Comparative Analysis of Image Segmentation Techniques for Sea Turtle Identification

Xiang Li., Zhijie Wei, Shukun Chen, Chuansheng Xu, Zengqi Hao

1. Introduction

The purpose of this group project is to develop and evaluate various computer vision techniques for segmenting turtles in underwater photographs. These techniques are applied to the SeaTurtleID2022 dataset. Identifying individual animals from images is essential for wildlife research. Such research includes monitoring population sizes, studying behavioral patterns, and designing effective conservation strategies. Currently, this identifications process is carry out manually by expert. However, manual processing requires significant times and effort, especially for large dataset collected over extended period. To address these limitations, this project employ advanced computer vision technologry to automate the segmentation process, aiming to enhance both accuracy and efficiency.

1.1 motivation

Until now, experts have generally carried out the identification process manually. However, for large data sets collected over a long period of time, the act of manual processing will take a lot of time and effort. In order to improve the accuracy and efficiency of the experiment and solve some limitations, we use advanced computer vision technology to automate the segmentation process

2. Literature Review

Image segmentation is a basic area of computer vision. Its fisrt goal is to label each pixel in an image to distinguish between different objects. This ability is necessary for tasks such as object localization and recognition. Traditional segmentation methods, including K-means clustering and Mean Shift, are simple and widely used but have serious limitations.

1.2 Problem Statement

Choosing a portion of sea turtles for underwater imaging is difficult in the concept of segmentation. Underwater images are quite complex due to their typical conditions, which are lighting that changes, camouflaged objects, distortion caused by water, and unnaturally colored backgrounds which could be the turtles. The SeaTurtleID2022 dataset, consisting of 8,729 images of 438 distinct turtles, which have been collected over 13 years, serves as one of the major assets for sea turtle identification. Every photo has labels for the different parts, for example, heads, shells, and flippers. Given the complexity of the problem, it is essential to assess and develop multiple segregation mechanisms to figure out which is appropriate for the job. This research also seeks to analyze classic techniques of image segmentation for sea turtles and compare them to a new powerful deep learning-based strategy. Examples of the traditional techniques I will be testing include K-means clustering and Mean Shift while the models developed on deep learning, namely U-Net, DeepLabV3, and YOLOv8-segment will also be used. While critiquing these methods, the report will also shed light on their areas of use for wildlife identification cases.

They often fail to achieve the goal in complex environments like underwater scenes.

K-means clustering and Mean Shift are based on using color and texture features. K-means is a convenient way to divide an image into different regions by grouping similar pixels. However, it has some trouble with underwater images where shadows, water distortion, and similar colors. According to Dhanachandra et al. (2015) [1] , K-means improve the

image quality before segmentation when used with contrast adjustments. Also, subtractive clustering can help improve the initial placement of clusters, making K-means more useful. On the other hand, Mean Shift adapts to the data without need a fixed number of clusters. However, it can be slow and heavy, especially when dealing with high-resolution images or detailed backgrounds.

U-Net is different from traditional models, because its architecture was a breakthrough in image segmentation, particularly in medical applications where annotated data is often scarce. As Ronneberger says et al. (2015)[2], U-Net uses a symmetric encoder-decoder structure that allows it to effectively capture both local and global information in an image. Its huge success comes from using data augmentation, which helps train the model even when there are only a few examples. In underwater environments, where lighting and distortion can vary, U-Net has a sense of ability to adapt.

Due to the utilization of attention mechanisms, the convolutional neural networks (CNNs) can be enhanced by concentrating the network on the main objects of an image. As described by Woo et al. (2018) [3], the Convolutional Block Attention Module (CBAM) takes into account both spatial and channel attention to emphasize the significant features. Channel attention is about the "what" that is significant in a picture, and spatial attention is related to the "where" that important features are located. This dual-level method has proved to yield better segmentation even in the most complex scenarios, for example, the underwater condition. CBAM is a compact and effortless add-on for existing CNNs that is good for partial identification of turtles, such as limbs or shell, without long or additional computation. Besides making them higher or wider, the latest breakthroughs in deep learning have gone beyond making architectures stretch. Photo Harmonics like ResNet's residual connections help to quicken the process of training the deep networks that do not suffer from issues like vanishing gradients. He et al. (2016) [4] proposed an image recognition residual learning, which has been shown to be suitable for training extremely deep neural networks and enhancing their performance. Relying solely on CBAM can attain superior results in terms of precision and/or recall when combined with other well-performing models such as ResNet on multiple datasets that comprise complex images, ie. ImageNet and MS COCO. This gets CBAM to the position of the most probable segmentation method for turtles, as the turtle's background is very similar to the turtle and hard to differentiate. Activation functions play acgreat role in the performance of neural networks. Sagarap (2018) [5] suggested that, in contrast to the classic SoftMax activation function, a classification uses the ReLU activation function. However, ReLU is often implemented in the hidden layers, though it has also been demonstrated to be effective in the final layer for some data. This could pave the way for the underwater segmentation experiment, not only because it may allow for faster training, but also provide for similar or even greater accuracy. Such a light-weight alteration can take advantage while using on-device imagery (this is particularly true for underwater drones) since it reduces the memory and processing requirements of the model.

