MOG: a background extraction approach for data augmentation of time-series images in deep learning segmentation

Jonas Nagell Borgersen[1], Aya Saad[1], Annette Stahl[1]

[1]Dept. of Engineering Cybernetics, Norwegian University of Science and Technology, NTNU, Trondheim, Norway

ABSTRACT

Image segmentation is one of the key components in systems performing computer vision recognition tasks. Various algorithms for image segmentation have been developed in the literature. Among them, more recently, deep learning algorithms have been remarkably successful in performing this task. A downside with deep neural networks for segmentation is that they require a large amount of labeled dataset for training. This prerequisite is one of the main reasons that led researchers to adopt data augmentation approaches in order to minimize manual labeling efforts while maintaining highly accurate results. This paper uses classical non-deep learning methods for background extraction to increase the size of the dataset used to train deep learning attention segmentation algorithms when images are presented as time-series to the model. The method presented adopts the Gaussian mixture-based (MOG2) foreground-background segmentation followed by dilation and erosion to create masks necessary to train the deep learning models. It is applied in the context of planktonic images captured in situ as time series. Various evaluation metrics and visual inspection are used to compare the performance of the deep learning algorithms. Experimental results show higher accuracy achieved by the deep learning algorithms for time-series image attention segmentation when the proposed data augmentation methodology is utilized to increase the training dataset.

Keywords: data augmentation, image analysis, deep learning, background extraction algorithms, in-situ plankton taxa classification, segmentation

1. INTRODUCTION

Image segmentation is a classical problem in computer vision. Proposed algorithms in the literature for image segmentation, such as in [1, 2], tend to assign each pixel in a given image a class. In recent years, deep convolutional neural networks for semantic segmentation have achieved improved segmentation accuracy over classical, non-deep-learning methods. The introduction of the Fully Convolutional Network (FCN) architecture [3] and the legendary U-net [4] showed great potential for deep neural networks in image segmentation. Several other network architectures such as the DeeplabV3+ [5] and the Unet++ [6], among many others, have since shown improved results over the U-net and the FCN for various segmentation tasks. However, there are certain drawbacks with the deep neural networks for semantic segmentation, e.g., the networks require a sizable training dataset of correctly labeled images to achieve satisfactory accurate results. Attempts for data augmentation were proposed in the literature to compensate for the small datasets used in the training process of the deep learning models. Those attempts generally exert some operations on the images provided in the dataset for training; such operations include geometric transformations, color space augmentations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, generative adversarial networks, neural style transfer, and meta-learning [7]. Still, data augmentation efforts do not consider the massive amount of unlabeled data captured during the data collection process, especially in the context when the data is provided as time-series.

The work presented in this paper aims at performing in-situ detection of planktonic data captured as time-series by a mobile platform, lightweight autonomous vehicle (AUV), described in [8]. The entire process of pixel-wise classification and detection of plankton could be done through instance segmentation [9, 10]. However, deep learning methods for instance segmentation detect planktonic organisms that belong to predefined classes manually labeled by domain experts. In real-life situations, the AUV may encounter thousands of planktonic species [11] that have never been seen nor included in the labeled dataset.

The contributions in this paper consist of implementing data augmentation in a novel manner, making use of the massive amount of the data captured in situ. At the same time, minimize the manual labeling efforts exerted by domain experts, which is time-consuming and subject to human error. The data augmentation method utilized adopts classical background extraction methods over images captured as time series to perform attention segmentation by extracting the foreground objects from the background. The output of this segmentation module consists of masks used to increase the dataset for the deep learning segmentation algorithm. The method presented applies the Gaussian mixture-based (MOG2) [12, 13] to perform foreground-background segmentation followed by dilation and erosion [14] to remove detected noise in the masks. The reason behind applying this method on the time series captured images is that such algorithms do not require labeled images for training. At the same time, they provide decent up-sizing to the dataset aiming at improving the accuracy of the deep learning segmentation model.

