Semantic Segmentation of Underwater Images

Sanika Paranjpe
Luddy School of Informatics and Computing
Indiana University
sparanjp@iu.edu

Yamini Priya Kodeboyina
Luddy School of Informatics and Computing
Indiana University
ykodeboy@iu.edu

Shardul Dabhane
Luddy School of Informatics and Computing
Indiana University
sdabhane@iu.edu

Under Guidance of Prof. Md Alimoor Reza and Prof. David Crandall

Abstract

Convolutional Neural networks are most widely used powerful tools in the domain of computer vision. Our work presents semantic segmentation on underwater images with the most diverse dataset of underwater animals. Our focus during this work was to establish a baseline performance on our dataset using the state of the art convolutional neural networks designed for semantic segmentation. Our dataset consists of 31 classes on total which consist of 21 animal classes and 9 background classes and one other class. We present a baseline evaluation on three networks which are Fully Convolutional Neural Network(FCN) with a ResNet50 backbone, DeepLabV3 with a ResNet50 backbone and LR-ASPP with MobileNetV3-Large backbone.

1. Introduction

In the Computer Vision domain there are two distinct segmentation algorithms which are Semantic segmentation and instance segmentation. Semantic segmentation treats multiple objects of the same class as a single entity whereas Instance segmentation treats such objects as different instances of the same class.

Semantic Segmentation is a widely studied topic in the world of computer vision and deep learning. It is used in object detection, robot vision and scene geometry identification [1]. The success in semantic segmentation is due to the development of state of the art very deep neural networks. Semantic Segmentation can be used on underwater images which have numerous applications and that was our motivation to take up this topic.

Underwater imagery has many applications in domains such as studying marine biology, the places where certain marine animals stay, the characteristics of the animals etc.

Humans can only travel up to a certain depth underwater, the other way to be able to study the depth of the oceans is to create underwater automatic vehicles. But to be able to identify certain species underwater, we need special computer vision tools. One such tool is Semantic segmentation. This can be used in robots which automatically navigate and collect data from the deep sea. But for such robots which navigate in the underwater environment, the existing solutions for semantic segmentation are not yet trained on this specific domain. The contents in the underwater images are very different from the normal day-to-day scenes as they have different background patterns, some unique optical distortions and domain-specific objects.[1] There has been some research in coral reef detection and fish detection, but there is not much work in segmenting marine animals of different species along with the background information.

2. Background and Related Work

There has been some work in dealing with underwater images like removing the distortions due to underwater photography from the images, underwater object detection [2] and segmenting images, etc. There has been some work in the domain of segmenting underwater images. Some researchers have worked on instance segmentation such as Jiwoon et al in [9] and on semantic segmentation by Drews-Jr et al in [8] and as mentioned in later paragraphs.

The work presented In [1] consists of total 8 classes with 7 main classes which have human divers, robots, ruins, plants, fish, seafloor/rooks and it has one background class. The authors from [1] have set a benchmark on experimenting on some existing model and also created their own model called SUIMNet. There also has been some work in segmenting Coral reefs as per [3]. They have used Fully Convolutional Neural networks to segment coral reefs from images from a survey of the health of the coral reefs.

We are using the dataset which was created by Prof. Md Alimoor Reza et al. Also, some students under his guidance at Indiana University have previously worked on few-shot segmentation on this dataset. They worked on segmenting foreground and background classes. Our projects extends this concept to segment the image into 31 annotated classes using Multi-class semantic segmentation.

3. Methods

The aim of this project is to establish a baseline performance using some state of the art Semantic segmentation models and perform experiments on it. We also had planned to implement some novel ideas on this dataset, but due to shortage of time and many challenges that we faced while working on the baseline performance, we were not able to work on a novel model.

We have used pretrained models from Pytorch torchvision library to train our dataset on those models and perform our baseline analysis. The pretrained models consists of Fully Convolutional Neural Network(FCN) with a ResNet50 backbone, DeepLabV3 with a ResNet50 backbone and LR-ASPP with MobileNetV3-Large backbone. All of these models have been pretrained on a subset of COCO train2017, on the 20 categories that are present in the Pascal VOC dataset [10].

