COMP9517 Group Project
Sea Turtles Image Segmentation

Darren Pradhan z5421298
Murtaza Pakawala z5405719
Sunit Sunit Ravi z5436640
Thi Huyen Trang Tran z5054327
Tung Nhi Tran z5509226

***Abstract*—This study explores and compares computer vision techniques to automate segmentation of key body parts of sea turtles in underwater images, using the SeaTurtleID2022 dataset. Accurate segmentation is vital for wildlife monitoring and research, traditionally reliant on manual analysis. We implemented three deep learning-based architectures: U-Net, R2U-Net, and DeepLabv3, known for their effectiveness in complex image segmentation tasks. U-Net, with its encoder-decoder structure and skip connections, is adept at capturing spatial details. R2U-Net, a recurrent residual variant, enhances this framework by retaining context through recurrent layers and residual connections, improving segmentation accuracy for smaller features. DeepLabv3, leveraging atrous convolutions and multi-scale feature extraction through its ASPP module, excels at handling diverse object sizes and scales. Our results indicate that while all models successfully segment the primary turtle structure, R2U-Net performs best for smaller, detailed parts, especially the head. Computational limitations required training on a subset of 4,000 images, which may have impacted performance. Future work could involve training on the full dataset and resolving technical issues to implement Mask R-CNN, which holds promise for instance segmentation of individual body parts in complex natural scenes.These findings demonstrate the potential of deep learning to improve efficiency in ecological research.**

***Index Terms*—iimage segmentation, U-Net, R2U-Net, DeepLabv3, Mask R-CNN, sea turtles, computer vision, SeaTurtleID2022 dataset, wildlife monitoring.**

I. Introduction

A. Task Summary

This project aims to develop and compare computer vision methods to segment key body parts of sea turtles—the head, flippers, and carapace—from underwater photographs. Automated segmentation is critical for applications in wildlife monitoring and research, where traditionally these tasks have relied on manual labour. By leveraging advancements in computer vision, we can reduce human effort and improve efficiency in tracking and studying individual turtles over time.

The primary methods explored in this project are U-Net and R2U-Net, both of which fall under deep learning-based architectures rather than traditional computer vision techniques. U-Net is well-known for its effectiveness in biomedical and environmental image segmentation, while R2U-Net, a residual recurrent variant of U-Net, enhances performance by incorporating residual and recurrent layers. These architectures provide a foundation for segmentation accuracy and model flexibility, making them suitable for complex datasets like SeaTurtleID2022.

B. Main Issues

The project faces several significant challenges which include the following:

- Variability in Images: The dataset images exhibit a wide range of environmental conditions, such as differences in lighting, water depth, and visibility. These factors impact image quality and can obscure key features.
- Changes Over Time: Turtles may undergo natural changes in appearance, such as scarring or changes in pigmentation, while some identifiers, like the shape of facial scales, remain consistent. This variability adds difficulty to reliable re-identification and segmentation.
- It is challenging to distinguish between the turtle's various body parts in many of the images due to severe noise or blurriness. The fact that the photos were taken from different viewpoints makes precise identification even more difficult. Important physical features of the turtle are obscured or not fully visible in some of the pictures. The analysis is made harder by the presence of barnacles on the turtle's skin and shell, which can be confused with the body's natural features. Consequently, it becomes more challenging to draw accurate segmentations based on the photos.

C. Approach

Our approach involves testing and comparing the performance of U-Net and R2U-Net architectures, both of which are inspired by recent advancements in segmentation techniques. Specifically:

- U-Net and R2U-Net were selected due to their established success in handling image segmentation tasks with complex features, such as those present in medical imaging, which closely parallel the challenges of sea turtle identification.
- Various configurations of U-Net and R2U-Net were tested, examining aspects like residual layers in R2U-Net, which allows for better feature learning and handling of complex variations in turtle images. We evaluate each model's performance based on Intersection over Union (IoU) to quantify segmentation accuracy across body parts.

