Segmenting Snow Scene from CCTV using Deep Learning Approach

Punyanuch Borwarnginn [1], **Worapan Kusakunniran** [2], **Parintorn Pooyoi** [3], and **Jason H. Haga** [4]

ABSTRACT

Recently, data from many sensors has been used in a disaster monitoring of things, such as river water levels, rainfall levels, and snowfall levels. These types of numeric data can be straightforwardly used in a further analysis. In contrast, data from CCTV cameras (i.e. images and/or videos) cannot be easily interpreted for users in an automatic way. In a traditional way, it is only provided to users for a visualization without any meaningful interpretation. Users must rely on their own expertise and experience to interpret such visual information. Thus, this paper proposes the CNN-based method to automatically interpret images captured from CCTV cameras, by using snow scene segmentation as a case example. The CNN models are trained to work on 3 classes: snow, non-snow and non-ground. The non-ground class is explicitly learned, in order to avoid a confusion of the models in differentiating snow pixels from non-ground pixels, e.g. sky regions. The VGG-19 with pre-trained weights is retrained using manually labeled snow, non-snow and non-ground samples. The learned models achieve up to 85% sensitivity and 97% specificity of the snow area segmentation.

Keywords: Deep Learning, CNN, Snow Scene, CCTV, Segmentation

1. INTRODUCTION

For decades, attempts of impact reduction and early detection of natural disasters have gained high attention from around the world, in order to save human lives. Therefore, up-to-date and related data from several types of sensors is needed for a warning system to monitor the likelihood of natural disasters. For example, in Japan, such information is provided publicly via a website, including river water levels, rainfall levels, and snowfall levels [1]. It also provides videos capturing river conditions in real-time, from CCTV cameras.

However, these videos and images are reported to users without any additional meaningful information or interpretation. The users must rely on their own experience and judgment to interpret images and videos seen. An automatic way of interpreting meaningful information from an image would be helpful to general users. This paper uses snow scene detection as a case study to show how to provide useful information for disaster management.

As mentioned in [2][3], snow can be a key cause of several disasters. Early related warning would be useful for disaster management. Snow depth, snow cover area, and other related snow conditions could be used in combination with more information from other sensors (e.g. water level sensors [4][5], rainfall sensors [6][7], and dam level sensors [8]) to predict an early warning of disasters such as snow avalanches and floods. Then, a warning could be issued to notify citizens or human operators for further management.

This research project aims to develop a technique to automatically segment snow areas in images. This segmented information could be later used for automatic interpretation of snow levels in scenes captured from CCTV cameras. There have been several relevant works published recently. Bossu et al. [9] developed a technique to detect the presence of snow or rain through visual information. The classical Gaussian mixture model was used for foreground/background segmentation in videos. A foreground part was then used in a process of rain and snow detection, using a novel feature called a histogram of orientations of rain or snow streaks (HOS). The feature was extracted using geometric moments and followed a model of Gaussian uniform mixture. The technique of expectation maximization was used to distinguish two different distributions of rain and snow.

The next related work was published to detect clouds in RGB color remote sensing images, using deep pyramid networks [10]. The spatial texture information was computed to be used in pixel-level decision making. Also, the pre-trained CNN model was integrated to enhance accuracy of classification masks in an encoder layer. Experimental results showed that it outperformed other baselines and obtained

Manuscript received on September 28, 2019 ; revised on October 31, 2019.

Final manuscript received on November 12, 2019.

[1,2,3]The authors are with Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom, Thailand 73170, Phone(+66-2)441-0909, E-mail: punyanuch.bor@mahidol.edu, worapan.kun@mahidol.edu and parintorn.poo@gmail.com

[4]The author is with National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki, Japan 305-8560, Phone(+81-29)861-2000, E-mail: jh.haga@aist.go.jp

