Evaluation of Deep Object Detectors for Pointing Gesture Classification for Underwater Human-Robot Communication

Preeti Pidatala

Submitted under the supervision of Junaed Sattar to the University Honors Program at the University of Minnesota-Twin Cities in partial fulfillment of the requirements for the degree of Bachelor of Science, summa cum laude in Computer Science.

2023

Acknowledgements

First and foremost, I would like to thank my advisor, Junaed Sattar, for his endless guidance and support throughout this project. I would also like to thank my thesis readers, Catherine Zhou and Karthik Desingh for their insightful feedback and time. Additionally, I'd like to extend a huge thank you to Chelsey Edge for all her advice and patience with my many, many questions. Finally, I would like to thank my family for their encouragement and love always.

Abstract

Autonomous underwater vehicles, or AUVs have a growing number of applications in today's world. One such application is assisting divers to aid with difficult or risky underwater tasks. In this setting, communication between the robot and human diver is of utmost importance. However, due to the restrictions of the underwater environment, divers must largely depend on non verbal communication. Pointing gestures are a natural choice as they are both intuitive and effective in conveying direction. The success of an underwater expedition relies on an AUV's ability to distinguish various pointing gestures. This motivates the objective to find a model which can be utilized for underwater pointing gesture classification tasks with high accuracy. Based on their success in related terrestrial gesture classification projects, four machine learning models (UNET, PSPNet, DeepLabV3 and Mask R-CNN) were analyzed. They were trained on various pointing gestures including go here, pick up, general point, and take a picture. Finally, they were evaluated on their ability to classify these pointing gestures. This project presents an evaluation of UNET, PSPNet, DeepLabV3 and Mask R-CNN for the task of classifying underwater pointing gestures.

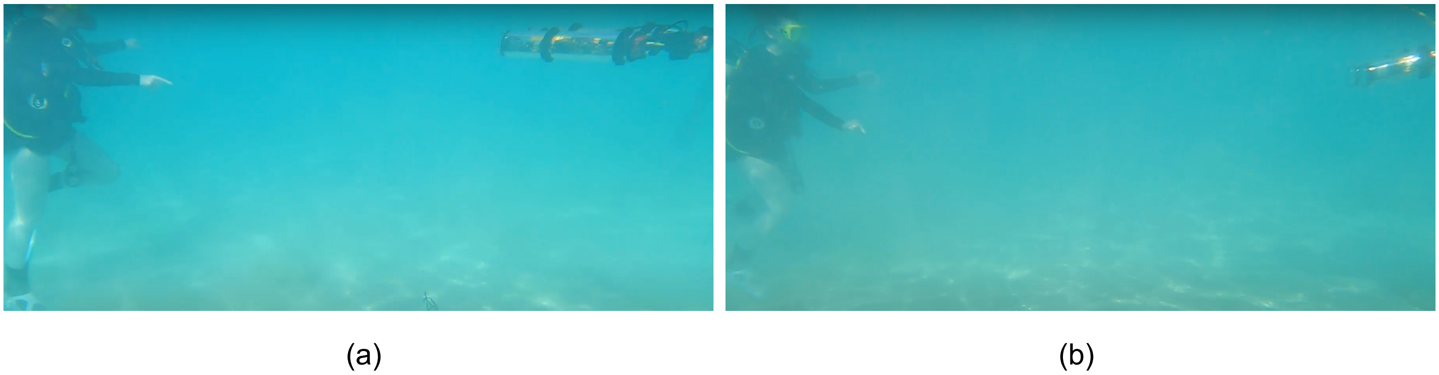
Table of Contents

1. Introduction

Autonomous vehicles have been on the rise within society. Applications such as self-driving cars, vacuums, drones and rockets are being integrated into daily life. They are often capable of autonomously carrying out tasks that are traditionally time, energy, or resource consuming- bringing both convenience and productivity to users. Some applications, such as self-driving cars, support human users through intelligent assistance- which can reduce human error. More recently, underwater robotics, and more specifically, autonomous underwater vehicles (AUV's) have gained significant attention for various applications such as underwater exploration, monitoring, and even environmental clean up [1]–[3]. These robots can operate in challenging environments where human intervention is either impractical or dangerous. An AUV assistant can make these expeditions safer and more efficient. However, AUVs face a unique set of challenges due to the unknown and dynamic environment that they primarily operate in. These vehicles must be able to adapt to high pressure and low visibility environments. As a result, communication between the AUV and human diver, one of the most crucial aspects of a successful expedition, becomes increasingly difficult. In order for this communication to be successful, the AUV must be able to recognize when a diver is attempting to communicate, understand the diver's objectives, and respond appropriately. In terms of robot to human communication, text displays, LED patterns, and sound are amongst the strategies currently being explored [4]. For humans to relay information effectively to their AUV aids, gestural communication has been emerging as a strong and reliable means of communication.

This approach involves two main components: detecting human hand pointing gestures using sensors and interpreting them to control the robot. The first part of this revolves upon accurate pointing gesture detection using computer vision techniques to recognize a hand forming a pointing gesture. The second can be achieved using machine learning algorithms, such as neural networks to train the system to recognize what command the pointing gesture is associated with. In this way, the AUV and human diver can communicate with each other in an intuitive and effective manner. Previous research has explored various machine learning techniques for terrestrial gesture classification, but underwater pointing gesture classification has not been explored much at all.

