

DIFFERENT COLOR SPACES IN DEEP LEARNING-BASED WATER SEGMENTATION FOR AUTONOMOUS MARINE OPERATIONS

Jussi Taipalmaa¹, Nikolaos Passalis², Jenni Raitoharju³

¹Department of Computing Sciences, Tampere University, Finland

²Aristotle University of Thessaloniki, Greece

³Finnish Environment Institute, Jyväskylä, Finland

ABSTRACT

For autonomous unmanned surface vehicles (USV) operations, it is important to be able to observe the surroundings using visual information. Water segmentation is a task where the water surface is recognized and separated from everything else. The algorithm performing the segmentation must be robust, because safety is the most important feature of autonomous USVs. This is especially challenging in many USV applications, where the rapidly changing weather and lighting conditions can cause significant distribution shifts. In this study, we analyze the robustness of different color spaces (e.g., RGB and HSV) for water segmentation and consider how to use different color channels in training and testing to maximize the robustness. We evaluate the segmentation performance on a challenging completely unseen test dataset, recorded in vastly different conditions and with different equipment.

Index Terms— Water Segmentation, Deep Learning, Autonomous Navigation

1. INTRODUCTION

Autonomous unmanned surface vehicles (USV) are often used for different types of missions, such as transportation, which includes navigating on the pathway, and search and rescue missions, in which the focus is finding an object within the search area. For these types of missions, maintaining safety is the most important feature of a USV. Therefore, it is important to be able to observe the surroundings using visual information, which can be used to perform water segmentation. Water segmentation is the process, in which the water surface is recognized and separated from stationary or moving objects. The purpose of water segmentation is to offer a USV the perception capabilities needed to detect nearby objects, such as other surface vessels, swimmers, docks, and obstacles, like rocks or shoreline.

To ensure safe operation, the algorithm performing the segmentation must be robust. For the scope of this study, robustness is defined as the ability of an algorithm to adapt to different conditions, which can often cause significant distribution shifts to the input data, lowering the water segmentation accuracy. Therefore, the algorithm must be robust even in unseen situations that can be caused by a limited training set, changing weather and lighting conditions, or a change of camera equipment. Such changes usually decrease the performance of the algorithm, so it is important to study how to minimize such performance degradation.

In this study, we analyze the effect of different color spaces (e.g., RGB, HSV) for water segmentation and propose simple, yet effective ways to use and combine different color channels in training and testing to maximize the robustness. As we demonstrate, the performance of the algorithm improves by finding the best combination of different color channels. We evaluate the segmentation performance on a completely unseen test dataset, recorded in different conditions and with different camera equipment compared to the training data as shown in Figure 1. The training is done with a hand-annotated dataset of 300 images recorded with a Go-Pro4 Session camera, and the testing is done with 50 images from a USV and 50 images from an unmanned aerial vehicle (UAV) recorded with different cameras. Also, the training set is recorded during winter and the testing set during summer, which leads to significant changes in the scenery.

2. RELATED WORK

Semantic image segmentation, or pixel-level classification, aims at grouping together the image regions belonging to the same semantic category and assigning each pixel to one of the categories [1]. Semantic segmentation plays an important role in different image understanding tasks. While some applications aim at fine-grained labeling [2, 3], a simpler task of road segmentation has gained a lot of attention due to various applications in Advanced Driver Assistance Systems (ADAS) and self-driving cars. State-of-the-art algorithms rely on different deep learning approaches [3, 4, 5, 6].

Thanks to Future makers aCOLOR-project for equipment and data collection and to Academy of Finland project no. 328754 (AutoSOS) for funding experiments and analysis.

For autonomous marine operations, water segmentation is a task with a similar importance as road segmentation for self-driving cars. However, as efforts on the development of USVs have not been as extensive as those for self-driving cars, the research on water segmentation is still sparse. Most of the existing works on water segmentation rely on low-level features, which have been utilized with Decision Forests [7], Expectation Maximization [8], and Support Vector Machines [9]. Bovcon et al. use an inertial measurement unit to assist the segmentation method proposed in [8]. Lopez-Fuentes et al. [10] apply deep learning for water segmentation to detect floods in rivers. In our previous work [11], we published a water segmentation dataset collected in Finnish lake environment along with deep learning based segmentation results. This study extends the previous work by examining the effects of different color spaces on the segmentation accuracy.

