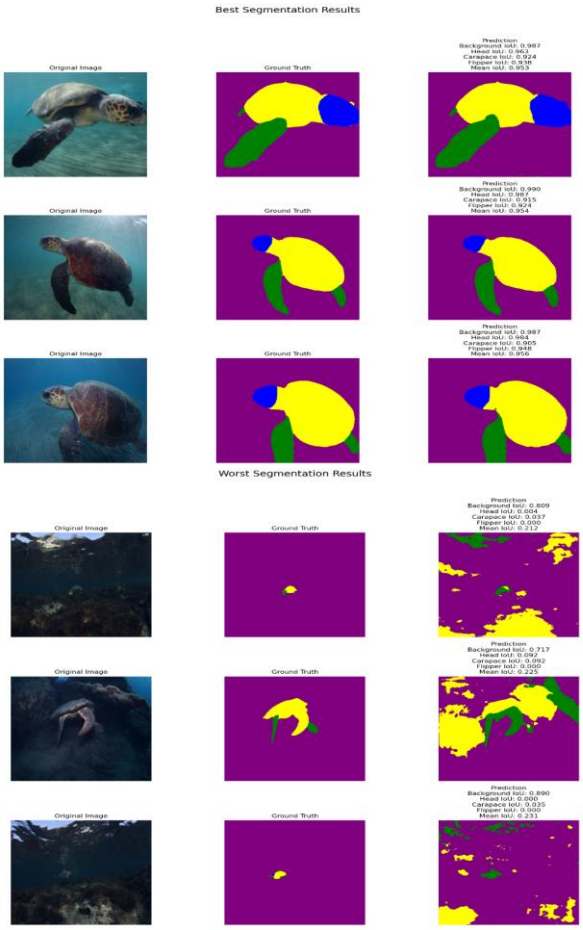


Fig. 14. Three Best and Three Worst Model Predictions for U-Net

For training the U-Net model, we evaluate the IoU of each class and the IoU of all classes to obtain the data for the model throughout its training history, which is then graphed in Fig. 13. By training the model for 20 epochs, we incrementally decreased the training loss to improve its accuracy.

Additional to quantitative analysis of model via IoU calculations, we obtained the model segmentation results for the three best and three worst mean IoUs for qualitative analysis, as detailed in Fig. 14. By gathering the images with the highest mIoU and lowest mIoU, it allows analysis into the characteristics that negatively impacted the model's ability to segment the image. The model achieved a mean IoU of 0.8161 on the test set.

G. *Summary of Mean IoU for All Models*

TABLE I. MEAN IOU TABLE FOR ALL MODELS

Mean IoU	Mask-RCNN	FCN	DeepLabv3+	PP-LiteSeg	U-Net
Background	0.810	0.9477	0.9836	0.9857	0.9800
Head	0.611	0.7632	0.7035	0.8356	0.8382
Carapace	0.442	0.4331	0.3477	0.7018	0.6773
Flipper	0.145	0.6838	0.4170	0.7623	0.7688
Combined	0.502	0.7069	0.6129	0.8214	0.8161

V. DISCUSSION

A. *Mask-RCNN*

The Mask R-CNN achieves an average IoU score of 0.502. The per class IoU are as follows: Head IoU 0.611, Flippers IoU 0.145, Carapace IoU 0.442, and Background IoU 0.810. Due to the model requiring an extensive amount of time to train, it was only trained for 10 epochs and the results reflect that. The model performs marginally worse than other models in this study which were trained for 20 epochs. The model struggles most with flippers class with an IoU score of only 0.145. The flippers class is more complex because there are multiple flippers in one image, and some are also obscured by the carapace. In images where the sea turtle is far away, the model failed to distinguish distinct parts of the sea turtle and only output one mask and one bounding box. Even with Mask R-CNN advanced RoIAlign() that enhances mask placement accuracy, with training on only a subset of 10% of the original data and the use of only 10 epochs, the model did not produce satisfactory segmentation results.