D. Dataset

The SeaTurtleID2022 dataset, sourced from Kaggle, is one of the largest and longest-spanned datasets available for sea turtle identification, consisting of 8,729 images of 438 individual loggerhead turtles observed over a 13-year period [1]. This dataset contains rich metadata, including:

- *Annotations*: Each image includes segmentation masks for the head, flippers, and carapace, facilitating detailed model training for body-part-specific segmentation.
- *Temporal and Environmental Variation*: Images were captured under varied natural conditions in Laganas Bay, Greece, with changes in lighting, water depth, and camera equipment over the years, making it an ideal dataset for robust segmentation model evaluation. [1]

Exploratory Data Analysis (EDA)

Our EDA provides insight into the SeaTurtleID2022 dataset, focusing on the distribution and frequency of photographs, yearly encounters, and observation patterns of individual turtles. The following visualizations highlight key aspects:

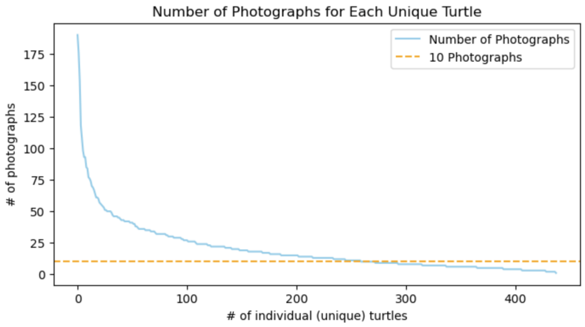

Fig. 1. Number of photographs for each unique turtle

The graph 1 illustrates the distribution of the number of photographs across individual turtles. The dataset is highly imbalanced, with some turtles having significantly more images than others. Most individuals have around 10 photographs or fewer, though some have upwards of 100 images, which can aid in robust model training and evaluation.

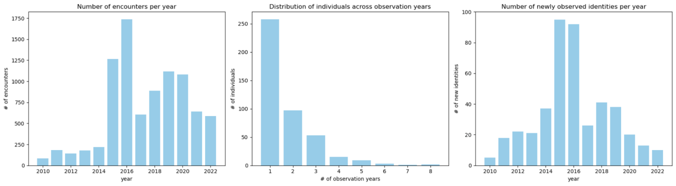

Fig. 2. The encounters and their distribution

- *Number of encounters per year*: the graph Fig.2 shows the annual encounter count for turtles in the dataset. Encounter frequency varies across years, with the number increasing significantly around 2016. This variation reflects changes in data collection intensity over time, which could impact the dataset's temporal diversity and model training effectiveness.
- *Distribution of individuals across observation years*: The second bar graph in Fig.2 displays the number of unique turtles observed over multiple years. A large proportion of turtles were observed only in a single year, while a smaller subset was observed across several years. This distribution allows for studying individual identification over time and handling challenges associated with changes in appearance.
- *Number of newly observed identities per year*: The third bar plot in Fig.2 presents the number of newly encountered turtles each year, peaking around 2016. The addition of new identities each year helps in assessing the model's ability to generalise to previously unseen individuals and is relevant for open-world re-identification tasks.

II. Literature review

Image segmentation is a fundamental task in computer vision, especially valuable for applications requiring precise object delineation in complex backgrounds, such as wildlife monitoring and biomedical imaging. U-Net, R2U-Net, DeepLabv3 and Mask R-CNN are among the most widely used architectures for segmentation, each offering unique strengths for various segmentation challenges.

U-Net , introduced by Ronneberger et al. (2015) in [2], has become a landmark model in semantic segmentation, particularly in biomedical and environmental domains. U-Net's architecture consists of a contracting path to capture features and an expansive path to enable precise localisation. This U-shaped structure is enhanced by skip connections, which link layers in the encoder to corresponding layers in the decoder, ensuring that important spatial details are preserved across the network. This design makes U-Net highly effective in segmentation tasks that demand high-resolution outputs. Its success has led to numerous adaptations and widespread adoption in tasks where both accuracy and efficiency are essential. [2]

Since the introduction of U-Net, its effectiveness in image segmentation has driven continuous advancements and improvements in the field. As a result, numerous variations of the original U-Net architecture have been developed. One such variation is the R2U-Net.