There are some earlier studies on effects of color space on image classification and segmentation. However, the choice of a particular color space is largely application dependent. For instance, color pixel classification [12] and soccer image analysis [13] have been done using a hybrid color space. YCbCr color space has been used for skin detection [14]. Other analysis on skin detection found that RGB color space was the model that gave the best results [15]. Further studies suggested that, skin pixel classification using a Bayesian model gave the best results using LAB color space [16]. The effect of different color spaces was studied in segmentation of aerial images over planted fields, suggesting that a reduction in the complexity of the segmentation procedure is achievable when it is operating on a single color space domain [17]. Automatic segmentation of image into natural objects based on different color space models was studied in [18], resulting that RGB color space is the best color space representation for the set of the images used. Furthermore, importance of color spaces for image classification was investigated in [19], suggesting that using several different color spaces as input to individual networks significantly improves the result. To best of our knowledge, this is the first work that examines the effect of using and combining different color spaces on water segmentation performance.

3. METHODS AND EXPERIMENTS

3.1. Color space conversion

In this paper, we analyze the the effect of different color spaces, namely RBG, HSV, and grayscale, for water segmentation. The original images were in RGB format and we converted them into grayscale and HSV color spaces as described below.

3.1.1. RGB to grayscale conversion

RGB values were converted to grayscale values by forming a weighted sum of the R (red), G (green), and B (blue) compo-

nents:

$$grayscale = a \cdot R + b \cdot G + c \cdot B.$$

For a , b and c we used values: $a = 0.2989$, $a = 0.5870$, $c = 0.1140$. These values come from the BT.601 standard for use in colour video encoding, where they are used to compute luminance from an RGB-signal [20].

3.1.2. RGB to HSV conversion

HSV presentation of an image consists of H (*hue*), S (*saturation*), and V (*value*) components. From RGB components, maximum (M), minimum (m), and chroma (C) values can be obtained as:

$$\begin{aligned} M &= \max(R, G, B), \\ m &= \min(R, G, B), \\ C &= M - m. \end{aligned}$$

The *hue* component, presented it in degrees $[0^\circ, 360^\circ]$, is calculated as:

$$H' = \begin{cases} \text{undefined}, & \text{if } C = 0 \\ \frac{G-B}{C} \bmod 6, & \text{if } M = R \\ \frac{B-R}{C} + 2, & \text{if } M = G \\ \frac{R-G}{C} + 4, & \text{if } M = B \end{cases}$$

$$H = 60^\circ \cdot H'.$$

Value component is calculated as $V = M$, while *saturation* component is defined as:

$$S = \begin{cases} 0, & \text{if } V = 0 \\ \frac{C}{V}, & \text{otherwise} \end{cases}$$

Hue distinguishes the color of the source, *saturation* distinguishes a pure spectral light from a pastel shade of the same *hue* and *value* describes the brightness of an area.

3.2. Network architecture

The network architecture used for water segmentation is based on the KittiSeg road segmentation model [4], which employs a fully convolutional deep learning architecture [2] and was originally developed for self-driving cars. KittiSeg consists of a VGG-16 [21] based encoder which performs feature extraction, and a segmentation decoder, which performs the actual water segmentation.

In this work, we follow the same architecture, but replace the encoder with a lighter and faster version, as originally proposed in [11]. The encoder architecture is derived from the VGG16 architecture by reducing the number of convolution layers. The lightweight encoder is composed of 5 convolutional layers with 32, 64, 128, 256, and 512 filters, respectively. These layers were randomly initialized [22], i.e., no pretraining was used.

set	rotations	training images	validation images
1	$[0^\circ]$	240	60
2	$[0^\circ, 180^\circ]$	480	120
3	$[0^\circ, 90^\circ, 180^\circ, 270^\circ]$	960	240

Table 1. Set numbers for different rotations and corresponding number of images

Then, we employed a segmentation decoder following the approach proposed in [2]. Given the feature maps produced by the encoder, a series of three transposed convolution layers are used to upsample the output [23]. Those features are first processed by a 1×1 convolution layer and then added to the partially upsampled results. The output of the network corresponds to the probability of each pixel depicting a region that contains water. Since the classification problem at hand is a two-class problem, *water / not water*, the predicted probability can be thresholded in different ways to finetune the behavior of the model according to the needs of each application, e.g., to minimize the false negatives.