The deep learning semantic segmentation network architectures are trained on the datasets that contain both types of masks manually and automatically generated. The original manually labeled dataset consists of 312 planktonic image scenes with corresponding ground truth image masks labeled by domain experts, biologists. The manually labeled masks are not a perfect pixel-by-pixel representation of foreground and background; most of the extracted regions detected as foreground are larger than the actual object size due to human error. Furthermore, some planktonic organisms present in images were mistakenly not labeled. As the manual labeling is not entirely accurate, we do not expect any model to produce perfect pixel-by-pixel attention segmentation on the data.

This paper further presents a comparison between the segmentation results of networks trained on manually labeled data with a network trained on the augmented dataset using Gaussian mixture-based (MOG2) segmentation to confirm the usefulness of the novel augmentation approach. Imperial results show that the proposed data augmentation methodology improves the trained models accuracy.

The rest of the paper is organized as follows: Section 2 introduces some preliminary knowledge related to this paper. Section 3 presents related work in data augmentation. Section 4 explains the methodolgy. Section 5 presents the experimental results. Finally, section 6 concludes the paper and presents some future directions.

2. BACKGROUND

Image segmentation is the process of partitioning an image into multiple segments or objects aiming at analyzing and understanding the image and its context for computer vision recognition systems [15]. This section covers the essential background methods for image segmentation proposed in the literature and utilized in this paper.

2.1 Classical segmentation

Classical segmentation refers to methods that do not rely on deep learning algorithms. There is a broad spectrum of classical segmentation methods based on thresholding, morphological operators, edge detection, color space or background extraction. The segmentation algorithms, which are based thresholding techniques, measure the pixel intensity on the gray scale, then classify the scales based on global or local threshold values [16]. The segmentation methods based on edge detection detect discontinuity in the local features to generate maps with edges of objects [16]. On the other hand, the segmentation that relies on color spaces identifies different colors and maps them into separate classes.

Another approach for segmentation is based on background extraction algorithms. This approach can be used as a segmentation mechanism in applications where images are captured and provided as time series. In this technique, all detected objects that are static are classified as part of the background, while all moving particles are considered as foreground [17]. A Gaussian mixture-based (MOG2) model is proposed in [12, 13] utilizes properties of statistical distributions to create an improved background extraction algorithm that can overcome the complexity of the pixel value distribution in the image.

2.2 Deep learning segmentation models

Deep learning models have in recent years considerably increased performance for image segmentation tasks [15] over classical methods. The fully Convolutional network (FCN) proposed by Long *et al.* in [3] is considered a milestone for creating segmentation maps for images of varying dimensions using deep convolutional networks. However, the FCN is too slow when applied in real-time; moreover, it is not effective in capturing global contextual information [15]. In recent years, researchers have created a wide variety of deep learning models for image segmentation to improve the accuracy and efficiency of neural network models. This section evaluates a selection of deep learning models for semantic segmentation in the planktonic domain context.

2.2.1 U-net

U-net has obtained its name from the u-like structure of the network architecture. This architecture consists of one contracting path, an encoding path, and one expanding path, a decoding path. A key point in the U-net is that the feature maps generated in the contracting path are passed to the expanding path through skip connections. The skip connections help the expanding path construct the segmented output. The purpose of the contracting path is to capture contextual information, while the purpose of the expanding path is to construct the segmentation map output. The U-net architecture [4] was developed to solve a segmentation task with very little available training data, only 30 labeled images [4]. To generate a sufficient amount of training data, the author in [4] uses extensive data augmentation, and obtains significantly improved segmentation accuracy doing so.

2.2.2 Unet++

Unlike the Unet architecture, the U-net++ [6] introduces dense nested skip pathways between the encoder and decoder of the U-net architecture. The convolutional layers within the pathways are densely connected; this means that every neuron is connected to all neurons in the previous layer. The convolutional layers are preceded by concatenation layers which fuse the output from the preceding layer of the same dense block with the up-sampled output from the proceeding layer of the lower level dense block. This is done to make the encoder and decoder feature maps similar, under the assumption that this would lead to improved segmentation. U-net++ is in [6] trained and tested on four different datasets containing biomedical images. With the Intersection over Union (IoU) metric, it outperforms the original U-net on all four datasets.

