International Journal of Control, Automation and Systems 15(6) (2017) 2942-2949
http://dx.doi.org/10.1007/s12555-016-0332-z
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Multi-task Convolutional Neural Network System for License Plate Recognition

Hong-Hyun Kim, Je-Kang Park, Joo-Hee Oh, and Dong-Joong Kang*

Abstract: License plate recognition is an active research field as demands sharply increase with the development of Intelligent Transportation System (ITS). However, since the license plate recognition(LPR) is sensitive to the conditions of the surrounding environment such as a complicated background in the image, viewing angle and illumination change, it is still difficult to correctly recognize letters and digits on LPR. This study applies Deep Convolutional Neural Network (DCNN) to the license plate recognition. The DCNN is a method of which the performance has recently been proven to have an excellent generalization error rate in the field of image recognition. The proposed layer structure of the DCNN used in this study consists of a combination of a layer for judging the existence of a license plate and a layer for recognizing digits and characters. This learning method is based on Multi-Task Learning (MTL). Through experiments using real images, this study shows that this layer structure classifies digits and characters more accurately than the DCNN using a conventional layer does. We also use artificial images generated directly for training model.

Keywords: Deep convolutional neural network, license plate recognition, machine learning, multi task learning.

1. INTRODUCTION

License plate recognition technology has been studied actively as demands sharply increase with the development of the ITS. In particular, it is the technology already widely used in the application fields such as freeway tollgate, parking vehicle access control, and traffic data collection. Since the license plate recognition is greatly affected by the font, color and viewing angle of the license plate, the number of license plates included in one image, and illumination, it still comes under a challenging and difficult field. Existing traditional approach for license plate recognition is based on image processing method and can be classified broadly into three phases such as license plate feature extraction, LP segmentation, and the recognition of digits and characters on the plate. The advantages and disadvantages of the algorithms used in each phase are described in detail in the review paper [1].

However, all of the algorithms and methods described in this paper work correctly only under constrained conditions. Usually, the geometric shape model and feature, according to the properties of the license plate to be used in image processing, should be decided directly by men with experience in the field, and the image processing procedure, too, should follow the predefined method. Since the image processing procedure empirically decided by men cannot express all cases appearing in the actual scene, recognition performance deteriorates, and the feature selected and extracted by the empirical method hardly can correspond to various environmental changes such as complicated background or illumination change.

This study applies Deep Convolutional Neural Network (DCNN), of which the performance has recently been proven in image classification, speech recognition, and object recognition to license plate recognition. The image recognition method using DCNN does not distinguish between feature extraction, image segmentation, and recognition. The DCNN directly extracts classification parameters optimized for the given problem by self-learning of low- to high-level features from several cases of training sample data with labels. Extracted feature can be directly applied to the detection and recognition of digits and characters of the license plate concerning the input image. In Summary, DCNN can recognize license plate only with the acquired image.

This study combines Multi-Task Learning (MTL)

Manuscript received June 1, 2016; revised December 31, 2016; accepted April 5, 2017. Recommended by Associate Editor Myung Geun Chun under the direction of Editor PooGyeon Park. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (2016R1A2B4007608), and National IT Industry Promotion Agency(NIPA) grant funded by the Korea government(MSIT) (S0602-17-1001).

Hong-Hyun Kim, Je-Kang Park, and Dong-Joong Kang are with the School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, 46241, Korea (e-mails: {hong12, jkp, djkang}@pusan.ac.kr). Joo-Hee Oh is with the Company of LG Electronics, 170, Seongsanpaechong-ro, Seongsan-gu, Changwon-si, Gyeongsangnam-do, 51533, Korea (e-mail: joohee.oh@lge.com).
* Corresponding author.

framework with existing Convolutional Neural Network (CNN) structure and applies it to the issues of the distinction of the existence of digits and characters of the license plate and the individual classification of digits and characters. The MTL is a method of simultaneous learning of two or more related issues by including in one network structure [2]. Zhang *et al*. applied this method to the issue of face detection and proved that its performance improved from the method using a single CNN [3]. Thus, this study designs a structure for the application of the MTL to license plate recognition and proves that this structure is proven through experiment that it is more effective than the method using a single CNN structure. On the other hand, in order to expect high accuracy of DCNN, it is important to prepare enough training data having the target value. In this paper, we show how to prepare the license plate data directly without taking its pictures or downloading it on the web.