[2]Corresponding author: worapan.kun@mahidol.edu

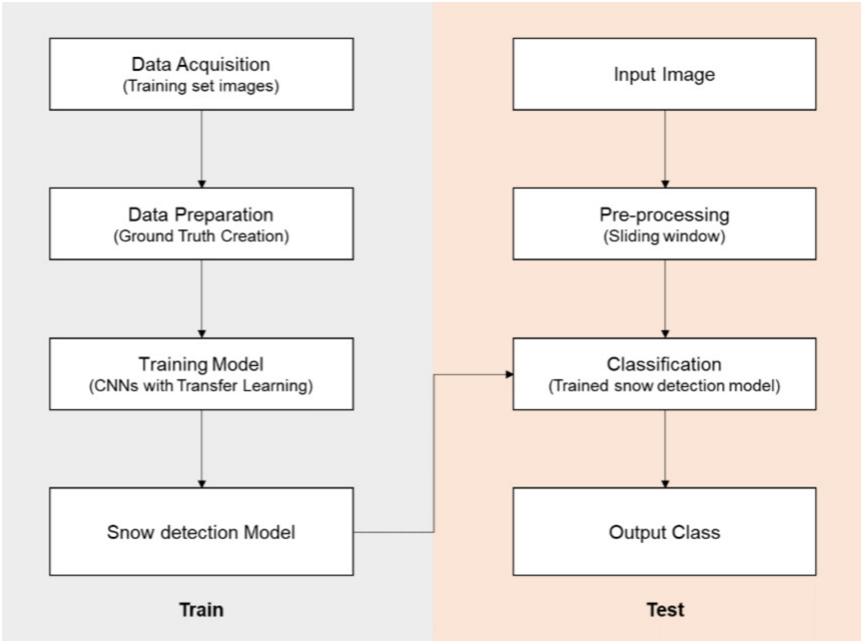


Fig.1: *Overview of the proposed method.*

perceptually superior performance on the datasets used. In addition, Liu et al. [11] proposed the DesnowNet based on a multi-stage deep network to remove translucent and opaque snow particles from images. The multi-scale design was also embedded into the network to deal with a diversity of snow types.

Similarly, Li et al. [12] proposed the composition generative adversarial networks to remove snowflakes from images. The networks contained a background module and a snow mask estimate module. The composition loss between the input snow image and the combination of generated clean image and estimated snow mask was used in the training process. In [13], Zhan et al. applied the deep convolution network to differentiate cloud regions from snow regions in multi-spectrum satellite images. The low-level spatial information and high-level semantic information were integrated using a multi-scale prediction strategy. Manually labeled cloud and snow samples were used in the training process.

Lately, in [14], the method of snow scene segmentation in CCTV images was introduced using a convolutional neural network with transfer learning. The performance was significantly enhanced using a post-processing step. The pre-defined masks which could not possibly be snow, such as sky regions, were excluded from the segmentation results. The specificity could move up from 78% to 98%. However, this would be too sensitive when the viewing angle of the CCTV camera is changed, since the masks must be pre-defined. In this paper, the whole process is autonomous and does not require any pre-defined masks. It is more robust in the event of a shifting viewpoint of the camera.

In the proposed method, the VGG-19 is adopted with transfer learning with two different scenarios. In the first scenario, it is trained for three classes including snow, non-snow, and non-ground classes. Thus, pixels classified as snow will be used to compute the final segmentation. In the second scenario, two models are trained for two classes each. The first model is trained for snow and non-snow classes, while the second model is trained for ground and non-ground classes. Then, the final snow segmentation will be made up of pixels belonging to both the snow class in the first model and the ground class in the second model. This is done to avoid a confusion differentiating between snow and sky (i.e. non-ground) regions. Experimental results show promising and comparable performance, when compared with the method proposed in [14].

The remainder of this paper is organized as follows. Section 2 explains details of the proposed method. Then, section 3 describes and discusses experimental results. Finally, conclusions are drawn in section 4.

2. PROPOSED METHOD

The overview of the proposed method of snow scene detection is illustrated in Figure 1. The framework consists of a training and a testing process. For the training process, it consists of three phases, including data acquisition, data preparation, and model training. The output is a snow detection model to be used in the testing part. A test image is pre-processed into small patches before submitting it to the trained model. Therefore, the model will detect these patches as snow, non-snow, or non-ground.