3.3. Training setup

The dataset used for training is Tampere-WaterSeg dataset [11] publicly available at <http://urn.fi/urn:nbn:fi:att:eafdb99c-4396-4591-80e0-24219875b5b6>. It consists of 600 labeled HD-quality (1920×1080 pixels) images that contain views from a USV on a lake. The dataset was recorded with GoPro Hero 4 Session camera during wintertime in Tampere, Finland. Thus, the images depict snowy conditions and they contain three subcategories: open lake, channel area, and docking situations, 200 images each. In this paper, we followed the training phase of the test setup 7 defined in [11]. This allowed us to use data from all the three different subcategories for training the model.

For these images, we performed conversions from RGB color space to HSV and grayscale color spaces resulting in 9 different training sets (RGB, R, G, B, HSV, H, S, V and gray). In addition, we increased the number of training and validation images by adding different rotations of the images. We added horizontally and vertically flipped images as described in Table 1 to obtain three different training sets for each considered color space. Thus, we used a total of 27 different training sets for training our segmentation model.

In addition to the cross-entropy loss function, we used focal loss [24] to train our model. Focal loss reduces the relative loss for well-classified examples putting more focus on hard, misclassified samples. Cross-entropy (CE) and focal loss (FL) are defined as follows:

$$\begin{aligned} \text{CE}(p_t) &= -\log(p_t), \\ \text{FL}(p_t) &= -(1 - p_t)^\gamma \log(p_t). \end{aligned}$$

In our experiments, we set $\gamma = 1$. Using both loss functions

for each training set, we ended up having a total of 54 different training setups.

3.4. Test setup

For testing, we collected two different sets of images. The first set, named “USV”, was recorded with Xiaomi Yi 4K+ Action camera from the same USV as the training images. The images were downsampled from 3840×2160 to 1920×1080 pixels. The test set was recorded on a clear summer day. As we used snowy images for training, the conditions were very different (Fig. 1). Also the point of view is slightly different between training and testing data.

Our second test set, named “UAV”, was recorded with an unmanned aerial vehicle flying above water with a DJI Phantom 4 Pro camera facing strictly downwards. This creates a completely different setup compared to the training setup. The aim of these tests is to determine how the trained model behaves with slightly different data (USV in different conditions) and completely different data (USV vs. UAV). Specifically, we analyze how the used color space affects the segmentation accuracy.

We tested models by converting the test images into the same color space as the model was trained with. The number of test images is 50 for both USV and UAV sets. The images were annotated accurately by marking the water areas using polygons [25]. We will make the test sets available with Tampere-WaterSeg dataset.

In addition, we formed different ensemble classifiers in order to test different combinations of color channels. In this approaches, we calculated the average of the outputs from multiple different models before determining if a pixel is classified as water or non-water. We formed different ensembles including the average of the output from R, G and B models, and the average of the output from all trained models for one rotation set. We also formed one ensemble consisting of three classifiers that obtained the best scores on the test set to demonstrate the results that can be obtained using this method.

4. RESULTS AND DISCUSSION

We formed receiver operating characteristic (ROC) curves and calculated the area under curve (AUC) for each test setup. The ROC-curves for all color spaces for both USV and UAV sets in the setup that includes rotation set 1 and focal loss are provided in Figure 1. The AUC scores for all the different test setups are presented in Table 2.

These results suggest that the color does not seem to be an important factor for USV set and often just individual channels works as well or better than RGB and HSV color spaces. For the UAV set, the outcome is completely opposite since only RGB and HSV color spaces achieve high performance and all the other sets lead to mediocre scores. This can mean