All of these models have been pretrained on terrestrial data, so one of the research question we tried to answer in this project is to experiment on how well the models which are pretrained on terrestrial data is able to segment underwater images.

3.1. Dataset

The dataset we are using is provided to us by Prof. Md Alimoor Reza. Dataset Link @ [12]. The dataset consists of diverse set of images of many animal species; 575 images of underwater environments from the internet, fully annotated at the pixel level.

All images and their ground truth masks are present in form of ".mat" files with data present in form of dictionary with keys including the "image_array" which consists of the image as an np.array in RBG format, "class" which tells the class of main object in the image and "mask_array" which consists of an image with size same as the original, but its pixel values represent the class of that pixel according to our 31 classes.

Each image contains at least one of the animals from the following 21 categories: Crab, Dolphin, Frog, Turtle, Whale, Nettles, Octopus, Sea Anemone, Shrimp, Stingray, Penguin, Sea Urchin, Seal, Shark, Nudibranch, Crocodile, Otter, Polar Bear, Seahorse, Starfish, Squid. The dataset also consists of background classes such as coral, rock, water, sand, plant, human, iceberg, reef, fish and other. Almost all foreground classes have 30 images each. Some classes such as sea anemone, sea urchin, sea horse, frog have only 15 images each. For background class of coral, it is present in only 15 images.

Some example images from the dataset:

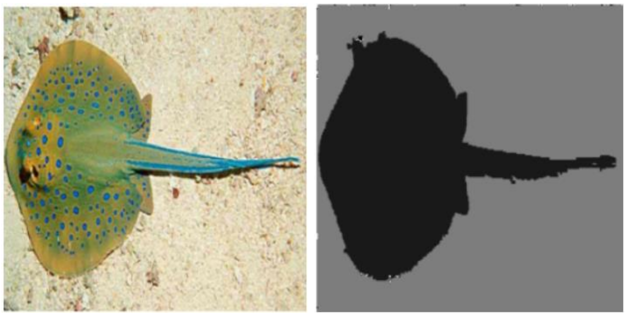

Fig 1: Image and Ground truth mask of Sting-Ray class (image and mask resized to 256 x 256 image)

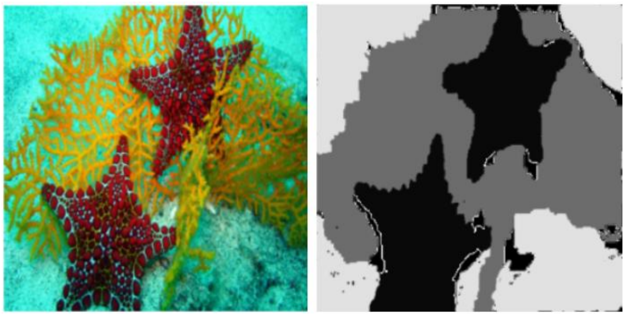

Fig 2: Image and Ground truth mask of Star-fish class (image and mask resized to 256 x 256 image)

3.2. Data Augmentation and Transforms

Data Augmentations:

We decided to do data augmentation on the training images on our dataset. The classes in the dataset are imbalanced. Each animal will have around 30 images each and some classes have even about 15 images. But on the other side, the background class such as water or seafloor is present in almost all the images. Such imbalance causes issues while training the algorithm. Thus we decided to add augmented images to our training set so that the model gets more data to train on for classes with less data.

Operations performed while augmenting images:

1. Horizontal flip (on random 200 images of training data)
2. Shifting to right by 20 pixels (on random 200 images of training data)
3. Shifting to left by 20 pixels (on random 100 images of training data)
4. Shifting to up by 20 pixels (on random 100 images of training data)
5. Shifting to down by 20 pixels (on random 100 images of training data)

The reason for using these operations for data augmentation was that performing these operations on the ground truth will not change the label values, hence these operations can be performed on both image and ground truth.

Transformations:

We have applied some transformations to the image and ground truth labels.

