

Distance Sensitive AdaBoost using Distance Weight Function

Wonju Lee*, Minkyu Cheon*, Chang-Ho Hyun**+, and Mignon Park*+

* School of Electrical and Electronic Engineering, Yonsei University

** Division of Electrical Electronic and Control Engineering, Kongju National University

Abstract

This paper proposes a new method to improve performance of AdaBoost by using a distance weight function to increase the accuracy of its machine learning processes. The proposed distance weight algorithm improves classification in areas where the original binary classifier is weak. This paper derives the new algorithm's optimal solution, and it demonstrates how classifier accuracy can be improved using the proposed Distance Sensitive AdaBoost in a simulation experiment of pedestrian detection.

Key Words: Machine Learning, AdaBoost, Distance, Weights.

1. Introduction

In machine learning, classifiers are needed that lead to accurate and fast decisions in a wide variety of environments. Commonly used classifiers for real-time detection tests are AdaBoost and support vector machines, and recent studies have sought to improve the accuracy and detection speed of these classifiers [1], [2]. This paper proposes a method to increase the accuracy of the AdaBoost learning algorithm by adjusting weak classifiers to use distance weights based on parameters of closeness to the detection object. This modification improves overall performance, as AdaBoost combines collections of weak classifiers to form stronger classifiers.

AdaBoost's detection method is to find features of a detection object in the training data set. This can be difficult when the positive data points of the training data are close to the wide image, which includes both the detection object and a background. This problem may involve occlusions with uncertain contours, and it can cause undesirable classification results [3]. Thus, the proposed, advanced AdaBoost, which adds distance weight parameters related to the specific detection object, will improve detection accuracy.

The proposed method works, not by extracting a new feature, but rather by inserting distance weight parameters related to the detection object into AdaBoost, improving the weak classifiers. Additionally, a new region close to threshold is determined based on the original algorithm's classification errors, which improves accuracy where the original binary classifier is weak. Thus, our proposed method complements the binary classifier.

We introduce Boundary classifier briefly in Section 2, show the procedures of our proposed system in Section 3, and describe the results of our simulation experiment in Section 4. Finally, some concluding remarks are made in Section 5.

2. Background

2.1. AdaBoost

According to AdaBoost algorithm, we can find the strong classifier $f(x)$ combined weak classifiers h with training data x at the step t . So, training error ε will be a summation of incorrectly classified weights as follows [3-5].

$$\varepsilon_t = \sum_{i=1}^m w_i |h_t(x) - y_i| \quad (1)$$

where i is the index of training data, t is a training step, m is the number of training data, $y_i \in \{+1, -1\}$ is the true class of a classification. The normalized weight w can be adjusted by the classification result of samples at a step.

2.2. Binary Classifier

According to Machine Learning theory, the binary classifier is the decision function with two classes as shown in Fig. 1 (a). Thus, the binary classifier h will be presented [6-10].

$$h_t(x) = \begin{cases} +1 & \text{if } x > \theta \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

where θ is a threshold which is a criterion of a decision.

2.3. Boundary AdaBoost

As shown in Fig. 1 (b), Boundary classifier has zero at the region close to a threshold. Then,

$$b_t(x) = \begin{cases} 1 & \text{if } x - \theta > s \\ 0 & \text{if } |x - \theta| < s \\ -1 & \text{otherwise} \end{cases} \quad (3)$$

Manuscript received Nov. 1, 2011; revised Jun. 7, 2012; accepted Jun. 11, 2012

*Corresponding Author: Chang-Ho Hyun(hyunch@kongju.ac.kr)

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0024914)

© The Korean Institute of Intelligent Systems. All rights reserved.

where, s is the boundary of this classifier which has restricted the range $[-1, +1]$. In this way, Boundary classifier can set a new zero point, thus adjusting the feature weight of this region. Note that the region close to the threshold has a low rate of confidence, meaning that features in the boundary region may be classified incorrectly, since they are close to this threshold [11-16].

