

Number Plate Detection with a Multi-Convolutional Neural Network Approach with Optical Character Recognition for Mobile Devices

Christian Gerber* and Mokdong Chung**

Abstract

In this paper, we propose a method to achieve improved number plate detection for mobile devices by applying a multiple convolutional neural network (CNN) approach. First, we processed supervised CNN-verified car detection and then we applied the detected car regions to the next supervised CNN-verifier for number plate detection. In the final step, the detected number plate regions were verified through optical character recognition by another CNN-verifier. Since mobile devices are limited in computation power, we are proposing a fast method to recognize number plates. We expect for it to be used in the field of intelligent transportation systems.

Keywords

Convolutional Neural Network, Number Plate Detection, OCR

1. Introduction

Acquiring real-time number plate localization with a high detection rate in a natural traffic environment is still a widely researched area in computer science. Recent research results [1-3] show that convolutional neural networks (CNNs) provide a high detection and a low false positive rate among classifying images. Chen et al. [1] proposed a CNN-based verifier for number plate detection by processing a small, number plate conformed sliding sub-window on the entire input image. Instead of detecting license plates directly, we reduced the neural network computations by applying a bigger sliding window to localize cars first. The detected car regions served as the input for the second supervised CNN to detect number plates. Number plates were then verified by applying optical character recognition (OCR) on the detected plate regions. Our approach as compared to single CNN-based number plate detectors, provides a high detection rate by reducing the overall neuron calculations. Therefore, number plates can be detected on mobile devices, based on neural network classifiers, within a fast-changing real world environment. Intelligent transportation systems (ITSs) are becoming more important as the amount of traffic is also growing. Number plate detection is important for modelling and tracking traffic flow. In Section 2, we compare our proposed method to similar

* 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.

Manuscript received April 13, 2015; accepted August 5, 2015.

Corresponding Author: Mokdong Chung (mdchung@pknu.ac.kr)

* Dept. of Computer Engineering, Pukyong National University, Busan, Korea (christian.gerber@sunrise.ch, mdchung@pknu.ac.kr)

approaches. Section 3 shows how we applied our method and the results are provided in Section 4. The discussion is presented in Section 5 and the conclusion is given in Section 6.

2. Related Work

Chen et al. [1] proposed a number plate detection using a single-stage, single-scale CNN. This approach looks for text features by applying a square-shaped sliding window over the full input image. The sliding window looks for two full characters, which requires minimum resolution and a small sliding window step size (vertical, horizontally). Approaches using a sliding window for object detection are slow due to the evaluation of overlapping image regions. Our method uses a single-scale CNN for both car and plate detection. The searching area for license plates is reduced to a car detected image region by applying vehicle detection first. Vehicle detection uses a large sliding window step size and a large input image for convolution, which reduces the overall classification amount. Li et al. [2] proposed a multi-scale CNN architecture approach. The classifier can be fed with the features extracted in multiple stages. This gives the advantage of feeding the classifier with different scales of receptive fields. The entire input image has to be divided into small sliding window steps to detect the license plate. With our approach, we reduced the input image classification by applying a CNN for car detection first, as shown in Table 1.

Table 1. Comparison of similar CNN number plate detection approaches

Li et al. [2]		Chen et al. [1]	multi CNN
CNN	Multi-scale, multi-stage	Single-stage, single-scale	Multi-stage, single-scale
Advantage	Reduced classification steps	Fast learning	Reduced classification steps, fast learning
Disadvantage	Slow learning	Requires minimum resolution, small sliding window steps	Plate detection depends on car detection results

CNN=convolutional neural network.

3. Methods

3.1 Multi-CNN Architecture

The license plate is detected by sequentially applying multiple CNN-verifiers. We first performed a supervised CNN for car detection (CNN1) and second, a supervised CNN for plate detection (CNN2). Plate validation was performed with OCR by detecting digits with a third CNN-verifier (CNN3). A sliding window was used to generate input images for all classifiers. Before the classification of the generated input image, the algorithm performed image preprocessing for car-, plate- and digit-classifiers. Car image preprocessing was done by applying grayscale, Gaussian blur, Sobel, and subsampling. Gaussian blur followed by a Laplacian convolution and a subsampling step was used for plate detection. The plate and digit image preprocessing was done by applying Gaussian blur, Laplacian, and subsampling. After the car classification step, CNN2 took the detected car regions as input in order

to detect plates. The detected plates were saved and we applied CNN3 for digit recognition. If there were detected digits, the classified plate image region of CNN2 was saved as plate and the next car region was applied for plate detection. If there were no more existing car regions, the algorithm terminated, as shown in Fig. 1.

