River water quality estimation based on convolutional neural network

Takahiro Oga*, Yo Umeki†, Masahiro Iwahashi* and Yoko Matsuda‡
* Dept. of Electrical, Electronics and Information Engineering, Nagaoka Univ. of Tech.,
Nagaoka, Niigata, Japan.
E-mail:s163119@stn.nagaokaut.ac.jp, iwahashi@vos.nagaokaut.ac.jp
† Dept. of Information Science and Control Engineering, Nagaoka Univ. of Tech.,
Nagaoka, Niigata, 940-2137 Japan.
E-mail:umeki@stn.nagaokaut.ac.jp
‡ Dept. of Civil and Environmental Engineering, Nagaoka Univ. of Tech.,
Nagaoka, Niigata, 940-2137 Japan.
E-mail: ymatsuda@vos.nagaokaut.ac.jp

***Abstract*—We propose an estimation method of water quality of a river based on convolutional neural networks (CNNs), a semantic segmentation method, and a new dataset. Since floods are serious problems for riversides, various technologies have been proposed as a safety system. So far, measurement methods of water level are proposed as one of various systems. However, these methods cannot estimate a flood which is caused by debris flows because the flood does not follow the increasing water level. To estimate floods, water quality is more important information because water become muddy before floods. Hence, we estimate water quality using monitoring images and CNNs which trained new dataset. Furthermore, we combine a pre-processing using a semantic segmentation method. Since regions of no-river water often produces misdetections, we exclude these regions using a semantic segmentation method before CNNs. In simulations, it was observed that the proposed method objectively outperforms state-of-the-art methods in accuracy based on our dataset.**

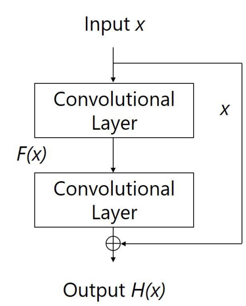


Fig. 1. Residual block architecture.

I. Introduction

Floods are serious problems for riversides and various methods of flood estimation have been proposed as a safety system [1], [2]. Since floods and water level have strong correlation, several methods of flood estimation use water level estimation [3]–[6]. Traditionally, water level indicator based methods are proposed [3], [4]. Although the methods realize to set indicators into water, the system are too expensive.

Recently, flood estimation methods based on water level are proposed by using monitoring images and edge detection methods. A water level estimation based on edge detection is proposed [5]. They remove effects of weather by using frame addition for accurate estimation. Another type of the method considers light reflections using difference of inter-frame reflectance [6]. These methods realize inexpensive and accurate estimation methods of floods.

Unfortunately, water level based flood estimation methods cannot completely estimate floods because a flood which is caused debris flows do not follow increasing water levels [7]. Hence, we need to consider other factors of floods for the estimation. Floods have an another sign which is water quality to become muddy [8], [9]. For the above reasons, river water quality is effective to estimate floods.

To estimate water quality, we propose a new dataset and compared CNNs architectures using our dataset. We compared CNNs such as AlexNet, NIN, GoogLeNet, VGGNet, ResNet WideResNet and ResNeXt [10]–[16]. We trained these CNNs architectures using our dataset. The dataset has natural river images which have two classes “Clear” and “Muddy”. Then, input images are classified into “Clear” and “Muddy” classes using trained CNNs model.

Furthermore, we apply a pre-processing to estimate water quality accurately. Usually, natural images of river contain different regions such as grass, sand, sky and tree. These regions often products worse results of water quality estimation and these are easily remove using semantic segmentation methods. Hence, we apply a preprocessing before training of CNNs using semantic segmentation methods. In experiment, compared with state-of-the-art CNNs, the proposed method shows the better results in accuracy.

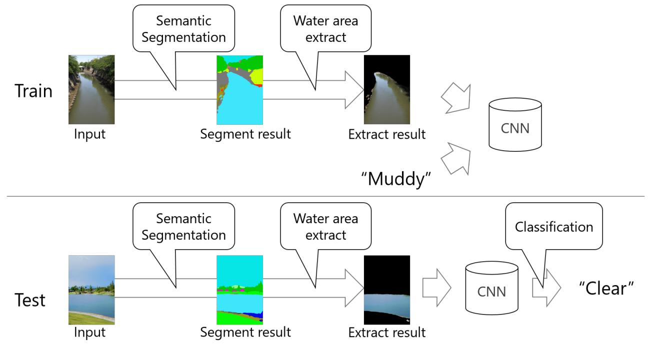


Fig. 2. Overview of proposed method.