YOLOv8 is an improved version of the original YOLO model, which directly integrates segmentation capabilities into the object detection process. This enhancement makes it highly effective for underwater detection, which is invaluable for our task of recognizing sea turtles underwater. With YOLOv8, we can process our project’s image dataset more efficiently, as it can accurately segment features such as the sea turtle’s head in underwater conditions.

In their 2016 paper, Redmon et al. [6] highlighted the advantages of YOLOv2, which improved speed and accuracy over YOLOv1, making it more suitable for real-time monitoring tasks[7]. YOLOv2 enhanced feature extraction and added batch normalization, allowing for more stable training. YOLOv8 further elevates these capabilities, making it an excellent choice for underwater detection tasks that require accurate object recognition.

Bochkovskiy et al. (2020) [8] introduced YOLOv4, which achieved significant advancements in speed

and accuracy. Compared to YOLOv1, YOLOv4 incorporates features like weighted residual connections, cross-stage partial connections, and the Mish activation function. It also introduced new data augmentation techniques, such as mosaic enhancement and self-confrontational training. These improvements enabled YOLOv4 to achieve high accuracy while maintaining real-time performance, making it highly suitable for underwater environments like those in our project.

Additionally, YOLOv4 introduced the concepts of “free package” and “special package” to enhance training and detection performance. The "free package" utilizes techniques that improve training efficiency without increasing model complexity, such as advanced data augmentation and improved loss functions. The “special package” includes small modules, like Spatial Pyramid Pooling, which enhance detection performance without adding significant computational cost. These innovations allow YOLOv4 to perform efficient real-time detection and segmentation, making it ideal for tasks such as underwater wildlife monitoring and perfectly aligned with the needs of our project.

The literature highlights several challenges unique to underwater image segmentation. These challenges include different lighting conditions, obstacles such as plants or other marine life, and backgrounds in similar colors and textures to the sea turtles. Methods such as U-Net and CBAM offer promising solutions by helping the model focus on important parts of the image, while architectures such as YOLOv8 are effective for real-time segmentation. Future research can explore hybrid models that combine attention mechanisms with real-time detection frameworks to overcome current constraints. Generating synthetic data using generative adversarial networks (GANs) can also help address class imbalance problems and improve model robustness.

3. Method

3.1 Traditional method

Image segmentation based on K-means

For unsupervised learning algorithms, one of the classical methods is K-means clustering. K-means clustering is applicable to a large number of variables, but when the number of clusters is different, its corresponding results are also different. So it is required to initialize the proper number of number of cluster, k2. Again, it is required to initialize the k number of centroid. Different value of initial centroid would result different cluster. [1]

OpenCV has a built-in K-means function to make the task easier. Through this method, we first perform background removal on the original image, then use morphology to process the original image, and finally generate foreground mask for the turtle. The brightness, color information, and pixel data of the image are then separated by converting the image into a Lab color space into a two-dimensional array, each of which is represented by the three components of the Lab. For the k value, first set the cluster number to 4, corresponding to the background, head, flippers, and carapace, and specify the maximum number of iterations of the task to be processed, accuracy requirements, and termination conditions. Once the clustering is completed using OpenCV's K-means function, the resulting labels are reshaped to fit the image dimensions, producing the final segmentation output. The results as shown in figure 1.