Fig.2: *Screenshot of river.co.jp.*

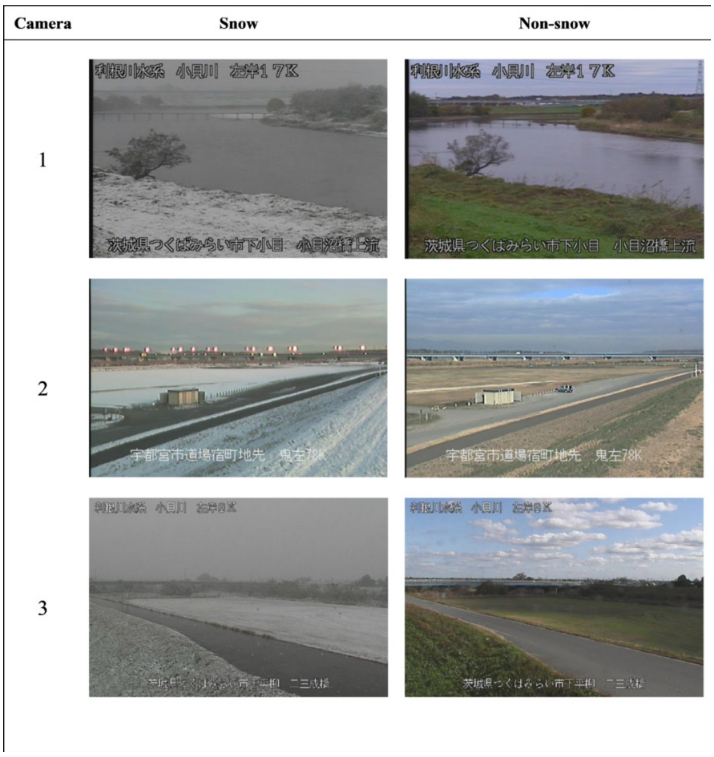


Fig.3: *Sample images from each camera.*

Fig.4: *A sample image of snow (left), the possible area of snow marked as white (middle), the actual area of snow marked as white (right).*

Fig.5: *A sample image of non-snow (left), possible area of snow marked as white (middle), the actual area of snow marked as white (right).*

Fig.6: *Sub-image of the original image SUB_O, the possible area of snow SUB_P, and the actual snow area SUB_S.*

2. 1 Training Method

2.1.1 Data Acquisition

The dataset is acquired from CCTV cameras on the Japanese disaster prevention information of river website [1] shown in Figure 2. Most CCTV cameras are positioned along the river. The images from each camera have the same size and the same angle. However, they are captured at different times and in different weather conditions. We chose daytime images with separation of snow and non-snow images.

2.1.2 Data Preparation

For the data preparation, we selected images from 3 cameras in which 36 images were snow and 9 images were non-snow images per camera, as shown in Figure 3. Since there is no additional information from the original images, such as snow area, the ground truth is required in order to train the model. We manually defined the regions of possible snow area and actual snow area as shown in Figures 4 and 5. With the original image and 2 region images, we performed a mapping process to generate the ground truth as follows:

- Using 50×50 pixels of a sliding window technique [15][16] on the original image O, the possible area P, and the actual snow S to generate a series of sub-images SUB_O, SUB_P, SUB_S was determined as seen in Figure 6.
- For each SUB_O, SUB_P, SUB_S
  - If 60% of pixels in SUB_P are black AND SUB_P is located on the top half, then SUB_O is NON-GROUND.
  - If 60% of pixels in SUB_S are white, then SUB_O is SNOW.
  - If 60% of pixels in SUB_P are white AND 60% of pixels of SUB_S are black, then SUB_O is NON-SNOW.
- SUB_O is saved into its class folder.
- Repeat for the entire image.

Therefore, the final classes are snow, non-ground (e.g. sky) and non-snow, with over 10,000 images per class.

2.1.3 Model Training

In deep learning [17][18][19][20], CNNs have been successfully applied in computer vision tasks such as image classification, object detection, and video recognition. Hence, a convolutional neural network (CNN) was selected for this work. There are several CNN architectures such as VGG-19 [21][22], AlexNet [23][24], InceptionV3 [25][26], and MobileNetV2 [27][28]. However, VGGNet was selected as the network model for this work because of its performance and lower complexity. VGG-16 and VGG-19 architectures are illustrated in Figure 7.