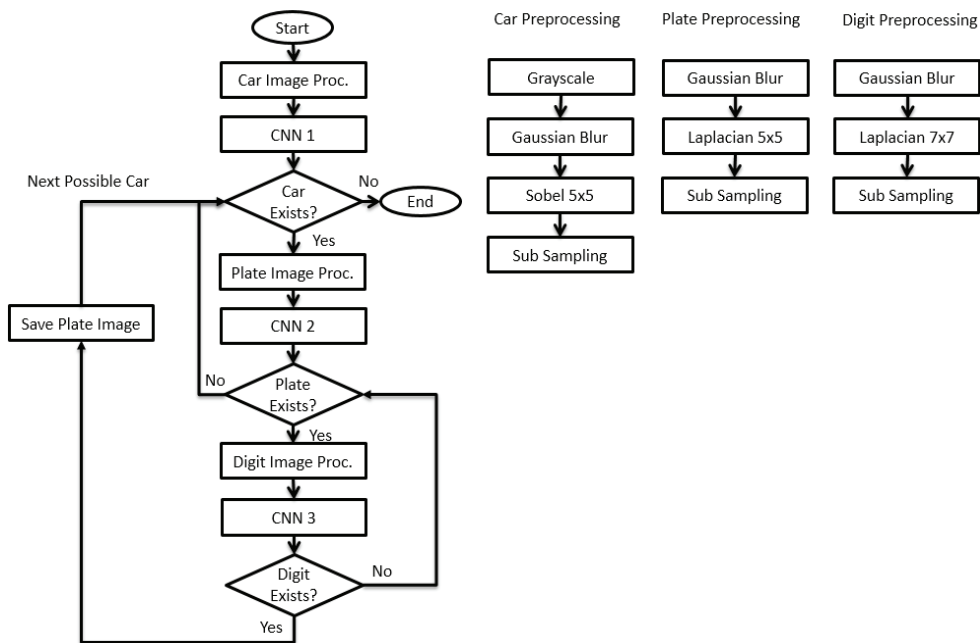


Fig. 1. Multi-convolutional neural network (CNN) number plate detection flow diagram.

3.2 Convolutional Neural Network

A CNN consists of one or multiple stages of image processing and a neural network as a classifier. One stage consists of a convolution step followed by a subsampling step. The convolution step usually convolutes the input data with multiple, different filters to extract features. The subsampling layer summarizes detected features into a feature map and reduces the dimension of the convoluted images from the previous step. These layers are arranged in a feed-forward cascade structure.

The convolutional layers create a feature map of each given image convolution algorithm. A convolution covers a convolution kernel of an uneven number of pixels in the horizontal and vertical directions. In our approach, we used a 5x5 pixels Sobel algorithm for feature and corner extraction. Feature maps result after the convolution of the input image with the convolution algorithm.

Reducing the resolution of the feature maps by applying subsampling make the features less sensitive to translation. Averaging a pixel area that is 2x2-neighboring-pixels in size from the previous convolution layer usually performs this reduction. Nevertheless, reducing the resolution by subsampling also reduces the amount of high-level features. Generally, a CNN with multiple stages considers low-level features. Li et al. [2] introduced a multi-stage, multi-scale CNN to classify low-features and high-features containing convolutions. More convolution and subsampling layers reduce the detail of the origin image. After the feature extraction, the resulting image is applied to a feed-forward multilayer perceptron [4].

3.3 Car Localization

The video input stream was acquired using a session capture that was 480×640 in pixel resolution for color images. The images of the video stream were converted to gray and convoluted by a Sobel 5×5 kernel filter to extract edges and corners. The 320×240-pixel sliding window began at the upper left corner and proceeded at 16 pixel steps both vertically and horizontally with four different resolutions. We acquired the input image for CNN1 by downscaling the convoluted sliding window output to a resolution of 20×15 pixels. Therefore, CNN1 is a single-stage CNN with only one convolution layer for feature extraction. The classification for CNN1 is a fully connected multilayer perceptron (MLP), with one hidden layer and 10 hidden nodes. It was trained by applying back propagation [5] with a given training set of 121 front car images and 2,578 non-car images. We used sliding windows of different resolutions to detect cars of variable sizes.

