Journal of Korean Institute of Intelligent Systems, Vol. 23, No. 2, April 2013, pp. 166-171
http://dx.doi.org/10.5391/JKIIS.2013.23.2.166

An Improvement of AdaBoost using Boundary Classifier

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

The method proposed in this paper can improve the performance of the Boosting algorithm in machine learning. The proposed Boundary AdaBoost algorithm can make up for the weak points of Normal binary classifier using threshold boundary concepts. The new proposed boundary can be located near the threshold of the binary classifier. The proposed algorithm improves classification in areas where Normal binary classifier is weak. Thus, the optimal boundary final classifier can decrease error rates classified with more reasonable features. Finally, this paper derives the new algorithm's optimal solution, and it demonstrates how classifier accuracy can be improved using the proposed Boundary AdaBoost in a simulation experiment of pedestrian detection using 10-fold cross validation.

Key Words: Machine Learning, AdaBoost, Boundary, Binary classifier.

1. Introduction

In machine learning, the main factors required for training the machine are samples and classifiers. Standard training data are used to make a clear decision for both detection and tracking algorithms. For example, these standard training data include CBCL of MIT, CMU, and others. Furthermore, for various experiments, we can use unusual training data which contain a variety of angles and brightness values.

The classifier is a decision function for events. Binary classifiers can be used to determine events based on a binary function [1], [2]. Classifiers must be trained in diverse environments to increase the accuracy using adjusted variables. In this paper, we improve the performance of a binary classifier, and we test a modified version of AdaBoost that applies the proposed boundary classifier. According to AdaBoost theory, the Boosting algorithm can use a strong classifier that combines modified weak classifiers [3-6]. Generally, weak binary classifiers will classify detecting objects into two classes. However, the proposed boundary classifier can classify objects into three classes, that is, into two classes and a meaningless class. The meaningless class indicates a region with a low confidence rate even though the object is correctly classified. In other words, this class is chosen as the neighborhood domain of the threshold which is a criterion of a binary classifier. Thus, the proposed boundary classifier can improve performance, since this meaningless class will be excluded from a correctly classified class. We will test the AdaBoost algorithm, which modifies the weak classifier using a boundary classifier and tune the size of a boundary. The proposed method works, not by extracting a new feature, but rather by inserting weight parameters related to the detection object into AdaBoost, improving the weak classifiers. Additionally, a new domain of threshold is determined based on Normal algorithm's classification errors, which improves accuracy where Normal binary classifier is weak. Thus, our proposed method complements the binary classifier.

We introduce Normal Adaboost briefly in Section 2, show the procedures of our proposed Boundary AdaBoost in Section 3, and the result of experiment using pedestrian training data in Section 4. Finally, some concluding remarks are made in Section 5.

접수일자: 2013년 4월 2일
심사(수정)일자: 2013년 4월 4일
게재확정일자 : 2013년 4월 5일
†Corresponding author
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)

2. Background

2.1 Optimal Solution of Normal AdaBoost

Normal AdaBoost algorithm is the solution of the optimization problems using the additive linear model [7], [8].

$$(\alpha_t, h) = \underset{\alpha,h}{\arg\min} \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h(x_i)} \tag{1}$$

where x is training samples, h is the classifier, α is the confidence rate of the classifier, D is weights of the training data and t is a step of the calculation. We can separate into two cases, correctly classified and incorrectly classified, as follows.

$$\begin{aligned} &\sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h(x_i)} \\ &= \sum_{i:yh>0} D_t(i) e^{-\alpha_t h(x_i)} + \sum_{i:yh<0} D_t(i) e^{\alpha_t h(x_i)} \\ &= e^{-\alpha_t} \sum_{i=1}^{m} D_t(i) e^{h(x_i)} + \left(e^{\alpha_t} - e^{-\alpha_t}\right) \sum_{i:yh<0} D_t(i) e^{h(x_i)} \end{aligned} \tag{2}$$

Let $\varepsilon_t = \frac{\sum_{i:yh<0} D_t(i) e^{h(x_i)}}{\sum_{i=1}^{m} D_t(i) e^{h(x_i)}}$, then we can obtain the following equation.

$$\begin{aligned}&\sum_{i=1}^{m} D_t(i)e^{-\alpha_t y_i h(x_i)}\\ &= e^{-\alpha_t}\sum_{i=1}^{m} D_t(i)e^{h(x_i)} + \left(e^{\alpha_t} - e^{-\alpha_t}\right)\varepsilon_t \sum_{i=0}^{m} D_t(i)e^{h(x_i)}\end{aligned} \tag{3}$$

where

$$\sum_{i=1}^{m} D_t(i)e^{h(x_i)} = 1$$

If the weight D is normalized for a next step calculation, then we can obtain the following equation.