Pointing gestures are incredibly pertinent to nonverbal communication. They communicate direction and offer another layer of clarification to commands. In collaborative settings, pointing gestures are often used to intuitively direct attention to an object, area, or person. This translates to underwater human to robot communication as well. Underwater pointing gestures can be used to communicate the intent to travel to an area, pick up an object, take a picture of an item, or follow a specific route [5]. AUVs must be able to recognize and understand various pointing gestures to be able to adequately assist the human divers. Thus, it is important to find an algorithm that is capable of classifying various underwater pointing gestures accurately. Within this paper, various machine learning models will be evaluated in order to optimize underwater pointing gesture classification.



(a) (b)

Figure 1. *Demonstration of underwater pointing gestures for human-robot communication with the LoCO AUV* [6]

2. Background

2.1. Terrestrial Applications

Several studies have investigated the use of hand gestures in human-computer interaction. Hand gestures as a means of communication have been shown to be extremely effective due to their intuitive nature. They allow the communicator to quickly convey tasks in environments where verbal communication is challenging. In terrestrial applications, various machine learning algorithms have been explored to efficiently detect and classify gestures. These algorithms have been used in a wide range of fields and applications.

For example, one of the most common gesture based communication is sign language. Researchers have developed technological translators to ease the communication gap between those who are and are not fluent in sign languages. The majority of these rely on gesture recognition. Many proposed methods use Convolutional Neural Networks (CNN's) to extract specific gestures from images [7]. Moreover, in a comparative review between the Convolutional Neural Network and Support Vector Machine approaches, CNN models yielded a higher accuracy in recognizing specific ASL gestures [8].

Another application of hand gesture recognition is interactive gaming. This field pushes the boundaries of entertainment by allowing the user to control gameplay without hardware. Rather, various machine learning and computer vision techniques are employed to detect and interpret hand gestures in place of traditional hardware controls [9]. In one implementation, first the hand region is extracted from the image using skin color. The model, YCbCr, segments the image as per skin color to retrieve just the hand. Then, specific hand gestures are identified by computing the angles between detected fingers in the input image. An advantage of this approach is that there is no need for training samples unlike many other machine learning approaches [10]. This method works well when categorizing gestures into open hand versus closed hand, and the results of this paper show high accuracy in interactive game play applications.

Finally, both of the previous applications employ computer vision based approaches to detect the hand gestures. However, there are many hand gesture recognition algorithms which utilize sensors to detect these gestures. One paper explores 'data gloves', a simple alternative to vision based hand detection. In this approach, inertial sensors on the glove track finger placements. Then, a neural network is used with this data in order to map various placements to different hand gesture meanings [11].

2.2. Machine Learning Models

There are various machine learning models designed to perform these tasks. These algorithms train on a labeled dataset of images to complete image processing tasks such as classification, object detection, semantic segmentation, and instance segmentation. Classification is the task of assigning a specific category or label to a given image. Detection refers to identifying a bounding box for each object. Semantic segmentation is the process of assigning an object label to each pixel in an image, and instance segmentation builds upon this to additionally differentiate between various instances of the same object. In terms of gesture classification, semantic segmentation is a popular choice due to its powerful ability to distinguish between specific hand positions.

There is research supporting many different machine learning algorithms capable of semantic segmentation. Most of these build upon convolutional neural networks- or CNN's. Though CNNs are useful in image classification and processing tasks, their output is often a single label which classifies the input image. However, semantic segmentation tasks additionally require the bounds of each object within the image to be identified. Moreover, CNNs require a large amount of training data, which is impractical in various different fields [12]. In the case of input images consisting of multiple objects, a different method will need to be employed. Region based CNN, or R-CNN, was designed to address this. In this algorithm, object regions are identified and then passed into a convolutional neural network. This method supports the prediction of labels and bounding boxes for multiple objects in a single image [13]**.** However, since it requires the use of CNNs, it quickly becomes inefficient with complex inputs. As a result, Fast R-CNN and Faster R-CNN were proposed to improve the efficiency through the use of feature maps and a single CNN. Faster R-CNN uses an RPN- or region proposal network- to output a bounding box and its classification label for each object in an input image [12]**.**

One common algorithm for image processing tasks that builds upon this is Mask R-CNN. This was initially proposed for instance segmentation tasks. Mask R-CNN extends Faster R-CNN by optimizing this algorithm for semantic segmentation. It introduces pixel-pixel alignment and adds an object mask to the existing Faster R-CNN algorithm output. In this way, it is able to efficiently predict masks, labels, and bounding boxes for objects [14]. Another such algorithm that accomplishes semantic segmentation is UNet, originally developed for medical applications. UNet has a "U-shaped" architecture and works by utilizing feature mapping to build a segmented image output. This algorithm also utilizes data augmentation to provide strong results with smaller data sets [15]. Next, PSPNet, or the Pyramid Scheme Parsing Network, also builds upon this idea by providing a solution for scene parsing. Similarly to UNet, this network also employs the feature map extracted with a CNN. Then, it applies the pyramid pooling module to extract deeper contextual features from sub-regions which aids to produce the final output [16]. Finally, DeepLab is a family of deep learning models for semantic image segmentation. Each version incorporates new techniques to enhance the model performance. DeepLabV3 is a version of this model, and it applies dilated convolution and atrous convolution to efficiently extract multi-scale contextual information in input images. This technique is especially useful for semantic segmentation tasks, where objects can vary significantly in size and shape [17].

