COASTAL HABITAT MAPPING WITH UAV MULTI-SENSOR DATA: AN EXPERIMENT AMONG DCNN-BASED APPROACHES

Yi Liu[1]*, Qinghui Liu[1], James Edward Sample[2], Kasper Hancke[2], Arnt-Børre Salberg[1]

[1] Norwegian Computer Center, P.O. Box 114 Blindern, No-0314 OSLO, Norway
[2] The Norwegian Institute for Water Research, Økernveien 94, 0579 Oslo, Norway

Commission III, WG III/10

KEY WORDS: deep learning, semantic segmentation, UAV, multi-sensor data, data fusion, environmental monitoring

ABSTRACT:

With recent abundant availability of high resolution multi-sensor UAV data and rapid development of deep learning models, efficient automatic mapping using deep neural network is becoming a common approach. However, with the ever-expanding inventories of both data and deep neural network models, it can be confusing to know how to choose. Most models expect input as conventional RGB format, but that can be extended to incorporate multi-sensor data. In this study, we re-implement and modify three deep neural network models of various complexities, namely UNET, DeepLabv3+ and Dense Dilated Convolutions Merging Network to use both RGB and near infrared (NIR) data from a multi-sensor UAV dataset over a Norwegian coastal area. The dataset has been carefully annotated by marine experts for coastal habitats. We find that the NIR channel increases UNET performance significantly but has mixed effects on DeepLabv3+ and DDCM. The latter two are capable of achieving best performance with RGB-only. The class-wise evaluation shows that the NIR channel greatly increases the performance in UNET for green, red algae, vegetation and rock. However, the purpose of the study is not to merely compare the models or to achieve the best performance, but to gain more insights on the compatibility between various models and data types. And as there is an ongoing effort in acquiring and annotating more data, we aim to include them in the coming year.

1. INTRODUCTION

After its development for military applications, unmanned aerial vehicle (UAV) has become a popular tool for civil applications [Pajares, 2015]. As UAVs can also operate at much lower altitudes as to satellites, equipped with various sensors, it provides a flexible and cost-effective approach is to acquire a lot of high resolution information over an area of interest.

As part of a Norwegian infrastructure project, the goal is to establish drone-based mapping and monitoring of the coastal environment. Automated image analysis via deep learning needs to be implemented for mapping habitats including seafloor substrate types, subsurface vegetation and other management-relevant species. This task of mapping every pixel in an image can be referred as *semantic* segmentation in computer vision.

The current main stream approach is based on Deep Convolutional Neural Networks (DCNNs) [LeCun et al., 1989, Krizhevsky et al., 2012, Simonyan and Zisserman, 2015, Szegedy et al., 2015, He et al., 2016, Sandler et al., 2019], deployed in a fully convolutional fashion [Long et al., 2015]. These Fully convolutional Networks (FCNs) replaces the fully-connected layer in a classification network with convolution layers, effectively extends image-level classification to pixel-level classification.

The repeated combination of max-pooling and striding of consecutive layers of DCNNs [LeCun et al., 1989, Long et al., 2015] is a most common technique in DCNN models, but the repeated use of it is known to significantly reduces the spatial resolution of the resulting feature maps. Two different approaches have been employed to address this problem. One of them is by using transposed convolution or upsampling [Zeiler et al., 2011, Noh et al., 2015, Long et al., 2015, Ronneberger et al., 2015]. In-network upsampling has been observed [Long et al., 2015] to be fast and effective for learning dense prediction. A typical example for using upsampling is UNET [Ronneberger et al., 2015], which has an symmetric encoder-decoder architecture. The other approach is by using dilated convolution (atrous convolution) [Holschneider et al., 1990, Chen et al., 2017]. And a well-known example for using atrous convolution is DeepLabv3 [Chen et al., 2018b], which has an asymmetric encoder-decoder architecture. Its Astrous Spatial Pyramid Pooling (ASPP) module uses atrous convolution and pooling operation to capture features at multiple scales. [Chen et al., 2017] further extend Deeplabv3 to Deeplabv3+ to recover detailed object boundaries by concatenating the low-level features with bilinearly upsampled features from ASPP encoder. Instead of a parallel design in ASPP module, [Liu et al., 2020b] present Dense Dilated Convolutions Merging (DDCM) Network which employs dilated convolution in a cascading structure [Yu and Koltun, 2016a]. Their model achieves good performance on ISPRS Potsdam and Vaihingen data [Kaiser et al., 2017], as well as the DeepGlobe land cover [Demir et al., 2018] dataset.