This paper is organized as follows. Section 2 introduces license plate recognition and related works of deep learning. Section 3 describes the DCNN structure used in this study and the concrete methodology for the application of license plate recognition. Section 4 presents experiment environment, data preparation, and experimental results, and lastly, Section 5 mentions the conclusions and the possibility of improvements in the future.

2. RELATED WORKS

2.1. License plate recognition

License plate recognition is a field where studies have actively been conducted, in the method using the traditional image processing method of the past to the method using deep learning recently. Most of the traditional methods for license plate recognition consist of three phases such as license plate feature extraction, LP region segmentation, and license plate recognition. Of these, the feature is selected on the basis of the degree of the distinction between the license plate and the background part, so geometric shapes, colors and textures of the license plate are considered [1,4,5]. Accordingly, recognition performance is greatly affected by hand-crafted image features. LP region segmentation is the process of differentiating the area, including the license plate from the background using the extracted image feature. In the recognition phase, the method using template matching is an intuitive and universal method [6, 7]. This method recognizes digits and characters, comparing the similarity of the pixel intensity between the template and the license plate. The template matching is easy to use, but it has a weakness that it is sensitive to the rotation or noise of the target of detection. For license plate recognition, the machine learning-based method is used, too [8]. This has a relatively stronger performance than the template matching described above. However, this method, too, is affected by hand-crafted image features. As a recent approach for classification and machine learning problems, CNN is one of the methods suggested as another solution in order to overcome the problem occurring in the hand-crafted image features.

2.2. Deep learning

This study would solve the problem of license plate recognition by making the system learn the DCNN structure that uses a deep neural network. Yann LeCun proposed a convolutional deep learning algorithm for learning a complicated, deep neural network [9]. Using this algorithm, a good result was obtained in hand-written character recognition. Later, deep learning has brought about the rapid performance improvement in the research fields such as image classification and speech recognition [10,11]. In addition, CNN has been applied to object surface inspection to improve the inspection performance [12].

Recently, big companies such as Google, Baidu and Facebook applied the CNN to face detection, voice recognition, street view, drones and autonomous vehicle driving in various ways [13]. In addition, Google won Large Scale Visual Recognition Challenge (LSVRC-2014) that is a competition for the recognition of images and finding the correct labels of objects in images using deep learning, reaching an accuracy of 93.3% [14].

3. DEEP CONVOLUTIONAL NEURAL NETWORKS FOR LICENSE PLATE RECOGNITION

DCNN performs feature extraction and classification simultaneously unlike the existing hand-crafted image processing-based method that divides feature extraction and classification. In other words, it can take images as an input, not image feature and do end-to-end learning directly to obtain the target value of the user as an output. In addition, with the progress of learning, the proposed model directly extracts the optimal features from low-level features like oriented edge and boundary to high-level expression in which these features are combined. The extracted features are robust for local position variation, rotation, scale and brightness change. Since these features are used in classification, it has performance much stronger (robust or reliable) as compared to the recognition using the existing shallow learning based on hand-crafted feature. The performance of the method using the DCNN has already been proven in various recognition fields such as character recognition, object recognition, and pedestrian recognition [18,19].