According to AdaBoost algorithm, Boosting method can be presented as the following strong classifier which combines a lot of weak classifiers.

$$f(x) = \sum_{t=1}^T \alpha_t b_t(x) \quad (4)$$

where, α_t is the confidence rate of the t -th weak classifier. Generally, we can obtain the lower confidence rate at the high step. In other words, we will be hard to correctly classify the sample since the unclear samples remain at the high step. And, we can derive the cost C of AdaBoost using the lost function L as follows.

$$C = \sum_{i=1}^m L(y_i, f(x)) \quad (5)$$

where, i is regarded as an index of training data. For the simple calculation, we will apply a greedy approach, a forward additive model as follows.

$$C = \sum_{i=1}^m L(y_i, f_{t-1}(x_i) + \alpha b(x_i)) \quad (6)$$

In order to find a minimum error, we should obtain the minimum values of both a confidence rate α and a weak classifier.

$$(\alpha, b) = \arg \min_{\alpha, b} \sum_{i=1}^m D_t(i) e^{-\alpha y_i b(x_i)} \quad (7)$$

where

$$D_t(i) = e^{-y_i f_{t-1}(x_i)} \quad (8)$$

Then, we will use an exponential function as a loss function. Boundary AdaBoost which applies Boundary classifier can be presented as follows.

$$y_i b_t \in \{-1, 0, +1\} \quad (9)$$

Then, Boundary AdaBoost is

$$\begin{aligned} \sum_{i=1}^m D_t(i) e^{-\alpha y b} &= \sum_{i: yb > 0} D_t(i) e^{-\alpha} + \sum_{i: yb < 0} D_t(i) e^{\alpha} + \sum_{i: yb = 0} D_t(i) \\ &= \sum_{i=1}^m D_t(i) \left\{ e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t + (1 - e^{-\alpha}) \delta_t \right\} \end{aligned} \quad (10)$$

where,

$$\varepsilon_t = \frac{\sum_{i: yb < 0} D_t(i)}{\sum_{i=1}^m D_t(i)}, \quad \delta_t = \frac{\sum_{i: yb = 0} D_t(i)}{\sum_{i=1}^m D_t(i)}$$

Then, we can obtain the following optimal solution using the maximum likelihood estimation. Consequently, we will obtain the optimal α as follows.

$$\begin{aligned} \alpha_t &= \arg \min_{\alpha} \left\{ e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t + (1 - e^{-\alpha}) \delta_t \right\} \\ &= \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t - \delta_t}{\varepsilon_t} \right) \end{aligned} \quad (11)$$

The boundary-weak classifier b will be chosen by the minimum error of summation which is calculated by both the false positive and the false negative. Also, Boundary classifier can on the region of $\theta \pm s$ to set the error. As expected, we can set $b = 0$ to ensure the region has a low confidence rate.

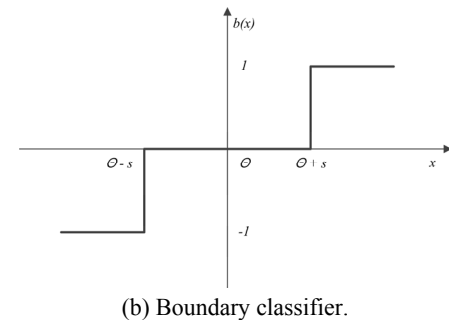
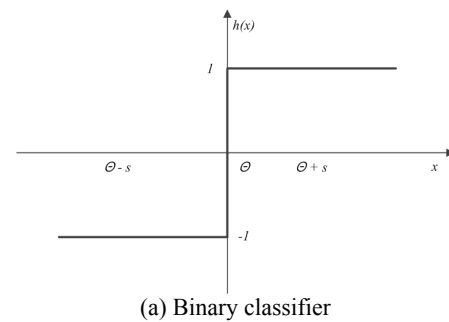


Fig. 1. Differences between two classifiers.

2.4. Weak points of Boundary AdaBoost

The drawback of Boundary AdaBoost is that it may not determine a size of the region close to the threshold. Therefore, this method is hampered by its necessary reliance on the experience of an expert to determine this size.