When designing models to carry out these segmentation tasks, it is also important to take pre-training into consideration. Pre-training and transfer learning are methods employed in machine learning to improve model performance. They are especially valuable when there is a smaller amount of labeled data. Pre-training consists of training a model on a large set of data to learn the general features of the data. After pre-training, the model is fine-tuned on a smaller dataset with labeled examples for a specific task. Transfer learning, on the other hand, entails adapting a pre-trained model for a different task by using it as a starting point and then training it further on a smaller task-specific dataset. Both pre-training and transfer learning can improve the accuracy of machine learning models and save computational resources and time, especially when labeled data is limited [18]. One such model that is commonly used in image processing tasks is ResNet, or Residual Network [19]. ResNet is a deep neural network which uses residual connections to enable training on deeper neural networks with improved accuracy. This network has been trained on ImageNet, a large dataset of labeled images. However, the need to utilize pre-training for strong results has been questioned by recent studies. Specifically, it has been shown that in object detection and image segmentation tasks, a Mask R-CNN model initialized randomly yields similar results as when initialized based on ImageNet pre-training [20]**.** Although the accuracy of results may not be vastly impacted, another study explored model robustness and uncertainty- finding that both of these were improved with pre-training [21].

From existing studies on terrestrial hand gesture classification tasks, UNET, PSPNet, DeepLabV3 and Mask R-CNN have emerged as the strongest models for such tasks [22]–[24]. Thus, they were chosen to be evaluated for the underwater pointing gesture classification in this project.

3. Objective

In this thesis, four machine learning models will be compared, namely UNET, PSPNet, DeepLabV3 and Mask R-CNN for pointing gesture classification. The overall objective is to identify a model which can be utilized for underwater pointing gesture communication tasks with high accuracy. Results with and without pretraining on ImageNet will be compared and analyzed using various evaluation methods. The results of this study will provide valuable insights into the effectiveness of these models for this underwater application and will further the exploration of underwater human-robot communication systems.

4. Approach

4.1. Dataset

To evaluate and compare models for underwater pointing gesture classification, underwater data were collected to compile a dataset of labeled images. Volunteers were recorded performing four right-handed gestures, as outlined in figure 1, at different angles to the camera in an indoor swimming pool. These images were then manually parsed into frames and annotated with a pointing gesture mask and class label to form a dataset of 3498 images (figure 2). This data was used in the next steps of the evaluation process.

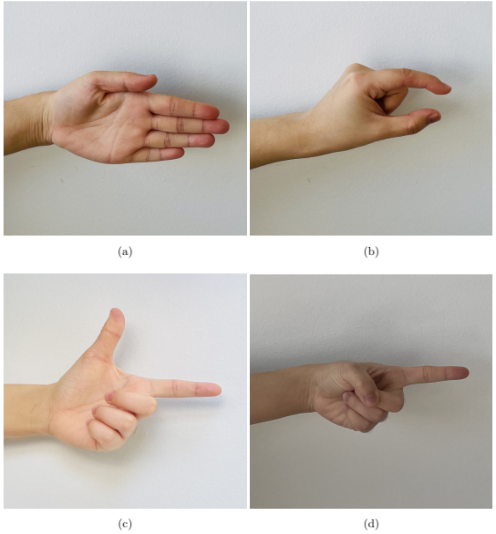


Figure 1. *The four types of* pointing *gestures in the study include: a) go somewhere, b) pick up, c) take a picture, d) general point*

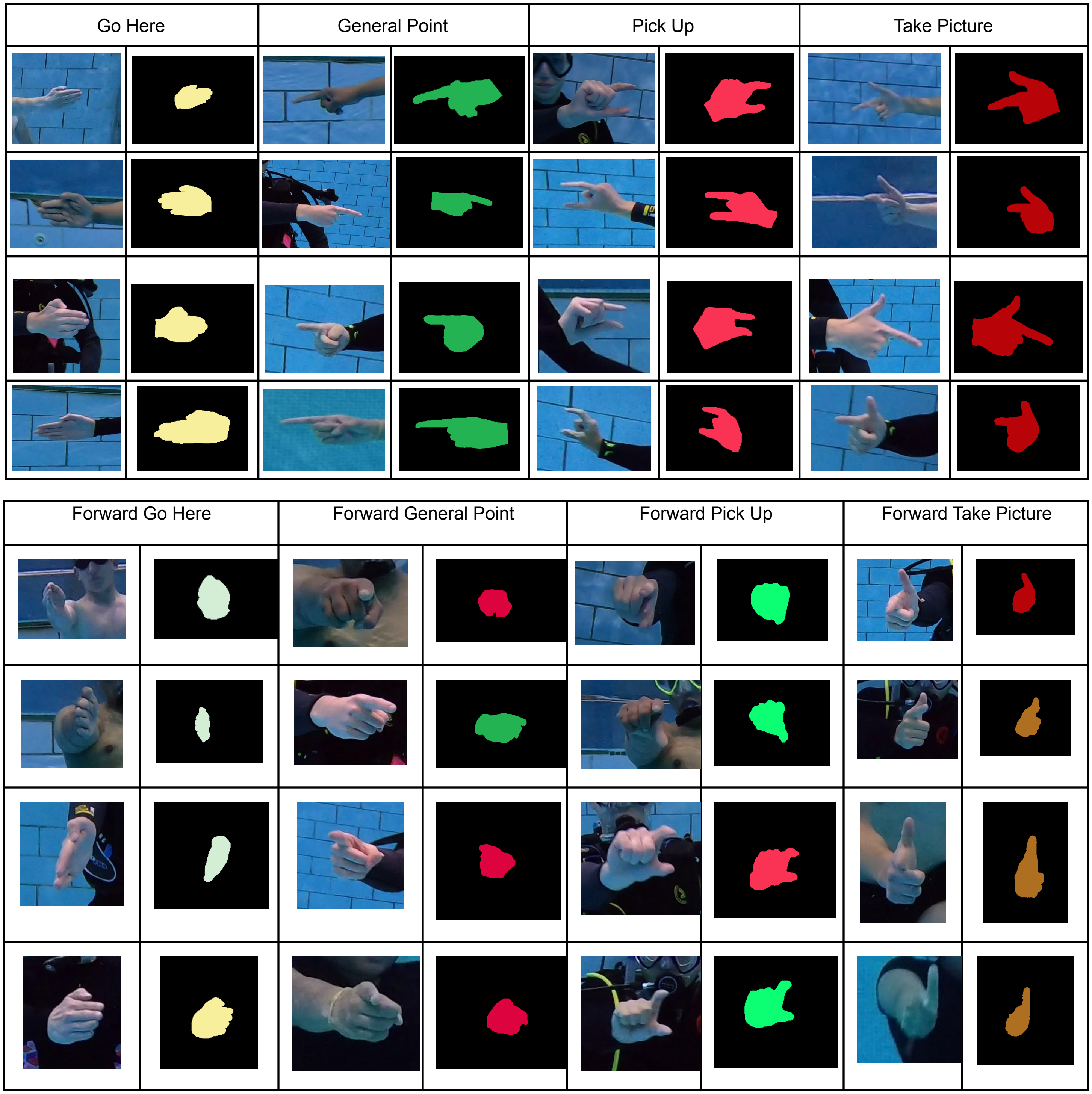
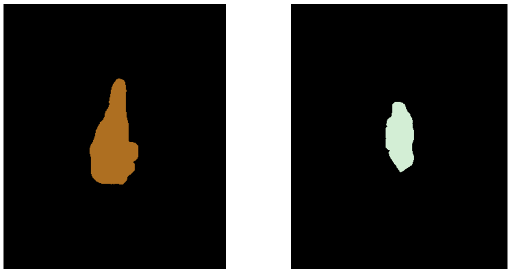