In this paper, we have trained DCNN using directly generated license plate images. Applying the concept from MTL, the structure of DCNN consists of layers for judging the existence of license plates and layers for digit and character recognition. The learned model determines

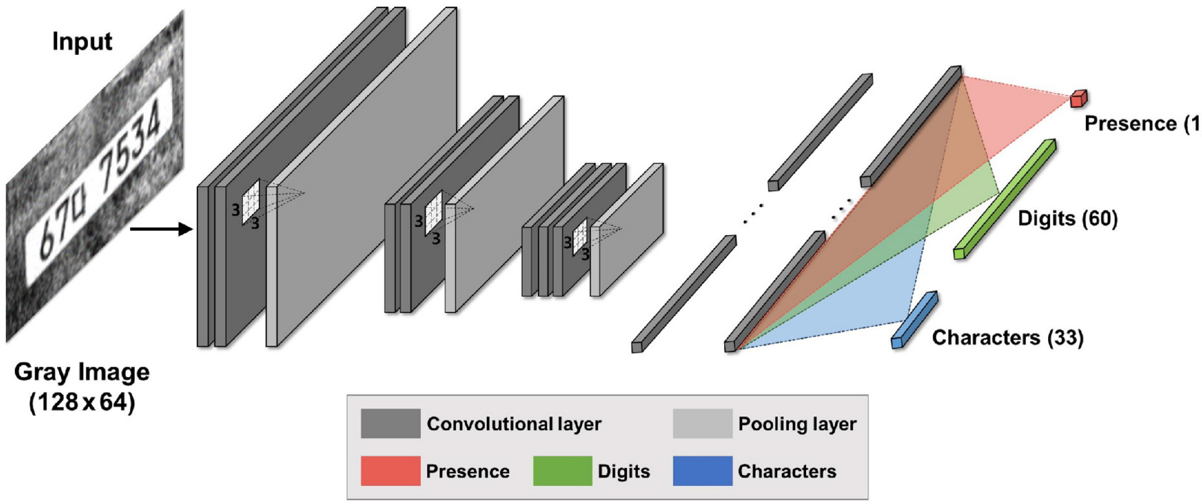


Fig. 1. DCNN structure.

whether a license plate exists for a given input image. At the same time, the numbers and characters of the license plate are recognized. If it is determined that there is no license plate in the image, all recognition processes are rejected. On the contrary, if it is determined that a license plate exists, the digits and character in the license plate are recognized. In addition, the layers for judging the existence of license plates can be used to detect the position of the license plate when a full scene image is given.

3.1. Layers

In the structure of DCNN, in general, a convolutional layer and a pooling layer are repeatedly arranged, and a fully connected layer is located at the most bottom of layers. Here, the convolutional layer is the key layer in CNN, which generates a mask or kernel that extracts the features that best differentiate the input images during the process of learning. At this time, shared weights in the feature map of the layer allow effective performance of learning by reducing the number of parameters to learn. The pooling layer provides translation invariance of input images. In addition, it has an effect on the reduction of the number of parameters to learn by down-sampling the input. A kind of the pooling layer includes max-pooling, average pooling and LP pooling layers. This study uses the max-pooling. The max-pooling is a method taking the largest value in the area of the kernel with specified size.

In this paper, we placed the convolutional layer at the bottom of the DCNN instead of the fully connected layer. This form is called a fully convolutional layer. The layer replaced with the convolutional layer performs the same function as the fully-connected layer. That is, it plays a role of classification. However, it does not depend on the size of the input image. When a full scene image larger than the image used for training is input, the whole area of the image is searched in the sliding window manner [20]. This method is more efficient than the conventional sliding window method because it shared the calculated results with each other in searching.

3.2. Networks structure

The DCNN structure is as shown in Fig. 1. First, a 128×64 gray-scale image is fed to the network. With the input image, a 3×3 sized kernel generates 48 feature maps in the first convolution layer. In the second convolution layer, too, a 3×3 sized kernel generates 48 feature maps, which is subsequently connected to a 2×2 sized max pooling layer. In the third convolution layer, a 3×3 sized kernel generates 64 feature maps. The fourth convolution layer is also the same as the third convolution layer. Then, a 2×2 max pooling layer is connected. The fifth convolution layer has 128 3×3 sized kernels. Subsequently, the sixth convolution layer and the seventh convolution layer have the same 128 kernels of size 3×3. Then, a 2×2 max pooling layer is connected. The eighth convolutional layer then generates 2048 feature maps through a 16×8 kernel. The ninth convolutional layer generates 2048 feature maps with a 1×1 size. Then, dropout [21] is applied. Dropout increases the generalization performance of the classifier by cutting weight values between layers at a proper ratio.