Following transforms were applied to images:

1) Resizing:
   All images in the dataset had different dimensions. To make the dimensions consistent, we resized all images in the dataset to size 256x256. Used “Bilinear” Interpolation for resizing.
2) Normalized all the images as per ImageNet standards by using mean = [0.485, 0.456, 0.406] and std = [0.229, 0.224, 0.225].

Following transforms were applied to images:

1) Resizing:
   All ground truth labels were also resized to size of 256x256 to match the resized image size. The interpolation used here was “Nearest”.

Both images and ground truth labels were converted to pytorch tensors in the end.

3.3. Models Used

We are using three pretrained models from torchvision library with completely different convolutional network architecture which were designed to perform semantic segmentation on terrestrial data to perform our analysis and answer our research question. Each of these models is a convolutional neural network with different structures.

1) FCNResNet50:

Fully Convolutional neural Network is a semantic segmentation model built using only convolutional layers producing dense output predictions and show excellent state-of-the-art segmentation on PASCAL VOC [4].

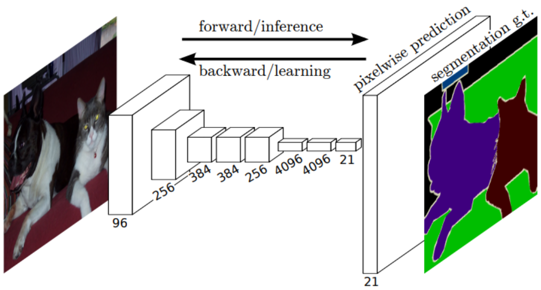

Fig 3: Fully Convolutional Neural Network structure [4]

Details on the pretrained model:

This pretrained model from the torchvision is pretrained on subset of COCO train2017 dataset which consists of 20 classes. This model uses ResNet50 as a backbone classifier.

The predictions returned by this model have masks for each class as an image. So the predictions are of size [batch_size,20, (image size)].

Changes we made on the pretrained model:

1. We have removed the last layers from the model and worked with the output of the classifier.
2. The dense classifier from the FCN model returns predictions in following format [b,c,h,w]. Here b = batch size, c = number of channels = 512 and (h,w) = image size = 32x32. So up sampling is needed here to get 31 channels at the output and output image size as 256x256.
3. We have used Relu Activation, Batch normalization layers and Transpose convolutions. We used a series of BatchNorm -> Deconvolution -> Relu with appropriate input output channels ans kernel size and stride to achieve correct dimensions as output predictions. We have used a 1x1 2-d Convolution at the end to achieve smoothness.

2) DeepLabV3:

The pretrained model from the torchvision library uses ResNet50 as a backbone.

The DeepLab network is created using atrous convolutions which is powerful as it widens the field-of-view of the model which enables learning of features at multiple scales.

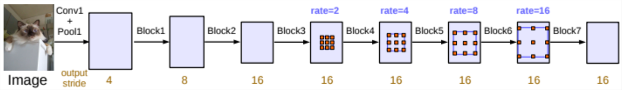

Fig 4: Atrous Convolutions [5]

Changes we made on the pretrained model:

1. We have removed the last layers from the model and worked with the output of the classifier similarly as with FCN to get output image masks of size 256 *256

3) LR-ASPP MobileNetV3 Large:

LR-ASPP stands for Lite Reduced Atrous Spatial pyramid pooling which is presented in this model to perform semantic segmentation.

This model extends the MobileNet to add in the concept of LR-ASPP to produce state-of-art-performance on ImageNet and CityScapes standard datasets. [6]

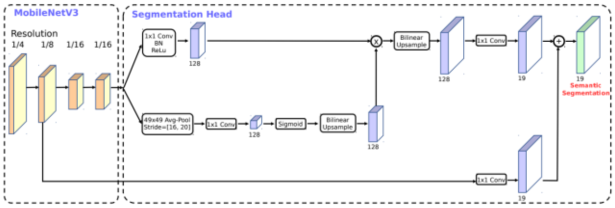

Fig 5 : Architecture of LR-ASPP MobileNetV3-large[6]

Changes we made on the pretrained model:

2. We have removed the last layers from the model and worked with the output of the classifier similarly as with FCN to get output image masks of size 256 *256

3.4. Experiments

We did experiments to compare the performance of the three models on our dataset with and without data augmentation. We also experimented with different optimizers and learning rates, with and without weight decay to get insights on the performance of the three models.