II. Fundamental methods for water quality estimation

A. Basic stractures of CNNs

In this section we explain CNNs architectures [17], [18]. CNNs architectures are mainly constructed convolutional layers, pooling layers and a fully-connected layer. Convolutional layers extract feature maps from an input image and pooling layers choose features which explain the image well. CNNs architecture have some convolutional layers and pooling layers alternatively. In the last of architecture, a fully-connected layer classify images using extracted image features.

Past a decade, CNNs gave large effects for image processing and large amount of CNNs architectures proposed. We introduce main architectures of CNNs. AlexNet is baseline method of CNNs architectures. AlexNet has 8 layers and the architecture can learn large datasets with short times [10]. NIN is the stacking of multiple mlp layers [11]. The mlp layer includes the multi-layer perceptron. GoogLeNet has the inception module [12]. The inception module combines each results after uses some convolutional layers of various parameters. VGGNet has a simple construction by consecutive small convolutional layers [13]. This architecture is often used classification tasks. ResNet can learn in substantially deep architecture because of including the residual block [14]. The residual block is shown in Fig. 1. The residual block which adds input data to output data is effective to construct deep networks. Recently, as derivation of the ResNet, WideResNet and ResNeXt are proposed [15], [16]. WideResNet is constructed to increase filters of the ResNet. ResNeXt transforms convolutional layers into parallel convolutional layers.

B. Semantic segmentation

We explain the state-of-the-art method of semantic segmentation which uses the ADE20K dataset and PSPNet [19], [20]. PSPNet is a CNNs architecture which have some pooling layers after a convolutional layer. To extract different features using some pooling layers, the architecture can classify complex classes. To train the architecture with the ADE20K dataset, the method has high accuracy of segmentation for these complex classes.

III. Proposed method

A. Framework

We propose an estimation method of water quality of a river based on CNNs, a semantic segmentation method, and a new dataset. Fig. 2 shows the flow of the proposed method. First, we train CNNs architectures using our dataset whose images are natural river and given classes "Clear" and "Muddy". The training images are cropped by semantic segmentation method in the pre-processing. Second, the proposed model estimate water quality of river images. Input images are pre-processed as with training and trained models classify input images into "Clear" or "Muddy" classes.

B. Pre-processing

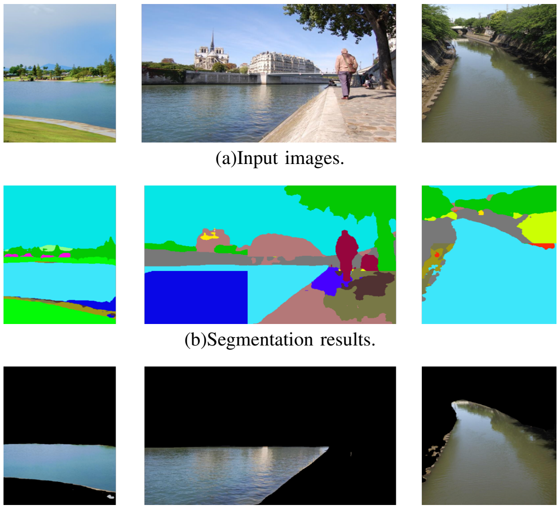
In this section, we explain pre-processing using semantic segmentation method. Although river images usually contain not water regions, these regions often products worse results. Hence, we remove these regions using the state-of-the-art semantic segmentation method which uses explained in Section II-B. The method classifies pixels of an input images into 150 classes. Since likely water regions are almost "river" in a river image, we crop water, lake, river, sea, and natatorium as river regions. Thanks to the strategy, an input image has regions of river water only.

C. Dataset construction

We construct a dataset for the water quality estimation. We collected river images from "river" class of the Place365 dataset [21] and images of Nagaoka camera monitoring system, and classified images into two classes, "Clear" and "Muddy". "Clear" images are defined images whose river bottom can be seen and water colors are blue and green [22]. Conversely, "Muddy" images defined as having brown water. Based on the above definition, the ground truth is provided by various persons with the majority rule.

IV. Experiment

For experiments, we compare CNNs architectures and use images in the proposed dataset as training, validation images. There are 400 training images, 50 validation images. We use



(a)Input images.

(b)Segmentation results.

(c)Extract results.

Fig. 3. Input images.