Thereafter, the network is split into three branches. The first branch is to decide the existence of the license plate. The second branch is to classify digits. The third branch is to classify characters. In other words, the proposed network structure shares the features extracted from the front part. At the bottom of the layer, the presence or absence of the license plate is determined and the license plate is recognized.

Table 1. Architecture specifics for proposed layers.

Layer	1	2	3	4	5	6	7
Stage	Conv	Conv	Pool	Cov	Conv	Pool	Conv
# channels	48	48	48	64	64	64	128
Filter size	3×3	3×3	-	3×3	3×3	-	3×3
Conv. Stride	1×1	1×1	-	1×1	1×1	-	1×1
Pooling size	-	-	2×2	-	-	2×2	-
Pooling stride	-	-	2×2	-	-	2×2	-
Zero-Padding size	1×1	1×1	-	1×1	1×1	-	1×1
Spatial input size	128×64	128×64	128×64	64×32	64×32	64×32	32×16

Layer	8	9	10	11	12	Output presence	Output digit	Output character
Stage	Con	Conv	Pool	Conv	Conv	Conv	Conv	Conv
# channels	128	128	128	2048	2048	1	60	33
Filter size	3×3	3×3	-	16×8	1×1	1×1	1×1	1×1
Conv. Stride	1×1	1×1	-	1×1	1×1	1×1	1×1	1×1
Pooling size	-	-	2×2	-	-	-	-	-
Pooling stride	-	-	2×2	-	-	-	-	-
Zero-Padding size	1×1	1×1	-	-	-	-	-	-
Spatial input size	32×16	32×16	32×16	16×8	1×1	1×1	1×1	1×1

3.3. Branched layer

3.3.1 Layer to judge the existence of the license plate

The first branched layer judges the existence of the license plate in the input image. At the bottom of the network, a convolution layer with 1x1 sized 2048 kernels and a layer with one node are connected. The judgment of the existence of the license plate may be considered binary classification. The output node uses sigmoid activation. Therefore, for the objective function, the binary-entropy loss is used.

$$L_1 = -t\log y, \tag{1}$$

where the input data containing the license plate fully has a label value of $t = 1$ and other data has a value of $t = 0$. y is the output value of the sigmoid function.

3.3.2 Layer of the classification of digits

The second branched layer classifies is responsible for digits recognition in the license plate. The license plate consists of six digits. The digit corresponding to each position can have 10 values from 0 to 9. Thus, the output layer has a total of 60 nodes. Like the first branched layer, a convolution layer with 1x1 sized 2048 kernels is connected to a layer with 60 nodes. Each digit outputs a 10-way softmax [22]. All output values are summed. For objective function, the following cross-entropy loss is used.

$$L_2 = -\sum_{j=0}^{5}\sum_{i=0}^{9} t_i \log y_i, \tag{2}$$

where t_i is one-hot vector form with 10-binary values. It has 1 in the relevant value and 0 in other values. j represents the index of each digit in the license plate. That is, $j = 0$ means the first number in the license plate, and $j = 5$ means the last number on the license plate.

3.3.3 Layer of the classification of characters

The third branched layer classifies 33 characters. Like above-mentioned layer, 1×1 sized 2048 kernels are connected to a layer with 33 kernels. Here, the output is provided in 33-way softmax form. For objective function, the following cross-entropy loss is used.

$$L_3 = -\sum_{i=0}^{32} t_i \log y_i, \tag{3}$$

where t_i is a vector form with 33-binary values. It has 1 in the relevant value and 0 in other values.