This study does not allow for a direct comparison of Mask-RCNN with the other models as it was only trained on 10% subset of the SeaTurtleID2022 data set. However, it can be deduced that the model is functional in producing qualitatively correct segmentation masks as evidenced by results detailed in Section IV.A. This proves promising results for future development of this method as training on only 10% has yielded relatively high mIoUs, with further optimization of the training method for increasing training speeds and training on a larger percentage of the data set, the Mask-RCNN model can be refined to be a viable image segmentation method.

B. *Fully Convolutional Network (FCN)*

After 20 epochs, the mean IoU of the FCN is 0.7069, ranking third in mean IoU. Its per-class IoU scores show excellent segmentation of the background (average IoU is 0.9477), solid accuracy for the head and flipper classes (average IoU is 0.7632 and 0.6838 respectively), but lower accuracy for the carapace (IoU around 0.4331). These mean IoU values stem from obscurity within the image (low contrast, and so forth), where additional terrain in the testing class, unexpected lighting image conditions and other unexpected noise all affect the accuracy of the model. Further analysis of the results in Fig. 15, evidences the model experiences difficulty when producing segmentation masks for images with dark backgrounds and backgrounds with terrain interference. Qualitative analysis of the first image containing the brightly lit turtle against the dark background evidences the turtle in the image is mostly segmented correctly, however it mistakenly identifies the background as turtle anatomy, thus majorly diminishing the prediction's mean IoU.

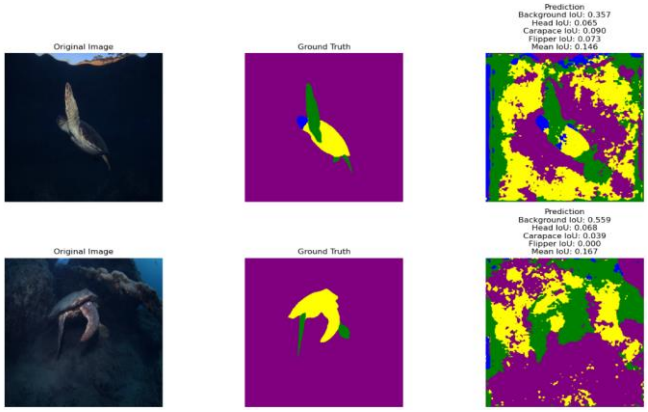


Fig. 15. Worst IoU Segmentation Predictions for FCN

Examining Fig, 7, the training graphs show the FCN struggled with improving the turtle carapace mean IoU, whilst the head and flipper mean IoUs all increased as more epochs were trained. This evidenced FCN weaknesses relating to carapace detection, which could be explained by the images containing similar carapace shaped or colored objects, thus negatively affecting the effectiveness of the mIoU. The head and flippers were less affected likely due to less objects having similar characteristics.

C. *DeepLabv3+*

The DeepLabv3+ model ranks last in terms of mean IoU. Whilst the model demonstrates exceptional performance in background segmentation (IoU 0.9836) and robust head detection (IoU 0.7035), likely due to the distinctive features of these regions, the relatively lower performance in carapace (IoU 0.3477) and flipper (IoU 0.4170) segmentation reveals the challenges the model faces when handling complex geometric patterns and variable appearances of turtle anatomy. Despite ASPP's multiscale processing capabilities, the significant size variations in flippers and carapaces across images remain challenging, and the similar textures between these regions often lead to boundary confusion.

Qualitative analysis of Fig. 16 exemplifies the model's shortcomings, with the first and third images depicting the turtle above the dark terrain, obscuring the outline of the turtle, thus resulting in segmentation predictions that completely misidentify the outline of the turtle. The second image (IoU values of head, carapace, and flipper are zero) further supports the model's difficulty with similar textures as it completely misidentified the turtle, instead suggesting the terrain beside is the turtle.