Due to the nature of CNN structures, the receptive field is limited to local regions [Luo et al., 2016], which imposes an adverse effect on the performance of FCNs. To address this, several approaches have been proposed. Dilation convolution [Holschneider et al., 1990] operation is a common technique to expand the receptive field [Yu and Koltun, 2016b, Chen et al., 2017, Zhao et al., 2018, Liu et al., 2020b], but there is a heavy overhead cost for large dilation rate. The use of pooling operation is another approach for capturing long-range dependency. Global average pooling module is proposed in ParseNet [Liu et al., 2015], different-dilation based atrous spa-

* Corresponding author: yiliu@nr.no

tial pyramid pooling (ASPP) module is in DeepLab [Chen et al., 2018a] and different-region based pyramid pooling module (PPM) is in PSPNet [Zhao et al., 2017].

In terms of network structure, most models adopt an encoder-decoder structure [Badrinarayanan et al., 2017, Noh et al., 2015, Ronneberger et al., 2015], as it not only helps refine segmentation masks but also helps building contextual information.

Besides the complexity of networks, another factor that affects the result is the type of input data. As drones are equipped with multiple sensors, there is usually extra information such as multispectral data in addition to conventional RGB. How to make full use of the data available given the choices of many available neural network models is worth of investigation.

In this paper, we re-implement and adapt three encoder-decoder structure models, UNET [Ronneberger et al., 2015], DeepLabv3+ [Chen et al., 2018a] and Dense Dilated Convolutions Merging Network (DDCM) [Liu et al., 2020b], to include multi-sensor data. In particular, we train with conventional RGB and near infrared (NIR) data, and compare with the results that are trained with RGB only. A consistent ResNet50 [He et al., 2016] is used as backbone. The models are different in terms of the use of dilation, pooling operation, and how low-level features and high-level ones are combined. We show both overall and class-wise score for all models trained with and without the NIR data and discuss the result. In addition, an adaptive class weighting loss is implemented to account for the high imbalance of label categories.

2. METHODS

2.1 Deep Neural Network Models

2.1.1 UNET UNET [Ronneberger et al., 2015] features a symmetric encoder-decoder architecture that is originally proposed for cell segmentation in microscopy images, but has become a baseline model in remote sensing. According to its original design, the encoder (the red contracting path in Fig. 1a) consists of repeated convolutional layers (3x3, unpadded), followed by a rectified linear unit (ReLU) and max pooling (2x2 with stride 2) for downsampling. The resulting condensed high-level feature maps are then processed in the decoder (the green expansive path in Fig. 1a). In the decoder, each level consists of bilinear upsampling, followed by a 2x2 convolution, a concatenation with the cropped feature map from the corresponding encoder, and two 3×3 convolutions (each followed by a ReLU). In this implementation, the contracting path is replaced by the backbone ResNet50 (down to layer3).

2.1.2 DeepLabv3+ DeepLabv3+ [Chen et al., 2018a] features Atrous Spatial Pyramid Pooling (ASPP) module and the combination of dilated convolution and spatial pyramid pooling [Grauman and Darrell, 2005, Lazebnik et al., 2006, He et al., 2014, Zhao et al., 2017]. Dilated convolution can be viewed as a generalized convolution, which modifies filter's field-of-view by the rate value, and it has been widely used in modern convolution neural networks. The ASPP module (denoted by the red dashed box in Fig. 1b) applies several parallel dilated convolution with different rates and uses adaptive pooling subsequently to capturing multi-scale features.