The final objective function used in learning is as follows:

$$L = w_1L_1 + w_2L_2 + w_3L_3. \tag{4}$$

w_1, w_2, and w_3 respectively mean the weights multiplied in each objective function. This study conducted experiments by setting all to 1.0. For activation function coming after all convolution layers, ReLU [23] was applied. The proposed architecture information applied to the experiments can be found in Table 1.

3.4. Branched layer

The following post-processing is performed on the output result using the trained network model. The size of the images used for training is 128×64, while the test images used for license plate recognition are often larger.

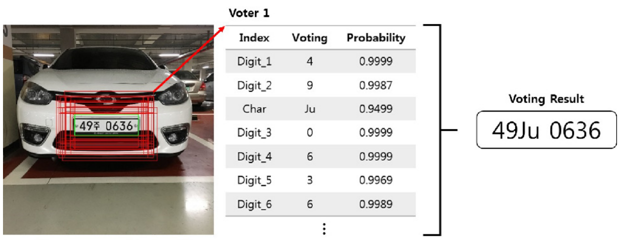


Fig. 2. Voting method for the license plate recognition.

Applying the convolution network such as the model proposed for the full scene image yields the same results as applying a sliding window to search the target object [18]. At this time, the image pyramid is used to consider various scales of the image. A set of images generated by gradually changing the size of an image is called the image pyramid. The learned model predicts the area where the license plate will exist for multiple scales of the image. A region where the probability that a license plate exists is 95% or more is outputted.

In Fig. 2, the red box is the area where the probability of the existence of the license plate is more than 95%. The voting concept is used for license plate recognition. In the red box areas where the license plate is judged to exist, only regions where all the numbers and characters have a predicted probability value of 90% or more have voting rights. In this way, the digits and the character with the most votes are selected as the final value. At this time, the threshold value 90% is an empirically determined value. In the Fig. 2, the green box means the position of the plate detected as outputting the intersection of the red box.

4. EXPERIMENTS

All experiments were implemented using Keras: with Theano backend [24]. The specifications of the PC used in the experiments are i5, Memory 16GB, GTX 980 ti, and the experiments were implemented in CUDA Toolkit and the cuDNN environment. Main factors set up in the model learning are summarized in Table 2. At this time, the learning rate is decreased to 1/10 in the 10th training epoch and decreased again to 1/10 in the 15th training epoch. This has the effect of quickly training weights early and fine tuning them later.

4.1. Dataset

4.1.1 Data generation

Collecting large sets of data is one of the most essential for successful deep learning. However, taking hundreds of thousands of pictures necessary for learning requires considerable time and labor. Even if it is collected from the internet, labeling each image is also a difficult task.

In this paper, we created the data based on the standard information of the license plate provided by the National information Center of the Republic of Korea and used it for DCNN training. Fig. 3 is the standard information for non-business-use registration plates.

Table 2. Key learning parameter factors.

Parameter factor	Value
Batch size	64
Momentum	0.90
Weight decay	0.0005
Learning rate	0.01, 0.001, 0.0001
Epoch	20

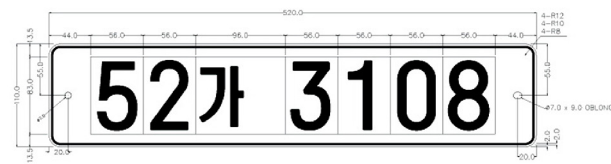


Fig. 3. Registration plate specification for non-business use (520 mm $\times$ 110 mm).

4.1.2 Generation mechanism

First, we prepared a sample image of each number and character. Sample images also use image files provided by the National information Center. The prepared sample image is created considering the position according to the license plate specification. The image generated at this time takes into consideration by the following conditions:

- In order to cope with various lighting changes, the color of the white area and number area of the license plate are randomly applied.

- In order to take into account the various angels and scales, the homography transformation is applied to the generated license plate image.