R2U-Net , developed by Alom et al. in [3], builds on the U-Net architecture by introducing recurrent and residual layers. The recurrent layers in R2U-Net allow the network to retain contextual information across multiple passes, while the residual blocks address the vanishing gradient problem, enabling deeper and more robust learning. These enhancements make R2U-Net particularly suited to tasks requiring high accuracy in complex environments, as it captures finer image details and handles small-scale features more effectively than standard U-Net. As a result, R2U-Net has been successful in segmentation tasks involving intricate structures. [3]

DeepLabv3 , introduced by Chen et al. in [5], enhances semantic segmentation by employing atrous (dilated) convolutions to capture multi-scale contextual information without increasing computational cost. The model utilises an Atrous Spatial Pyramid Pooling (ASPP) module, which probes convolutional features at multiple scales, effectively capturing both local and global context. This design enables DeepLabv3 to achieve high accuracy in segmenting objects at various scales, making it particularly effective in complex scenes with diverse object sizes. Notably, DeepLabv3 achieves this performance without relying on DenseCRF post-processing, simplifying the segmentation pipeline. Its balance of accuracy and efficiency has led to widespread adoption in tasks requiring detailed semantic segmentation, including environmental monitoring and biomedical imaging. [5] [6]

Mask R-CNN Mask R-CNN, introduced by He et al. in [7], extends the capabilities of Faster R-CNN by adding a branch specifically for predicting segmentation masks alongside object detection and classification. This design enables Mask R-CNN to perform instance segmentation, distinguishing between different instances of the same class, which is particularly useful for complex scenes with overlapping objects. Mask R-CNN's ability to produce high-resolution masks while simultaneously identifying and classifying objects makes it highly effective for tasks requiring precise boundary definition and instance-level segmentation. However, its computational demands are substantial, which can be challenging for large datasets without access to high-performance computing resources. [7]

The U-Net, R2U-Net, DeepLabv3 and Mask R-CNN models each provide valuable approaches to segmentation, with varying strengths depending on the specific task requirements. U-Net and R2U-Net excel in tasks requiring precise pixel-level segmentation with moderate computational demands, making them ideal for biomedical and environmental applications. DeepLabv3 stands out for its ability to handle multi-scale contextual information through atrous convolutions and its simplified pipeline, making it highly effective for semantic segmentation in diverse and complex environments. Mask R-CNN is advantageous for instance segmentation, effectively distinguishing individual objects in complex scenes, but requires significant computational resources. Together, these models represent a robust foundation for tackling segmentation tasks, though their performance may vary with the dataset's size and complexity.

III. Methods

A. U-Net Architecture

The U-Net model is widely used for image segmentation tasks, particularly in medical image processing. It combines the strengths of convolutional neural networks (CNNs) and encoder-decoder structures to capture both local and global context information in images.

The architecture of U-Net consists of two primary parts: encoding and decoding paths, often visualized as a U-shaped structure. The encoding path, also known as the contracting path, compresses the input image by reducing its spatial dimensions while increasing the feature depth. Conversely, the decoding path, or expanding path, restores the original spatial dimensions, producing a segmented output.

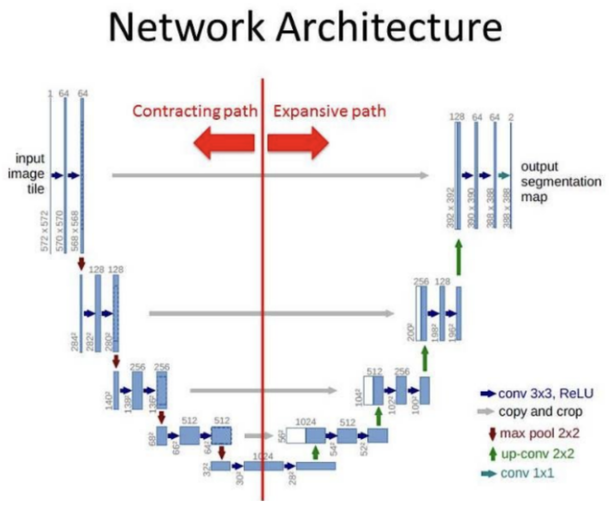

Fig. 3. U-Net architecture with contracting path and expansive path [2]