In this paper, we propose a new approach to determining the size of the region using distance weights. These weights are determined by distance a feature from the threshold. With this method, we increase the weights of training data that are far from the threshold, where confidence rates are higher, while we decrease the weights of training data that are close to the threshold, where confidence rates are lower. In other words,

our proposed AdaBoost algorithm selects a strong classifier which scatters the training data far from the threshold.

2.5. Histogram of Oriented Gradients

The aim of the Histogram of Oriented Gradients (HOG) is to describe an image by a set of local oriented gradients histogram. These histograms represent occurrences of specific gradient orientation in a local part of images. The HOG can be calculated by three steps, which are gradient computation, orientation binning, and histogram generation [6], [7].

In gradient computation step, the gradient of an image is obtained by filtering it with two one-dimensional filters, which are $(-1 \ 0 \ 1)$ for horizontal direction and $(-1 \ 0 \ 1)^T$ for vertical direction. The gradient could be signed or unsigned, in our case, we use signed gradient whose values are from $-\pi$ to π .

The next step is orientation binning, which is to compute the histogram of orientation. For orientation binning, the image should be divided into predefined size of blocks. In addition, we should define the range of the bins for histogram generation. For example, if the signed gradient is divided into 6 bins with same range, then the range of each bin is 60 degree ($\pi/3$ radian). In histogram generation step, we impose values to the each histogram of block. According to the orientation of the gradient, the magnitude of the gradient is accumulated to the bin of the histogram. In our proposed algorithm, the size of data is 64×64 pixels, each data divided into four blocks which are 32×32 pixels, and range of each bin is 45 degree. Therefore, each data has four histograms and four eight-dimensional feature vectors.

3. Distance Sensitive AdaBoost

3.1. Definition

Distance Sensitive AdaBoost (DSA) can choose an optimal classifier for scattering training data far from the threshold. We define distance as the distance features from the threshold. In this section, we demonstrate how we alter weak classifiers to a proper form to apply our distance. If we select a linear relationship between weights and distances then

$$h_j(x) = x - \theta_j \quad (12)$$

where x is the training data, i is an index number of the training data, θ is a threshold of the weak classifier and j is a feature index number. This distance weight function has the distance plotted along the x-axis, and weight plotted along the y-axis. Thus, if we choose $h = x - \theta$, then we can derive the following weak classifier that applies distance weights.

$$\sum_{i=1}^m D_t(i) e^{-\alpha y_i h(x_i)} = \sum_{i=1}^m D_t(i) e^{-\alpha y_i (x_i - \theta_t)} \quad (13)$$

In order to reduce the time complexity of classifiers, we can obtain new extracted features using a translation of x-axes.

$$x_s = x - \theta_j \quad (14)$$

3.2. Generalization

In this section, we derive the generalized form of DSA which selects the optimal classifier for scattering features far from threshold. Also, the distance weights function is an alternate form of classifier h which has distance from threshold on the x-axis and weight on the y-axis. Thus, if we choose $x_s = x - \theta_j$, then the origin of the feature's x-axis is moved to θ . In addition, if we use the translation feature as equation (14) and determine arbitrary distance weight function, then we derive the basic form of AdaBoost as follows.

$$\sum_{i=1}^m D_t(i) e^{h(x_i)} = \sum_{i=1}^m D_t(i) e^{g(x_{i,s})} \quad (15)$$

where $g(x_{i,s})$ is the distance weight function. We can separate into two cases, correctly classified and incorrectly classified, as follows.

$$\begin{aligned} & \sum_{i:y_g > 0} D_t(i) e^{-\alpha y_i g(x_i)} + \sum_{i:y_g < 0} D_t(i) e^{-\alpha y_i g(x_i)} \\ &= e^{-\alpha} \sum_{i:y_g > 0} D_t(i) e^{g(x_i)} + e^{\alpha} \sum_{i:y_g < 0} D_t(i) e^{g(x_i)} \\ &= e^{-\alpha} \sum_{i=1}^m D_t(i) e^{g(x_i)} + (e^{\alpha} - e^{-\alpha}) \sum_{i:y_g < 0} D_t(i) e^{g(x_i)} \end{aligned} \quad (16)$$