$$\sum_{i=1}^{m} D_t(i)e^{-\alpha_t y_i h(x_i)} = e^{-\alpha_t}\left(e^{\alpha_t} - e^{-\alpha_t}\right)\varepsilon_t \tag{4}$$

Finally, we derive the optimal solution of the confidence rate α as follows.

$$\begin{aligned}&\frac{\partial}{\partial\alpha}\ln\left\{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t\right\}\\ &= \frac{-e^{-\alpha} + e^{\alpha}\varepsilon_t + e^{-\alpha}\varepsilon_t}{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t}\\ &= \frac{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t + -2e^{-\alpha} + 2e^{-\alpha}\varepsilon_t}{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t}\\ &= 1 + \frac{-2e^{-\alpha} + 2e^{-\alpha}\varepsilon_t}{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t}\\ &= -e^{-\alpha} + e^{\alpha}\varepsilon_t + e^{-\alpha}\varepsilon_t\\ &= e^{-\alpha}\left(-1 + e^{2\alpha}\varepsilon_t + \varepsilon_t\right)\\ &= \frac{1-\varepsilon_t}{\varepsilon_t} - e^{2\alpha}\end{aligned} \tag{5}$$

As expected, it is shown that the optimal α is

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{6}$$

Consequently, Normal AdaBoost has exponentially decreasing error rates [5].

2.2 Updating Weights of Normal AdaBoost

As expected, we can obtain the weight D by updating in Normal AdaBoost as follows.

$$\begin{aligned}D_{t+1}(i) &= e^{-y_i f_t(x_i)}\\ &= e^{-y_i f_{t-1}(x_i)} e^{-y_i \alpha_t h_t(x_i)}\\ &= D_t(i)e^{-y_i \alpha_t h_t(x_i)}\\ &= \begin{cases} D_t(i)e^{-\alpha} & if\ yb > 0\\ D_t(i)e^{\alpha} & if\ yb < 0\end{cases}\end{aligned} \tag{7}$$

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.

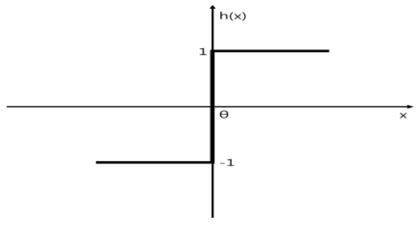
3. Boundary AdaBoost

3.1 Boundary Classifier

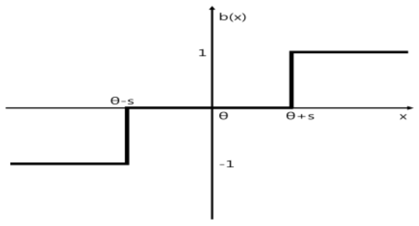
In this section, we will explain the modified binary classifier, a boundary classifier as shown in Fig. 1. According to machine learning theory, the binary classifier h is the decision function with two classes. Thus, the binary classifier at step t will be presented as follows [3], [5].

$$h_t(x) = \begin{cases} 1 & if\ x > \theta\\ -1 & otherwise\end{cases} \tag{8}$$

where θ is a threshold which is a criterion of a decision. In this paper, we propose a boundary classifier which has three classes {-1, 0, 1}, depending on an input value.



(a)



(b)

Fig. 1. Comparison of the binary classifier and a boundary classifier (a) Normal binary classifier (b) Boundary classifier.

The meaningless class of a boundary classifier can be determined as the neighborhood region [-*s*, *s*] of a threshold θ. Thus, the size of a boundary will be 2*s* and a boundary classifier *b* can be presented as follows.

$$b_t(x) = \begin{cases} 1 & if\ x - \theta > s\\ 0 & if\ |x - \theta| < s\\ -1 & otherwise\end{cases} \tag{9}$$

The training samples *x* belonging to the boundary *s* will be the feature with a low confidence rate, since these samples are located in the neighborhood of a threshold. In other words, we cannot determine if classified samples are correctly classified even if these samples are classified to their true classes. Fig. 2 shows three classifiers: two black classifiers and one red classifier. We will choose a red classifier even if all three classifiers have the 0 error rate. The number of samples far from the threshold is an important factor, which determines an optimal classifier. In other word, a red classifier will be classified correctly even if samples are affected by a noise [9-13].

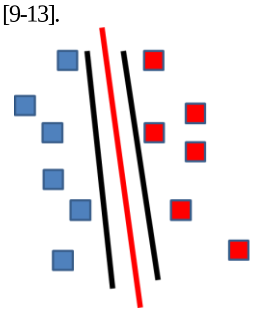


Fig. 2. Optimal classifier to scatter training data.