On the ImageNet dataset, VGG-19 achieved slightly better results compared to VGG-16. Therefore, we applied a transfer learning technique [21][29]

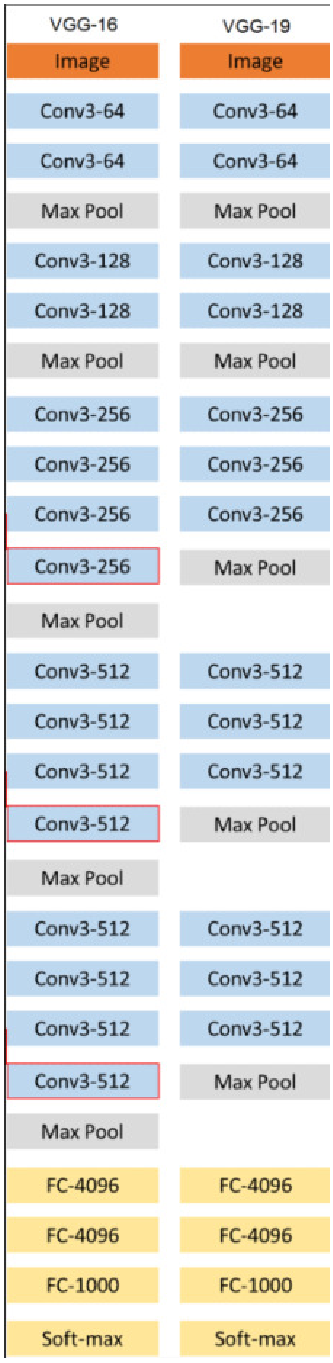


Fig. 7: *VGG-16 and VGG-19 architectures.*

using pre-trained weights from the ImageNet dataset to find the initial weights of the network and allowed it to calculate new weights for the training data. Once the training is done, it returns a model that will be used to split a test image into snow, non-snow, and non-ground parts.

In this paper, the network was applied with two different scenarios, as shown in Figures 8 and 9. In scenario 1, the network was re-trained for 3 classes: snow, non-snow, and non-ground. The non-ground class contained samples of regions that had very similar characteristics to snow. The non-ground samples cover sky and river regions. In scenario 2, two models were re-trained. The first model was re-trained for two classes: snow and non-snow. The second model was re-trained for two classes: ground and non-ground. We used two models in order to avoid confusion between snow and non-ground (i.e. sky and river) regions. As can be seen in Figure 9, the output snow region contains pixels belonging to the snow class and the ground class in the first and second models respectively.

2.2 Testing Method

In testing our model, an input image is pre-processed by using a sliding window with no overlapping to generate small sub-images of 50×50 pixels, as illustrated in Figure 10. Each sub-image is input to the trained snow detection model to classify whether this area is snow, non-snow or non-ground. Then the final output has marked regions of snow, non-snow, and non-ground as in Figure 11. All experiment settings are explained in the next section.

3. EXPERIMENTS

In our experiments, 39 images were taken from 3 CCTV cameras (13 images from each camera) [1]. Each image was split into patches as described in the proposed method section. They were used in both the training and validating processes. In each camera, patches from 10 images are used in the training, while patches from the remaining 3 images were used for validation. Validating patches were generated using a sliding window, which is not used for training the model. The ground truth was generated manually.

Two metrics are used to evaluate performance of the proposed method: sensitivity and specificity. They are calculated as shown in equations (1) and (2) [30].

$$sen = \frac{TP}{TP + FN} \tag{1}$$

sen is a sensitivity showing a true positive percentage. TP is a true positive counting a number of pixels of snow detected as snow. FN is a false negative counting the number of pixels of snow detected as non-snow.

$$spec = \frac{TN}{TN + FP} \tag{2}$$

$spec$ is a specificity showing a true negative percentage. TN is a true negative counting a number of pixels of non-snow detected as non-snow. FP is a false positive counting the number of pixels of non-snow detected as snow. Values of sensitivity and specificity range from 0 to 1, where 1 represents the highest score.