Figure 2. *Sample images and labeled masks from each of the classes of the dataset used in this project*

These images were then split into training, validation, and test sets in a 75% - 10% - 15% split. Each model was trained on set hyperparameters for a fair comparison on model strength. Four of these corresponded to the four different pointing gestures and the remaining four were the classes for forward pointing gestures. Since these forward pointing gestures were captured at a straight forward angle to the camera, only the fingertips of the pointing gesture were visible in most cases. Thus, manually drawing forward pointing gesture masks were difficult and their ground truth masks were significantly different from other masks for the same pointing gesture, as shown in figure 2. Moreover, the forward go there and forward take picture pointing gesture masks look fairly similar (figure 3). Therefore, to improve results, forward pointing gestures were labeled in separate classes.



Forward Take Picture Gesture

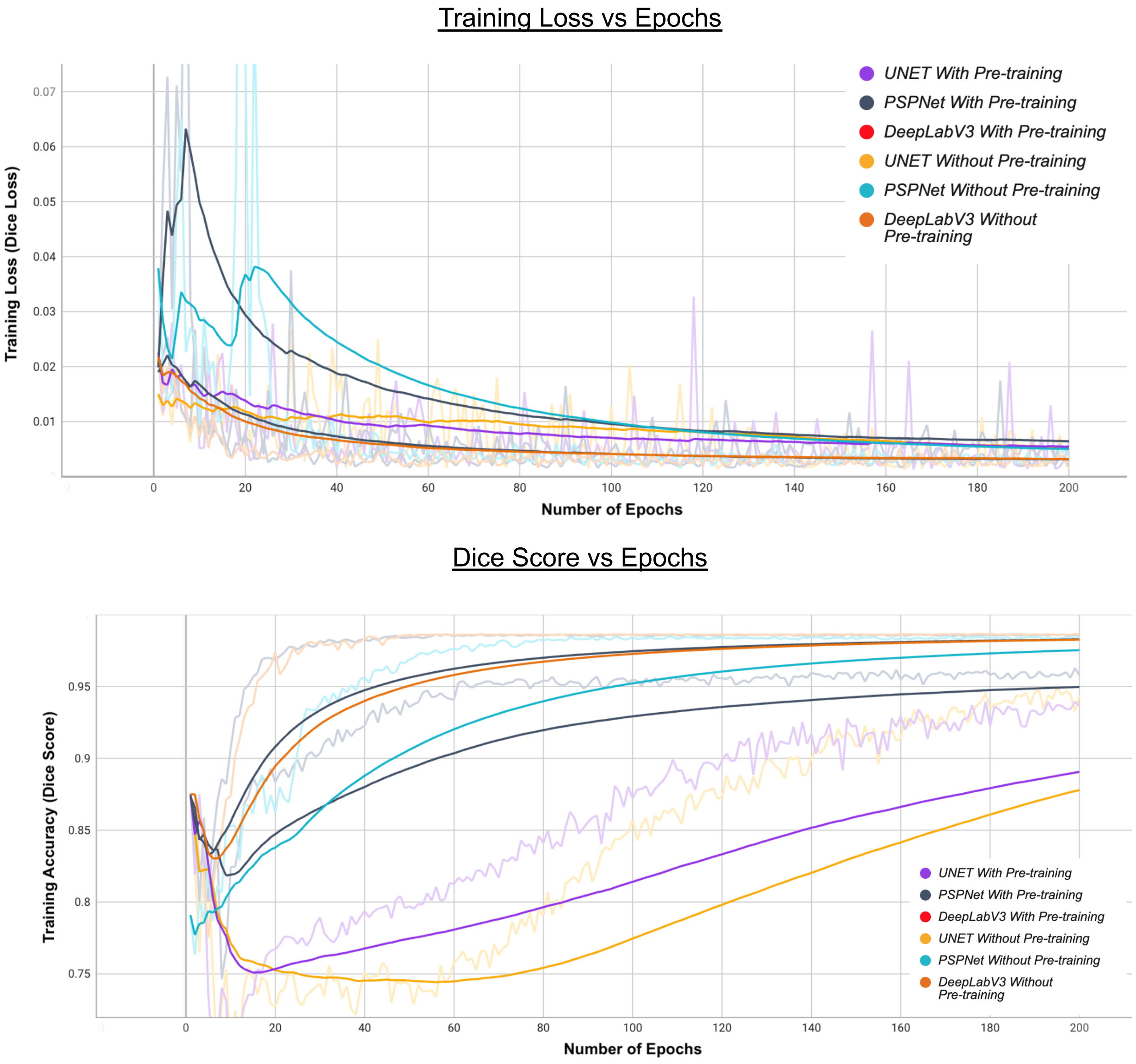
Forward Go Here Gesture

Figure 3. Forward go here and Forward take picture pointing gesture masks

4.2. Training

Each of the four models was run on 200 epochs with and without pretraining on ImageNet with an initial learning rate of 10^{-5}. A scheduler was also used to decrease the learning rate when the model stopped improving. As demonstrated in figure 4, the accuracy shows no significant improvement around 200 epochs and the validation loss plateaus around 100 epochs. Dice loss was used to validate the model. This loss function is region based and functions by calculating the similarity in predicted and expected mask regions [25]. During training, the best model was saved for evaluation.

Upon successful completion of the training stage, an evaluation script was run on each of the pretrained and non-pretrained models with the validation images. Various evaluation benchmarks including: dice similarity coefficient, intersection-over-union (IOU), and average precision (AP) score were computed on the models.