If ε is defined as equation (18), then we obtain the following equation.

$$e^{-\alpha} \sum_{i=1}^m D_t(i) e^{g(x_i)} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t \sum_{i=1}^m D_t(i) e^{g(x_i)} \quad (17)$$

where

$$\varepsilon_t = \frac{\sum_{i:y_g < 0} D_t(i) e^{g(x_i)}}{\sum_{i=1}^m D_t(i) e^{g(x_i)}} \quad (18)$$

If $\sum_{i=1}^m D_t(i) e^{g(x_i)} = 1$, then the weight is normalized for a next step calculation as follows.

$$e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t \quad (19)$$

Finally, we derive the optimal solution to be the same as in Original AdaBoost as follows.

$$\begin{aligned} \frac{\partial}{\partial \alpha} \ln \left\{ e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t \right\} &= \frac{-e^{-\alpha} + e^{\alpha} \varepsilon_t + e^{-\alpha} \varepsilon_t}{e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t} \\ &= \frac{e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t + -2e^{-\alpha} + 2e^{-\alpha} \varepsilon_t}{e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t} \\ &= 1 + \frac{-2e^{-\alpha} + 2e^{-\alpha} \varepsilon_t}{e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t} \\ &= -e^{-\alpha} + e^{\alpha} \varepsilon_t + e^{-\alpha} \varepsilon_t \\ &= e^{-\alpha} (-1 + e^{2\alpha} \varepsilon_t + \varepsilon_t) \\ &= \frac{1 - \varepsilon_t}{\varepsilon_t} - e^{2\alpha} \end{aligned} \quad (20)$$

As expected, it is shown that the optimal α is the same as in Original AdaBoost.

$$\begin{aligned} \alpha_t &= \arg \min \left\{ e^{-\alpha} + (e^{\alpha} - e^{-\alpha}) \varepsilon_t \right\} \\ &= \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \end{aligned} \quad (21)$$

Therefore, we can use the same optimal solution, even though we have altered the weak classifier to include the distance weight function. Thus, our DSA has exponentially decreasing error rates similar to Original AdaBoost [5].

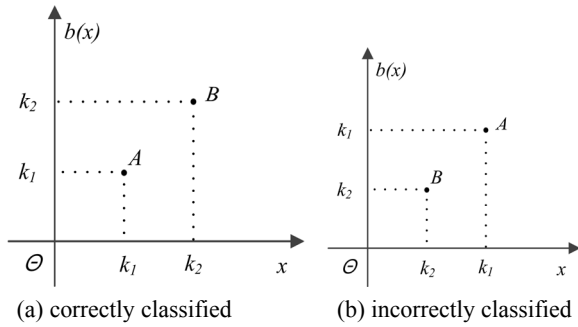


Fig. 2. Relationship between weights and classification results.

3.3. Updating Weights

If we choose $g(x_s) = h_j(x)$, we can alter the classifier h to match the classifier g as follows.

$$\begin{aligned} D_{t+1}(i) &= e^{-y_i f_t(x_i)} \\ &= e^{-y_i f_{t-1}(x_i)} e^{-y_i \alpha_t h_t(x_i)} \\ &= e^{-y_i f_{t-1}(x_i)} e^{-y_i \alpha_t g_t(x_i)} \\ &= D_t(i) e^{-y_i \alpha_t g_t(x_i)} \end{aligned} \quad (22)$$

where

$$D_t(i) = e^{-y_i f_{t-1}(x_i)}$$

As expected, we can obtain the weight D by updating the same equation as in Original AdaBoost. According to AdaBoost theory, the feature weight of correctly classified data is decreasing and the feature weight of incorrectly classified data is increasing, so the summation of weights can determine the error rates of classifiers. Thus, we can choose the strong classifier that correctly classifies as that which has high weights.