- Randomly selected background image and license plate image is composited. Where the background image is an arbitrary image downloaded from the internet.

- Finally, in order to improve generalization ability, a noise is added to the generated image. The generalization is the ability of the trained model to fit untrained data.

Some sample plate images generated based on this generation mechanism are shown in Fig. 4. The size of the generated image is 128×64 and it is a gray image. The number of generated images is $200,000$ positive images containing license plate and $200,000$ negative images. Negative images consist of images with part of the plate cut off and no plate. The data generation method is one of the contributions in this paper.

4.2. Evaluation

There is no consistent way to evaluate the structure we have proposed. Because there is also no public Korean li-

Positive Negative

Fig. 4. Some positive and negative samples of the training dataset.

cense plate dataset for comparison, we carried out a qualitative evaluation of images collected under various conditions. To verify the performance of the proposed multi-layer model, we compare the performance of the single-layer model learned with the same dataset. Additionally, in order to evaluate the generalization performance of the multi-layer model, we tested real images captured in various angles, lighting, size and pose.

4.3. Results

We trained the single layer models with the same dataset. But the single layer models without the presence layer do not decrease in loss value during training. So the performance can not be evaluated. This means that the model wihout the presence layer can't recognize the license plate. When the model is verified using images of the same size as those used in the training, the classification accuracy of the license plate is less than 10%. On the other hand, the proposed method works well. We can guess the reason by checking whether the loss value decreases or not during training. The loss value with regarding to the presence of the license plate is decreased before the loss value associated with digits and character recognition decreases. But, the single layer model without the presence layer does not work for decrease of the loss value. That is, the presence layer helps the network to recognize the license plate.

As we do not have reference data set for comparison, there is no consistent evaluation criterion for evaluating the generalization ability of the proposed method. Therefore, we use many images taken in different angles in various environments for the performance evaluation. Fig. 5 shows the images used in the test. The Images were taken at night, rainy day, sunny day, indoor and shade environments. Fig. 6 shows the license plate recognition results and the location where the license plate was detected. Experimental results show that the proposed method accurately recognizes all the images taken from the front as well as the images taken from multiple angles. Table 3 shows the results of classification error using 302 test images. The accuracy of the candidate region indicates the classification performance of all the candidate regions

Table 3. Classification performance of our model.

Type	License plate (300/302)	Candidate region (12766/13023)
Classification Error	0.006623	0.019734
Accuracy	0.993377	0.980266

where the license plate is present. After applying the voting concept in the post-process, the accuracy rose from 98% to 99%.

5. CONCLUSION

This study applies DCNN to license plate recognition. In particular, the proposed DCNN structure was designed to simultaneously train the layer for determining the existence of the license plate and the layer for digit recognition in the image. It is based on MTL. In the structure of DCNN designed in this way, the presence layer of the license plate not only improves the performance of the digit recognition but also is used to detect the position of the license plate in the image. We directly generated the images necessary for training the network. The generated images were considered various conditions that can occur in the actual environment. As a result, the experimental results using real images showed good recognition performance.

To verify the performance of the proposed multilayer model, we trained the single layer model using the same data, but the results were not good. On the other hand, there was no consistent benchmark for comparing license plate recognition performance, we evaluated qualitatively performance using a large amount of data acquired under various conditions. As a result, The proposed model accurately can recognize license plate taken at various conditions on a rainy day, sunny day, day and night time.

In this paper, plate recognition is limited to the European style non-business Korean license plate. If we additionally generate plates with different shapes and colors to learn the model, we can expect all type of plates to be recognized. Further, it is a future work to allow two or more license plate types to be recognized at the same time.

REFERENCES

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp.311-325, 2013. [click]

[2] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp.41-75, 1997. [click]

[3] C. Zhang and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," *Proc. of IEEE Winter Conf. Applications of Computer Vision (WACV)*, pp. 1036-1041, 2014. [click]

(a) (b)

Fig. 5. Test images taken in various conditions(at night, rainy day, sunny day, indoor and shade for (a). Test images taken in multiple angles for the same vehicle in (b).