Training Loss vs Epochs

Dice Score vs Epochs

Figure 4. This illustrates the relationship between epochs, loss, and accuracy during the training of the models being evaluated.

5. Results

5.1. Evaluation Methods

The performance of the models were evaluated based on dice similarity coefficient, intersection-over-union (IOU), and average precision (AP) score. The dice coefficient and IOU measures indicate similarity between predicted and ground truth masks, while the AP score reflects the accuracy of predictions based on precision and recall values. The dice coefficient is calculated using the overlap area between the predicted and ground truth masks and the total number of pixels between the masks. The IOU is calculated using the overlap and union of the predicted and ground truth masks. Finally, the AP score is calculated by computing the area under the precision-recall curve. The computations for the three evaluation values are as follows and visualized in figure 5**:**

$IOU \;=\; \frac{|A \cap B|}{|A \cup B|}$, where A and B represent the prediction and ground truth respectively

Dice coefficient = 2 * (Precision * Recall) / (Precision + Recall)

AP = $\int_{k=0}^{n-1} precisions\,(k) \,*\, [recalls(k) \,-\, recalls(k\,+\,1)]$, n = thresholds

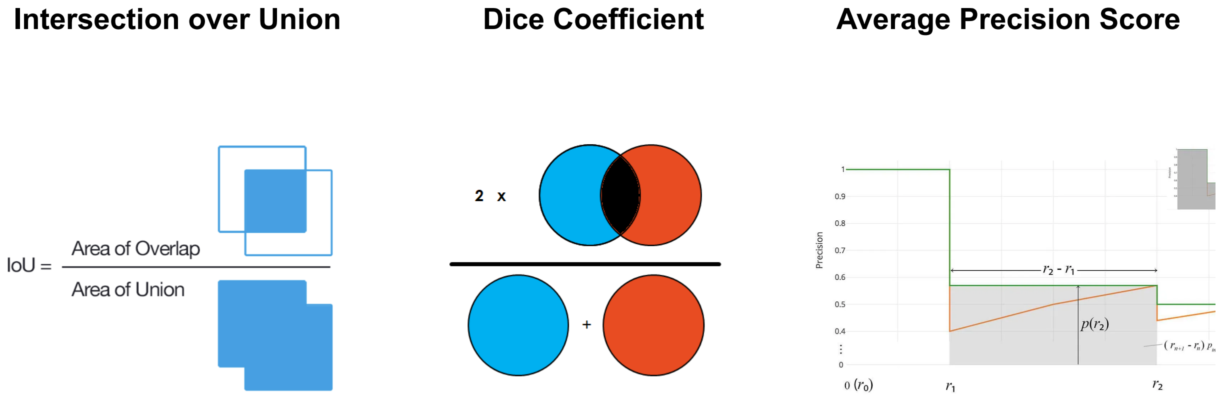

Intersection over Union **Dice Coefficient** **Average Precision Score**

Figure 5. This visually illustrates the calculations of IOU, Dice Coefficient and AP score. *Image source: Wikimedia*

5.2. Quantitative and Qualitative Analysis

The quantitative results are outlined in table 1**.** The table compares the dice coefficient, IOU, and AP scores for the semantic segmentation predictions of each model. It shows the scores from the Mask R-CNN, DeepLabV3, UNET, and PSPNet models with and without pretraining.