2.2.3 Pyramid attention network

Just like the U-net, the Pyramid attention network [18] has an encoder decoder architecture. In addition, it utilizes a feature pyramid attention (FPA) module and global attention upsampling (GAU) modules. The proposed FPA module fuses features from different scales, to give more accurate segmentation. The GAU modules provides global context for decoding. Together with the encoder, these modules help the network capture global context, while at the same time they capture different scale of the feature information. A strength of the pyramid attention network is the ability to localize and classify small objects.

2.2.4 Deeplab V3+

The Deeplab V3+ uses dialations, so called atrous convolution, in the decoding of the segmentation maps [5, 19]. This enables, more computational efficient decoding, with a larger field of view for filters. In addition, Deeplab V3+ utilizes spatial pyramid pooling for robust segmentation of objects at multiple scales. Deeplab V3+ also combines methods from deep convolutional networks with probabilistic graphical models in order to improve the localization of object boundaries [5, 19].

2.2.5 Linknet

For real time applications that apply semantic segmentation, the run time is crucial. Linknet [20] is a simple network architecture designed to reduce the run time for predictions. It is 10 times faster than Segnet [21], another light-weight network that achieves higher accuracy for commonly used Cityscapes datasets [22].

3. RELATED WORK

Data augmentation is commonly used in computer vision to improve generalization for deep learning models. This applies especially when there are few labeled training images available [7]. A model with little available training data tends to overfit, meaning that the model is well adapted to the training set, while the performance drops significantly on the validation set. Data augmentation increases the number of data points for training, decreasing the distance between the training set and the validation set. This fact often yields improved model performance on the validation set [7]. There exists several methods to avoid overfitting for small datasets without data augmentation, such as batch normalization, dropout, pre-training, and transfer learning. These methods focus on the network architecture to increase the ability of generalization [7]. Data augmentation, on the other hand, handles the problem of small-sized training data at its core by creating additional training data.

Data augmentations is carried out through oversampling or data warping. Oversampling involves creating synthetic data through methods like feature space augmentations, mixing images, and generative adversarial networks (GANs) [7]. Data warping preserves the original labeling of the images. Geometric transformations, neural style transforms, random erasing, and color transformations are some examples of data warping [7].

3.1 Data augmentation using basic image manipulations

Two widely used groups of data warping manipulations are color space transformations and geometric transformations. The former transformations involve changing the color space within training images to make models more robust towards variations in lighting and color [23]. On the other hand, geometric transformations change geometric properties of the training images to make the models more robust to changes in position and orientation [23]. Examples of geometric transformations are rotation and flipping. Oversampling techniques using basic image manipulations include mixing images and random erasing. Mixing images combines sections of images into synthetic images [24]. Random erasing is done by selecting patches of training images while assigning all pixel values randomly or with predefined values [25]. Random erasing is done to overcome overfitting due to some objects or parts of images being unclear [7].

3.2 Data Augmentation using deep learning methods

Neural networks have the ability to map high dimensional inputs into lower dimensional representations [7]. Low dimensional feature maps can be extracted and isolated, opening up possibilities to use the neural networks in data augmentation [7]. Oversampling can be exerted by Generative adversarial network (GANs) as an example of deep neural network. GANs are used to generate artificial instances from a dataset while retaining similar characteristics of the original dataset. Another method is called neural style transfers can recreate an image so that it is displayed in a different style, while still retaining the original image motive [7].

4. METHODOLOGY

Proposed approaches for semantic segmentation having deep neural networks as their backbone architectures are extensively studied in the literature. This paper provides the experimental results based on the implementation, training, and performance comparison of the following network architectures.

- U-net [4]
- U-net++ [6]
- Linknet [20]
- Deeplab V3+ [5]
- Pyramid Attention Network (PAN) [18]

The training set consists of 284 images, and the validation set consists of 28 images. The labeling is done manually by domain experts, biologists. The original image size is 2448x2050. Figure 1 shows two input images along with their corresponding manually labeled masks. It is evident that the masks do not represent a perfect segmentation map. Figure 1 shows that the regions representing plankton in the masks are slightly bigger than the actual planktonic organisms sizes.