TABLE I
NUMBER OF LAYERS AND LEARNING RATE FOR CNNS.

	Number of layers	Learning rate
AlexNet [10]	8	0.001
NIN [11]	6	0.001
GoogLeNet [12]	22	0.1
VGGNet [13]	19	0.05
ResNet [14]	20/50/101	0.01
WideResNet [15]	16	0.1
ResNeXt [16]	50	0.1

two test datasets in this experiment. The proposed dataset and the SUN dataset are used as test datasets [25]. The SUN dataset has 100 river images which are classified into two classes the same as the proposed dataset. We applied several data augmentations and contrast normalization. Data augmentations are random cropping and horizontal flipping [10]. Training images are randomly cropped into 224×224 images at 5 times. After that, the cropped images are flipped right and left. Consequently, the size of training images is 224×224, and the number of ground truth and image pairs is 4000. In the training, we use the MomentumSGD as an optimization and we set $momentum = 0.9$. The networks trained with 100 epochs. In Table I, we show number of layers and learning rate for CNNs architectures. We decrease learning rate 0.1 times when the $epochs = 50$. We use a weight decay [23] of 0.0005 and a mini-batch size of 32. We use the He Initialization as initial values of convolution filters [24]. Also, we do not change other conditions from original paper.

We train CNNs using the proposed dataset and evaluate the proposed method using two test datasets. We show resultant accuracy of the proposed dataset for each architecture in the Table II, where “Prop.” means accuracy of the proposed method which applying pre-processing and “Only CNNs” means accuracy of CNNs classification which not applying

TABLE II
TEST ACCURACY OF THE PROPOSE DATASET[%].

	Prop.	Only CNNs	Difference
AlexNet [10]	88.0	71.0	+17.0
NIN [11]	92.0	84.0	+12.0
GoogLeNet [12]	91.0	86.0	+5.0
VGGNet [13]	93.0	85.0	+8.0
ResNet20 [14]	91.0	82.0	+9.0
ResNet50 [14]	95.0	84.0	+11.0
ResNet101 [14]	93.0	70.0	+23.0
WideResNet [15]	92.0	84.0	+8.0
ResNeXt [16]	94.0	84.0	+10.0
Average	92.1	82.2	**+9.9**

TABLE III
TEST ACCURACY OF THE SUN DATASET[%].

	Prop.	Only CNNs	Difference
AlexNet [10]	86.0	76.0	+10.0
NIN [11]	86.0	79.0	+7.0
GoogLeNet [12]	85.0	82.0	+3.0
VGGNet [13]	89.0	75.0	+14.0
ResNet20 [14]	86.0	77.0	+9.0
ResNet50 [14]	90.0	83.0	+7.0
ResNet101 [14]	86.0	80.0	+6.0
WideResNet [15]	83.0	82.0	+1.0
ResNeXt [16]	92.0	83.0	+9.0
Average	87.0	79.6	**+7.4**

pre-processing. “Difference” means the difference accuracy of the proposed method and accuracy of only CNNs. The compared CNNs are state-of-the-art ones, AlexNet [10], NIN [11], GoogLeNet [12], VGGNet [13], ResNet [14], WideResNet [15], and ResNeXt [16] in this paper. Compared with only CNNs, the proposed method shows high accuracy with all of CNNs architectures. In the proposed method, almost all of the CNNs shows over 90% of accuracy on the proposed dataset. The propose method is higher than only CNNs by 9.9% on average of accuracy. Similarly, we show resultant accuracy of the SUN dataset in the Table III. Accuracy of the SUN dataset is improved by pre-processing as with the proposed dataset. The propose method is higher than only CNNs by 7.4% on average of accuracy.

Classification results of water quality are shown in Fig. 4. In the images, several CNNs produce misdetection in “Only CNNs”. We can see improved results using our strategy in “Prop.”. From these results, it is observed that the proposed method out performs state-the-art ones for water quality estimation as mentioned in Section I.

V. CONCLUSION

In this paper, we proposed an estimation method of water quality of a river based on CNNs, semantic segmentation, and a new dataset. Our dataset has river images and classes which are “Clear” and “Muddy”. The proposed method train images which are extracted water regions using a semantic segmentation method. In experiment, the proposed method provides better results than the state-of-the-art one on the proposed dataset and the SUN dataset.