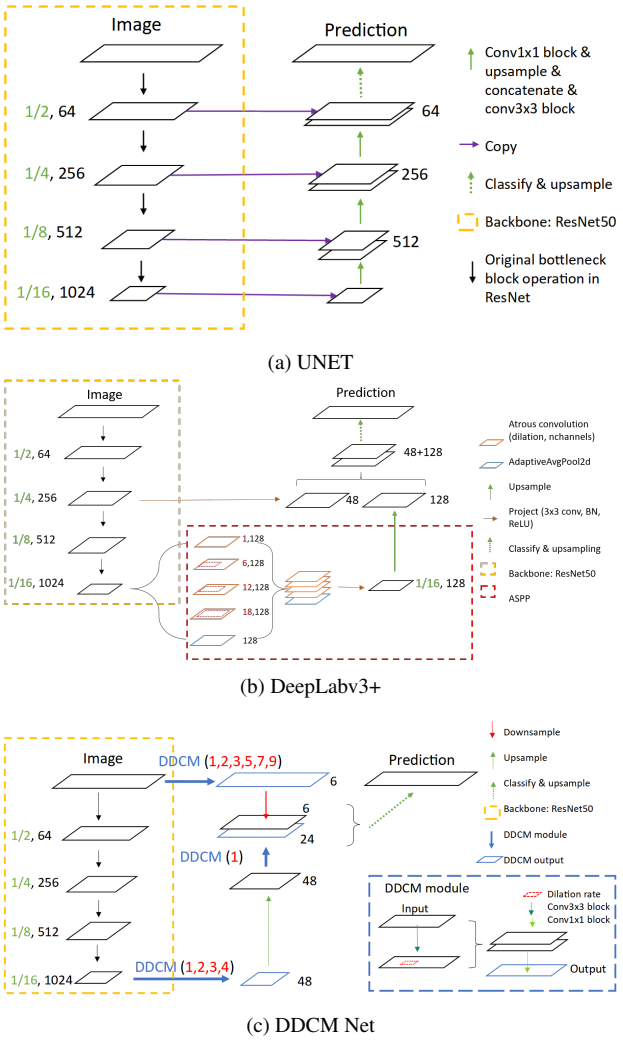

Figure 1. Architecture illustration of a) UNET, b) DeepLabv3+ and c) DDCM-Net. Upsample refers to bilinear upsampling. Backbone feature extraction is denoted by yellow dashed boxes. Model specific modules are denoted by colored dashed boxes. The size of feature maps are noted as fraction of its original size and reflected by the size of the planes. The numbers in green denotes image scale, red denotes dilation rate and black denotes number of feature channel.

2.1.3 Dense Dilated Convolutions Merging (DDCM) Network DDCM [Liu et al., 2020b] is another example of achieving contextual aggregation through dilated convolution. There are two differences compared to DeepLabv3+ architecture. One is the design of DDCM module. In its simplest form with only one dilation rate value, it consists of dilated convolution with the given rate, followed by PReLU [He et al., 2015] non-linear activation, batch normalization (BN) [Ioffe and Szegedy, 2015] and concatenation with the input. Given a sequence of rates, the DDCM module repeats this operation in a cascade manner, so the number of feature maps increases with the number of rates given. The rates are indicated by the red numbers in Fig. 1c. The final step in DDCM module consists of of 1×1 convolution, BN and PReLU to reduce the number of output features. The other difference is that the DDCM module is applied multiple times in the network (see Fig. 1c), first directly on the image level, then twice on the high-level features from the backbone layers. Maxpool2d is used as downsample method, indicated by the red arrow in Fig. 1c. The (1/2)x size feature maps extracted by DDCM in the two branches are concatenated as input to a 3x3 convolution layer, before applying bilinear interpolation to recover its full resolution (denoted by blue dashed arrow as "classify and upsample" in Fig. 1).

2.1.4 Multi-channel input modification To make the models general for UAV multi-sensor data, the number of input channels in the backbone is modified. And in addition, to benefit from a pre-trained network, the initial weights of the first three channels (RGB) is kept and copied for the extra channel. For the experiment in this study, all three models are implemented based on the architectures illustrated in Fig. 1, with a pre-trained ResNet50 as backbone, and trained using consistent protocols for comparison.