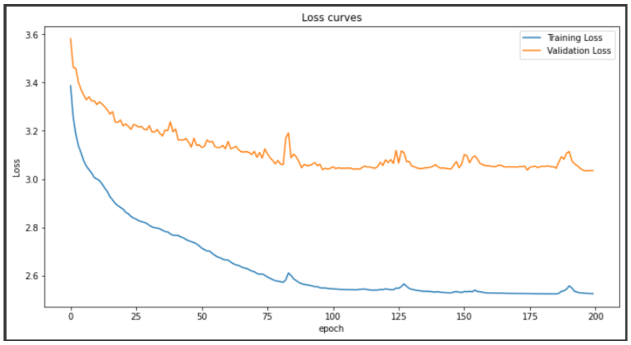

Fig 6: Training and Validation loss curve for FCN model with Adam optimizer

4. Results

Following are the results of our experiments conducted to establish a baseline on the dataset.

4.1. Without Data Augmentation

We wanted to compared the impact of implementing augmentation on the data. Hence we also carried out experiments without augmentation of the data.
To obtain the below results we have used the following hyperparameters:
Batch size = 20
Learning rate: 0.1
Weight Decay = 0.0001
Optimizer = SGD (Stochastic Gradient Descent)
Epochs = 200
Loss Function used: Cross Entropy Loss

Table 1: Results without data augmentation

Model	**mIOU**	**Accuracy**
FCNResNet50	0.1499	0.5179
DeepLabV3-ResNet50	0.0900	0.5090
LR-ASPP MobileNetV3	0.0384	0.5291

4.2. With Data Augmentation

We have augmented the training data and obtained results on the dataset using 3 different optimizers and different hyperparameters. To obtain the below results we have used the following hyperparameters:
Batch size = 20, Epochs = 200
Loss Function used: Cross Entropy Loss

Table 2: Experiments for FCN model

Optimizer	Learning Rate	Weight Decay	mIOU	Accuracy
SGD	0.1	0.0001	0.2078	0.5820
Adam	0.0001	-	0.3319	0.6431
RMSProp	0.0001	-	0.2540	0.5735

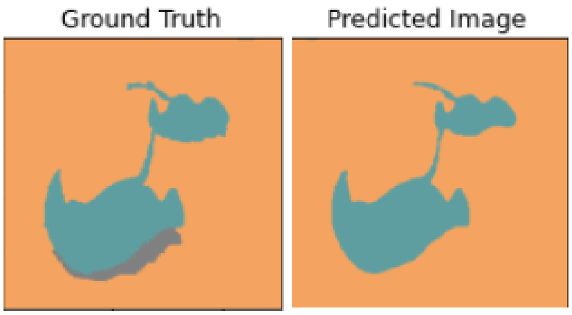

Fig 7: Stingray and background Class correctly predicted by FCN model

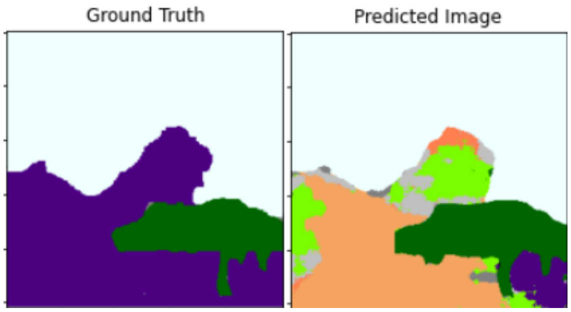

Fig 8: Correctly segmented animal class but background class incorrect

Table 3: Experiments for Deeplabv3 Resnet50 model

Optimizer	Learning Rate	Weight Decay	mIOU	Accuracy
SGD	0.1	0.0001	0.1354	0.5123
Adam	0.0001	-	0.3053	0.6097
RMSProp	0.0001	-	0.2843	0.6360