Model		Intersection over Union (IOU)	Dice Coefficient	Mean Average Precision (mAP)
With Pre-training	Mask R-CNN	–	.929	0.921
	UNET	0.9981	0.9269	0.9422
	PSPNet	0.9978	0.9756	0.9429
	DeepLabV3	0.9985	0.9717	0.9859
Without Pre-training	Mask R-CNN	–	.805	0.809
	UNET	0.9982	0.9319	0.9431
	PSPNet	0.9984	0.9814	0.9622
	DeepLabV3	.9984	0.9852	0.9708

Table 1. The quantitative results are outlined. The intersection over union, dice coefficient, and mean average precision evaluation metrics are shown.

The results suggest that the DeepLabV3 model with pre-training performs slightly better than the other 3 models for the underwater pointing gesture semantic segmentation task. This model showed the highest IOU and Mean Average Precision scores. However, the DeepLabV3 model without pre-training performs the second best, with only a slightly lower Mean Average Precision score. As the results demonstrate, there isn't a significant quantitative improvement in pretraining the UNET, PSPNet, or DeepLabV3 models. These results align with the findings in [19], that pretrained models may not always perform better than their counterparts. This could be due to significant differences between the model pre-training dataset, imagenet in this case, and the current dataset. On the other hand, the metrics suggest that the Mask R-CNN model performs better with pre-training, as reflected in the higher dice coefficient and mean average precision evaluation metrics. The qualitative results presented in table 2 and table 3 also provide visual insight into the performance of these models.

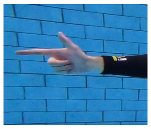
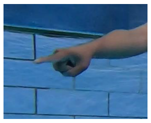
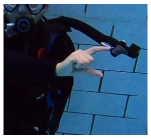
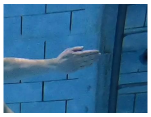
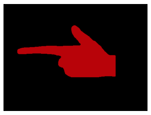
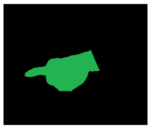
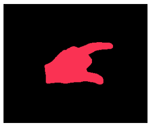
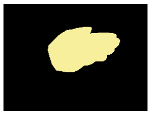
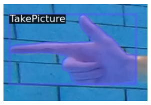
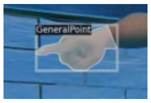
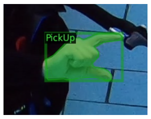
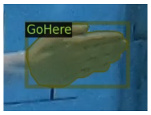
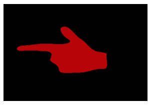
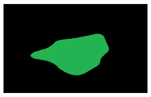
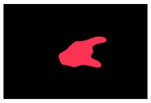
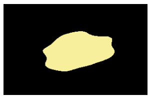
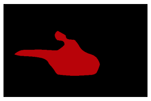
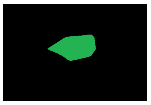
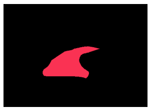
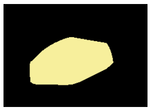
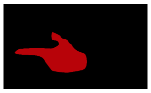
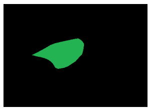
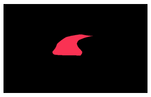
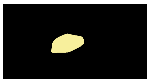
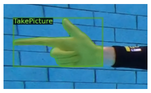
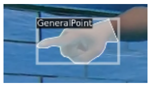
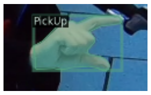
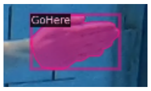
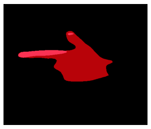
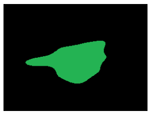
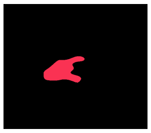
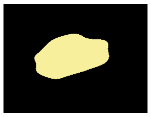
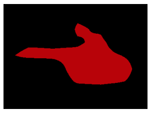
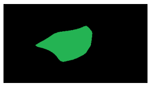
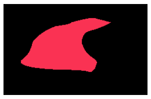
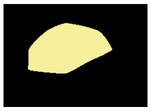
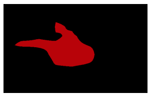
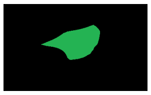
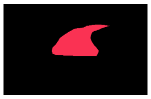
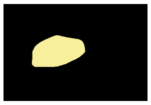
	Input				
	Ground Truth				
With Pre-training	Mask R-CNN				
	UNET				
	PSPNet				
	DeepLabV3				
Without Pre-training	Mask R-CNN				
	UNET				
	PSPNet				
	DeepLabV3				

Table 2. This shows the predictions of each model on an input image from various classes along with the ground truth mask.