2.2 Adaptive Class Weighting Loss

2.2.1 Main loss function As label classes are highly imbalanced in this dataset, how to obtain meaningful updates on the weights for the minority classes needs to be addressed for efficient training. We address this problem by using an adaptive class weight loss. A customized loss function [Liu et al., 2020a], where a median frequency [Eigen and Fergus, 2015] class weight sampling method based on iterative batch-wise class rectification [Kampffmeyer et al., 2016], is used. The total loss function is formulated as a combination of a positive and negative class balance (PNC) function $\mathcal{L}_{pnc}$ and dice loss $\mathcal{L}_{dice}$ [Milletari et al., 2016],

$$\mathcal{L}_{acw} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}(w_{i,j}\mathcal{L}_{pnc}) + \log\left(\frac{1}{C}\sum_{j=1}^{C}\mathcal{L}_{dice}\right) \quad (1)$$

where $w_{i,j}$ is the pixel-wise adaptive class weights for the i-th pixel of the j-th class, N the total number of pixels, C the total number of classes, and $*$ denotes element-wise multiplication.

The PNC function is based on the L2 least squares error

$$\mathcal{L}_{pnc} = \mathcal{L}_2 - \log\left(\frac{1-\mathcal{L}_2}{1+\mathcal{L}_2}\right), \quad (2)$$

where $\mathcal{L}_2 = \sum_j^C |y_{i,j} - \tilde{y}_{i,j}|_2^2$, with $y_{i,j} \in (0,1)$ as the probability of the i-th pixel to be j-th class and $\tilde{y}_{i,j} \in \{0,1\}$ the ground truth.

The dice loss emphasizes the measure of intersection over union and can be written as

$$\mathcal{L}_{dice} = 1 - \frac{2\sum_i^N y_{i,j}\tilde{y}_{i,j}}{\sum_i^N y_{i,j}^2 + \sum_i^N \tilde{y}_{i,j}^2}\ . \quad (3)$$

2.2.2 Iterative median frequency class weights We first compute the pixel frequency of class j at iteration $n \in \{1,2,3,...\}$

$$f_j^n = \frac{\hat{f}_j^n + (n-1)f_j^{n-1}}{n} \quad (4)$$

where $\hat{f}_j^n$ denotes the pixel frequency of the j-th class, the number of pixels of class j divided by the total number of pixels, at the current n-th iteration,and $f_j^0 = 0$. Then we update the iterative median frequency class weights by

$$fw_j^n = \frac{\text{median}(\{f_j^n | j \in C\})}{f_j^n + \epsilon} \quad (5)$$

with a damping factor ϵ of 1e-5. Finally, the pixel-wise adaptive class weights is computed by

$$w_{i,j} = \frac{fw_j^n}{\sum_j(fw_j^n)}(1 + y_{ij} + \tilde{y}_{ij})\ , \quad (6)$$

3. EXPERIMENTS

3.1 Dataset

The data are acquired over the coast of Akerøya (shown in Fig. 2) on 27 August 2019. The sensors used are Sensefly S.O.D.A. 3D RGB sensor and Micasense RedEdge-MX Multispectral sensor, with flight altitude of 85 m and 117 m, respectively. After pre-processing procedures such as orthorectification, image stitching, radiometric calibration quality assessment of the data, removal of personal/sensitive information and metadata, a total of six sub-images are made available to us.

The raw RGB dataset is supplied as a single, 4-band (RGBA) GeoTiff with 2.2 cm cell resolution and 8-bit uint data type. The multispectral dataset has a 9.3 cm cell resolution and values are stored as 32-bit float data type. Due to the differences, a single, consistent dataset containing all the bands of interest is first created, using a 5 cm cell resolution and a bit-depth conversion of 32-bit to 8-bit. The combined dataset has 8 bands (rgb-red, rgb-green, rgb-blue, multispec-red, multispec-green, multispec-blue, multispec-nir and multispec-rededge). We will focus on the RGB and NIR in this study, but the other bands, such as red edge, will also be used in the future.

There are in total 9 classes that are annotated. An statistical overview of the images and classes is shown in Tab. 1. We can see that there is a very large class imbalance, where the minority classes such as green algae, red algae and lichen each accounts for less than 0.5% of the total number of pixels. Based on the statistics, image 1&6 are selected as the test set and the rest as the training set. However, as part of the NIR data is missing in image 6, the scores for models trained with NIR data are calculated using image 1 only.

3.2 Training Protocols

3.2.1 Data Set Parameters Due to the large size of the images and to increase model robustness, random image crop is

Table 1. Ground truth statistics. The numbers indicate class percentages (%), with minority classes highlight.