Encoding Path (Contracting Path)

In the encoding path, U-Net applies multiple convolutional layers with ReLU activation functions followed by max-pooling layers. This series of operations reduces the spatial dimensions (height and width) of the feature maps while increasing the depth. Each convolutional block consists of two consecutive convolutions, each followed by a ReLU activation.

- Convolution + ReLU: Each step in the encoding path involves two convolutions with a ReLU activation function. These convolutions increase the feature representation at each layer.
- Max Pooling: After each convolutional block, max-pooling is applied to reduce the spatial dimensions, allowing the network to focus on higher-level features.

Decoding Path (Expanding Path)

The decoding path mirrors the encoding path, gradually reconstructing the spatial dimensions by up-sampling the feature maps. Each step in the decoding path involves an up-sampling operation, followed by a series of convolutions to refine the segmentation mask.

- Up-Sampling: U-Net uses transposed convolutions or up-sampling operations to increase the spatial dimensions of the feature maps.
- Skip Connections: One unique aspect of U-Net is the inclusion of skip connections between corresponding

layers in the encoding and decoding paths. These skip connections provide additional context information from the encoding path, allowing the model to recover fine-grained details lost during down-sampling.

U-Net's design makes it well-suited for tasks requiring precise localization, such as image segmentation, by preserving spatial details through skip connections and progressively up-sampling features in the decoding path.

B. R2U-Net Architecture

R2U-Net is inspired by the deep residual model, Recurrent Convolutional Neural Network (RCNN), and U-Net. The architecture of R2U-Net and the pictorial representation of the unfolded RCL layers with respect to time-step are shown in Fig.4 [4]. Here $t = 2$ refers to the recurrent convolutional operation that includes one single convolution layer followed by two sub-sequential recurrent convolutional layers.

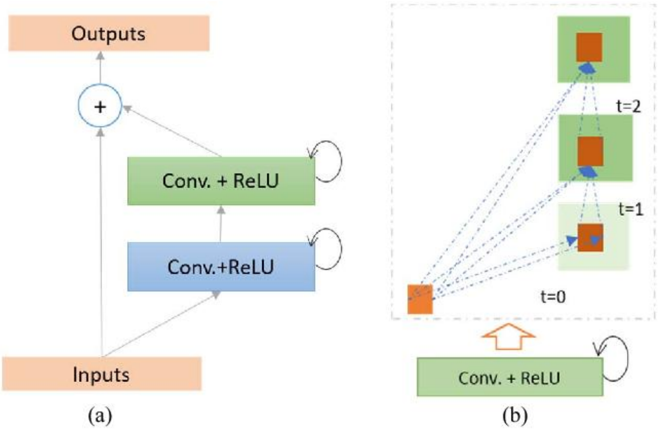

Fig. 4. Recurrent Residual convolutional units (RRCU) and the unfolded recurrent convolutional units for t = 2. [4]

In details, consider x_l as the input sample of the l^{th} layer of the residual RCNN block, a pixel located at (x, y) of the k^{th} feature map in the RCL. Moreover, the output of the network at the time step t is $O^l_{ijk}(t)$. This output is fed to the standard ReLu activation function f, therefore:

$$\mathcal{F}(x_l, w_l) = f(O^l_{ijk}(t)) = \max(0, O^l_{ijk}(t))$$

For R2U-Net, the final output of RCNN units are passed through the residual unit. Assume that, the output of RRCNN blocks is x_{l+1} and computed as:

$$x_{l+1} = x_l + \mathcal{F}(x_l, w_l)$$

Then, x_{l+1} is used for down-sampling and up-sampling layers in the encoding and decoding units of the model.