3.4 Plate Localization

Once the car image regions were detected, we applied CNN2 for number plate detection onto the detected car subimage from CNN1. A sliding window with a vertical and horizontal step of 5 pixels was used as for CNN2 with two different resolutions. For the single CNN plate classification we used sliding windows that were 100×40 pixels. CNN2 is a single-stage CNN with Gaussian blur followed by a Laplacian 5×5 filter convolution and a subsampling step, as shown in Fig. 1. The classification layer for CNN2 was a supervised MLP with 10 hidden nodes, where the training was done by back propagation with a given training set of 20 narrow plate images and 1,699 non-plate images.

3.5 Plate Verification with OCR

By applying OCR for digit recognition with a multi-stage, single-scaled CNN and 10 output neurons on the plate-detected image region, we verified the car's number plate, as shown in Fig. 2.

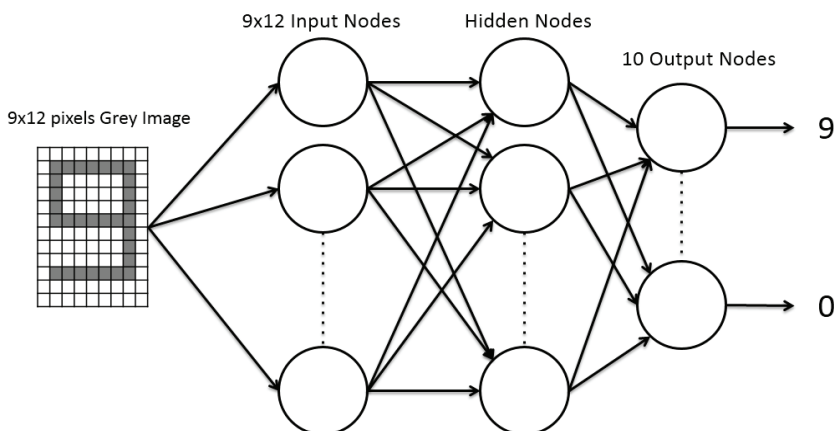


Fig. 2. Architecture for a digit classifier based on convolutional neural network.

There is an output neuron for every digit. With our method, existing digits on a number plate region signaled the existence of a Korean number plate, as shown in Fig. 7. Therefore, once the plate image

regions were detected, we applied CNN3 for digit detection onto the plate-detected subimage from CNN2. A sliding window with a vertical and horizontal step of 2 pixels was used as for CNN3 with 18×24 pixels. CNN3 is a single-stage CNN with a Gaussian blur followed by a Laplacian 7×7 filter convolution and a subsampling step. The classification layer for CNN3 was a supervised MLP with 30 hidden nodes, where the training was done by back propagation with a given training set of 10 images per digits and 712 non-digit images, as shown in Fig. 3. If there were one or more detected digits within the number plate region, the algorithm recognized this region as a number plate.

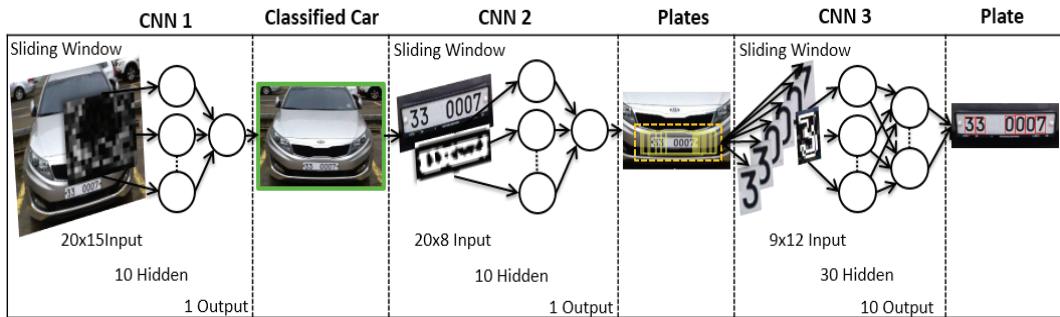


Fig. 3. Number plate classification based on multiple convolutional neural networks (CNNs).