File Name	Height, Width	No data	Background	Green algae	Red algae	Rock	Sand	Lichen	Vegetation	Beach
ne_akeroya_5cm_area_1	2201, 3400	6.08	41.79	0.73	0.34	32.15	2.43	0.93	8.43	7.13
ne_akeroya_5cm_area_2	3600, 3700	0.59	24.65	0.22	0.15	17.55	45.07	0.01	5.82	5.94
ne_akeroya_5cm_area_3	3001, 2400	1.71	30.05	0.6	0.28	26.82	26.53	0.05	6.38	7.58
ne_akeroya_5cm_area_4	5201, 3700	8.44	50.07	0.01	0.05	23.11	6.42	0.03	10.0	1.87
ne_akeroya_5cm_area_5	2501, 3500	0.04	51.41	0.13	0.07	28.61	7.45	0.03	11.13	1.13
ne_akeroya_5cm_area_6	3600, 1700	3.46	50.71	1.16	0.46	36.05	3.71	0.47	2.07	1.9
Summary	-	4.02	41.55	**0.34**	**0.18**	25.49	16.44	**0.18**	7.87	3.94

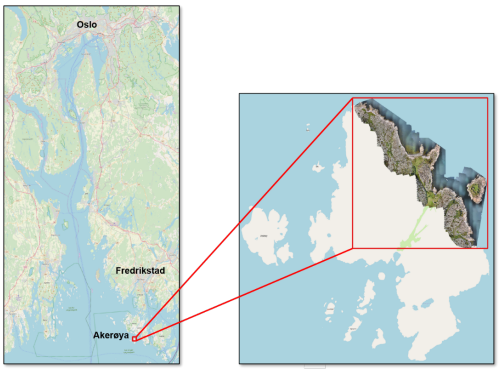

Figure 2. Illustraion of the study area

used during training. Each model is trained twice, one with conventional RGB and one with the fourth channel being the NIR data. For initialization, the pretrained weights on ImageNet are used, and for the weights of NIR channel, we use the same as for the third channel. Data augmentation [Shorten and Khoshgoftaar, 2019] is used both for training and testing, details of which are summarized in Tab. 2.

3.2.2 Optimizer, Learning Rate and Loss The Adam [Kingma and Ba, 2014] with AMSGrad [Reddi et al., 2018] is used as the optimization algorithm, where the weight decay for non-bias weight parameters is set as 2e-5. A multi-step learning rate (LR) scheduler is used

$$LR = LR_0 \times \gamma^{epoch//steps} \tag{7}$$

where $\gamma = 0.8$, $steps = 2$, and an initial LR of 6e-5. The LR to bias weight parameters are set as twice as non-bias weight parameters. Within each epoch, a polynomial decay $(1 - iter/iter_{max})^{0.9}$ is used to adjust LR, where $iter_{max}$ is the maximum number of iteration. When LR becomes smaller than 3.28e-6, a constant LR of 1.8937e-6 is used.

The models are implemented with PyTorch and run on a workstation with two NVIDIA GeForce RTX 2080Ti 12GB GPUs.

4. RESULTS

For inference, a 448×448 tiling window with a stride of 100 is used on the test images (area 1&6). In addition, horizontal and vertical flip are applied to each image patch. The inference result then is reverted back to the original orientation before `np.argmax` is applied on the class axis for final prediction map. Under the same initial settings, all models are trained for 10 epochs and the last updated model is used for inference. A visual comparison of the test image (area 1) is shown in Fig. 3.

4.1 Evaluation Metrics

For quantitative evaluation, standard segmentation metrics that are based on pixel accuracy and region intersection over union are used [Long et al., 2015]:

- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuracy: $(1/n_{cl}) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_{cl}) \sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$
- frequency weighted IU: $(\sum_k t_k)^{-1} \sum_i t_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$

where n_{ij} is the number of pixels of class i predicted to be of class j, n_{cl} the total number of classes and $t_i = \sum_j n_{ij}$ the total number of pixels of class i.

4.2 Overall performance

The overall evaluation metrics are summarized in Tab. 3 & 4. They show the scores for RGB-only models (evaluated on area 1&6) and NIR models (evaluated on area 1), respectively. In both tables, the highest score per metrics (column-wise) is highlighted in blue. For RGB-only models, DeepLabv3+ outperforms DDCM marginally, but both outperform UNET by at least 6%. However, this margin is significantly reduced in Tab. 4 by the inclusion of NIR. Although the best scores are still from the more sophisticated models (DeepLabv3+ and DDCM), the margin is greatly reduced. Within the same model category, UNET(NIR) outperforms the RGB-only version in all measures, while DeepLabv3+(NIR) underperforms in all measure and DDCM(NIR) shows a mixed result.