Color channel	USV						UAV					
	cross-entropy			focal loss			cross-entropy			focal loss		
	set 1	set 2	set 3	set 1	set 2	set 3	set 1	set 2	set 3	set 1	set 2	set 3
RGB	0.9540	0.8683	0.8041	0.8567	0.7660	0.7906	0.9207	0.9483	0.9482	0.9325	0.9026	0.9402
R	0.9564	0.9279	0.8846	0.9613	0.9161	0.9125	0.7730	0.8071	0.7049	0.7123	0.7591	0.7249
G	0.9533	0.9245	0.9208	0.9538	0.9552	0.7878	0.5345	0.7434	0.4959	0.7130	0.7091	0.6316
B	0.9060	0.9648	0.8924	0.9599	0.8605	0.8808	0.6475	0.6347	0.7748	0.7063	0.6831	0.6755
HSV	0.7936	0.7730	0.6280	0.7788	0.6825	0.7741	0.9189	0.9457	0.9504	0.9377	0.9533	0.9580
H	0.6387	0.5111	0.4504	0.6145	0.4651	0.4810	0.7869	0.8536	0.8951	0.7234	0.8318	0.8490
S	0.7468	0.6584	0.5510	0.6845	0.6868	0.6067	0.5430	0.7966	0.7560	0.6418	0.7709	0.8668
V	0.9535	0.9378	0.9314	0.9567	0.9631	0.9420	0.7574	0.7574	0.7068	0.7536	0.8022	0.6738
gray	0.9343	0.9521	0.9141	0.9624	0.9228	0.9098	0.7064	0.8278	0.6804	0.7206	0.7517	0.6666

Table 2. AUC scores for different rotations and loss-functions.

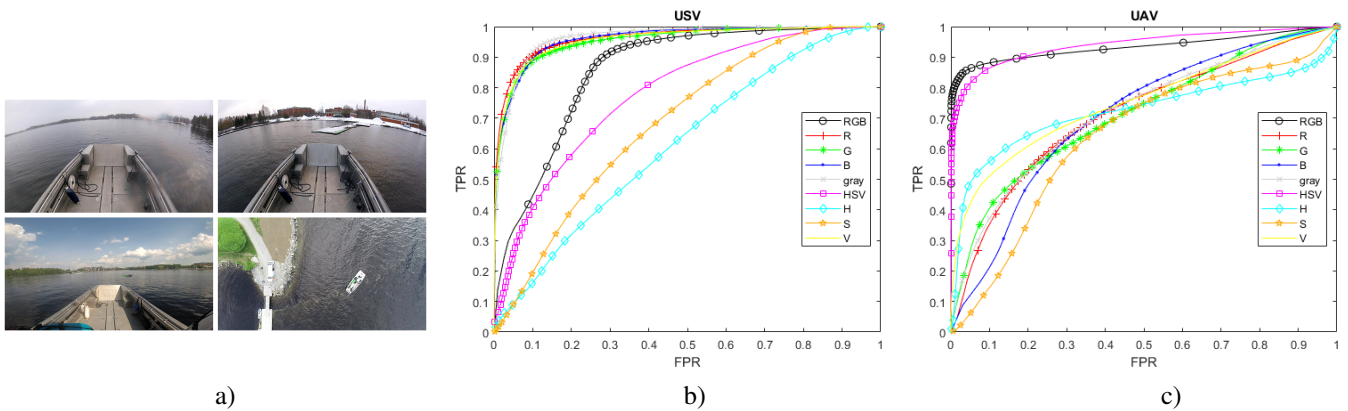


Fig. 1. a) top row: training images, bottom left: USV test image, bottom right: UAV test image b) ROC-curves for USV, set 1, focal loss c) ROC-curves for UAV, set 1, focal loss

channel combination	USV		UAV	
	CE	FL	CE	FL
R+G+B	0.9618	0.9639	0.7465	0.7287
ALL	0.9430	0.9500	0.8288	0.8257
TOP3	0.9704		0.9629	

Table 3. AUC scores for ensembles of classifiers

that the classification on UAV set relies on color information, possibly due to the varying illumination conditions (e.g., sun reflections) that are encountered in this test set.

Furthermore, rotations do not seem to increase the accuracy on the USV test set. This can be explained by the fixed position of elements in images, the bow of the USV is always on the bottom of the image and the sky is on top. The situation is different in the UAV test set, where the UAV can basically face any direction. Therefore, augmenting the training set with various rotations can indeed lead to significant improvements in the UAV test set.

The AUC scores for the ensemble classifiers are presented in Table 3 using rotation set 1. The important findings are that in the USV test set the combination R, G and B channels

provides better results than any single channel or RGB channel. Also combining all the channels provides good results. For the UAV neither of these combinations do not provide the best results, but when three channels with highest performance are combined, the results are better than for any single channel. The TOP3 test shows that by finding the best combination of color channels, the results can be further improved significantly.