The different deep neural networks for segmentation, mentioned above, are trained over 15 epochs with batch size set to 4 using the dataset containing only manually labeled masks. The images and their masks are downsized to 512x512 as a preprocessing step before the training. The training is exerted over different loss functions in order to optimize the segmentation accuracy. Fine tuning the loss function is an important step in the training process since there is no ideal loss function that generalizes to all segmentation tasks[26]. The utilized loss functions in the process are listed as: binary cross entropy loss (BCE), weighted BCE, focal loss, and lovasz loss. For the U-net and the U-net++, the optimal loss function is the weighted BCE with positive weights set to 2. For the other network architectures, the optimal loss function is the weighted BCE with positive weight set to 7.

Each network is trained with its optimal loss function. Transfer learning is applied by importing resnet-101 [27] pre-trained on imagenet [28]. The learning rate is set to 0.0001 and the Adam optimizer is used in the algorithm [29]. The following standard data augmentation is used for all images: rotation (0°-35°), horizontal flip and vertical flip. We compare dice coefficient, precision and recall for all networks. The results from this comparison is used to decide which network to use for further testing with a larger dataset containing masks generated by Gaussian mixture-based (MOG2) segmentation.

We modify two selected networks by using different encoders. This is done to see if we can obtain less complex models giving faster predictions without significantly decreasing the dice coefficient. In real time image processing, run time is