4. Results

We detected the car number plate with only one CNN as a classifier, which is similar to the method by Chen et al. [1], resulted in computing 8,500 multiplied by 11 neurons per image, with a total calculation amount of 93,500 neurons. With the multi-CNN approach, 250 multiplied with 11 neurons resulted in a neuron calculation amount of 2,750 neurons per car, 1,760 multiplied with 11 neurons resulted in a neuron calculation amount of 19,360 neurons per plate and 328 multiplied with 40 neurons, with a total of 13,120 calculations, for digit verification, in the case of only one detected car and plate region of size 320×240, 100×40 and 18×24 pixels for digits respectively. Therefore, 35,230 neurons were calculated per image with only one car and one plate region (Table 2).

Table 2. Number of neuron calculations of single CNN and multi CNN

	Single CNN	Multi CNN
CNN1 24×18 input neuron	-	2,750
CNN2 32×12 input neuron	93,500	19,360
CNN3 32×12 input neuron	-	13,120
Total neuron evaluation	93,500	35,230

CNN=convolutional neural network.

Both the car and the plate classifier consisted of one hidden layer with 10 hidden nodes, whereas, the digit classifier used 30 hidden nodes for classification. By increasing the amount of hidden neurons we also increased the overall calculation time, as shown in Fig. 4.

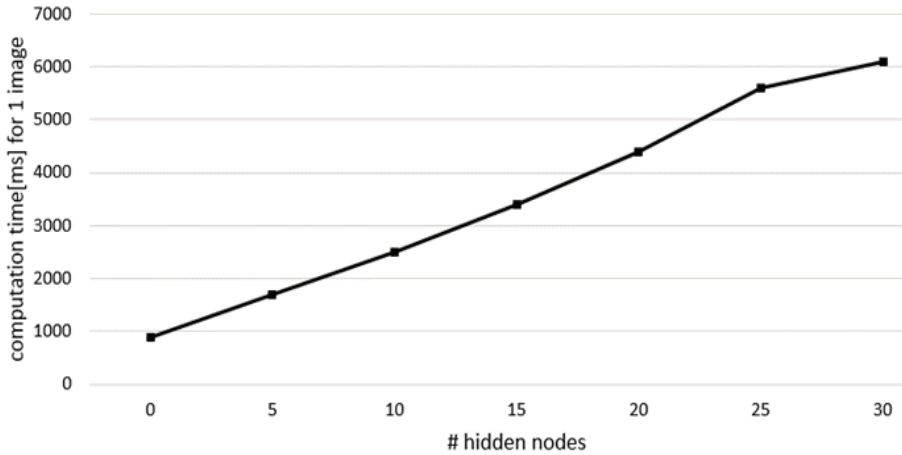


Fig. 4. Plate detection computation performance in relation with the number of hidden nodes (iPhone 4: 1 GHz Cortex A-8 CPU, 512 MB RAM [6], 480×640 pixels input).

We compared the computation performance of the multi-CNN approach with our single-CNN approach, which is similar to the approach used by Chen et al. [1] in terms of architecture and classification of real traffic environment front car images, as shown in Fig. 5.