3.2 Optimal Solution of Boundary AdaBoost

Boundary AdaBoost has the particular weak classifiers having three values different from Normal AdaBoost: true (1), false (-1) and meaningless (0). Thus,

$$y_i b_t \in \{-1, 0, 1\}$$

where *y* is true classes of the training data, *b* is the boundary classifier. Therefore, equation (2) can be separated into three cases: *yb* > 0, *yb* < 0 and *yb* = 0.

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

where

$$\varepsilon_t = \frac{\sum_{i:yb<0} D_t(i)}{\sum_{i=1}^{m} D_t(i)}, \; \delta_t = \frac{\sum_{i:yb=0} D_t(i)}{\sum_{i=1}^{m} D_t(i)} \quad (11)$$

The optimal solution of the confidence rate of classifier α can be calculated as follows.

$$\begin{aligned}&\tfrac{\partial}{\partial\alpha}\ln\left\{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t + \left(1 - e^{-\alpha}\right)\delta_t\right\}\\ &= \frac{-e^{-\alpha} + e^{\alpha}\varepsilon_t + e^{-\alpha}\varepsilon_t + e^{-\alpha}\delta_t}{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t + \left(1 - e^{-\alpha}\right)\delta_t} \\ &= 1 + \frac{-2e^{-\alpha} + 2e^{-\alpha}\varepsilon_t - \delta_t + 2e^{-\alpha}\delta_t}{e^{-\alpha} + \left(e^{\alpha} - e^{-\alpha}\right)\varepsilon_t + \left(1 - e^{-\alpha}\right)\delta_t}\\ &= -e^{-\alpha} + e^{\alpha}\varepsilon_t + e^{-\alpha}\varepsilon_t + e^{-\alpha}\delta_t \\ &= e^{-\alpha}\left(-1 + e^{2\alpha}\varepsilon_t + \varepsilon_t + \delta_t\right) \\ &= \frac{1 - \varepsilon_t - \delta_t}{\varepsilon_t} - e^{2\alpha}\end{aligned} \quad (12)$$

Thus, we easily obtain

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1 - \varepsilon_t - \delta_t}{\varepsilon_t}\right). \quad (13)$$

3.3 Updating Weights of Boundary AdaBoost

In this section, we will present a method of updating weights as follows [3].

$$D_{t+1}(i) = e^{-y_i f_t(x_i)} \quad (14)$$

Then, we can derive the following form using a forward additive model, which is a greedy approach.

$$\begin{aligned}D_{t+1}(i) &= e^{-y_i f_{t-1}(x_i)} e^{-y_i \alpha_t b_t(x_i)} \\ &= D_t(i)e^{-y_i \alpha_t b_t(x_i)} \\ &= \begin{cases} D_t(i)e^{-\alpha} & if\ yb > 0 \\ D_t(i)e^{0} & if\ yb = 0 \\ D_t(i)e^{\alpha} & if\ yb < 0 \end{cases}\end{aligned} \quad (15)$$

The weights of training data sets depend on classification results. The weights of adjacent features close to a threshold are increasing, which indicates that they are incorrectly classified data. With this method, we increase the weights of training data that are far from the threshold, where confidence rates are higher. On the other hands, 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. Consequently, we proposed the Boundary AdaBoost as choosing the optimal classifier to scatter training data far from the threshold associated with the detection objects [14].

3.4 Fuzzy Lookup Method for determining Boundary Size

We now need to consider how the boundary size *s* might be determined. In this paper we use the fuzzy lookup methods in order to determine the boundary size since these methods can easily be applied and implemented by the expert's experience [14-16]. Thus, the fuzzy rules R_i are chosen via the relationship between the input x_1, x_2 and output *y* of the target system as follows.

$$R_i : If\ x_1\ is\ A_{i1}\ and\ x_2\ is\ A_{i2}\ Then\ y\ is\ B_i \quad (16)$$

These fuzzy rules can be expressed by the fuzzy inference μ_B where includes the fuzzifier, defuzzifier and fuzzy rules. Thus,

$$\mu_{B'}(y) = \max\left[\sup_{x\in U}\left(\mu_{A'}(x)\prod_i \mu_{A_{il}}(x_i)\mu_{B_i}(y)\right)\right] \quad (17)$$

where

$$\mu_{A'}(x) = \begin{cases} 1 & if\ x = x^* \\ 0 & otherwise \end{cases}, \quad y^* = \frac{\sum_{l=1}^{M} y_l w_l}{\sum_{l=1}^{M} w_l} \quad (18)$$

where μ_A is the singleton fuzzifier, *y** is the centered average defuzzifier, *l* and *w* are the index and weights of the fuzzy rules, respectively. Several fuzzy rules can be crashed each other since they depend on the input-output pairs via the experience. This problem can be solved by assigning the degree *D* to each rule [].