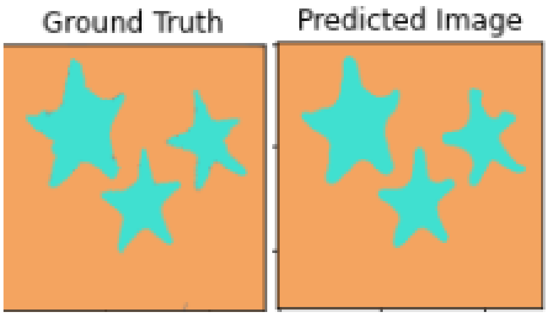

Fig 9: Starfish and background Class correctly predicted by DeeplabV3 model

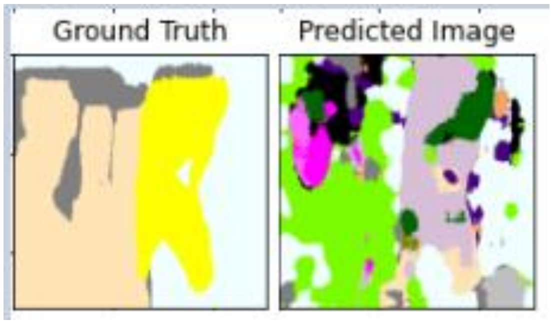

Fig 10: Example of failed Segmentation by DeeplabV3 model

Table 4: Experiments for LR-ASPP MobileNetV3 Large

Optimizer	Learning Rate	Weight Decay	mIOU	Accuracy
SGD	0.5	0.0001	0.1477	0.5609
Adam	0.0001	-	0.2102	0.5672
RMSProp	0.0001	-	0.1875	0.5885

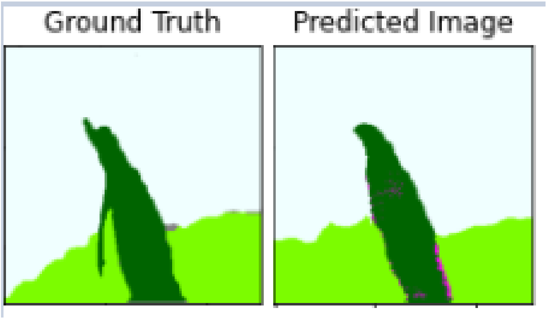

Fig 11: Example of Segmentation by LR-ASPP model

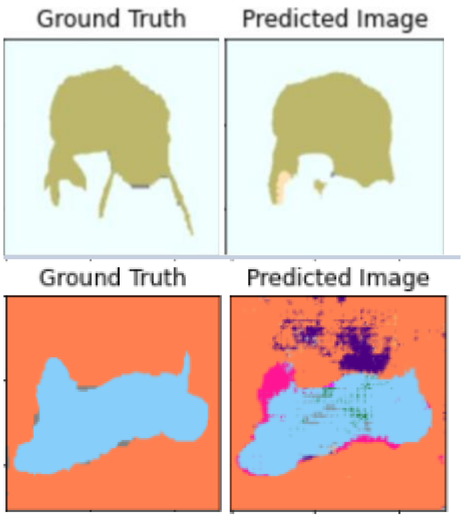

Fig 12: Examples of Segmentation by LR-ASPP model

Some Weird Outputs:

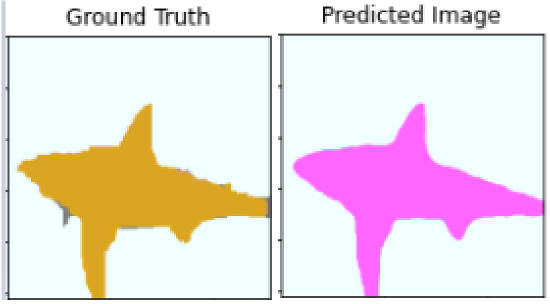

Fig 13: Correctly segmented animal, but wrong class predicted.