Recall that DeepLabv3+ and DDCM use various dilation rates and spatial pyramid pooling design, the result shows that such feature pyramid design with varying dilation rates is successful in capturing mutli-scale context and helps boosting model performance using RGB-only data.

4.3 Class-wise performance

For per-class performance, class ACC ($\sum_i n_{ii}/t_i$) and class IU ($n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$) are computed and shown in Tab. 5 & 6. The noticeable improvement (>5%) between the same model type is highlighted in gray, and the best score for selected classes is highlighted in blue.

In the ACC scores (Tab. 5), we see that the NIR data bring big improvement for UNET in classifying green algae, red algae, Rock, Vegetation, especially red algae, where it has the highest score among all model types. But for green algae, lichen and vegetation, it's still DeepLabv3+ and DDCM have the highest scores. In the IU scores (Tab. 6), it shows the similar results except for lichen, where UNET without NIR actually has the highest score, but only marginally to DeepLabv3+.

Overall, we find that introducing the NIR channel into training does not bring significant performance improvement for DeepLabv3+ and DDCM, but does make a difference for UNET. We suspect that this is because that the NIR channel brings the contextual information that UNET needs more than DeepLabv3+ and DDCM. The latter two, by design, have modules that enable them to capture mutli-scale context using RGB information alone and to achieve high performance.

Table 2. Data Set parameters

	Crop Size	Samples	Augmentation*
Train	(448, 448)	5000	RandomCrop (1.0), VerticalFlip (0.5), HorizontalFlip (0.5), RandomRotate90 (0.5), Transpose (0.5),ShiftScaleRotate (0.2), MedianBlur (0.2), CoarseDropout (0.2)
Validation	(448, 448)	1000	RandomCrop (1.0), VerticalFlip (0.5), HorizontalFlip (0.5),RandomRotate90 (0.5)

*Implemented using Albumentations library [Buslaev et al., 2020]

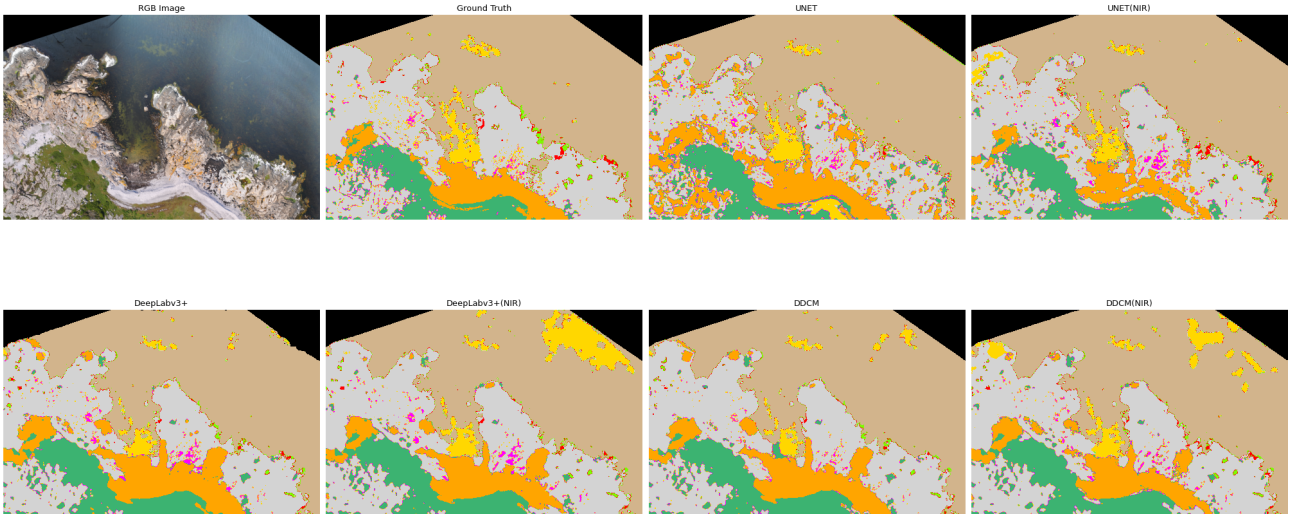

Figure 3. Example of predictions on a test image.