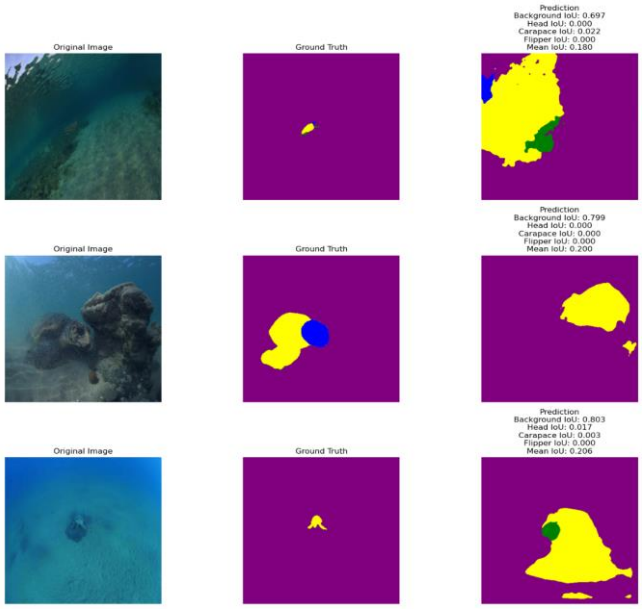


Fig. 16. Worst IoU Segmentation Predictions for DeebLabv3+

Furthermore, the class IoU metrics over training epochs as seen in Fig. 9, where background and head segmentation quickly reached high performance, while carapace and flipper segmentation improved more gradually, suggesting these classes require more sophisticated feature learning approaches.

D. PP-LiteSeg

On the test set, the PP-LiteSeg model achieves the highest mean IoU of 0.8214 compared to other models in this study. This model has the highest average background IoU of 0.9857 and average carapace IoU of 0.7018. It has the second highest average head IoU of 0.8356 and average flipper IoU of 0.7623. Thus, the model quantitatively performs the best among the four models, FCN, Deeplabv3+, U-Net, and PP-LiteSeg, in comparison.

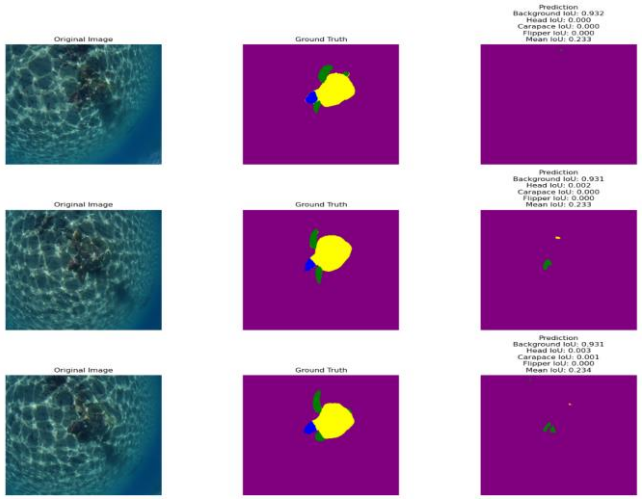


Fig. 17. Worst IoU Segmentation Predictions for PP-LiteSeg

As evidenced by Fig. 17, images with the worst IoU results share the same issue of struggling with complex lighting. The PP-LiteSeg model struggles in the complex scenes containing sunlight casting a ripple effect over the entire image. These challenging lighting conditions may cause the model to not recognize the turtle as it blends into the background, leading to low or zero IoU scores for the head, carapace, and flippers.

As shown in the graphs of Fig. 11, the model improves detection of the head, carapace, and flipper at roughly the same rate, indicating that it successfully improves the mean IoU as each epoch is trained. This indicates that this model correctly identifies an increasingly larger amount of segmentation predictions as more epochs are trained.

E. U-Net

The model achieved a mean IoU of 0.8161 on the given dataset, indicating the model has the second-best segmentation performance overall when compared to all other models. The background class has the highest IoU, reaching 0.9800, indicative of the background's nature of occupying large parts of the image and usually contrasts the turtle, improving segmentation results. The IoU of the head class of 0.8382, flipper class of 0.7688, and carapace class of 0.6773, are mostly the highest second highest mean IoU values in comparison to the other models.