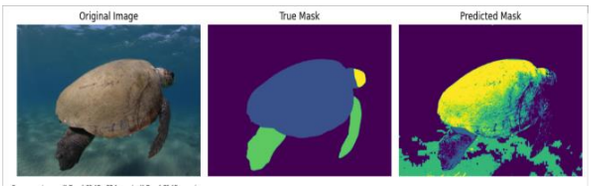


figure 1 K-means image segmentation results

the output results highlight the tipping point: since the K - means in random initialization process, for generating tags, the order can be arbitrary. As a result, we use color features or visual inspection to establish the correct correspondence between the labels and the turtle's anatomical parts.

For this project, we made the following assumptions: label 0 indicates the background, label 1 head, label 2 flippers, label 3 carapace.

Image segmentation based on Mean Shift:

For the use of traditional model, the method of using in addition to the K means clustering, we also make the Mean Shift clustering image segmentation. For Mean Shift, there is no need to pre-define the number of clusters, unlike K-means. Therefore, Mean Shift can dynamically adapt to the data structure, and it will perform better for complex image segmentation tasks. On the other hand, Mean Shift's flexibility in detecting variable patterns proved advantageous in cases where image data presented an irregular distribution. Mean Shift is chosen because it does not need to specify the number of clusters. It preprocesses the image by using MS algorithm to form segmented regions and maintain the ideal discontinuous features of the image.[9]The result is shown in figure 2.

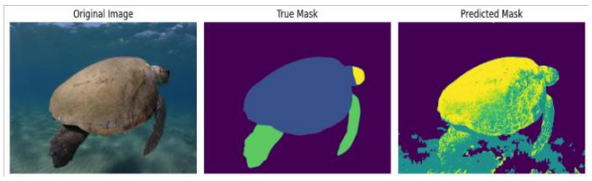


Figure 2 Mean Shift image segmentation results

We first perform background removal and morphological operations, then convert the image to Lab color space. Only pixels with non-zero masks are retained to reduce computation. Next, we use the estimate bandwidth function to determine an appropriate bandwidth, a key parameter for Mean Shift clustering. Using this bandwidth, we apply Mean Shift to obtain cluster labels and centroids. Since the label order is not fixed, we analyzed the color properties of each cluster to associate them with the turtle's anatomical parts.

3.2 Yolov8

Yolov8 (You Only Look Once version 8) is a new object detection model developed by the Ultralytics team, designed for real-time object detection, segmentation, and keypoint detection in computer vision tasks. YOLOv8 builds upon previous YOLO models with several enhancements[10], achieving significant improvements in both speed and accuracy.YOLOv8 has optimized its loss function, primarily using a combination of IoU (Intersection over Union) loss, classification loss, and confidence loss[11]. By introducing dynamic thresholds and weighted mechanisms, the model can adaptively adjust the weights of different objects during training, which enhances its detection capability and stability.[12].

YOLOv8 enhances multi-scale feature extraction and fusion[13], enabling precise object localization across diverse environments. Since sea turtle images are often taken under varying lighting and backgrounds, YOLOv8's multi-scale feature extraction supports accurate detection in different contexts. Building on this, YOLOv8-seg adds an instance segmentation branch that combines detection and segmentation, significantly improving segmentation accuracy and efficiency—ideal for fine-grained tasks like segmenting parts of sea turtles.

YOLOv8 also includes extensive data augmentation techniques, such as random cropping and color jittering[14], which boost the model's generalization ability and help it adapt to complex underwater conditions. Additionally, YOLOv8-seg supports multiple prompt types (point, box, text) to generate segmentation results, further enhancing its versatility. Given YOLOv8's high object detection accuracy, we chose YOLOv8-seg for segmenting different parts in the SeaTurtle2022 dataset. Its lightweight design allows deployment on resource-limited devices, making it highly suitable for real-time applications in marine research and conservation, providing a robust and efficient solution for sea turtle segmentation.

3.3 DeepLabV3 based on COCO API preprocessing

DeepLabV3-ResNet50[15]

DeepLabV3-ResNet50 is a deep learning model designed for semantic segmentation, combining the strengths of DeepLabV3 and ResNet-50 to achieve highly accurate results[16]. It utilizes several advanced techniques to enhance its segmentation capabilities. One key feature is the use of dilated convolution, which enlarges the receptive field without significantly increasing computational cost,

allowing the model to effectively segment larger or more distant objects. Additionally, the model employs Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual information, complemented by global average pooling to ensure robust feature representation.

The backbone of the model is ResNet-50, a 50-layer convolutional network known for its residual connections, which help efficiently extract features while mitigating the vanishing gradient problem. By combining the multi-scale feature extraction capabilities of DeepLabV3 with the efficient feature extraction of ResNet-50, DeepLabV3-ResNet50 excels at capturing fine details and providing detailed edge representations, making it particularly effective for segmentation tasks.