Table 3. Overall evaluation scores of RGB-only models. The best scores per column is highlighted in blue

Model	pixel Acc.	mean Acc	mean IU	f.w. IU
UNET	87.47	70.27	54.54	83.05
DeepLabv3+	**95.08**	**76.21**	**64.96**	**91.28**
DDCM	94.82	71.87	62.68	90.89

Table 4. Overall evaluation scores of models using RGB & NIR data. The best scores per column is highlighted in blue. The better score between the same model type is highlighted in gray.

Model	pixel Acc.	mean Acc	mean IU	f.w. IU
UNET	85.52	68.20	55.79	77.39
UNET(NIR)	**89.82**	**71.10**	**60.74**	**82.43**
DeepLabv3+	**90.24**	**71.91**	**61.50**	**83.25**
DeepLabv3+(NIR)	86.99	69.98	57.97	79.56
DDCM	**90.52**	67.40	**59.56**	**83.54**
DDCM(NIR)	88.85	**70.03**	59.33	81.56

5. CONCLUSION

We made a unified re-implementation of three neural network models with distinctive architecture and complexity for general use with UAV multi-sensor data. We tested on a high resolution dataset acquired for coastal habitat monitoring. The conventional RGB data and the NIR band from multisepctral sensor are used.

We observe that neural network models with high contextual information aggregation capacity are important for achieving satisfactory performance if there is only conventional RGB data available. And simply adding additional data from extra sensor, NIR in our example, in training existing complex deep neural networks does not warrant a performance gain as one might expect. This performance gain could be expected from baseline models though, namely UNET in this study, especially for vegetation related classes. In our experiment, the best performance is achieved from the more complex models using RGB-only data but the gap is much reduced in baseline model when NIR is included.

Furthermore, we verify that dilated convolutions with multiple rates on high-level features and its fusion with low-level features is an effective approach for contextual information aggregation. The use of customized loss function with adaptive class weighting is also found to be effective in training with the highly imbalanced data. We aim to include more UAV multi-sensor data to further investigate these findings in the coming year.

ACKNOWLEDGEMENTS

This work was supported by the Research Council of Norway under the project of Norwegian Infrastructure for drone-based research, mapping and monitoring in the coastal zone.

REFERENCES

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481-2495.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. A., 2020. Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2). https://www.mdpi.com/2078-2489/11/2/125.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2018a. 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.

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation.

Table 5. Class-wise ACC scores. Noticeable improvement (>5%) between the same model type is highlighted in gray. The best score of selected classes is highlighted in blue.

Model	No data	Background	Green algae	Red algae	Rock	Sand	Lichen	Vegetation	Beach
UNET	97.90	96.72	13.43	33.49	72.63	**69.95**	57.13	81.14	**91.44**
UNET(NIR)	97.88	97.37	**24.89**	**57.50**	**87.98**	61.0	54.24	**92.13**	66.92
DeepLabv3+	99.04	**96.26**	**33.63**	35.34	85.24	49.47	**62.46**	92.38	**93.43**
DeepLabv3+(NIR)	98.87	88.2	26.62	37.0	87.36	**57.16**	58.36	**93.87**	82.36
DDCM	97.78	96.33	31.06	27.95	88.7	47.04	38.44	92.16	87.15
DDCM(NIR)	98.15	93.3	**34.11**	**40.22**	87.62	**58.24**	**46.05**	91.74	80.88

Table 6. Class-wise IU scores. Noticeable improvement (>5%) between the same model type is highlighted in gray. The best score of selected classes is highlighted in blue.

