



# A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets

Van Hiep Phung<sup>✉</sup> and Eun Joo Rhee<sup>\*✉</sup>, *Member, KIICE*

Department of Computer Engineering, Hanbat National University, Daejeon 34158, Korea

## Abstract

Accurate classification of cloud images is a challenging task. Almost all the existing methods rely on hand-crafted feature extraction. Their limitation is low discriminative power. In the recent years, deep learning with convolution neural networks (CNNs), which can auto extract features, has achieved promising results in many computer vision and image understanding fields. However, deep learning approaches usually need large datasets. This paper proposes a deep learning approach for classification of cloud image patches on small datasets. First, we design a suitable deep learning model for small datasets using a CNN, and then we apply data augmentation and dropout regularization techniques to increase the generalization of the model. The experiments for the proposed approach were performed on SWIMCAT small dataset with  $k$ -fold cross-validation. The experimental results demonstrated perfect classification accuracy for most classes on every fold, and confirmed both the high accuracy and the robustness of the proposed model.

**Index Terms:** Cloud classification, CNN, Data augmentation, SWIMCAT dataset

## I. INTRODUCTION

Research on the cloud and its characteristics plays a very important role for many applications: e.g., climate modeling, weather prediction, meteorology study, solar energy production and satellite communication [1-6]. Cloud classification is an essential role in cloud observation. However, at present, existing cloud classification is done by professionally trained observers. This method is highly time consuming, and depends on the experience of observers; moreover, there are some problems that cannot be well handled by the human observers [7]. Therefore, automatic classification of the cloud is a much-needed task.

Much research on classification of cloud images has been conducted. Buch and Sun [8] applied binary decision trees to classify pixels in the whole sky imager (WSI) images into five cloud types. Singh and Glennen [9] proposed five differ-

ent feature extraction methods (autocorrection, co-occurrence matrices, edge frequency, Law's features and primitive length) and used the  $k$ -nearest neighbor and neural network for cloud classification. Calbo and Sabburg [10] applied a Fourier transform for cloud-type recognition. Heinle et al. [11] predefined several statistical features to describe color and texture and used a  $k$ -nearest neighbor classifier. Liu et al. [12] used an illumination-invariant completed local ternary pattern descriptor for cloud classification. Liu et al. [13] extracted some cloud structure features and used a simple classifier called the rectangle method for classification of infrared cloud images. Liu et al. [14] proposed a salient local binary pattern for cloud classification. Liu et al. [15] used a weighted local binary descriptor. Dev et al. [16] proposed a modified texton-based approach to categorize cloud image patches. Luo et al. [17] combined manifold features and text features and then used a support vector machine (SVM) to

Received 22 July 2018, Revised 31 August 2018, Accepted 31 August 2018

\*Corresponding Author Eun Joo Rhee (E-mail: [ejrhee@hanbat.ac.kr](mailto:ejrhee@hanbat.ac.kr), Tel: +82-42-821-1205)

Department of Computer Engineering, Hanbat National University, 125, Dongseo-daero, Yuseong-gu, Daejeon 34158, Korea.

Open Access

<https://doi.org/10.6109/jicce.2018.16.3.173>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

classify cloud images. Gan et al. [18] proposed cloud type classification using duplex norm-bounded sparse coding. Nevertheless, most of these approaches are based on hand-crafted features so each method needs to find its empirical parameters.

Recently, development of deep learning is increasing rapidly; in particular, convolutional neural networks (CNNs) have shown outstanding performance in image classification [19]. CNNs are able to “learn” features from the image data, so there is no need any feature extraction method. Some recent studies used CNNs for cloud classification. Shi et al. [20] proposed a CNN model to extract features and used SVM to classify cloud images. Ye et al. [21, 22] improved feature extraction by using both CNN and Fisher vector and used SVM to classify cloud images. Zhang et al. [23] proposed transferring deep visual information. Generally, these deep learning approaches achieved promising results; however, all the above deep learning approaches only utilized the CNN to extract features. They needed other methods for classification, and, they used pre-trained CNN models to extract features. However, for small datasets, the lack of sufficient image samples makes it difficult to converge in the end-to-end learning manner.

This paper proposes a deep learning approach for classification of cloud images on small datasets. First, we design a deep learning model that includes both the feature extraction part and classification part using CNN, and then we apply two regularization techniques, data augmentation and dropout, to generalize our model. Our experiment is performed on the SWIMCAT dataset.