3.4 U-net

Standard U-net

U-Net[1] is a convolutional neural network model designed for an image segmentation task, which is unexplained in the original text. Main features involve a symmetric down-sampling and up-sampling path and a skip connection. The skip connection is a direct feature transfer of the output from the encoder to the decoder to maintain the accuracy of the segmentation and details of the image. The encoder pays attention to subsequent layers of convolution and max pooling to get high level features, where convolution applies a 3x3 [9]kernel with ReLU activation function. The decoder recovers image resolution in a stepwise manner by transposed convolutions and upsampling interspersed by skip connections merging low-over abstract features from encoder, to keep objects details. A last, 1x1 convolution converts the output from the decoder into channels of pixel-wise segmentation. U-Net's structure with its symmetric and skip connections achieves fast convergence with small datasets, thus giving it an advantage for limited-sample applications. Hence it was chosen as the baseline for SeaTurtle2022

CBAM attention

CBAM (Convolutional Block Attention Module) is a light attention module, which actively filters out the important channels and details of spatial level, respectively. The entire mechanism contains a Channel Attention Module (CAM) and Spatial Attention Module (SAM). CAM uses global average and max pooling, where channel attention is assigned to the matrix, which is a result of applying a fully connected layer, called a channel-weight matrix. SAM takes this step further by checking the pool of channels while applying the max and average pool, which is followed by a 3x3 convolution that produces a spatial-weight matrix that aims to provide better focusing on the target. SeaTurtle2022 data with lighting variations and cluttering backgrounds has a more striking impact from CBAM contrasting the target against the background and increasing the system's robustness and performance.

Enhanced U-net

Enhanced U-Net is obtained by combining CBAM with U-Net [3] which builds more accurate segmentation. The inclusion of CBAM in U-Net ensures a better flow of information into the value function. Adding CBAM to the encoder's part encourages the model to focus on the most informative features and details, ultimately lending a hand to better extraction of information. CBAM integration in the decoding module is to avoid the loss of high-resolution features during transient background recovery and to focus on the target boundaries. Compared to original U-Net, this structure improves segmentation in complex and fine regions by capturing boundaries more accurately and enhancing robustness against noise and lighting changes. The lightweight CBAM structure only slightly increases parameters while significantly enhancing segmentation accuracy, making it an effective solution for precise segmentation in complex tasks.

4. Experimental Results

4.1 Traditional method

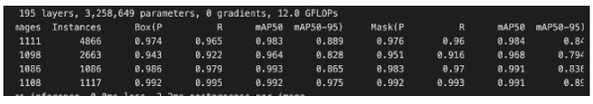
The experimental results are as follows, in SeaTurtle2022 dataset Mean Shift achieved 8.4% mean-IoU, turtle category IoU was 0.0%, flippers category IoU was 0.0%, head category IoU was 25.1%, number of training rounds was set to 10, batchsize was 32, optimizer was Adam, initial learning rate is 0.001, and image resolution is 256.Under the same circumstances, K-means achieved 7.3% mean-IoU, turtle category IoU was 7.8%, flippers category IoU was 0.1%, head category IoU was 14.0%.

The low performance of Mean Shift highlights its limitations in effectively segmenting the target classes under these experimental conditions. The results from both clustering methods demonstrate that traditional clustering-based approaches are inadequate for the complexity of this segmentation task.

4.2 Yolov8

The experimental results are as follows, in SeaTurtle2022 dataset YOlov8-segment achieved 88.8% mean-IoU, turtle category IoU was 97.5%, flippers category IoU was 82.7%, head category IoU was 86.2%, number of training rounds was set to 50, batchsize was 16, and image resolution is 1111.

The superior performance of YOLOv8 can be attributed to its advanced deep learning architecture and optimized parameter learning, making it highly suitable for accurate object detection and segmentation tasks.



```
  195 layers, 3,258,649 parameters, 0 gradients, 12.0 GFLOPs
 ages  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95)
 1111       4866      0.974      0.965      0.983      0.889      0.976       0.96      0.984      0.84
 1098       2663      0.943      0.922      0.964      0.828      0.951      0.916      0.968      0.79
 1086       1086      0.986      0.979      0.993      0.865      0.983       0.97      0.991      0.83
 1108       1117      0.992      0.995      0.992      0.975      0.992      0.993      0.991      0.85
```

Figure 3 Yolov8 results

4.3 DeepLabV3 based on COCO API preprocessing

The experimental results are as follows, in SeaTurtle2022 dataset achieved 83.1% mean-IoU, turtle category IoU was 87.6%, flippers category IoU was 73%, head category IoU was 72.5%, number of training rounds was set to 10, batchsize was 4, optimizer was Adam , initial learning rate is 0.001, and image resolution is 512.