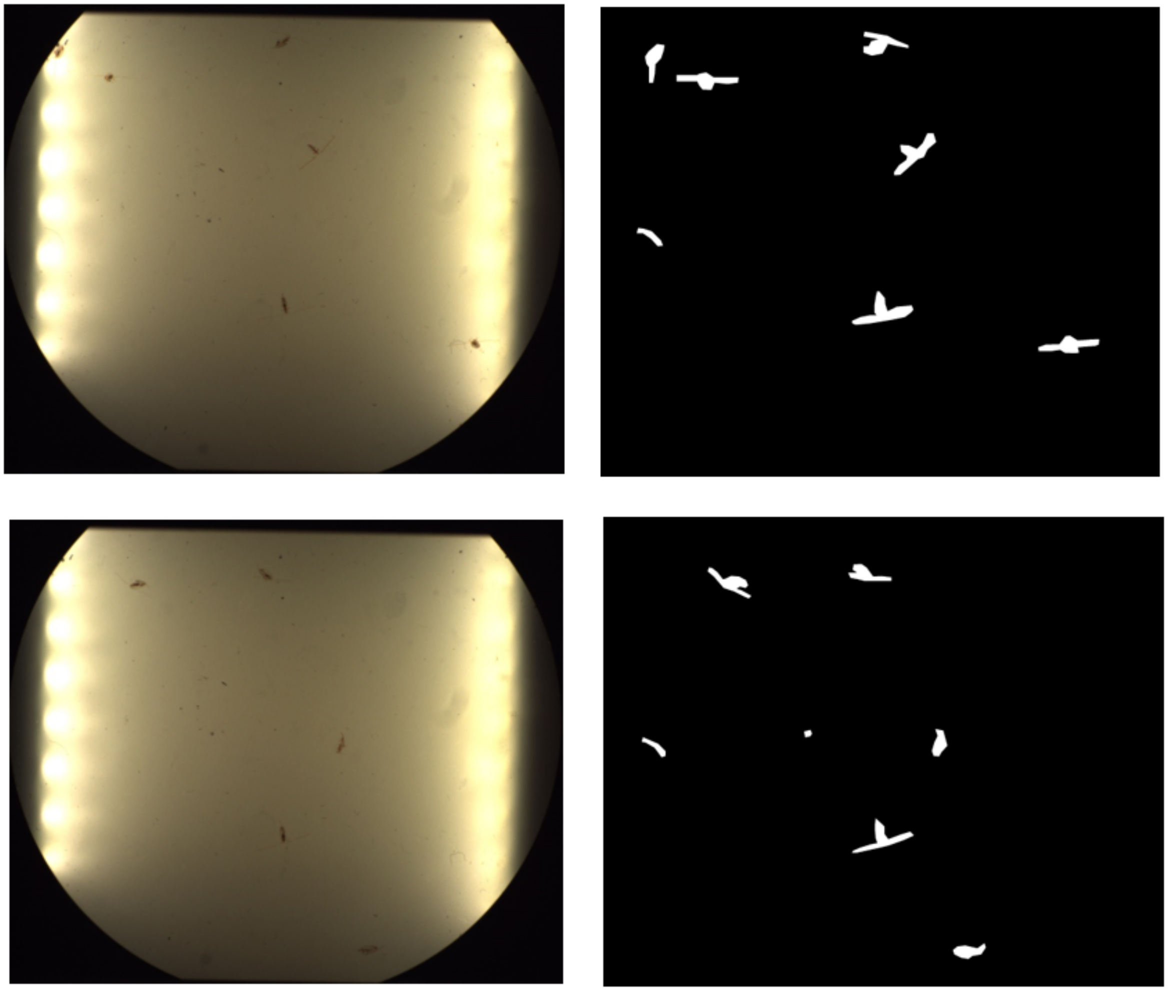

Figure 1: Input images and their corresponding manually labeled masks.

an important factor. We found that when using the pre-trained resnet 101 encoder, the Unet++ yields the highest dice coefficent, while the Linknet yields lowest run-time for prediction of images. We do further training for the U-net++ and the Linknet using different encoders. For U-net++ and Linknet we employ the following encoders; resnet 101 [27], resnet 34 [27], and mobilenet V2 [30].

The labeled dataset utilized is rather small with only 312 labeled images. Therefore we explore the use of Gaussian mixture-based (MOG2) segmentation to generate more training data. We carry out dilation [14] and erosion [14] to remove noise from the output masks. We use a 10×10 kernel for dilation and a 20×20 kernel for erosion. Gaussian mixture-based (MOG2) segmentation followed by dilation and erosion is carried out on the training data set, and can in that regard be viewed as a special case of data augmentation. Figure 2 shows the output from MOG2 before and after the removal of noise. We observe that 19% of the generated masks represent poor foreground and background segmentation; an example from the poor quality generated masks is shown in figure 3.

5. EXPERIMENTAL RESULTS

As a result of the training process, we observe that the Unet++ achieves the highest overall performance metrics when we train all networks using only manually labeled masks. We therefore choose the U-net++ for further training using the larger dataset containing both types of masks manually labelled, and masks generated through Gaussian mixture-based segmentation (MOG2). The hyper-parameters were fixed and the pre-trained resnet 101 is used as the encoder architecture. We measure the network accuracy using the dice coefficient, precision and recall.