Fig. 6. Examples of license plate recognition and detection on the test images.

[4] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: a survey," *IEEE Trans. on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377-391, 2008. [click]

[5] D. J. Kang, "Dynamic programming-based method for extraction of license plate numbers of speeding vehicles on the highway," *International Journal of Automotive Technology*, vol. 10, no. 2 pp. 205-210, 2009. [click]

[6] S. Z. Wang and H. J. Lee, "Detection and recognition of license plate characters with different appearances," *IEEE Proc. of Intelligent Transportation Systems*, vol. 2, pp. 979-984, 2003. [click]

[7] M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, "Saudi Arabian license plate recognition system," *IEEE Proc. of International Conf. on Geometric Modeling and Graphics*, pp. 36-41 2003. [click]

[8] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-

based approach for license plate recognition," *IEEE Proc. of Neural Networks for Signal Processing X, 2000, Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol. 2, pp. 614-623, 2000.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, pp. 396-404, 1990.

[10] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *Signal Processing Magazine*, vol. 29, no.6, pp 82-97, 2012. [click]

[12] J. K. Park, B. K. Park, J. H. Park, and D. J. Kang, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303-310, 2016. [click]

[13] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *arXiv:1312.6082*, 2013.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and A. C. Berg, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015. [click]

[15] D. Zang, Z. Chai, J. Zhang, D. Zhang, and J. Cheng, "Vehicle license plate recognition using visual attention model and deep learning," *Journal of Electronic Imaging*, vol. 24, no. 3, 2015.

[16] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMs," *arXiv:1601.05610*, 2016.

[17] Z. Zhao, S. Yang, and X. Ma, "Chinese license plate recognition using a convolutional neural network," *IEEE Workshop on Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, vol. 1, pp. 27-30, 2008.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Proc. of IEEE Conf. on Computer vision–ECCV*, pp. 818-833, 2014. [click]

[19] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," *IEEE Proc. of the Conf. on Computer Vision and Pattern Recognition*, pp. 2147-2154, 2014.

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv:1312.6229*, 2013.

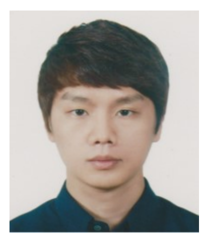
[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[22] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," *Proc. of the 24th International Conf. Machine Learning*, ACM, pp. 791-798, 2007. [click]

[23] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[24] F. Chollet. Keras, GitHub repository, https://github.com/fchollet/keras.

Hong-Hyun Kim received his B.S. degree in School of Mechanical Engineering from Pusan National University, Pusan, Korea in 2012. He is currently Unified Master and Doctor's course in the same graduate school. His research interests include deep learning, machine learning, and pattern recognition



Je-Kang Park received his B.S. degree in the school of mechanical engineering from Pusan National University, Pusan, Korea. He is currently a Ph.D. student in the same graduate school. His current research interests are computer vision, machine learning, deep neural network, and object detection.

Joo-Hee Oh received her B.S and M.S degrees in School of Mechanical Engineering from Pusan National University, Pusan, Korea, in 2014 and 2016, respectively. She is currently working for LG Electronics. Her research interests include deep learning and machine learning.

Dong-Joong Kang received his B.S. degree in Precision Engineering from Pusan National University, Pusan, Korea, in 1988 and his M.S. and Ph.D. degrees in Mechanical, and Automation & Design Engineering from KAIST, Korea, in 1990 and 1999, respectively. In 1997 to 2000, he was a research engineer at SAIT (Samsung Advanced Institute of Technology). Now he is a professor at the School of Mechanical Engineering in Pusan National University. He is also an associate editor of the International Journal of Control, Automation, and Systems since 2007. His current research interests are machine vision, pattern recognition, and visual inspection in factory.