Although DeepLabV3 demonstrated strong segmentation capabilities, it fell short in comparison to YOLOv8, particularly in the segmentation of flippers and head, likely due to differences in network architecture and depth.

```
Mean IoU (mIoU): 0.8310723225287171
IoU for class 0: 0.9923448108097916
IoU for class 1: 0.8761421069827441
IoU for class 2: 0.7303645394515524
IoU for class 3: 0.7254378328707805
```

Figure 4 DeepLabV3 results

4.4 U-net

Standard U-net

The experimental results are as follows, in SeaTurtle2022 dataset U-Net achieved 81.8% mIoU, turtle category IoU was 88.9%, flippers category IoU was 76.0%, head category IoU was 80.5%, number of training rounds was set to 10, batchsize was 32, optimizer was Adam , initial learning rate is 0.001, and image resolution is 256.

U-Net exhibited robust performance, particularly for the turtle category. The significant reduction in training loss over epochs suggests that the model was effective in learning features, though its overall segmentation performance was slightly inferior to that of YOLOv8 and DeepLabV3.

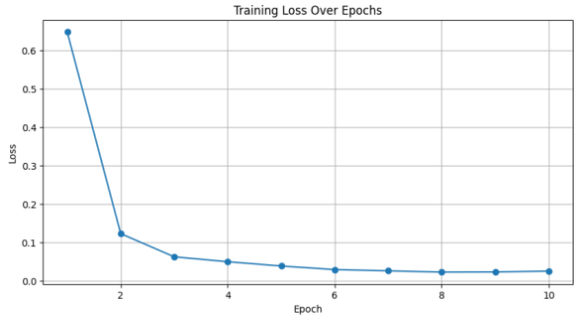


Figure 5 Standard U-net training loss

Figure 6 Standard U-net image segmentation results

Enhanced U-net

The experimental results are as follows, in SeaTurtle2022 dataset U-Net achieved 81.8% mean-IoU, turtle category IoU was 88.9%, flippers category IoU was 76.0%, head category IoU was 80.5%, number of training rounds was set to 10, batch size was 32, optimizer was Adam, initial learning rate is 0.001,and image resolution is 256.

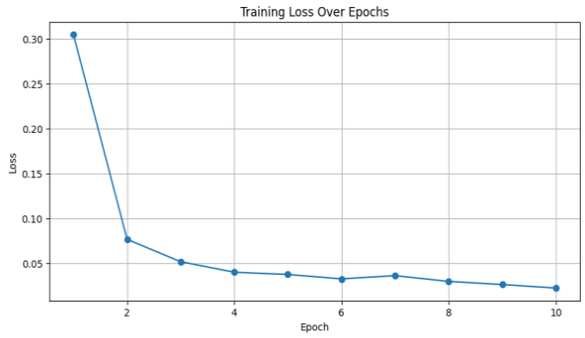


Figure 7 Enhanced U-net training loss

Figure 8 Enhanced U-net image segmentation results

After adding CBAM turtles all categories of IoU have increased, it can be clearly seen from the comparison of test results that the boundary of each part is more accurately localize.

5. Discussion

The comparative analysis of the experimental results on the SeaTurtle2022 dataset highlights are both the strengths and limitations of traditional methods and deep learning-based models for segmentation tasks. Below, we provide an integrated discussion of the four models—K-means, Mean Shift, YOLOv8-segment, DeepLabV3, and U-Net—in the context of their observed performance and challenges.