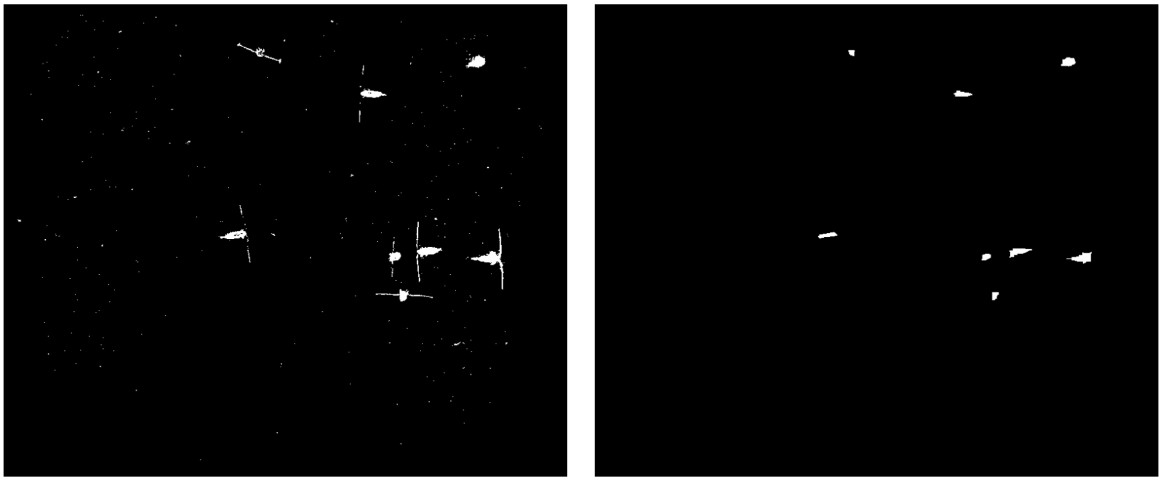

Figure 2: MOG2 output before and after noise removal

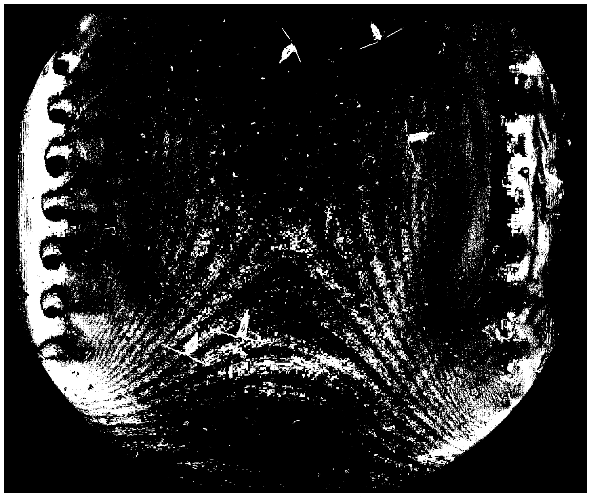

Figure 3: MOG2 poor quality mask output

We observe that the dice score decreases when we use the larger training set generated by the MOG2 algorithm. However we have seen that our original training data is not perfectly labeled. The masks have a slight overweight of false positives. The performance metrics, dice, precision and recall, are calculated by comparing the output masks from the model with manually labeled masks. This implies that a perfect segmentation will have high precision score, and a lower recall score. We observe that with our proposed MOG2 data augmentation we obtain increased precision and lowered recall. Figure 4 shows that this leads to improved segmentation. It is worth noting, that the antennas of the planktonic organisms in the masks to the right are thinner than those presented in the mask to the left. In reality, antennas of this species of plankton, copepods, are very thin. Both masks show accurate segmentation of the plankton species bodies.

Network architecture	Dice	Precision	Recall
U-net	0.7755	0.7704	0.7821
U-net++	0.8142	0.7837	0.8493
Linknet	0.7720	0.6756	0.9030
Deeplab V3+	0.7441	0.6558	0.8610
PAN	0.7515	0.6631	0.8696

Table 1: Performance metrics of the networks trained on the manually labeled dataset

Network architecture	Dice	Precision	Recall
U-net++	0.7779	0.8527	0.7170

Table 2: Performance metrics of the networks trained on the large dataset

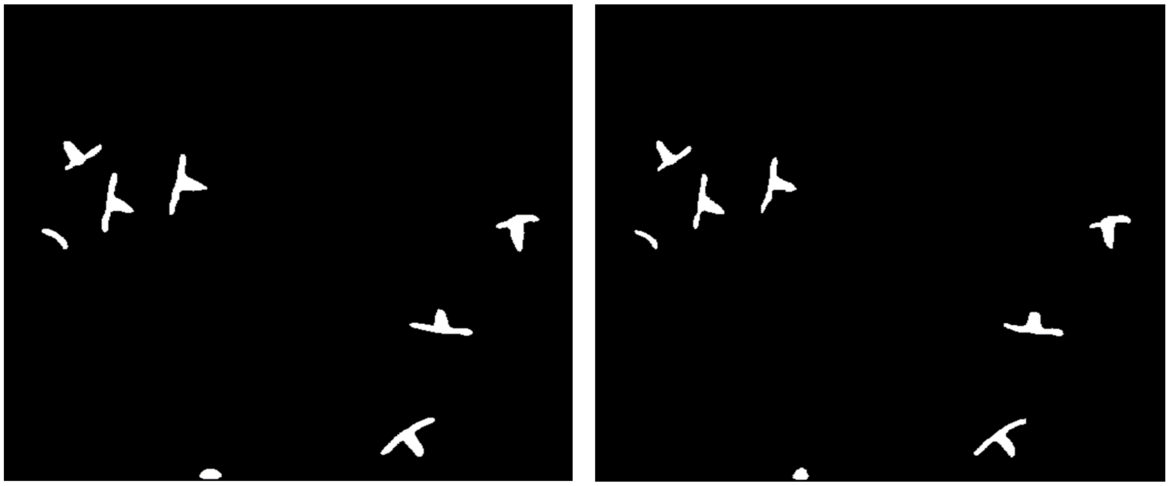

Figure 4: Resulting output masks U-net++. To the left is the result without applying the MOG2 augmentation, to the right the result after applying the MOG2 augmentation