By analyzing the model prediction results in Fig. 18, we can discover that when the turtle is occluded or in a complex background, as well as in images with substantial changes in posture or lighting conditions, the carapace and flippers are easily confused, or even the segmentation effect is poor. However, in comparison to worse performing models it was observed to have slightly higher mean IoU values, where the difference in IoU values may be attributed to the number of epochs being only 20. The improved accuracy of the U-Net model can be attributed to its nature of providing accurate training despite using smaller data sets, unlike FCN and DeepLabv3+ which typically benefit from larger data sets.

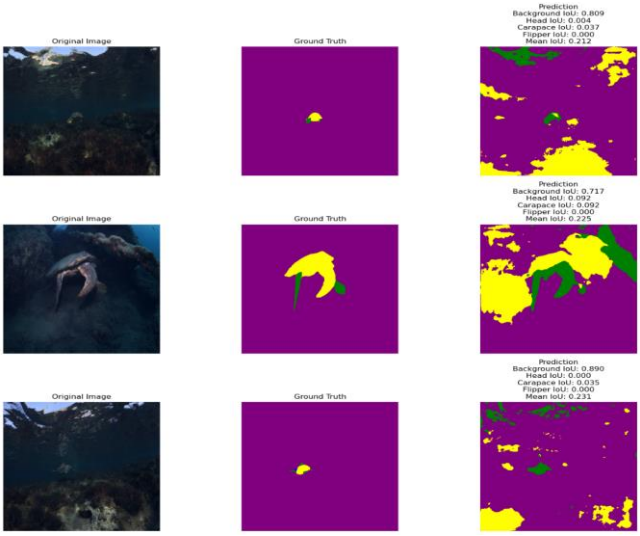


Fig. 18. Worst IoU Segmentation Predictions for U-Net

As shown in the graphs of Fig. 13, the model improves detection of the head and flipper at similar rates with some sudden mean IoU changes during the initial few epochs. However, the carapace IoU decreased in the last few epochs, which may be due to model overfitting or a lack of training epochs.

F. Leading mIoU Method

Observing the mean IoU results detailed in Table 1, it can be concluded that PP-LiteSeg has the best performance for

our specific training variables—batch size of four, learning rate of 0.0001, and 20 total epochs. This outcome of PP-LiteSeg having the best model performance may be attributed to several factors.

The minimal number of epochs used for training favors PP-LiteSeg as it is optimized to perform well while the other models usually require more epochs to stabilize and achieve more accurate segmentation masks. This is evidenced by the graphs detailing the mIoU of each class in Fig. 7, 9, 11, 13, where PP-LiteSeg is shown to have the smoothest curve for all classes and all turtle anatomy classes converge around the same time.

The batch size of four used in the dataset would typically negatively affect more complex models, however with PP-LiteSeg's simpler architecture, small batch sizes still result in acceptable convergence rates.

Using the low learning rate of 0.0001 enables all models to have a more stable and gradual learning curve, reducing overfitting or underfitting.

Utilization of three main lightweight modules FLD, UAFM, and SPPM is what enables the PP-LiteSeg to have the best performance out of the four different models for comparison—FCN, DeepLabv3+, PP-LiteSeg and U-Net.

G. Future Improvements

Limitations of the PP-LiteSeg model are still apparent, especially with the discussion of results in Section V.E, where incorrect segmentation predictions were still identified. The mean IoU for the head, carapace, and flipper can also still be improved not only for PP-LiteSeg but also other methods. Examining the other methods using models FCN, DeepLabv3+, and U-Net, it can be determined they all show different weaknesses as evidenced by the results. This indicates that some models may have an advantage in detecting certain image scenarios better than others.

However, based purely on quantitative results, future improvements should be focused on the “best model” which is PP-LiteSeg, deduced from mean IoU comparisons. Specific improvements to the model can be implemented by increasing training epochs, inclusion of additional data augmentation techniques, multiscale feature extraction, or training on a larger dataset.