Traditional Methods (K-means and Mean Shift)

According to the results of the project, we found that the traditional models K-means and Mean Shift have limitations in segmenting turtle images and are very easy to be disturbed. Because these traditional method mainly use pixel color or density information, they are vulnerable to background interference from the target object, such as water waves and changing light behind the turtle. So specifically, while the K-means model is simple, it is difficult to avoid interference and successfully separate the turtle from the water wave in the background, because the K-means model assumes that each pixel belongs to the nearest cluster center. These errors often occur in scenes with variations in water, shadows, and light, and the K-means model requires a certain number of clusters, which reduces its flexibility in different underwater images. In addition, requiring a certain number of clusters reduces its flexibility in different underwater images. For Mean Shift model, although not require a fixed number of clusters and more adaptable than K-means, as traditional model, it also face some problems, especially when deal with tasks with high computational costs and large images. Mean Shift rely on local pixel density, and water surface reflection often result in multiple peak causing part of the background to be incorrectly identified and labeled as an object. In general, traditional methods do not effectively capture spatial and contextual features, so they cannot handle complex fields and segment images accurately.

YOLOv8-segment

Based on the experimental results, the YOLOv8-segment model performed exceptionally well in terms of mean Average Precision (mAP) and Intersection over Union (IoU) scores, significantly surpassing traditional methods. The model achieved an overall Box mAP50 of 99.3% and Box mAP50-95 of 88.9%, demonstrating high detection performance across various categories. Additionally, it achieved a Mask mAP50 of 99.1% and Mask mAP50-95 of 89%, indicating excellent performance in segmentation tasks.

However, further analysis revealed a decline in segmentation precision at higher IoU thresholds, particularly in categories with complex boundaries. This suggests that the model still faces challenges in precise boundary localization, especially when segmenting smaller or more intricate areas, such as the flippers.

Furthermore, the impact of class imbalance on model performance is evident from the lower Mask mAP50-95 scores for certain categories, with the lowest being 79.4%. The imbalance in the dataset resulted in underrepresentation of smaller or less frequent categories during training, limiting the model's generalization ability for these classes.

DeepLabV3

The DeepLabV3 mode uses advanced feature extraction through its ResNet backbone, showed improved capability in complex backgrounds and specific targets. However, there are still a lot of limit:

- Complex Backgrounds and Low Contrast: DeepLabV3 do not work when the foreground background are similar color. Same if there is a low contrast. It result confusion and inaccurate segment.
- Diversity in Scale and Shape: When segmenting targets of significantly different scales, the model faced difficulties. It shows that a stable receptive field is hard to catch diverse object sizes.
- Occlusion and Visibility: When objects were hidden or partially visible, DeepLabV3 cannot accurately identify and segment them.
- Lighting Variations and Shadows: The model was sensitive to lighting changes. So augmentation techniques are ways to enhance robustness to different lighting conditions. Overall, while DeepLabV3 has certain feature extraction capabilities, it was still not good enough for some complex scenes.

U-Net

The U-Net model, particularly in its enhanced version with the Convolutional Block Attention Module (CBAM), showed significant improvements in segmentation performance. The incorporation of CBAM increased the mean IoU from 80.9% to 81.8%, with notable gains in the turtle, flipper, and head categories. The CBAM module enhanced the model's feature extraction and boundary recognition, especially in complex scenes with fine-grained details. However, U-Net with CBAM also had specific shortcomings:

- Background Complexity: Despite improvements, U-Net struggled with extreme lighting variations and noisy backgrounds. The attention mechanism sometimes failed to accurately focus on the target in the presence of significant noise, leading to blurred or incorrect segmentation boundaries.
- Category Imbalance: As observed in other models, category imbalance affected the effective-ness of CBAM in recognizing smaller targets, limiting its ability to enhance IoU for underrepresent-ed classes.
- Computational Cost: While CBAM is designed to be lightweight, it still introduced additional comutational costs, particularly for high-resolution images, which could affect the model's efficiency in real-time applications.

6. Conclusion

The comparative analysis of traditional methods and deep learning models for turtle image segmentation in the SeaTurtle2022 dataset reveals significant differences in their performance, strengths, and limitations. Below, we summarize the conclusions drawn for K-means, Mean Shift, YOLOv8-segment, DeepLabV3, and U-Net, along with future research directions.

Traditional Methods (K-means and Mean Shift)

For this project, in order to improve the accuracy of image segmentation, our future research should consider adding deep learning model. Deep learning

method, such as U-Net or Mask RCNN, learn advanced feature through end-to-end training provide more complex image segmentation method. These models can effectively differentiating between foreground and background by capture both spatial and contextual cues. Utiliz pre-trained models, increase labeled train data, and apply transfer learning can significantly enhance the robustness of segmentation models, particularly when deal with challenged backgrounds. Using deep model deal with the project, can help us better identify the goal, avoid to deal with the effect of background interference in the image, better image segmentation can better adapt images in complex scenes.