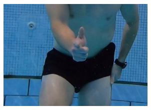
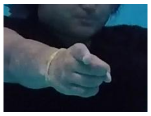
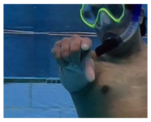
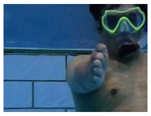
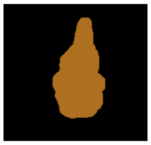
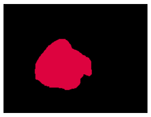
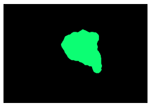
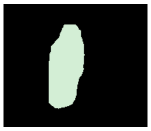
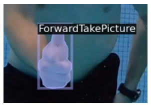
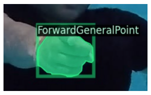
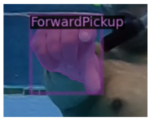
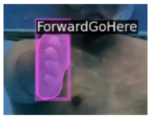
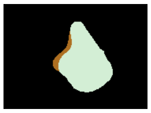
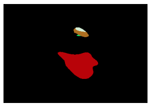
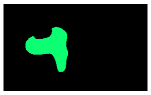
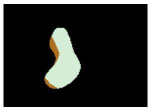
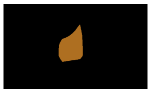
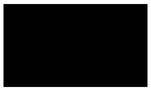
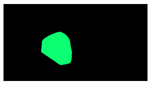
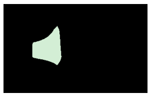
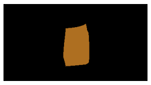
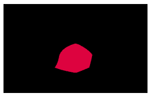
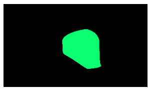
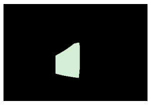
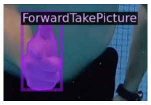
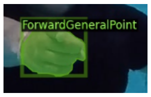
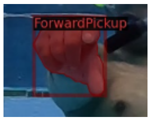
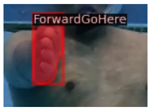
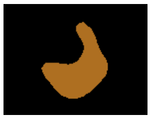
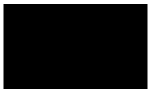
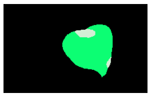
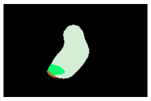
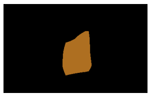
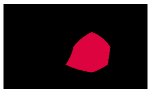
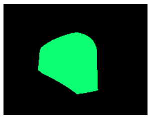
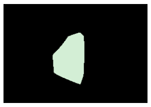
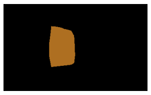
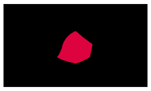
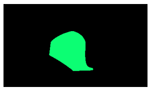
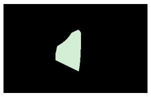
	Input				
	Ground Truth				
With Pre-training	Mask R-CNN				
	UNET				
	PSPNet				
	DeepLabV3				
Without Pre-training	Mask R-CNN				
	UNET				
	PSPNet				
	DeepLabV3				

Table 3. This shows the predictions of each model on an input image from each of the forward facing pointing gesture classes along with the ground truth mask.

The qualitative comparisons align with the quantitative ones. From table 2, it appears that there is no significant difference in mask predictions between pretrained models and their counterparts. Each mask color corresponds to a specific class, and tables 2 and 3 confirm that all of the models are able to correctly categorize each pointing gesture into its respective class. However, a visual comparison between tables 2 and 3 suggest that all of the models, with the exception of Mask R-CNN perform significantly better when tasked with predicting a pointing gesture mask at an indirect angle. As shown in table 3, the prediction masks belonging to the forward facing pointing gesture classes appear to vary significantly more than the ground truth. Given the visual results, Mask R-CNN seems to perform significantly better than the other models when predicting forward facing pointing gestures. Additionally, the pretrained PSPNet model and UNET model without pretraining both failed to predict the forward general point gesture. Overall, the qualitative and quantitative results suggest there is no overall significant performance difference between the models, but forward facing pointing gestures are better predicted by the Mask R-CNN models.

6. Conclusion

6.1. Summary

From the quantitative and qualitative results produced by the various models, DeepLabV3 model with pre-training emerged as the strongest model in terms of the intersection over union, dice coefficient, and average precision evaluation metrics. This model had an IOU score of 0.9985, a dice coefficient score of 0.9717, and an mAP score of 0.9859. Though all the models were able to correctly categorize the pointing gestures into their respective classes, the pretrained PSPNet model and UNET model without pre-training both failed to predict masks for the forward facing general pointing gesture. This, along with the quality of the other prediction masks, suggests that all the models perform better on predicting masks for non forward facing pointing gestures. However, given a forward facing pointing gesture input, the Mask R-CNN models perform the strongest.

6.2. Future Work

To expand upon the findings in this evaluation, a larger dataset should be trained including images from field settings. Since the dataset used in the evaluations within this paper consisted of lab data, the models may behave unexpectedly when given images with more diverse backgrounds, marine life, or debris. Future work would involve testing lab trained models on their ability to predict masks for field data. Finding a model that is able to generalize well to this new setting would be extremely beneficial in future underwater pointing gesture classification tasks. Another area of exploration is introducing augments within the training data. This would diversify the training data, and possibly improve the quality of mask predictions. Evaluating the performance of models trained on a set with augmentation would be another step in finding a stronger model for this task. Finally, exploring different model pre-training datasets would offer more insight into the results of this project. Understanding why various models perform similarly with and without pre-training can be useful in better fine tuning models for the underwater pointing gesture classification task.

7. Bibliography

[1] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, “Robotic Detection of Marine Litter Using Deep Visual Detection Models.” arXiv, Sep. 21, 2018. Accessed: Mar. 28, 2023. [Online]. Available: http://arxiv.org/abs/1804.01079

[2] B. Martinez *et al.*, “Technology innovation: advancing capacities for the early detection of and rapid response to invasive species,” *Biol. Invasions*, vol. 22, no. 1, pp. 75–100, Jan. 2020, doi: 10.1007/s10530-019-02146-y.

[3] A. G. Chavez, C. A. Mueller, T. Doernbach, D. Chiarella, and A. Birk, “Robust Gesture-Based Communication for Underwater Human-Robot Interaction in the context of Search and Rescue Diver Missions,” *J. Mar. Sci. Eng.*, vol. 7, no. 1, p. 16, Jan. 2019, doi: 10.3390/jmse7010016.