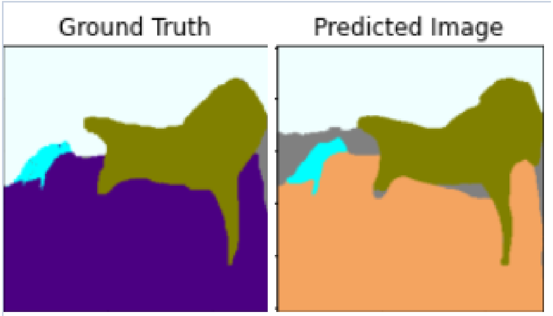

Fig 14: Both animals in cyan and green are segmented correctly, but wrong background class.

5. Discussion

Our results have answered the research question that models pre-trained on terrestrial data are able to perform semantic segmentation on diverse set of underwater image data. We wanted to extend our analysis to performing analysis on HRNet (High Resolution Network) [11] and we are working on it, but have not completed the analysis yet. We will extend our project to work on HRNet and also use some novel ideas to obtain better results.

Underwater imagery can be used in various research domains of robotic vision which will facilitate extended the study on marine biology. Our dataset consists of 21 species of marine animals, but there are thousands of different animal species underwater. This study can also be extended to include this variety of animals. To think about it, can this study be also extended to segmenting different varieties of underwater plants? Underwater images usually have the haziness effect which is due to capturing an image underwater. There are various algorithms and research work conducted to remove the haziness. This study raises a research question that how well can a semantic segmentation algorithm work after this haziness from the images is removed.

Further all this study in underwater images can be extended to videos and then real time vision in developing underwater robots.

6. Conclusion

We have successfully trained three networks which are Fully Convolutional Neural Network(FCN) with a ResNet50 backbone, DeepLabV3 with a ResNet50 backbone and LR-ASPP with MobileNetV3-Large backbone on the diverse dataset of underwater images provided to us by Prof Dr. Md Alimoor Reza et al. According to the experiments conducted by us, we observe that Fully Convolutional neural network performs best on the given dataset with usage of Adam optimizer with the mIOU of 33.89 % and Accuracy of 64.59%. The hyperparameters we used needs more finetuning, and we might get better results with fine-tuned parameters. Our work will act as a baseline comparison in future work on this dataset and on underwater imagery data in general. Our future work in this project will be to use HRNet and perform experiments and analysis using this dataset and then try to obtain better results using some novel ideas.

References

[1] Islam, M., Edge, C., Xiao, Y., Luo, P., Mehtaz, M., Morse, C., Enan, S., & Sattar, J.. (2020). Semantic Segmentation of Underwater Imagery: Dataset and Benchmark.

[2] A. Saini and M. Biswas, "Object Detection in Underwater Image by Detecting Edges using Adaptive Thresholding," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 628-632, doi: 10.1109/ICOEI.2019.8862794.

[3] J. P. Pierce, Y. Rzhanov, K. Lowell and J. A. Dijkstra, "Reducing Annotation Times: Semantic Segmentation of Coral Reef Survey Images," Global Oceans 2020: Singapore – U.S. Gulf Coast, 2020, pp. 1-9, doi: 10.1109/IEEECONF38699.2020.9389163.

[4] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation.

[5] Chen, L.C., Papandreou, G., Schroff, F., & Adam, H.. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation.

[6] Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q., & Adam, H.. (2019). Searching for MobileNetV3.

[7] Lambert, J., Liu, Z., Sener, O., Hays, J., & Koltun, V.. (2021). MSeg: A Composite Dataset for Multi-domain Semantic Segmentation. https://github.com/mseg-dataset/mseg-semantic

[8] Drews-Jr, P., Souza, I.d., Maurell, I.P. et al. Underwater image segmentation in the wild using deep learning. J Braz Comput Soc 27, 12 (2021). https://doi.org/10.1186/s13173-021-00117-7

[9] Ahn, Jiwoon et al. “Weakly Supervised Learning of Instance Segmentation With Inter-Pixel Relations.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 2204-2213.

[10] https://pytorch.org/vision/stable/models.html#semantic-segmentation

[11] Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., & Wang, J.. (2019). High-Resolution Representations for Labeling Pixels and Regions.

[12] Link to the Dataset shared with us: https://drive.google.com/drive/folders/1NhDgRIAGQtwBqWqAPzVLcBtrO4fcyEcf?usp=sharing