YOLOv8-segment

The YOLOv8-segment model achieved the highest segmentation accuracy among all models in our project. However, there is still room for improvement:

- Multi-Scale Feature Fusion: Introducing a multi-scale feature fusion mechanism can enhance detail capture at different scales, improving segmentation performance under various conditions.
- Data Augmentation: Techniques such as brightness adjustments and synthetic data generated by Generative Adversarial Networks (GANs) can improve the model's robustness and generalization ability.
- Lightweight Model Optimization: Techniques like pruning and quantization can increase inference speed while maintaining accuracy.
- Class Balancing: Implementing weighted loss functions or oversampling can improve performance for underrepresented classes.

DeepLabV3

DeepLabV3 has strong feature extraction. It can handle some complex backgrounds well, but has room for improvement:

- Multi-Scale Feature Fusion: This can balance local detail and global context. So, for some complex environment, especially underwater, it helps a lot.
- Class Imbalance and Occlusion Handling: Weighted loss functions or resampling techniques are used in this method. This can deal challenges with occlusions and not enough categories.
- Edge Detection Module: This module can improve boundary precision, reducing segmentation blurriness.
- Dataset Expansion: This function is to increase the diversity of datasets, such as lighting in underwater. This helps enhance generalization.

U-Net

U-Net, enhanced with CBAM, showed strong segmentation performance but can be further improved:

- Multi-Scale Feature Fusion: Combining CBAM with multi-scale fusion can help recognize both coarse and fine features, aiding segmentation of subtle targets.
- Data Augmentation and Synthetic Data: Using GANs and advanced augmentation can provide challenging training examples to enhance generalization.
- Lightweight Optimization: Optimizing CBAM or using efficient attention mechanisms can balance accuracy with computational efficiency, crucial for real-time applications.
- Category Balancing: Category-weighted loss functions can help focus on underrepresented classes, improving segmentation quality

Future work

The conclusions of traditional methods and deep learning models indicate that deep learning based methods are significantly superior to traditional clustering techniques in image segmentation tasks involving underwater images with complex backgrounds. Traditional methods such as K-means and Mean Shift are fundamentally limited by their inability to utilize global and contextual information,

resulting in poor performance in complex and dynamic environments.

In contrast, deep learning models have shown significant advantages, particularly through their ability to capture spatial and contextual features at different scales. Future research should focus on:

Enhanced multi-scale feature fusion: This technique is crucial for improving segmentation in different environments, enabling models to capture details at different granularity levels.

Addressing category imbalance: Category imbalance remains a key issue for segmentation accuracy. The use of weight reduction functions, resampling, and advanced data augmentation can help alleviate these challenges.

Using synthetic data generation: GAN and advanced data augmentation techniques can be used to create different training datasets, helping to improve the robustness of the model to different conditions.

Optimizing model architecture: Lightweight optimization techniques, including parameter reduction and efficient attention mechanisms, can ensure that deep learning models are both accurate and computationally feasible for real-time applications.

Combining these strategies will help improve the robustness and adaptability of segmentation models under different environmental conditions, paving the way for more effective solutions in complex real-world scenarios. By focusing on these directions, future research can further break through the boundaries of segmentation techniques, enabling them to address the challenges posed by dynamic and complex visual environments.

7. Reference

[1] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm," in *Proc. 11th Int. Multi-Conf. Information Processing-2015 (IMCIP-2015)*, 2015, vol. 54, pp. 764–771

[2] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//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: 234-241.

[3] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

[5] Agarap A F. Deep learning using rectified linear units (relu)[J]. arXiv preprint arXiv:1803.08375, 2018.

[6] Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7263-7271.

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection." Presented at CVPR 2016.

[8] Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934.

[9] W. Tao, H. Jin, and Y. Zhang, "Color Image Segmentation Based on Mean Shift and Normalized Cuts," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1382-1389, Oct. 2007.

[10] Ultralytics Documentation: YOLOv8 Documentation on Ultralytics

[11] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. Link.

[12] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2117-2125.

[13] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved Techniques for Training GANs. arXiv preprint arXiv:1606.03498. Link

[14] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 318-327. Link.

[15] https://blog.csdn.net/weixin_44816589/article/details/115266935 Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[16] https://pytorch.org/vision/main/models/generated/torchvision.models.segmentation.deeplabv3_resnet50.html