IV. Experimental Results

We have evaluated the U-Net and R2U-Net models on the dataset SeaTurtleID2022 to show their performances. Each photograph comes with annotations such as identities, encounter timestamps, and segmentation masks of the body parts.

A. The experimental setup

Due to computational limitations, we used 4,000 images (50% of dataset) and resized them to 256x256 pixels. The dataset is split into 3 parts : train, validation and test. The PyTorch framework is utilized for this implementation on a single GPU P100 (which is available to access in Kaggle) with 16GB of RAM.

Loss function

For training the models, the Dice coefficient was deployed as the loss function. It is computed for each class and the overall Dice score was averaged across all classes. The Dice coefficient is defined as:

$$DSC = \frac{2|S \cap T|}{|S| + |T|} = \frac{2|TP|}{|FP| + 2|TP| + |FN|}$$

Mean Intersection over Union (MeanIoU)

In our experiment, MeanIoU was computed as an additional evaluation metric. IoU is a widely used evaluation metric in semantic segmentation tasks that quantifies the overlap between predicted segmentation masks and the ground truth.

$$IoU = \frac{|S \cap T|}{|S \cup T|} = \frac{|TP|}{|FP| + |TP| + |FN|}$$

For a dataset (such as train dataset, validation dataset or test dataset), first of all, we'll compute the meanIoU for each class : *turtle*, *legs*, *heads* and *background*. Then, the final meanIoU (MulticlassIoU) is the average of the IoU scores across all classes:

$$\text{MulticlassIoU} = \frac{1}{4} \sum_{i=1}^{4} \text{meanIoU}_i$$

Training process

The training process was performed over 12 epochs for the U-Net model and 30 epochs for the R2U-Net model. Early stopping was implemented by monitoring the MulticlassIoU on the validation dataset to prevent overfitting and determine the optimal stopping point.

B. The results

The results of our models on test data are shown in the table I.

TABLE I
The Comparison of Two Segmentation Models

Model	**IoU**	**Dice loss**	**IoU of each class**			
			Background	***Turtle***	***Legs***	***Head***
U-Net	0.688	0.231	0.962	0.714	0.509	0.566
R2U-Net	0.739	0.192	0.943	0.654	0.621	0.738

V. Discussion

1) U-Net model: The U-Net model demonstrates solid segmentation capabilities for the turtle's primary body. This is evident from the IoU score for the "Turtle" class, which is approximately 0.7. This relatively high score highlights

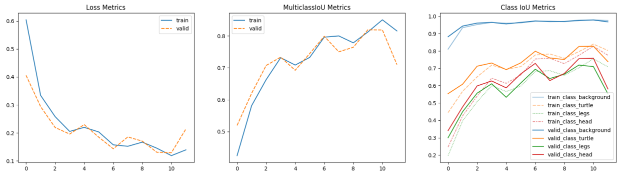

Fig. 5. The loss and IoU of U-Net model

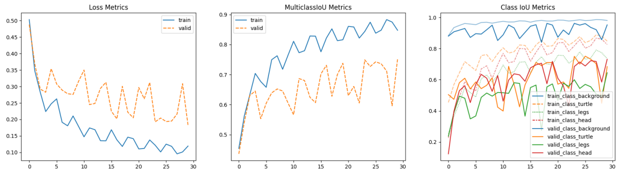

Fig. 6. The loss and IoU of R2U-Net model

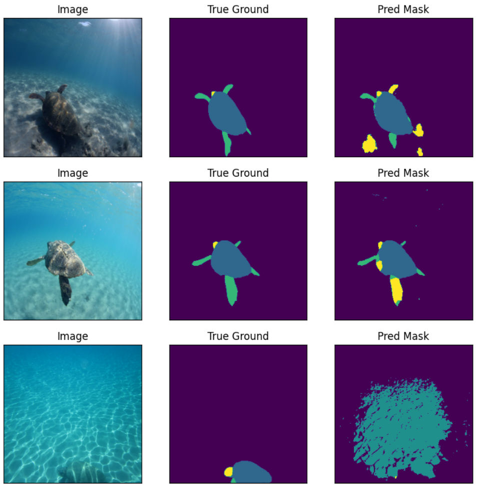

Fig. 7. The segmentation of U-Net.