The proposed method is applied in two different scenarios. In the first scenario, the proposed network

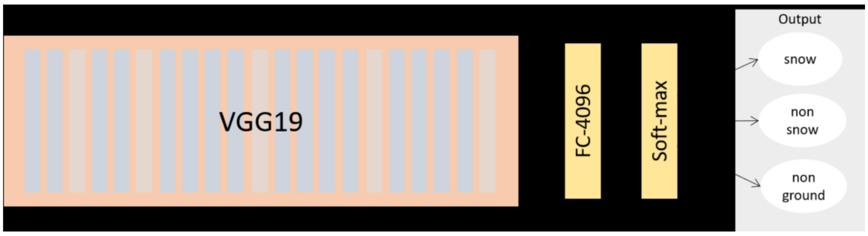


Fig.8: *Scenario 1 of the proposed method.*

/figures/model1.eps

Fig.9: *Scenario 2 of the proposed method.*

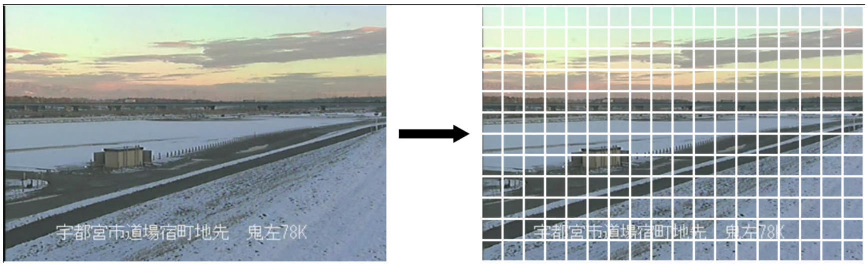


Fig.10: *A test image is pre-processed into small patches.*

Fig.11: *The output image containing snow area (white), non-snow area (black) and non-ground area (grey).*

is trained to classify patches into snow, non-snow, or non-ground classes. Sample patches of snow and non-ground classes could be confused. For example, a sky patch in the non-ground class could be very similar to a snow patch in the snow class. Thus, the non-ground class is split out explicitly to be the third class in the first scenario.

In the second scenario, two models are trained. The first model is trained to distinguish snow from non-snow pixels. The second model is trained to distinguish ground from non-ground pixels. The final snow pixels are pixels belonging to the snow class in the first model and the ground class in the second model. The experimental results of the proposed method are compared to the existing method [14], as shown in Table 1.

As shown in Table 1, the total number of pixels summing TP, TN, FP and FN is normalized to 100%. The proposed method clearly outperforms the existing technique [14]. This is mainly because the non-ground class is explicitly split out and trained separately from the snow class. In [14], several non-ground pixels (e.g. sky pixels) were wrongly classified as snow. This fault in segmentation made the specificity low. However, with the post-processing, the fault snow-pixels were removed using the pre-defined

Table 1: *Pixel-based accuracy of the snow scene segmentation.*

Method	TP (pixels)	TN (pixels)	FP (pixels)	FN (pixels)	sen (%)	$spec$ (%)
[14] without post-processing	13.71	64.51	17.78	4.09	77.02	78.39
[14] with post-processing	14.64	79.81	1.86	3.69	79.87	97.72
Proposed method: scenario 1	15.01	80.12	2.35	2.52	85.62	97.15
Proposed method: scenario 2	16.23	78.36	1.75	3.66	81.60	97.82

Table 2: *Pixel-based accuracy of the snow scene segmentation for the cross-cameras cases, using the proposed method.*

Training	Testing	TP (pixels)	TN (pixels)	FP (pixels)	FN (pixels)	sen (%)	$spec$ (%)
Cameras 1 & 2	Camera 3	12.59	75.63	8.57	3.21	79.68	89.82
Cameras 1 & 3	Camera 2	11.21	69.21	15.19	4.39	71.86	82.00
Cameras 2 & 3	Camera 1	13.17	72.78	8.84	5.21	71.65	89.13

Fig.12: *A sample result of snow segmentation. The three images are input image, groundtruth image, and resulted image respectively.*

non-ground regions. Thus, the specificity value was significantly increased. The pre-defined non-ground regions used in the post-processing step made the method proposed in [14] less sensitive to any shifting of a camera's viewpoint.

The proposed method achieved a promising performance of above 81% sensitivity and above 97% specificity. Scenario 1 gives better sensitivity with slightly lower specificity, when compared with scenario 2. The performance could be improved with a larger number of training samples, in order to avoid overfitting. A sample result of snow segmentation is shown in Figure 12.

In addition, the cross-camera cases are also evaluated, where two cameras are used in the training process and another remaining camera is used in the testing process. The proposed method using scenario 1 is used to evaluate the cross-cameras cases, since it achieves a better sensitivity when compared to the proposed method using scenario 2. The performance is shown in Table 2. It can be seen that the performance dropped when compared to the cases where all three cameras are used in the training process as reported in Table 1. This is because scenes seen in different cameras are varied, as shown in Figure 3.