Increasing the number of training epochs could result in a lesser gradient near the end of the IoU training graph in Fig. 11, as the model will have more epochs to train and converge. This prevents inadequate training durations or any overfitting, thus enhancing the model's ability to generalize a multitude of image scenarios.

According to the results of PP-LiteSeg in Fig. 17, the most challenging predictions for the model involved images with underwater lighting. Additional data augmentation techniques address the poor performance for samples with challenging lighting conditions, with augmentation of the training dataset by varying the exposure of training images, overlaying similar lighting conditions to other training images, and altering the contrast, the model is exposed to more samples with varying lighting conditions. This ensures the shortcomings of the PP-LiteSeg model are minimized to create a more accurate and efficient method.

Utilizing multiscale feature extraction similar to DeepLabv3+’s Atrous Spatial Pooling (ASPP) could also improve the segmentation of head, carapace, and flippers. Using multiscale feature extraction for filtering allows more detailed separation and training of each segmentation mask i.e., using large filters specifically for the turtle’s carapace and small filters for both the head and flippers, helping to improve the current carapace mIoU of 0.7018 and flipper mIoU of 0.7623.

Future research and application of these improvements to the PP-LiteSeg model could majorly improve its segmentation accuracy and performance, thus increasing the effectiveness of image segmentation for not only the SeaTurtleID2022 dataset, but also other similar datasets.

VI. Conclusion

The deep learning image segmentation models assessed in this study exhibit varied performance. Mask R-CNN, trained for fewer epochs and lesser proportion of the dataset due to extensive training time for bounding boxes, underperforms compared to other models. The FCN model efficiently segments the head from the background but struggles with the carapace and flippers due to background similarities and shape inconsistencies, making it suitable for head-background separation. DeepLabv3+ excels in background and head segmentation as well, with area of improvements suggested such as jump connections and adaptive loss weighting, to improve flipper and carapace segmentation. With the second highest mean IoU result, U-Net effectively segments most of the turtle body parts but struggles with complex scenes and low-light condition, resulting in a comparatively lower IoU for carapace and flipper class. Finally, the best performing model, PP-LiteSeg effectively segments all of the turtle body parts and reliably captures spatial relationships but falters in scenes with challenging lighting condition. The highest performing model is PP-LiteSeg, followed by U-Net, FCN, and Deeplabv3+. Mask R-CNN exhibits the poorest performance but cannot be compared to other models due to its substandard training. Each model exhibits distinct strengths and areas for improvement. To further improve upon our most effective model, PP-LiteSeg, we suggest further hyperparameter tuning, increasing training epochs, inclusion of additional data augmentation techniques, multiscale feature extraction, or training on a larger dataset.

References

[1] K. Papafitsoros, L. Adam, V. Čermák, and L. Picek, “SeaTurtleID: A novel long-span dataset highlighting the importance of timestamps in wildlife re-identification,” arXiv (Cornell University), Jan. 2022, doi: https://doi.org/10.48550/arxiv.2211.10307.

[2] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” arXiv (Cornell University), Nov. 2014, doi: https://doi.org/10.48550/arxiv.1411.4038.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in Eur. Conf. Comput. Vis., 2018, pp. 801-818.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” arXiv.org, Jan. 24, 2018. https://arxiv.org/abs/1703.06870v3

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Int. Conf. Med. Image Comput. Comput.-Assisted Intervention, 2015, pp. 234-241.

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," arXiv preprint arXiv:1706.05587, 2017.

[7] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1667-1683, Aug. 2019.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, Jun. 2017.

[9] J. Peng et al., “PP-LiteSeg: A Superior Real-Time Semantic Segmentation Model,” arXiv.org, Apr. 06, 2022. https://arxiv.org/abs/2204.02681

[10] M. Fan et al., “Rethinking BiSeNet For Real-time Semantic Segmentation,” arXiv.org, 2021. https://arxiv.org/abs/2104.13188 (accessed Nov. 09, 2024).

[11] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” arXiv (Cornell University), May. 18. 2015, doi: https://arxiv.org/abs/1505.04597v1