## II. SWIMCAT DATASET

The Singapore Whole-sky IMaging CATegories (SWIMCAT) dataset was introduced by Dev et al. [16] and the images were captured during 17 months from January 2013

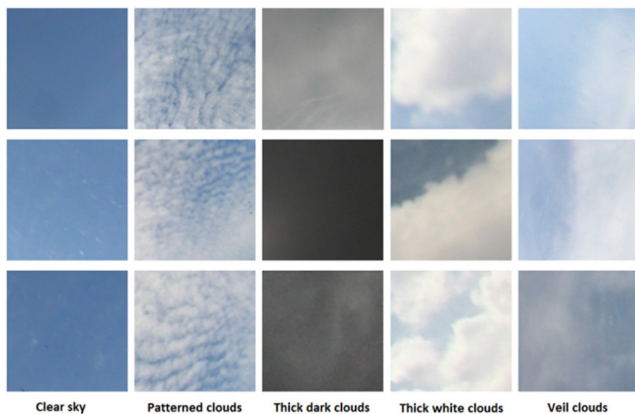


Fig. 1. Sample images of each class from the SWIMCAT dataset.

to May 2014 in Singapore using the Wide Angle High-Resolution Sky Imaging System (WAHRISIS), a calibrated ground-based whole-sky imager [24]. The dataset has five distinct categories. The five categories (clear sky, patterned clouds, thick dark clouds, thick white clouds, and veil clouds) are defined on the basic of visual characteristics of sky/cloud conditions and consultation with experts from the Singapore Meteorological Services.

The SWIMCAT dataset has 784 images of sky/cloud patches: clear sky, 224 images; patterned cloud, 89 images; thick dark cloud, 251 images; thick white cloud, 135 images; and veil cloud, 85 images. The dimensions of all the images are  $125 \times 125$  pixels. Some random images of each class from the SWIMCAT dataset are shown in columns in Fig. 1.

## III. METHODS

### A. Convolutional Neural Networks

CNNs were inspired by the human visual system [25, 26]. They are the state-of-the-art approaches not only for pattern recognition tasks but also for object detection tasks, especially with the development of computing capacity. Krizhevsky et al. [19] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 competition [27, 28] with brilliant deep CNNs that show the great power of deep CNNs.

Unlike many other pattern recognition algorithms, CNNs combine both feature extraction and classification. A schematic representation of a basic CNN (inspired by [29]) is shown in Fig. 2. The given network consists of five different layers: input, convolution, pooling, fully-connected, and output. The input layer specifies a fixed size for the input images, i.e., images may have to be resized accordingly. The image is then convolved with multiple learned kernels using shared weights. Next, the pooling layers reduce the size of the image while trying to maintain the contained information. These two layers comprise the feature extraction part. Afterwards, the extracted features are weighted and combined in the fully-connected layers. This represents the clas-

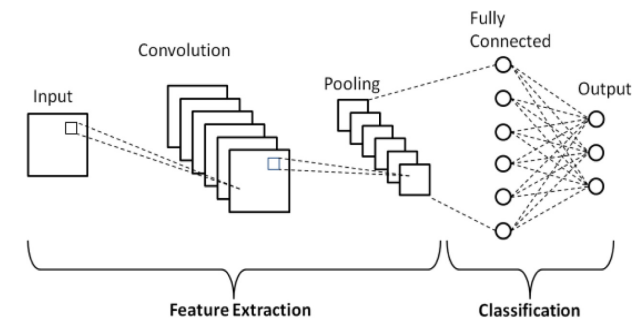


Fig. 2. Block diagram of a CNN.

sification part of the CNN. Finally, there is one output neuron for each object category in the output layer.

## B. Model Design

As we mentioned above, the architecture of CNNs follow this pattern:

$$IN \Rightarrow CONV \Rightarrow POOL \Rightarrow FC \Rightarrow OUT \quad (1)$$

where IN is the input layer, CONV is the convolution layer, POOL is the pooling layer, FC is the fully connected layer, and OUT is the output layer. However, for “deep learning” these layers are stacked together in a particular pattern that yields a CNN model. Basically, the most common form of CNN architecture is that many layers of CONV and POOL are repeated until the volume, width and height is small, at which point we apply some FC layers. The most common CNN architectures have the following pattern [30]:

$$IN \Rightarrow [CONV \Rightarrow POOL?]*M \Rightarrow [FC]*N \Rightarrow OUT \quad (2)$$

where “\*” indicates repetition and “?” indicates an optional pooling layer. For simplicity, we do not mention activation, but by default, the activation always follows CONV layers and FC layers. Theoretically, the larger number of convolutional layers ( $M$  is large) extracts more detailed features of input images; however, it needs a large amount of learning data. In this study, we selected less number of convolutional layers ( $M$  is small) and increased the generalization of our model by applying regularization techniques.