4. CONCLUSION

This paper proposes a CNN-based method of the snow scene segmentation. The VGG19 pre-trained weights are used and re-trained using 2 different scenarios. The first scenario trains the network to classify patches into 3 classes: snow, non-snow and non-ground. Pixels in the snow class are the final segmentation results. The second scenario trains two models to classify patches into two classes. The first model is trained for snow and non-snow classes. The second model is trained for ground and non-ground classes. The final segmentation result contains pixels belonging to the snow class in the first model and the ground class in the second model. The proposed method achieves up to 85% sensitivity and 97% specificity. It is also validated in the cross-camera cases, which obtain the sensitivity in a range of 71-79% and the specificity in a range of 82-89%. This would be practical where a camera used may not be seen and trained beforehand in the training phase.

ACKNOWLEDGEMENT

This research project was partially supported by the ICT International Collaboration Fund of the National Institute of Advanced Industrial Science and Technology.

References

[1] M. Japan. (2019) Disaster prevention information of river. [Online]. Available: `http://www.river.go.jp/kawabou/ipTopGaikyo.do`

[2] W. Wang, T. Liang, X. Huang, Q. Feng, H. Xie, X. Liu, M. Chen, and X. Wang, "Early warning of snow-caused disasters in pastoral areas on the tibetan plateau, *Natural hazards and earth system sciences*, vol. 13, no. 6, pp. 14111425, 2013.

[3] W. Haeberli and C. Whiteman, "Snow and icerelated hazards, risks, and disasters: a general framework," in *Snow and Ice-Related Hazards, Risks and Disasters.* Elsevier, 2015, pp. 134.

[4] J. A. Hernández-Nolasco, M. A. W. Ovando, F. D. Acosta, and P. Pancardo, "Water level meter for alerting population about floods," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA).* IEEE, 2016, pp. 879884.

[5] Z. S. M. Odli, T. N. T. Izhar, A. R. A. Razak, S. Y. Yusuf, I. A. Zakarya, F. N. M. Saad, and M. Z. M. Nor, "Development of portable water level sensor for flood management system," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, pp. 53525357, 2016.

[6] E. Trono, M. Guico, N. Libatique, G. Tangonan, D. Baluyot, T. Cordero, F. Geronimo, and A. Parrenas, "Rainfall monitoring using acoustic sensors," in *TENCON 2012 IEEE Region 10 Conference.* IEEE, 2012, pp. 16.

[7] R. Stumptner, C. Lettner, and B. Freudenthaler, "Combining relational and nosql database systems for processing sensor data in disaster management," in *International Conference on Computer Aided Systems Theory.* Springer, 2015, pp. 663670.

[8] J. Abdelaziza, M. Addab, and H. Mcheicka, "An architectural model for fog computing," *Journal of Ubiquitous Systems and Pervasive Networks*, vol. 10, no. 1, pp. 2125, 2018.

[9] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International journal of computer vision*, vol. 93, no. 3, pp. 348367, 2011.

[10] S. Ozkan, M. Efendioglu, and C. Demirpolat, "Cloud detection from rgb color remote sensing images with deep pyramid networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium.* IEEE, 2018, pp. 69396942.

[11] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, "Desnownet: Context-aware deep network for snow removal," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 30643073, 2018.

[12] Z. Li, J. Zhang, Z. Fang, B. Huang, X. Jiang, Y. Gao, and J.-N. Hwang, "Single image snow removal via composition generative adversarial networks," *IEEE Access*, vol. 7, pp. 2501625025, 2019.

[13] Y. Zhan, J. Wang, J. Shi, G. Cheng, L. Yao, and W. Sun, "Distinguishing cloud and snow in satellite images via deep convolutional network," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 17851789, 2017.

[14] P. Pooyoi, P. Borwarnginn, J. H. Haga, and W. Kusakunniran, "Snow scene segmentation using cnn-based approach with transfer learning," in *2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).* IEEE, 2019, pp. 9598.