[4] M. Fulton, C. Edge, and J. Sattar, “Robot Communication Via Motion: A Study on Modalities for Robot-to-Human Communication in the Field,” *ACM Trans. Hum.-Robot Interact.*, vol. 11, no. 2, pp. 1–40, Jun. 2022, doi: 10.1145/3495245.

[5] A. M. Walker, “Towards Natural Underwater Human-Robot Interaction: Pointing Gesture Recognition for Autonomous Underwater Vehicles,” May 2021, Accessed: Mar. 28, 2023. [Online]. Available: http://conservancy.umn.edu/handle/11299/225596

[6] C. Edge *et al.*, “Design and Experiments with LoCO AUV: A Low Cost Open-Source Autonomous Underwater Vehicle.” arXiv, Mar. 19, 2020. Accessed: Apr. 06, 2023. [Online]. Available: http://arxiv.org/abs/2003.09041

[7] S. Chavan, X. Yu, and J. Saniie, “Convolutional Neural Network Hand Gesture Recognition

for American Sign Language," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, Mt. Pleasant, MI, USA: IEEE, May 2021, pp. 188–192. doi: 10.1109/EIT51626.2021.9491897.

[8] V. Jain, A. Jain, A. Chauhan, S. S. Kotla, and A. Gautam, "American Sign Language recognition using Support Vector Machine and Convolutional Neural Network," *Int. J. Inf. Technol.*, vol. 13, no. 3, pp. 1193–1200, Jun. 2021, doi: 10.1007/s41870-021-00617-x.

[9] L. Zulpukharkyzy Zholshiyeva, T. Kokenovna Zhukabayeva, S. Turaev, M. Aimambetovna Berdiyeva, and D. Tokhtasynovna Jambulova, "Hand Gesture Recognition Methods and Applications: A Literature Survey," in *The 7th International Conference on Engineering & MIS 2021*, Almaty Kazakhstan: ACM, Oct. 2021, pp. 1–8. doi: 10.1145/3492547.3492578.

[10] K. Li, J. Cheng, Q. Zhang, and J. Liu, "Hand Gesture Tracking and Recognition based Human-Computer Interaction System and Its Applications," in *2018 IEEE International Conference on Information and Automation (ICIA)*, Aug. 2018, pp. 667–672. doi: 10.1109/ICInfA.2018.8812508.

[11] H. Min, C. Chen, S. Huang, X. Tian, Y. Yang, and Z. Wang, "Highly-accurate gesture recognition based on ResNet with low-budget data gloves," in *2021 3rd International Conference on Advanced Information Science and System (AISS 2021)*, Sanya China: ACM, Nov. 2021, pp. 1–6. doi: 10.1145/3503047.3503153.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." arXiv, Jan. 06, 2016. Accessed: Mar. 13, 2023. [Online]. Available: http://arxiv.org/abs/1506.01497

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation." arXiv, Oct. 22, 2014. Accessed: Mar. 06, 2023. [Online]. Available: http://arxiv.org/abs/1311.2524

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN." arXiv, Jan. 24, 2018. Accessed: Mar. 13, 2023. [Online]. Available: http://arxiv.org/abs/1703.06870

[15] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv, May 18, 2015. doi: 10.48550/arXiv.1505.04597.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid Scene Parsing Network.” arXiv, Apr. 27, 2017. Accessed: Mar. 06, 2023. [Online]. Available: http://arxiv.org/abs/1612.01105

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs.” arXiv, May 11, 2017. Accessed: Mar. 13, 2023. [Online]. Available: http://arxiv.org/abs/1606.00915

[18] Avinash, “Pre-Trained Machine Learning Models vs Models Trained from Scratch,” *Medium*, Sep. 23, 2021. https://heartbeat.comet.ml/pre-trained-machine-learning-models-vs-models-trained-from-scratch-63e079ed648f (accessed Mar. 14, 2023).

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/1512.03385

[20] K. He, R. Girshick, and P. Dollár, “Rethinking ImageNet Pre-training.” arXiv, Nov. 21, 2018. Accessed: Mar. 14, 2023. [Online]. Available: http://arxiv.org/abs/1811.08883

[21] D. Hendrycks, K. Lee, and M. Mazeika, “Using Pre-Training Can Improve Model Robustness and Uncertainty,” in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 2712–2721. Accessed: Mar. 14, 2023. [Online]. Available: https://proceedings.mlr.press/v97/hendrycks19a.html

[22] “Smart training: Mask R-CNN oriented approach | Elsevier Enhanced Reader.” https://reader.elsevier.com/reader/sd/pii/S0957417421009957?token=A5CDF8BD7301919A2E3F867906ADC24D91BBCF761E7BBA911E7711F6FE8F6773BD4CB5D59E4230FF5EE9C3AC9F3CC9FE&originRegion=us-east-1&originCreation=20230328222625 (accessed Mar. 28, 2023).

[23] T.-H. Tsai and S.-A. Huang, “Refined U-net: A new semantic technique on hand segmentation,” *Neurocomputing*, vol. 495, pp. 1–10, Jul. 2022, doi: 10.1016/j.neucom.2022.04.079.

[24] A. Dadashzadeh, A. T. Targhi, M. Tahmasbi, and M. Mirmehdi, “HGR-Net: A Fusion Network for Hand Gesture Segmentation and Recognition.” arXiv, Dec. 28, 2019. Accessed: Mar. 28, 2023. [Online]. Available: http://arxiv.org/abs/1806.05653

[25] S. Jadon, “A survey of loss functions for semantic segmentation,” in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Oct. 2020, pp. 1–7. doi: 10.1109/CIBCB48159.2020.9277638.