We selected  $M = 3$  and  $N = 2$ , and the feature extraction part consists of three groups of a CONV layer and a POOL layer. The classification part consists of two fully connected layers.

## C. Model Regularization

In this research, we used the SWIMCAT dataset which consists of only 784 image patches, which is a very small number for deep learning and makes it very easy to get overfitting. To overcome this problem we used two regularization methods. The first augments data passed into the network for training, and the second modifies the network architecture. They are data augmentation and dropout, respectively.

### 1) Data Augmentation

Data augmentation is a method that is used to generate new training samples from the original data samples by augmenting the samples via a number of random transformations such that the class labels are not changed. The purpose of data augmentation is to increase the generalizability of the model. With that, the robustness is improved and overfitting

**Table 1.** Augmentation parameters

No.	Parameter	Augmentation
1	Rotation (°)	40
2	Width shift (%)	20
3	Height shift (%)	20
4	Shear (%)	20
5	Zoom (%)	20
6	Horizontal flip	Yes
7	Vertical flip	Yes

is prevented.

In this study we did the data augmentation by applying random geometric transforms. The detailed configuration parameters of each augmentation method are shown in Table 1. We applied random rotation with range of 40°. We also applied random translation both vertically and horizontally with a range of 20%. Random shear transformation and random zoom are applied with the range of 20%. Finally, a horizontal flip and vertical flip were also performed randomly. By so doing, during the training time, our model will never see the exact same image twice. We only applied data augmentation at training time and did not apply data augmentation at the testing time and evaluation time of our trained networks.

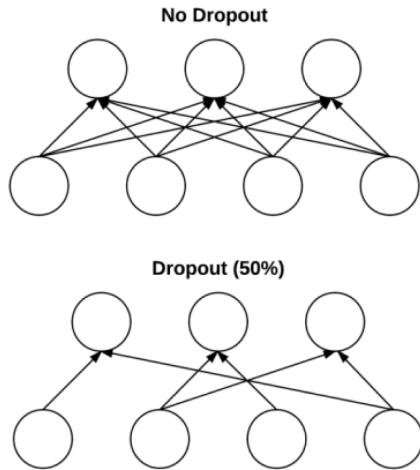
As mentioned above, by applying data augmentation, the network will never see the same input twice, but the inputs it sees are still heavily intercorrelated because they come from a small number of original images. The process cannot produce new information; it can only remix existing information so only data augmentation is not enough to completely get rid of overfitting. We applied one more regularization technique, dropout.

### 2) Dropout

Dropout was proposed by Srivastava et al. [31]; it randomly drops units with probability  $p$  from the neural network during training. Fig. 3 visualizes the dropout concept with dropout probability  $p = 0.5$ , the top parts are fully-connected layers without dropout, and the bottom parts are fully-connected layers with 50% of the connections dropped.

In this study, we used dropout for both the extraction part and the classification part. In the extraction part, we applied dropout between the second and third convolution groups with probability  $p = 0.25$ . In the classification part, we applied dropout  $p = 0.5$  between FC layers.

The detailed architecture of our implemented CNN model in which we included dropout layers and activation layers is shown in Table 2. We used 32 filters for the first and second convolutional layers and 16 filters for the third convolutional layer. Except that the activation of the output layer is softmax, all the other activations are ReLU. We apply  $3 \times 3$  ker-



**Fig. 3.** Drop out concept visualization: (top) no dropout and (bottom) dropout 50% of the connections.

**Table 2.** Architecture of our implemented convolutional network

No.	Layer	Output size	Filter/stride size	Dropout
1	Input	125×125×3	-	-
2	Convolution	123×123×32	3×3	-
3	ReLU	123×123×32	-	-
4	Max Pooling	61×61×32	2×2	-
5	Convolution	59×59×32	3×3	-
6	ReLU	59×59×32	-	-
7	Max Pooling	29×29×32	2×2	-
8	Dropout	29×29×32	-	0.25
9	Convolution	27×27×16	3×3	-
10	ReLU	27×27×16	-	-
11	Max pooling	13×13×16	2×2	-
12	Flatten	1×1×2704	-	-
13	Dropout	1×1×2704	-	0.5
14	Fully connected	1×1×32	-	-
15	ReLU	1×1×32	-	-
16	Fully connected	1×1×5	-	-
17	Softmax	1×1×5	-	-

nel sizes for CONV layers and 2 × 2 window sizes for the POOL layers.