6. CONCLUSION

This paper uses Gaussian mixture-based (MOG2) segmentation as a data augmentation approach to generate additional labeling in the training dataset used by deep learning attention segmentation models in their training process. The method is specifically applied to applications which provide captured images in a time series format. The idea is to extract the foreground from the background through excluding objects that are statically presented or not changing their positions in images placed in a sequence. As an example, we showcase the applicability of the method to the planktonic domain and more specifically to the platform described in [8]. We implement and train several deep neural network architectures for segmentation, and we train the networks using different loss functions to optimize their accuracy. Experimental results show that the use of the manually labelled masks augmented by the generated masks from the MOG2 yields improved segmentation. An interesting future direction would be to investigate other background extraction algorithms and use them for data augmentation. Another interesting future direction would be exploring the effect of this type of data augmentation on the different network architectures to determine which combination is more suited to which specific context and application.

ACKNOWLEDGMENTS

This research is funded by the Research Council of Norway (RCN) IKTPLUSS program (project number 262741) and supported by NTNU AMOS (RCN project number 223254).

REFERENCES

[1] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, “Semantic segmentation using regions and parts,” in 2012 IEEE conference on computer vision and pattern recognition, 3378–3385, IEEE (2012).

[2] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, “Attention-guided unified network for panoptic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7026–7035 (2019).

[3] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440 (2015).

[4] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention, 234–241, Springer (2015).

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proceedings of the European conference on computer vision (ECCV), 801–818 (2018).

[6] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in Deep learning in medical image analysis and multimodal learning for clinical decision support, 3–11, Springer (2018).

[7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data **6**(1), 1–48 (2019).

[8] A. Saad, A. Stahl, A. Våge, E. Davies, T. Nordam, N. Aberle, M. Ludvigsen, G. Johnsen, J. Sousa, and K. Rajan, "Advancing ocean observation with an ai-driven mobile robotic explorer," Oceanography **33**(3), 50–59 (2020).

[9] S. Bergum, A. Saad, and A. Stahl, "Automatic in-situ instance and semantic segmentation of planktonic organisms using mask r-cnn," in Global Oceans 2020: Singapore–US Gulf Coast, 1–8, IEEE (2020).

[10] A. Saad, S. Bergrum, and A. Stahl, "An instance segmentation framework for in-situ plankton taxa assessment," in Thirteenth International Conference on Machine Vision, **11605**, 1160511, International Society for Optics and Photonics (2021).

[11] P. Falkowski, "Ocean science: the power of plankton," Nature **483**(7387), S17–S20 (2012).

[12] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., **2**, 28–31, IEEE (2004).

[13] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," Pattern recognition letters **27**(7), 773–780 (2006).

[14] G. Bradski and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, " O'Reilly Media, Inc." (2008).

[15] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[16] S. Yuheng and Y. Hao, "Image segmentation algorithms overview," arXiv preprint arXiv:1707.02051 (2017).

[17] C. Wren, "Real-time tracking of the human body," SPIE proceeding, 1996 **2615**, 89–98 (1996).

[18] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," arXiv preprint arXiv:1805.10180 (2018).

[19] 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," IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017).

[20] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in 2017 IEEE Visual Communications and Image Processing (VCIP), 1–4, IEEE (2017).

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017).

[22] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in Proceedings of the IEEE conference on computer vision and pattern recognition, 3213–3223 (2016).

[23] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," arXiv preprint arXiv:1708.06020 (2017).

[24] H. Inoue, "Data augmentation by pairing samples for images classification," arXiv preprint arXiv:1801.02929 (2018).

[25] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in Proceedings of the AAAI Conference on Artificial Intelligence, **34**(07), 13001–13008 (2020).

[26] S. Jadon, "A survey of loss functions for semantic segmentation," in 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 1–7, IEEE (2020).

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778 (2016).

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, 248–255, Ieee (2009).

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 4510–4520 (2018).