				
Prop.	AlexNet [10]	“Muddy”	**“Muddy”**	**“Muddy”**
	NIN [11]	**“Clear”**	**“Muddy”**	“Clear”
	GoogLeNet [12]	**“Clear”**	“Clear”	“Clear”
	VGG [13]	**“Clear”**	“Clear”	**“Muddy”**
	ResNet50 [14]	**“Clear”**	**“Muddy”**	**“Muddy”**
	WideResNet [15]	**“Clear”**	“Clear”	“Clear”
	ResNeXt [16]	**“Clear”**	**“Muddy”**	“Clear”
Only CNNs	AlexNet [10]	“Muddy”	**“Muddy”**	“Clear”
	NIN [11]	“Muddy”	“Clear”	**“Muddy”**
	GoogLeNet [12]	“Muddy”	“Clear”	**“Muddy”**
	VGG [13]	“Muddy”	“Clear”	“Clear”
	ResNet50 [14]	“Muddy”	“Clear”	“Clear”
	WideResNet [15]	**“Clear”**	“Clear”	“Clear”
	ResNeXt [16]	“Muddy”	“Clear”	**“Muddy”**
Ground Truth		**“Clear”**	**“Muddy”**	**“Muddy”**

Fig. 4. Results of classification by proposed method.

ACKNOWLEDGMENT

This research is partially supported by SECOM Science and Technology Foundation.

REFERENCES

[1] P. Bonnet, J. Gehrke, and P. Seshadri, “Querying the physical world,” *IEEE personal Commun.*, vol. 7, no. 5, pp. 10–15, 2000.

[2] P. Jiang, H. Xia, Z. He, and Z. Wang, “Design of a water environment monitoring system based on wireless sensor networks,” *Sensors (Basel)*, vol. 9, pp. 6411–6434, 2009.

[3] K. Chetpattananondh, T. Tapoanoi, P. Phukpattaranont, and N. Jindapetch, “A self-calibration water level measurement using an interdigital capacitive sensor,” *Sensors and Actuators A: Physical*, vol. 209, pp. 175–182, 2014.

[4] J. Sunkpho and C. Ootamakorn, “Real-time flood monitoring and warning system.” *Songklanakarin Journal of Science & Technology*, vol. 33, no. 2, 2011.

[5] M. Iwahashi and S. Udomsiri, “Water level detection from video with fir filtering,” in *2007 16th Int. Conf. on Comput. Commun. and Networks*, 2007, pp. 826–831.

[6] J. Kim, Y. Han, and H. Hahn, “Embedded implementation of image-based water-level measurement system,” *Proc. IEEE Conf. Comput. Vis.*, vol. 5, no. 2, pp. 125–133, 2011.

[7] R. M. Iverson, M. E. Reid, and R. G. LaHusen, “Debris-flow mobilization from landslides,” *Annual Review of Earth and Planetary Sciences*, vol. 25, no. 1, pp. 85–138, 1997.

[8] F. Imaizumi, S. Tsuchiya, and O. Ohsaka, “Observation of the formation and dynamic process of debris flows in a mountainous debris torrent,” *Japan Society of Erosion Control Engineering*, vol. 55, no. 3, pp. 50–55, 2002.

[9] K. Senoo, T. Takahashi, Y. Araki, K. Furukawa, and T. Mizuyama, “Debris flow warning and evacuation system aided by gis a case study model for oshima county yamaguchi prefecture,” *Japan Society of Erosion Control Engineering*, vol. 53, no. 4, pp. 30–37, 2000.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[11] M. Lin, Q. Chen, and S. Yan, “Network in network,” *arXiv preprint arXiv:1312.4400*, 2013.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2015.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2016, pp. 770–778.

[15] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.

[16] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2017, pp. 5987–5995.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-performance neural networks for visual object classification,” *arXiv preprint arXiv:1102.0183*, 2011.

[19] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ade20k dataset,” *arXiv preprint arXiv:1608.05442*, 2016.

[20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2017, pp. 2881–2890.

[21] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Trans. Patt. Anal. and Mach. Int.*, 2017.

[22] C. V. Raman, “The molecular scattering of light,” in *Proc. of the Indian Academy of Sciences-Section A*, vol. 37, no. 3. Springer, 1953, pp. 342–349.

[23] A. Krogh and J. A. Hertz, “A simple weight decay can improve generalization,” in *Proc. Adv. in Neural Inf. Process. Syst. 4*. Morgan-Kaufmann, 1992, pp. 950–957.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[25] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2010, pp. 3485–3492.