## IV. EXPERIMENT

### A. K-Fold Cross-Validation

The gold standard for machine learning model evaluation is *k*-fold cross-validation. In this study we randomly split our data into 5 partitions (*k* = 5) of equal size. For each partition *i*, we trained our model on the remaining four partitions, and tested it on partition *i*. The final score was the averages of



**Fig. 4.** Schematic of 5-fold cross-validation.

all 5 scores obtained. The schematic of our 5-fold cross-validation is shown in Fig. 4.

### B. Experimental Environment

The hardware we used to implement the proposed network was a single PC Intel Core i5-7500 with the graphics processor NVIDIA GeForce GTX 1060 with 4 GB memory equipped. The software was developed using Python programming language based on Keras deep learning library [32] with back-end as TensorFlow [33].

### C. Experimental Results

We experimented on our approach as described in Section III on SWIMCAT database. We used batch size of 32 images, and RMSProp optimizer. The experimental result of five test sets after running 1,000 epochs for each fold, are shown in Table 3. We obtained an average accuracy of 0.986 with minimum and maximum accuracies at 0.975 and 0.994, respectively.

The confusion matrixes are shown in Fig. 5. In each confusion matrix, each column of the matrix represents the instances in a predicted class, and each row represents the instances in the actual class. Our proposed approach achieves perfect classification accuracy for most classes. The

**Table 3.** Experimental results

No.	Fold	Accuracy (%)
1	Fold 1	98.73
2	Fold 2	99.36
3	Fold 3	99.36
4	Fold 4	97.45
5	Fold 5	98.06
6	Average	98.59

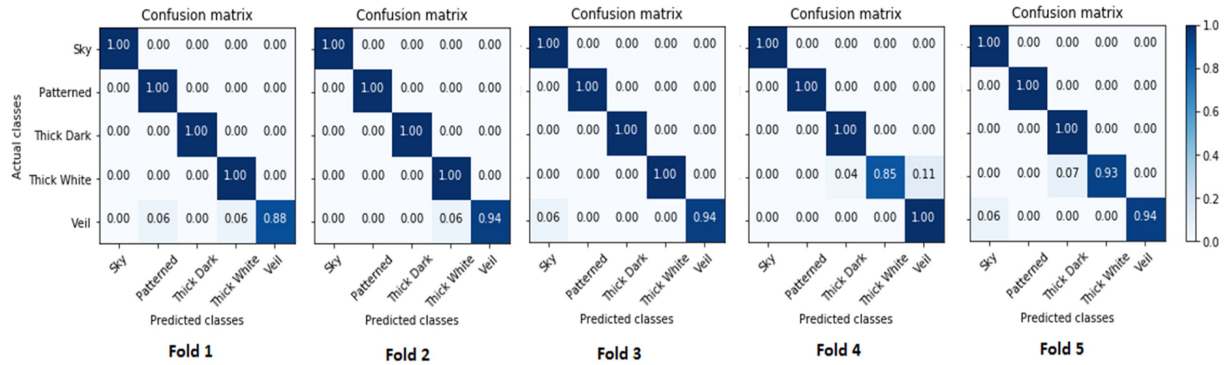


Fig. 5. Confusion matrixes using the proposed approach.

sky, patterned clouds and thick dark cloud classes always achieve 100% accuracy for all 5 folds.

## V. CONCLUSION

This paper presented a deep learning solution for classification of cloud image patches on small datasets. For this research we used the dataset SWIMCAT, which consist of only 784 images. That is very small for deep learning applications, and it is very easy to get overfitting. To solve this problem, we designed a CNN model with enough convolutional layers for small datasets, and we applied two regularization methods, data augmentation and dropout to thoroughly treat the overfitting problem. In addition, the gold standard for the machine learning model, the  $k$ -fold cross-validation was also applied. Experimental results confirmed that we achieved perfect classification accuracy for all classes and all folds, with a minimum accuracy of 97.45% and maximum accuracies of 99.36%. These results prove that the proposed model not only achieves high accuracy but is also robust.