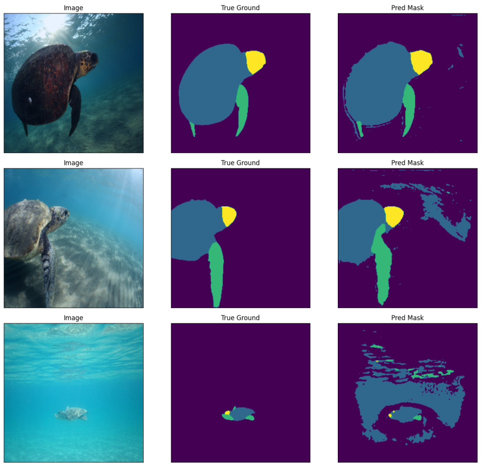

Fig. 8. The segmentation of R2U-Net

the model's ability to accurately segment the turtle's main body, as it is the most distinct part of the animal, often standing out in images due to its recognizable shape and size. However, certain challenges arise when dealing with background elements and finer details.

One common issue with the U-Net model is its tendency to misclassify certain background features, especially rocks, as parts of the turtle's head. This confusion is likely because rocks may have similar textures or colors to parts of the turtle, leading the model to mistake them for the head region. Additionally, the model struggles with images where the background is noisy or when the turtle's body parts, like the flippers and head, are not prominently visible. In such cases, it becomes challenging for the model to clearly identify and segment these smaller or subtler parts.

In conclusion, while the U-Net model performs well in identifying and segmenting the larger, more distinct parts of the turtle, it faces limitations with finer features and challenging backgrounds.

2) R2U-Net model: For the R2U-Net model, the number of output channels in the first block was optimized to identify the most suitable configuration for the model. After experimenting with 16, 32, and 64 channels, it was determined that 16 channels yielded the best performance. This configuration was subsequently used to construct the final R2U-Net model.

The R2U-Net model's segmentation output closely resembles the ground truth, successfully identifying the turtle's primary characteristics in the picture. The model can correctly segment the turtle's body parts, even if the turtle is comparatively small in the picture. This illustrates how the model can manage diverse object sizes and continue to function well at various scales. However, there are a few segmentation issues. For example, the model sometimes produces false positives in the background region by confusing the sea around the turtle with the turtle's body. Additionally, the segmentation of the flippers is not entirely accurate, because the model finds it difficult to accurately distinguish them, leading to some areas being incorrectly classed or combined with nearby regions. The model performs best in segmenting the turtle's head. This is not surprising because the head is typically the most recognizable and well-defined feature of the turtle, which facilitates accurate identification and segmentation by the model. The quantitative results shown in the table, where the MeanIoU for the head component is the highest among all the body parts, are consistent with the extremely accurate head segmentation.

Overall the R2U-Net model works well for segmentation task, there is definitely demand for improvement, especially when it comes to handling tiny parts like the flippers and separating the turtle from the environment when the image is too blurry even for human eyes.

3) Comparison: While both U-Net and R2U-Net demonstrate effective segmentation of the primary structure of the turtle, R2U-Net achieves higher accuracy, particularly in identifying and segmenting smaller, well-defined features like the head. R2U-Net's additional recurrent and residual layers help it retain more context, allowing it to perform better than U-Net on varied scales and complex images, even when the turtle appears smaller. However, R2U-Net also encounters issues with finer details, such as distinguishing the flippers from nearby regions, and sometimes misclassifies parts of the sea as part of the turtle.

In contrast, U-Net performs well in segmenting the main body but has more difficulty with small, detailed parts in noisy backgrounds. Overall, while R2U-Net exhibits more precise segmentation, especially for smaller features, both models face challenges in handling complex backgrounds and subtle details, indicating room for improvement in differentiating smaller body parts and filtering out background noise.