Model	No data	Background	Green algae	Red algae	Rock	Sand	Lichen	Vegetation	Beach
UNET	97.89	93.52	12.40	26.63	67.61	45.25	**46.03**	67.53	45.29
UNET(NIR)	97.87	93.05	**22.33**	**33.50**	**78.72**	47.84	40.75	**78.39**	**54.19**
DeepLabv3+	94.62	**91.80**	28.46	26.27	81.51	**43.34**	45.81	**81.26**	60.46
DeepLabv3+(NIR)	98.06	83.77	23.93	29.33	81.93	21.5	42.81	80.74	59.71
DDCM	97.75	91.66	25.77	22.76	82.97	**38.66**	35.36	79.37	61.78
DDCM(NIR)	98.1	88.69	**29.76**	**26.88**	80.72	29.44	**38.03**	80.96	61.44

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (eds), *Computer Vision – ECCV 2018*, Springer International Publishing, Cham, 833–851.

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R., 2018. Deepglobe 2018: A challenge to parse the earth through satelite images. *the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Eigen, D., Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2650–2658.

Grauman, K., Darrell, T., 2005. The pyramid match kernel: discriminative classification with sets of image features. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2, 1458–1465 Vol. 2.

He, K., Zhang, X., Ren, S., Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds), *Computer Vision – ECCV 2014*, Springer International Publishing, Cham, 346–361.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P., 1990. A real-time algorithm for signal analysis with the help of the wavelet transform. J.-M. Combes, A. Grossmann, P. Tchamitchian (eds), *Wavelets*, Springer Berlin Heidelberg, Berlin, Heidelberg, 286–297.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. F. Bach, D. Blei (eds), *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, 37, PMLR, Lille, France, 448–456.

Kaiser, P., Wegner, J., Lucchi, A., Jaggi, M., Hofmann, T., Schindler, K., 2017. Learning Aerial Image Segmentation From Online Maps. *IEEE Transactions on Geoscience and Remote Sensing*, PP, 1-15.

Kampffmeyer, M., Salberg, A.-B., Jenssen, R., 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 680–688.

Kingma, D. P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980. http://arxiv.org/abs/1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (eds), *Advances in Neural Information Processing Systems*, 25, Curran Associates, Inc.

Lazebnik, S., Schmid, C., Ponce, J., 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2, 2169–2178.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541-551. https://doi.org/10.1162/neco.1989.1.4.541.

Liu, Q., Kampffmeyer, M. C., Jenssen, R., Salberg, A.-B., 2020a. Multi-view self-constructing graph convolutional networks with adaptive class weighting loss for semantic segmentation. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Liu, Q., Kampffmeyer, M., Jenssen, R., Salberg, A.-B., 2020b. Dense Dilated Convolutions' Merging Network for Land Cover Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(9), 6309-6320.

Liu, W., Rabinovich, A., Berg, A., 2015. ParseNet: Looking Wider to See Better.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation.

Luo, W., Li, Y., Urtasun, R., Zemel, R., 2016. Understanding the effective receptive field in deep convolutional neural networks. D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (eds), *Advances in Neural Information Processing Systems*, 29, Curran Associates, Inc.

Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 565–571.

Noh, H., Hong, S., Han, B., 2015. Learning Deconvolution Network for Semantic Segmentation. *arXiv preprint arXiv:1505.04366*.

Pajares, G., 2015. Overview and Current Status of Remote Sensing Applications Based on Unmanned Aerial Vehicles (UAVs). *Photogrammetric Engineering & Remote Sensing*, 81, 281-330.

Reddi, S. J., Kale, S., Kumar, S., 2018. On the convergence of Adam and beyond. *CoRR*.

Ronneberger, O., P.Fischer, Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS, 9351, Springer, 234–241. (available on arXiv:1505.04597 [cs.CV]).

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2019. Mobilenetv2: Inverted residuals and linear bottlenecks.

Shorten, C., Khoshgoftaar, T. M., 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. https://doi.org/10.1186/s40537-019-0197-0.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.

Yu, F., Koltun, V., 2016a. Multi-scale context aggregation by dilated convolutions.

Yu, F., Koltun, V., 2016b. Multi-Scale Context Aggregation by Dilated Convolutions. *CoRR*, abs/1511.07122.

Zeiler, M. D., Taylor, G. W., Fergus, R., 2011. Adaptive deconvolutional networks for mid and high level feature learning. *2011 International Conference on Computer Vision*, 2018–2025.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230–6239.

Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C. C., Lin, D., Jia, J., 2018. Psanet: Point-wise spatial attention network for scene parsing. V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (eds), *Computer Vision – ECCV 2018*, Springer International Publishing, Cham, 270–286.