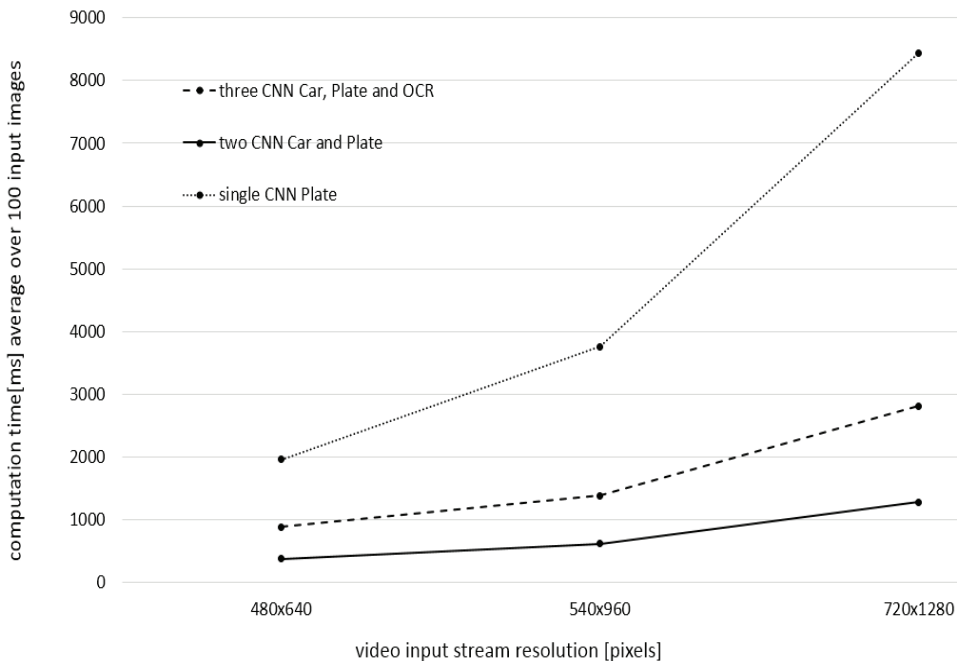


Fig. 5. Plate detection performance of single- and multi-CNN approach in relation to the input size (iPhone 4: 1 GHz Cortex A-8 CPU, 512 MB RAM [6]), with only one detected car region for the multi CNN approach. Resolution for car is 320×240, plate is 100×40 and for digits 18×24 are the same for all video input resolutions. The computation time is an average of over 100 input images for each evaluation. CNN=convolutional neural network.

Our single CNN plate detector detected more than 90% of the number plates in our car training data with narrow number plates. The detection rate of the combined classifier was lower than the single CNN plate classifier because the car detector failed to detect all of the cars, as well as failing to conduct digit detection within a positive number plate region. By applying OCR to our number plate classifier, we reduced the amount of false positives compared to our former approach [7], as shown in Table 3.

Table 3. Detection rate of our method applied on different still image databases

Test database	Detection rate (%)	False positives
Number plate		
Car training data (88 images)	91.25	84
Number plate with OCR		
Car training data (88 images)	88.63	27
Caltech [8] 99 (125 rear images)	96	6
Caltech [8] 01 (526 rear images)	88	26

However, real-time experiments in a natural traffic environment showed that our car detector detected 98% of the front and back of a car appearing on a 640×480-pixel video input stream, once the car was within range of our sliding windows, as shown in Fig. 6.

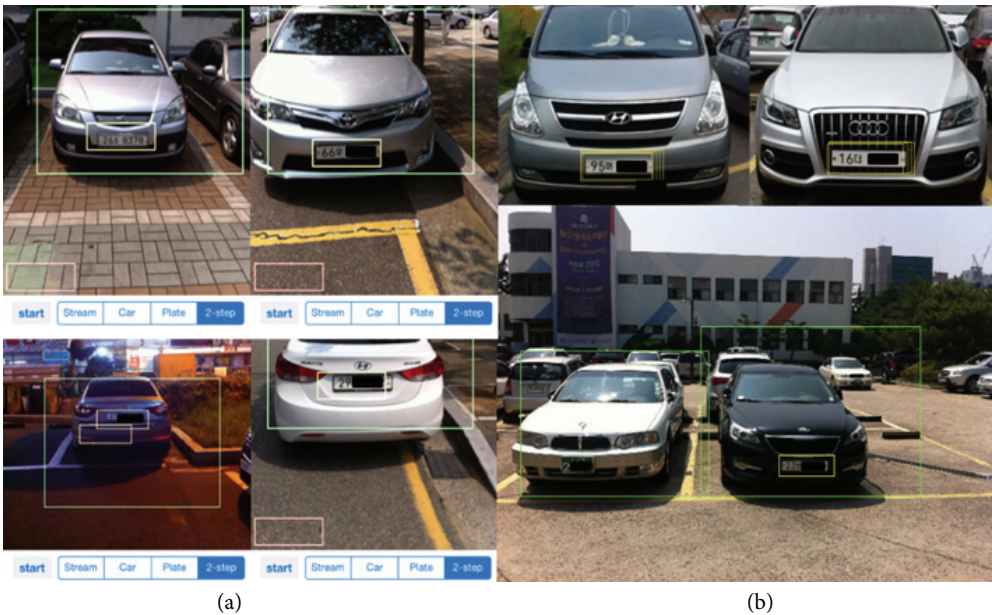


Fig. 6. Detection results from (a) our mobile application (results with an iPhone 4: 1 GHz Cortex A-8 CPU, 512 MB RAM [6], 640×480 pixels input) and (b) our car training data (results from a MacBook Air).