## REFERENCES

- [1] D. L. Hartmann, M. E. Ockert-bell, and M. L. Michelsen, "The effect of cloud type on Earth's energy balance: global analysis," *Journal of Climate*, vol. 5, no. 11, pp. 1281-1304, 1992. DOI: 10.1175/15200442(1992)005<1281:TEOCTO>2.0.CO;2.
- [2] G. A. Isaac and R. A. Stuart, "Relationships between cloud type and amount, precipitation, and surface temperature in the Mackenzie River Valley-Beaufort Sea area," *Journal of Climate*, vol. 9, no. 8, pp. 1921-1941, 1996. DOI: 10.1175/1520-0442(1996)009<1921:RBCTAA>2.0.CO;2.
- [3] F. Yuan, Y. H. Lee, and Y. S. Meng, "Comparison of radio-sounding profiles for cloud attenuation analysis in the tropical region," in *Proceedings of IEEE International Symposium on Antennas and Propagation*, Memphis, TN, pp. 259-260, 2004. DOI: 10.1109/APS.2014.6904461.
- [4] Y. Liu, J. R. Key, and X. Wang, "The influence of changes in cloud cover on recent surface temperature trends in the Arctic," *Journal of Climate*, vol. 21, no. 4, pp. 705-715, 2008. DOI: 10.1175/2007JCLI1681.1.
- [5] F. Cui, R. Ju, Y. Ding, H. Ding, and X. Cheon, "Prediction of regional global horizontal irradiance combining ground-based cloud observation and numerical weather prediction," *Advanced Material Research*, vol. 1073-1076, pp. 388-394, 2014. DOI: 10.4028/www.scientific.net/amr.1073-1076.388.
- [6] M. C. Naud, J. F. Booth, and A. D. Del Genio, "The relationship between boundary layer stability and cloud cover in the post-cold-frontal region," *Journal of Climate*, vol. 29, no. 22, pp. 8129-8149, 2016. DOI: 10.1175/JCLI-D-15-0700.1.
- [7] D. Pages, J. Calbo, J. A. Gonzalez, and J. Badosa, "Comparison of several ground-based cloud detection techniques," in *Abstracts of European Geophysical Society XXVII Assembly*, Nice, France, pp. 269-299, 2002.
- [8] K. A. Buch and C. H. Sun "Cloud classification using whole-sky imager data," in *Proceedings of the 9th Symposium on Meteorological Observations and Instrumentation*, Charlotte, NC, 1995.
- [9] M. Singh and M. Glennen, "Automated ground-based cloud recognition," *Pattern Analysis and Applications*, vol. 8, no. 3, pp. 258-271, 2005. DOI: 10.1007/s10044-005-0007-5.
- [10] J. Calbo and J. Sabburg, "Feature extraction from whole-sky ground-based images for cloud-type recognition," *Journal of Atmospheric and Oceanic Technology*, vol. 25, no. 1, pp. 3-14, 2008. DOI: 10.1175/2007JTECHA959.1.
- [11] A. Heinle, A. Macke, and A. Srivastav, "Automatic cloud classification of whole sky images," *Atmospheric Measurement Techniques*, vol. 3, no. 3, pp. 557-567, 2010. DOI: 10.5194/amt-3-557-2010.
- [12] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, "Illumination-invariant completed LTP descriptor for cloud classification," in *Proceedings of the 5th International Congress on Image Signal Processing (CISP)*, Chongqing, China, pp. 449-453, 2012. DOI: 10.1109/CISP.2012.6469765.
- [13] L. Liu, X. Sun, F. Chen, S. Zhao, and T. Gao, "Cloud classification based on structure features of infrared images," *Journal of Atmospheric and Oceanic Technology*, vol. 28, no. 3, pp. 410-417, 2011. DOI: 10.1175/2010JTECHA1385.1.
- [14] S. Liu, C. Wang, B. Xiao, Z. Zhang, and Y. Shao, "Salient local binary pattern for ground-based cloud classification," *Acta Meteorologica Sinica*, vol. 27, no. 2, pp. 211-220, 2013. DOI: 10.1007/s13351-013-0206-8.
- [15] S. Liu, Z. Zhang, and X. Mei, "Ground-based cloud classification using weighted local binary patterns," *Journal of Applied Remote Sensing*, vol. 9, no. 1, article no. 095062, 2015. DOI: 10.1117/1.JRS.9.095062.