[15] C. Wojek, G. Dorko, A. Schulz, and B. Schiele, "Sliding-windows for rapid object class localization: A parallel technique," in *Joint Pattern Recognition Symposium.* Springer, 2008, pp. 7181.

[16] A. Rosebrock, "Sliding windows for object detection with python and opencv," *PylmageSearch, Navigation*, 2015.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436, 2015.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT Press, 2016.

[19] E. Charniak, *Introduction to deep learning.* MIT Press, 2019.

[20] A. Gulli and S. Pal, *Deep Learning with Keras.* Packt Publishing Ltd, 2017.

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

[22] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint arXiv:1605.07678*, 2016.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 10971105.

[24] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 2016, pp. 28182826.

[26] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2017, pp. 783787.

[27] 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*, 2018, pp. 45104520.

[28] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 428438.

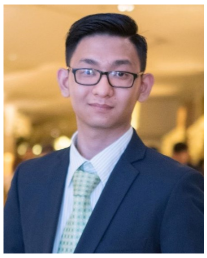
[29] B. Q. Huynh, H. Li, and M. L. Giger, "Digital mammographic tumor classification using transfer learning from deep convolutional neural networks," *Journal of Medical Imaging*, vol. 3, no. 3, p. 034501, 2016.

[30] W. Kusakunniran, Q. Wu, P. Ritthipravat, and J. Zhang, "Hard exudates segmentation based on learned initial seeds and iterative graph cut," *Computer methods and programs in biomedicine*, vol. 158, pp. 173183, 2018.

Punyanuch Borwarnginn received the B.Sc. degree in Information and Communication Technology from Mahidol University, Nakhon Pathom, Thailand, in 2009, and the M.Sc. degree in Informatics from the University of Edinburgh, Edinburgh, United Kingdom, in 2011. She is currently a PhD student in Computer Science, Faculty of Information and Communication Technology, Mahidol University. Her current research interests include image processing, biometrics, computer vision, pattern recognition, and machine learning.

Worapan Kusakunniran received the B.Eng. degree in computer engineering from the University of New South Wales (UNSW), Sydney, Australia, in 2008, and the Ph.D. degree in computer science and engineering from UNSW, in cooperation with the Neville Roach Laboratory, National ICT Australia, Kensington, Australia, in 2013. He is currently a Lecturer with the Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom, Thailand. He is the author of several papers in top international conferences and journals. His current research interests include biometrics, pattern recognition, medical image processing, computer vision, multimedia, and machine learning.

Dr. Kusakunniran served as a Program Committee Member for many international conferences and workshops. Also, he has served as a Reviewer for several international conferences and journals, such as the International Conference on Pattern Recognition, the IEEE International Conference on Image Processing, the IEEE International Conference on Advanced Video and Signal based Surveillance, the Pattern Recognition, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNET- ICS, PART B: CYBERNETICS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and the IEEE SIGNAL PROCESSING LETTERS. He was a recipient of the ICPR Best Biometric Student Paper Award in 2010, and also a winner of several national and international innovation contests.



Parintorn Pooyoi received the B.Sc. degree in information and communication technology from University of Mahidol, Thailand in 2019. He is currently a Java developer with Siam Commercial Bank, Bangkok, Thailand. His research interests include image processing, computer vision, machine learning, and deep learning.

Jason H. Haga is currently a Chief Senior Research Scientist of the Cyber-Physical Cloud Research Group at The National Institute of Advanced Industrial Science and Technology (AIST). For over 20 years, Dr. Haga has focused on multidisciplinary research. Past research includes the design and implementation of biomedical applications for grid computing environments and tiled display walls. He also has collaborated with cultural heritage institutions to deploy novel interactive exhibits to engage the public in learning. Current projects of interest include immersive visualization of data and applying machine learning in data intensive applications. He is actively involved with the Pacific Rim Application and Grid Middleware Assembly (PRAGMA) community, where he leads the Student Mentoring activities. Dr. Haga is continuing this mentorship effort at AIST by leading an international internship program that strategically positions AIST as an international hub for computer science research training. With over 14 years of global collaborative efforts with technologists and domain scientists in the PRAGMA community, he continues to look for interdisciplinary research opportunities connecting scientists to advance research on a global scale. Dr. Haga earned a PhD in biomedical engineering from the University of Tennessee, Memphis and did postdoctoral work at UC San Diego.