By applying OCR we reduced the amount of false positive detections (Fig. 7). If there were digits that were detected within the plate region, the region was classified as a number plate. If the digit classifier failed to detect positive digit regions, the positive-classified plate region was considered to be a false positive and not part of the detection result.



Fig. 7. Digit detection on Korean number plates.

5. Discussion

With our proposed method, we trained the car detector with car front images only. Extending this method with an unsupervised CNN would allow us to detect other kinds of vehicles, as well as vehicles and plates that are rotated or at an angle. The Laplacian and Sobel filters made our detector unaffected by variations in brightness, uneven illumination, and low contrast. Reducing the searching area for number plates within the detected car region further would reduce the overall classification steps. Cars that appear partially occluded within the video input image were not detected and therefore, the algorithm did not search for a number plate. Input images of cars covered with shadows had a higher false positive rate and a lower detection rate. Wider, more square-shaped plates and vehicles with a certain angle, rotation, or with different shapes, such as sports utility vehicles (SUVs), mini-vans, trucks, etc., had a low detection rate due to missing training data.

6. Conclusion

We proposed a neural network-based method for real-time number plate localization on mobile devices, including OCR. The real-time classification of image input data that is classified by neural networks and sliding windows is costly, and, therefore, not suited for mobile devices. By reducing the computation amount we showed that our proposed method could perform real-time license plate detection on a mobile device at a high detection rate. In the future, we intend to improve our classifiers in terms of rotation, angle, and shape.

Acknowledgement

This work was supported by a 2015 Research Grant from Pukyong National University.

References

- [1] Y. N. Chen, C. C. Han, C. T. Wang, B. S. Jeng, and K. C. Fan, "The application of a convolution neural network on face and license plate detection," in *Proceedings of 18th International Conference on Pattern Recognition (ICPR 2006)*, Hong Kong, 2006, pp. 552-555.

- [2] J. Li, C. Niu, and M. Fan, "Multi-scale convolutional neural networks for natural scene license plate detection," in *Advances in Neural Networks-ISNN 2012*. Heidelberg: Springer, 2012, pp. 110-119.
- [3] 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," Apr. 2014; <http://arxiv.org/pdf/1312.6082v4.pdf>.
- [4] L. M. Belue and K. W. Bauer, "Determining input features for multilayer perceptrons," *Neurocomputing*, vol. 7, no. 2, pp. 111-121, 1995.
- [5] 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.
- [6] Apple Inc., Part of iOS 6 Developer Library.
- [7] C. Gerber and M. Chung, "Two-step convolutional neural network approach for improved number plate localization on iOS," in *Proceedings of 2014 Korea Computer Congress (KCC2014)*, Busan, Korea, 2014, pp. 868-870.
- [8] California Institute of Technology, "Image_Datasets," 2015; http://www.vision.caltech.edu/Image_Datasets/.



Christian Gerber

He received B.S. degree in Computer Science from University of Bern in Switzerland in 2007. And he was a Master Student in POSTECH University, Korea from 2009 to 2011. Also he received M.S. degree in Computer Engineering from Pukyong National University, Korea in 2015. His research interests are in the areas of Machine Learning and Computer Graphics.



Mokdong Chung <http://orcid.org/0000-0002-3119-0287>

He received B.S. degree in Computer Engineering from Kyungpook National University, Korea in 1981. And he received M.S. and Ph.D. degrees in Computer Engineering from Seoul National University, Korea in 1983 and 1990, respectively. He was a professor at Pusan University of Foreign Studies from 1985 to 1996. And he has been a professor at Pukyong National University since 1996. His research interests are in the areas of Computer Security for Application, Context-Aware Computing, and Bigdata based Computer Forensics.