- [16] S. Dev, Y. H. Lee, and S. Winkler, "Categorization of cloud image patches using an improved texton-based approach," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, pp. 422-426, 2015. DOI: 10.1109/ICIP.2015.7350833.
- [17] Q. Luo, Y. Meng, L. Liu, X. Zhao, and Z. Zhou, "Cloud classification of ground-based infrared images combining manifold and texture features," *Atmospheric Measurement Techniques*, 2017. DOI: 10.5194/amt-2017-402.
- [18] J. Gan, W. Lu, Q. Li, Z. Zhang, J. Yang, Y. Ma, and W. Yao, "Cloud type classification of total-sky images using duplex norm-bounded sparse coding," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 7, pp. 3360-3372, 2017. DOI: 10.1109/JSTARS.2017.2669206.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing System*, vol. 25, pp. 1097-1105, 2012.
- [20] C. Shi, C. Wang, Y. Wang, and B. Xiao, "Deep convolutional activations-based features for ground-based cloud classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 6, pp. 816-820, 2017. DOI: 10.1109/LGRS.2017.2681658.
- [21] L. Ye, Z. Cao, Y. Xiao, and W. Li, "Ground-based cloud image categorization using deep convolutional visual features," in *Proceeding of 2015 IEEE International Conference on Image Processing*, Quebec City, QC, Canada, pp. 4808-4812, 2015. DOI: 10.1109/ICIP.2015.7351720.
- [22] L. Ye, Z. Cao, and Y. Xiao, "DeepCloud: ground-based cloud image categorization using deep convolutional features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 10, pp. 5729-5740, 2017. DOI: 10.1109/TGRS.2017.2712809.
- [23] Z. Zhang, D. Li, S. Liu, B. Xiao, and X. Cao, "Multi-view ground-based cloud recognition by transferring deep visual information," *Applied Sciences*, vol. 8, no. 5, article no. 748, 2018. DOI: 10.3390/app8050748.
- [24] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "WAHSIS: a low-cost high-resolution whole sky imager with near-infrared capabilities," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXV* (Vol. 9071). Bellingham, WA: International Society for Optics and Photonics, 2014. DOI: 10.1117/12.2052982.
- [25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998. DOI: 10.1109/5.726791.
- [26] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193-202, 1980. DOI: 10.1007/BF00344251.
- [27] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, "ImageNet large scale visual recognition competition 2012," 2012 [Internet], Available: [www.image-net.org/challenges/LSVRC/2012/](http://www.image-net.org/challenges/LSVRC/2012/).
- [28] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255, 2009. DOI: 10.1109/CVPR.2009.5206848.
- [29] L. Hertel, E. Barth, T. Kaster, and T. Martinetz, "Deep convolutional neural networks as generic feature extractors," in *Proceedings of 2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, pp. 1-4, 2015. DOI: 10.1109/IJCNN.2015.7280683.
- [30] A. Karpathy, "Convolutional Networks," [Internet], Available: <http://cs231n.github.io/convolutional-networks/>.
- [31] H. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [32] Keras: The Python Deep Learning library [Internet], Available: <https://keras.io/>.
- [33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al., "TensorFlow: large-scale machine learning on heterogeneous distributed systems," 2016 [Internet], Available: <https://arxiv.org/abs/1603.04467>.



### Van Hiep Phung

received his B.S. degree in Automatic Control from Hanoi University of Science and Technology, Hanoi, Vietnam in 2005. In 2009, he received his M.S. degree in Automation & Control in Graduate Institute of Automation and Control from National Taiwan University of Science and Technology (TaiwanTech). He is currently conducting research in the area of deep learning for computer vision in the Artificial Intelligence and Computer Vision Lab in the Graduate School of Information and Communications, Hanbat National University. He is interested in deep learning, computer vision, and pattern recognition.



### Eun Joo Rhee

is a Professor of Department of Computer Engineering at College of Information Technology, Hanbat National University, Daejeon, Korea, since 1989. He received the Ph.D. degree in electronics engineering from Chungnam National University in 1989. He was a postdoctoral fellow in Graduate School of Image Science and Technology of Tokyo Institute of Technology in Japan from 1994 to 1995, and a visiting professor in Oregon Graduate Institute of Science and Technology in America from 1998 to 1999. His research interests are in image processing, pattern recognition, computer vision and artificial intelligence.