4) Other project Insights:

a) DeepLabv3: In addition to U-Net and R2U-Net, we also explored DeepLabv3 which leverages atrous convolutions and the Atrous Spatial Pyramid Pooling (ASPP) module to capture multi-scale contextual information, as described in the literature. In this project, DeepLabv3+ was implemented using the Segmentation Models PyTorch library with a ResNet-18 encoder pre-trained on ImageNet. This lightweight backbone was selected to balance computational efficiency with segmentation accuracy, accommodating the constraints of limited training time and resources. The model architecture was configured with:

- Input channels: 3 (for RGB images),
- Output classes: 4 (head, flippers, carapace, and background).

To enhance learning, a custom loss function combined Cross-Entropy Loss and Dice Loss, with a weighting factor ($\alpha = 0.3$) to prioritise overlapping regions of the segmented masks. The optimiser employed was Adam with a learning rate of $1x10 - 4$, chosen for its effectiveness in handling complex non-linear loss landscapes. The following loss graph illustrates the model's performance during training, highlighting the effectiveness of the combined Cross-Entropy and Dice loss function.

DeepLabv3's capability for flexible multi-scale feature extraction via ASPP layers and its seamless integration with pre-processing pipelines enables efficient data handling and model adaptability.

Despite progress in implementing the DeepLabv3 model and plotting training and validation loss curves, the project faced limitations in plotting advanced metrics such as Multi-class IoU and Class-wise IoU due to time constraints. These metrics require additional evaluation code and computational resources to iterate over the validation set, compute per-class segmentation performance, and average the results.

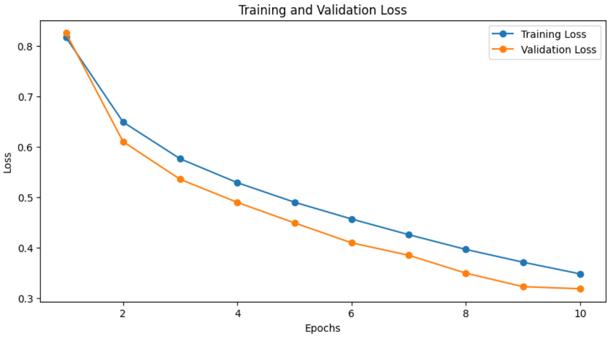

Fig. 9. Training and validation loss of DeepLabv3 model

Future work should focus on expanding metric calculations and optimising the training process. Incorporating IoU metrics will provide deeper insights into the model's performance and allow more comprehensive comparisons with U-Net and R2U-Net.

b) Mask R-CNN: Furthermore, we also explored Mask R-CNN for its advanced capabilities in instance segmentation, which could be advantageous for accurately segmenting the individual parts of the turtle.

The Mask R-CNN model builds upon foundational principles of CNNs, R-CNN, and Faster R-CNN. CNNs serve as the backbone of computer vision, transforming input images into feature maps via convolutional layers, summarising features through ROI pooling, and connecting neurons across layers with fully connected layers.

R-CNN extended this by using selective search to generate approximately 2,000 region proposals per image, applying a CNN to classify these regions. However, the inefficiency of performing CNN operations on numerous proposals led to the development of Fast R-CNN, which uses a Region Proposal Network (RPN) and ROI pooling, significantly speeding up the process. Faster R-CNN further refined this by integrating RPN for region proposal generation.

Mask R-CNN enhances Faster R-CNN by adding instance segmentation capabilities, enabling precise pixel-level classification of objects. It achieves this by predicting four outputs for each candidate: a class label, bounding box coordinates, and a segmentation mask. This functionality makes Mask R-CNN well-suited for tasks requiring detailed segmentation.