The weights of feature A and B depend on classification results, as shown in Fig. 2. The weights of adjacent features close to a threshold are increasing, which indicates that they are incorrectly classified data. So, the relationship between a distance and a weight is as follows.

$$\begin{aligned} D_{t+1}(i) &= D_t(i) e^{-y_i \alpha_t g_t(x_i)} & \text{if } yg > 0 \\ D_{t+1}(i) &= D_t(i) e^{y_i \alpha_t g_t(x_i)} & \text{if } yg < 0 \end{aligned} \quad (23)$$

Thus, it is shown that the updated weight equation is the same as that of Original AdaBoost.

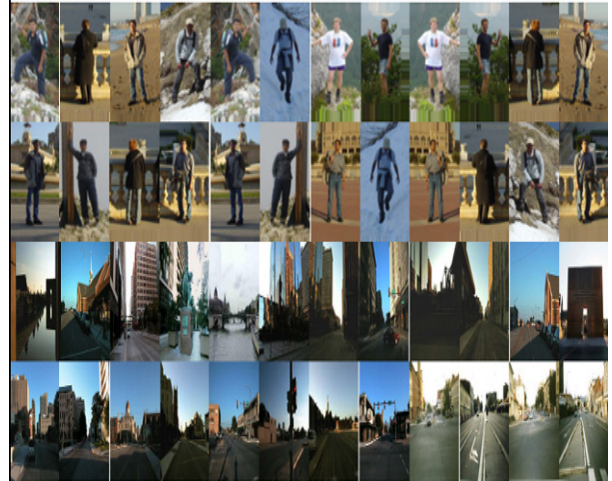


Fig. 3. Image data for training.



Fig. 4. Image data for testing.

4. Simulation Results

We simulated the performance of DSA using MATLAB. For this experiment, we chose the training images of a pedestrian and a road, as shown in Fig. 3. The number of positive pedestrian data points is 2416 images; the number of negative road data points is 2218 images. The test data for the simulation is 1132 positive images and 1453 negative images, as shown in Fig. 4. In addition, we chose the histogram of oriented gradients for the extraction algorithm which obtains sample features [17], [18].

The partition of HOG was chosen to be three on the x-axis, six on the y-axis and eight on the orientation. We examined various distance weight functions g for the step t as shown in Table 1. Note that we are able to choose from a variety of diverse functions to meet specific needs. For example, we can select a logistic function to decrease the neighborhood region of the threshold. Choosing the distance weight function can involve considerable risk: the error rate of Distance Sensitive AdaBoost for some distance weight functions is higher than that of Original AdaBoost.

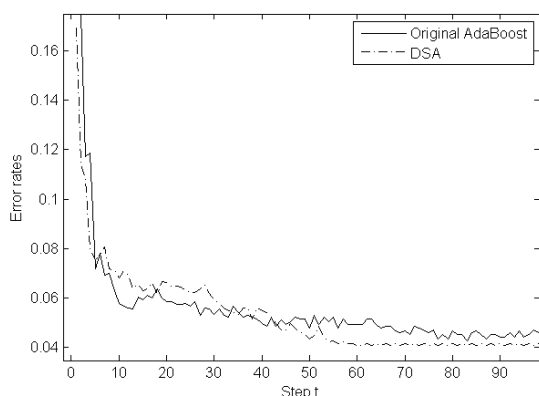


Fig. 5. Error rates of Original AdaBoost and DSA.

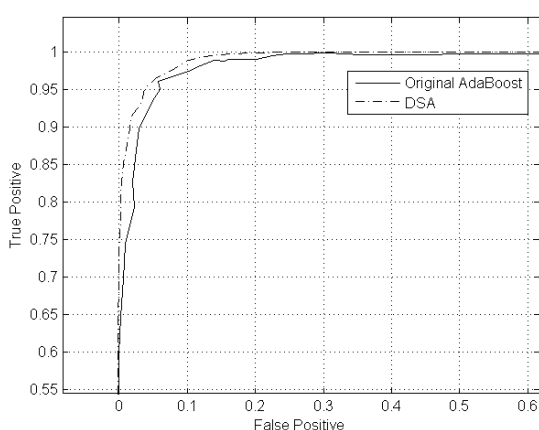


Fig. 6. ROC curve of Original AdaBoost and DSA.