5. CONCLUSIONS

This study concludes that semantic water segmentation can be performed for a diverse range of conditions, which can differ significantly compared to the training setup, with good results, given that the appropriate color space is used. The findings also demonstrates that the commonly used RGB color space might not always be the the best possible choice, as it is outperformed in 10 out of 12 different test setups. These findings also hint to further research direction on developing robust color-based ensembles, since carefully selecting a combination of different color spaces can yield to better results than any model trained on a single color space.

6. REFERENCES

- [1] X. Liu, Z. Deng, and Y. Yang, “Recent progress in semantic image segmentation,” *Artificial Intelligence Review*, 2018.
- [2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [3] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, “Understanding convolution for semantic segmentation,” in *IEEE Winter Conference on Applications of Computer Vision*, 2018, pp. 1451–1460.
- [4] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun, “Multinet: Real-time joint semantic reasoning for autonomous driving,” in *IEEE Intelligent Vehicles Symposium*, 2018, pp. 1013–1020.
- [5] D. Levi, N. Garnett, and E. Fetaya, “Stixelnet: A deep convolutional network for obstacle detection and road segmentation,” in *British Machine Vision Conference*. 2015, pp. 109.1–109.12, BMVA Press.
- [6] F. Zohourian, B. Antic, J. Siegemund, M. Meuter, and J. Pauli, “Superpixel-based road segmentation for real-time systems using cnn,” in *VISIGRAPP*, 2018.
- [7] P. Mettes, R. T. Tan, and R. Veltkamp, “On the segmentation and classification of water in videos,” in *International Conference on Computer Vision Theory and Applications*, 2014, vol. 1, pp. 283–292.
- [8] M. Kristan, V. Sulic, S. Kovacic, and J. Perš, “Fast image-based obstacle detection from unmanned surface vehicles,” *IEEE Transactions on Cybernetics*, 2015.
- [9] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh, “Self-supervised segmentation of river scenes,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 6227–6232.
- [10] L. Lopez-Fuentes, C. Rossi, and H. Skinnemoen, “River segmentation for flood monitoring,” in *IEEE International Conference on Big Data*, 2017, pp. 3746–3749.
- [11] J. Taipalmaa, N. Passalis, H. Zhang, M. Gabbouj, and J. Raitoharju, “High-resolution water segmentation for autonomous unmanned surface vehicles: a novel dataset and evaluation,” in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. 2019, IEEE.
- [12] N. Vandenbroucke, L. Macaire, and J.-G. Postaire, “Color pixels classification in a hybrid color space,” in *International Conference on Image Processing (ICIP)*, 1998, vol. 1, pp. 176 – 180.
- [13] N. Vandenbroucke, L. Macaire, and J.-G. Postaire, “Color image segmentation by pixel classification in an adapted hybrid color space. application to soccer image analysis,” *Computer Vision and Image Understanding*, vol. 90, pp. 190–216, 2003.
- [14] D. Chai and A. Bouzerdoum, “A bayesian approach to skin color classification in ycbcr color space,” in *IEEE TENCON*, 2000, vol. 2, pp. 421 – 424 vol.2.
- [15] M. Shin, J. Chang, and L. Tsap, “Does colorspace transformation make any difference on skin detection?,” in *Conference: Applications of Computer Vision*, 2002, pp. 275 – 279.
- [16] B. Zarit, B. Super, and F. Quek, “Comparison of five color models in skin pixel classification,” in *Conference: Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999, pp. 58 – 63.
- [17] N.M. Kwok, Q. Ha, and G. Fang, “Effect of color space on color image segmentation,” in *Conference: Image and Signal Processing*, 2009, pp. 1 – 5.
- [18] D. Khattab, H. Ebied, A. Hussein, and M. Tolba, “Color image segmentation based on different color space models using automatic grabcut,” *TheScientificWorldJournal*, 2014.
- [19] G.N. Shreyank and Y. Chun, “Colornet: Investigating the importance of color spaces for image classification,” *CoRR*, vol. abs/1902.00267, 2019.
- [20] Radiocommunication Sector of International Telecommunication Union, “Recommendation itu-r bt.601-7,” 2011.
- [21] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 818–833.
- [22] K. He, X. Zhang, Ren S., and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015.
- [23] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv:1603.07285*, 2016.
- [24] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” *CoRR*, vol. abs/1708.02002, 2017.
- [25] Labelbox, “Labelbox,” 2019, [Online]. Available: <https://labelbox.com>.