For the turtle dataset, Mask R-CNN offers advantages such as semantic segmentation for overlapping turtles, accurately distinguishing their head, body, and flippers. While instance segmentation may not be critical for this task, the model's

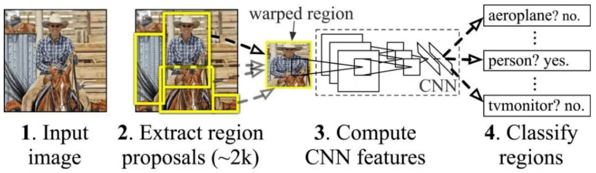

Fig. 10. Concept of R-CNN [15]

efficiency and robustness to variations in size, orientation, and environmental conditions make it ideal for processing the large and diverse SeaTurtleID2022 dataset.

However, due to time constraints, we were unable to resolve technical issues in the code that prevented the model from running successfully. Limited time and resources restricted our ability to investigate these errors fully, including finding the root cause of code errors and managing the increased computational requirements. Future work could focus on troubleshooting and implementing Mask R-CNN, as its potential for precise segmentation, particularly with complex objects in natural environments, makes it a promising model for further enhancing turtle body-part segmentation accuracy.

VI. Conclusion

1) Project Summary: In this project, we developed and compared deep learning methods, specifically U-Net and R2U-Net, to segment key body parts (head, flippers, and carapace) of sea turtles from underwater photographs in the SeaTurtleID2022 dataset. Both models demonstrated the capability to accurately identify the main structure of the turtle, with R2U-Net showing improved performance on smaller, more detailed parts, especially the head. However, challenges remained, particularly in differentiating finer features like the flippers in complex backgrounds and managing instances where background elements were mistaken for parts of the turtle.

2) Future work: While the project achieved its goals, due to limited computing resources, we trained our models on a subset of 4,000 images out of the 8,729 available, which may have contributed to lower overall performance. Expanding the training dataset to include all images could improve model generalisation and segmentation accuracy.

DeepLabv3 was successfully implemented but faced limitations in calculating advanced metrics such as Multi-class IoU and Class IoU due to time constraints, which restricted comprehensive performance evaluation. Addressing these limitations in future work could provide more robust insights into the model's capabilities.

Additionally, Mask R-CNN was not implemented successfully due to unresolved technical issues and computational demands. Overcoming these challenges and fully integrating Mask R-CNN could further improve segmentation precision, particularly for distinguishing complex objects in natural environments.

Future work could also focus on leveraging more robust data augmentation techniques to improve model resilience across different environmental conditions, such as varied lighting and water visibility. Overall, while this project demonstrates promising results, additional time and resources could enable more comprehensive evaluations and improvements, potentially achieving higher accuracy in sea turtle segmentation.

References

[1] Papafitsoros, Kostas, et al. ”SeaTurtleID: A novel long-span dataset highlighting the importance of timestamps in wildlife re-identification.” *arXiv preprint arXiv:2211.10307* (2022).

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

[3] Alom, Md Zahangir, et al. ”Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation.” *arXiv preprint arXiv:1802.06955 (2018).*

[4] Alom, Md. Zahangir et al. “Nuclei Segmentation with Recurrent Residual Convolutional Neural Networks based U-Net (R2U-Net).” *NAECON 2018 - IEEE National Aerospace and Electronics Conference* (2018): 228-233.

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

[6] Chen, Liang-Chieh, et al. ”Encoder-decoder with atrous separable convolution for semantic image segmentation.” Proceedings of the European conference on computer vision (ECCV). 2018.

[7] He, Kaiming, et al. ”Mask r-cnn.” Proceedings of the IEEE international conference on computer vision. 2017.

[8] Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. ”V-net: Fully convolutional neural networks for volumetric medical image segmentation.” 2016 fourth international conference on 3D vision (3DV). Ieee, 2016.

[9] Repo: https://www.kaggle.com/creazyeeeeli/code/acsnansck/edit

[10] Repo: https://github.com/navamikairanda/R2U-Net/tree/main

[11] https://pytorch.org/vision/main/models/mask_rcnn.html

[12] https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html

[13] https://haochen23.github.io/2020/05/instance-segmentation-mask-rcnn.html

[14] https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/

[15] https://viso.ai/deep-learning/mask-r-cnn/