However, if we examine the selecting weak classifiers, we can see that this undesired state occurs when the proposed algorithm has higher distance weights than classification weights. That is, the AdaBoost algorithm chooses some weak classifiers repetitively due to a constant weight. We have discovered that this status can occur for $g = 0.3$ and $t = 50$. When we have low distance weights, we can reduce this undesired state. For example, for $g = 0.3$ and $t = 100$, we compared error rates between Original AdaBoost and proposed algorithm. In Fig. 5 Also, Fig.6 shows ROC curve between two algorithms. These figures show that the proposed method had lower error rates than in Original AdaBoost after 40 steps. Calculating more steps of AdaBoost, did not have large additional effects.

Table 1. Error rates of DSA with distance weight function.

T \ g	0.2x	0.5x	x	2x	Original
5	8.20	7.51	7.57	7.57	8.19
10	6.97	6.88	6.56	6.18	6.80
50	5.25	4.73	5.05	4.86	5.79

We would need a more sensitive feature to decrease error rates further. As expected, we obtained the largest performance and speed at 60 steps, when using DSA. The accuracy of DSA

at 60 steps was improved 0.8% compared to Original AdaBoost, as shown in Fig. 7. Furthermore, the error rate of the original was 4.48% at 100 steps.

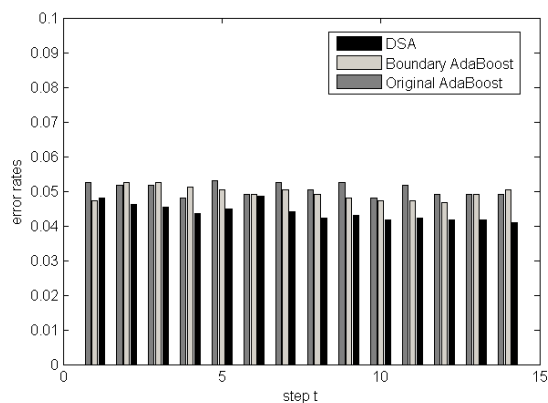


Fig. 7. Accuracy comparison of Original AdaBoost, Boundary AdaBoost, and DSA.

The training time of Original AdaBoost was 825 seconds, and the test time was 0.008 seconds, when the experiment was run on the Quad-core, 3 GHz CPU, 8G RAM system. In comparison, Distance Sensitive AdaBoost took 1071 seconds for training and 0.010 seconds for testing. Thus, test time increased 0.002 seconds using the proposed method.

5. Conclusions

In this paper, we have proposed Distance Sensitive AdaBoost which inserts new weights. These weights can solve the problem of Original AdaBoost which is hard to predict the true class in the region close to the threshold of the weak classifiers. Thus, we have improved the performance about 0.8 percentages more than that of Original AdaBoost. For the next study, we will use the linear discriminant analysis for combining features obtained from multiple sensors. Thus, we will study to use the combined features at the high steps.

References

- [1] P. Viola, M. Jones, "Rapid object detection using boosted cascade of simple features," *Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.
- [2] W. Yongqiao, W. Shouyang and K. K. Lai, "A New Fuzzy Support Vector Machine to Evaluate Credit Risk," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 16, pp. 820-831, 2005.
- [3] W. Chunpeng, D. Lijuan, M. Jun, F. Faming and W. Xuebin, "Detection of Front-View Vehicle with Occlusions Using AdaBoost," *ICIECS Conference on Information Engineering*

and *Computer Science*, pp. 1-4, 2009.

- [4] Y. Freund, Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.
- [5] Y. Freund and R. E. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771-780, 1999.
- [6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, 2005.
- [7] Z. Qiang, Y. Mei-Chen, and C. Kwang-Ting, "Fast human detection using a cascade of histograms of oriented gradients," *Computer Vision and Pattern Recognition*, vol. 2, pp. 1491-1498, 2006.
- [8] F. Suard, A. Rakotomamonjy, A. Bemsrhair, and A. Broggi, "Pedestrian detection using infrared images and histogram of oriented gradients," *Intelligent Vehicles Symposium*, pp. 206-212, 2006.
- [9] S. Shan, P. Yang, X. Chen and W. Gao, "AdaBoost Gabor Fisher Classifier for Face Recognition," in *Proc. AMFG*, pp. 279-292, 2005.
- [10] P. Viola and M. Jones, "Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade," in *Proc. NIPS*, pp. 1311-1318, 2001.
- [11] T. Onoda, G. Rätsch and K. R. Müller, "An Asymptotic Analysis of AdaBoost in the Binary Classification Case," in *Proc. ICANN*, pp. 195-200, 1998.
- [12] Räetsch, G, Onoda, T, Müller and K. R. Müller, "An Improvement of AdaBoost to Avoid Overfitting," in *Proc. ICONIP*, pp. 506-509, 1998.
- [13] L. Li and H. Lin, "Ordinal Regression by Extended Binary Classification," in *Proc. NIPS*, pp. 865-872, 2006.
- [14] T. Takenouchi and S. Eguchi, "Robustifying AdaBoost by Adding the Naive Error Rate," in *Proc. Neural Computation*, pp. 767-787, 2004.
- [15] G. Rätsch, T. Onoda and K. Müller, "Soft Margins for AdaBoost," in *Proc. Machine Learning*, pp. 287-320, 2001.
- [16] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, and A.R. Figueiras-Vidal, "Boosting by weighting boundary and erroneous samples," in *Proc. ESANN*, pp. 85-90, 2005.
- [17] C. R. Wang and J. J. Lien, "AdaBoost Learning for Human Detection Based on Histograms of Oriented Gradients," in *Proc. ACCV*, pp. 885-895, 2007.
- [18] J. Shen, C. Sun, W. Yang and Z. Sun, "Fast Human Detection Based on Enhanced Variable Size HOG Features," in *Proc. ISNN*, pp. 342-349, 2011.



Wonju Lee received a B.S. degree in Computer Engineering from Hyupsung University, Gyeonggi-do, Korea, in 2005, and M.S. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2007. He is currently a Ph.D. candidate in the Department of Electrical and Electronic Engineering at Yonsei University. His current research interests include machine learning, sensor network, and image processing.



Minkyu Cheon was born in Busan, Korea, in 1981. He received a B.S. in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2006. He is currently a Combined Master's and Doctoral Program candidate in Electrical and Electronic Engineering at Yonsei University.

His main research interests include machine learning, pattern recognition, and computer vision.



Chang-Ho Hyun received a B.S. degree in Control and Instrumentation Engineering from Kwangwoon University, Seoul, Korea, in 1999, and he completed M.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2002 and 2008. From 2008 to

2009, he was a Senior Engineer at Samsung Electronics. He then joined the faculty of the School of Electrical, Electronic and Control Engineering at Kongju National University, where he is currently an assistant professor. His current research interests include intelligent control and application, nonlinear control, robotics, and mobile robots.

E-mail: hyunch@kongju.ac.kr



Mignon Park received a B.S. degree and an M.S. degree in Electronics from Yonsei University, Seoul, Korea, in 1973 and 1977, and a Ph.D. from the University of Tokyo, Tokyo, Japan, in 1982. He was a researcher in the Institute of Biomedical Engineering, University of Tokyo, from 1972 to 1982, as

well as at the Massachusetts Institute of Technology, Cambridge, and at the University of California Berkeley, in 1982. He was a visiting researcher in the Robotics Division, Mechanical Engineering Laboratory Ministry of International Trade and Industry, Tsukuba, Japan, from 1986 to 1987. He has been a professor in the Department of Electrical and Electronic Engineering, Yonsei University, since 1982. His research interests include fuzzy control and applications, robotics, and fuzzy biomedical systems.

E-mail: mignpark@yonsei.ac.kr