$$D(R_i) = \mu_{A_{il}}(x_i^p)\mu_{B_i}(y^p) \quad (19)$$

4. Experimental Classification Results

We simulated Boundary AdaBoost to ensure performance using real-world images. For testing, we used both the pedestrian for positive data and road background for negative data, and the image size is 128 by 64 pixels. Thus, the goal of this simulation is to detect the pedestrian on the road. We chose the histogram of oriented gradients [17] for the extraction algorithm which obtains sample features. The partition of HOG was chosen to be three on the x-axis, six on the y-axis and eight on the orientation [17-19]. Thus, the resolution of these divided images will be 42 × 10 pixels. Furthermore, the orientation of these segment images can be divided into eight bins. Thus, the total dimension of features will be 3 × 6 × 8 bins. In other words, one training image will be separated into 18 segments. Similarly, each segment will be divided into 8 orientations [18].

For the first experiment, we measured the error rates using various boundary size *s*. Table 1 shows the initial value of both *α* and *s*, and shows the change of parameters at each step *t*. We assume that s = 0.04 and that both the minimum and maximum values of features will be assumed to be almost constant. Then, we set the boundary *s* to the various sizes in order to know the influence on the error rates of the final strong classifier. For testing, we have simulated Boundary AdaBoost with both 1945 positive data and 2360 negative data in MATLAB R2008a. Furthermore, 3145 images were tested that consist of 1105 positive data and 2040 negative data. We used 100 boundary weak classifiers, and have taken 2280 seconds for training and 0.01 seconds for testing with the 2.4 GHz Pentium 4 Dual core and 4G RAM. Fig. 3 shows the number of features belong to boundary using the proposed algorithm when s = 0.04. This result means the constant boundary size cannot reduce the number of the features belong to the boundary. Hence, we need to find the relevant boundary size.

The error rates of between Normal AdaBoost and Boundary AdaBoost, which have various boundary sizes using 10-fold cross validation, are shown in Fig. 4. As expected, constant boundary sizes have decisive effects on the error rates. Thus, it is important for experts to decide on the proper boundary size. Therefore, this method is hampered by its necessary reliance on the experience of an expert to define the boundary that should be used.

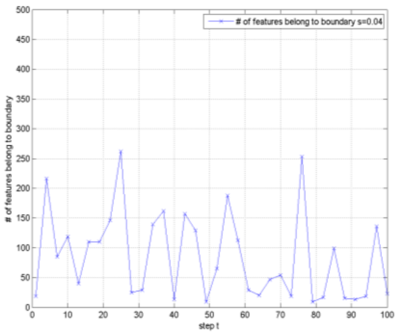
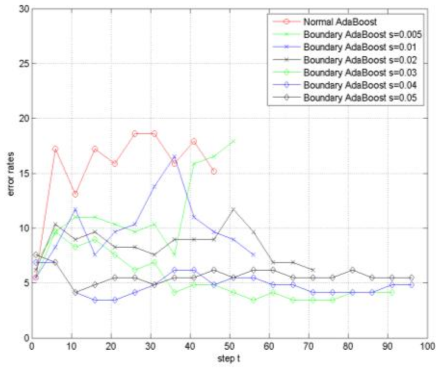


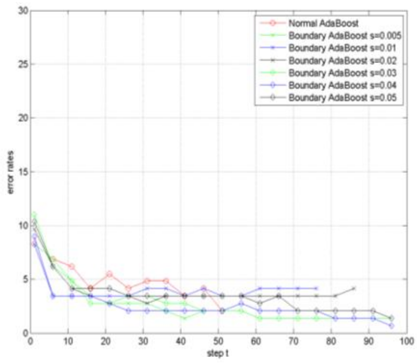
Fig. 3. Number of features belong to boundary using 10 fold cross validation.

Table 1. Parameters for training (*r* is the number of features belong to *s* and *α* is the confidence rate of the classifier).

t	1	2	3	4	5
s	0.012	0.013	0.014	0.012	0.013
α	0.780	0.709	0.569	0.447	0.347
r	135	180	169	181	163



(a)



(b)

Fig. 4. Comparison of error rates with various boundary sizes (a) when the error rates cannot exponentially reduce (b) when the error rates exponentially reduce.

To assign the boundary size, we applied the fuzzy lookup method. This method needs the input-output pairs for the IF-THEN rules, which were obtained by the pre-test. Thus, we chose the two fuzzy inputs as both the classification error ε and the number of the features belong to the boundary. Also, the output of the fuzzy method was chosen as the boundary size. The each input consists of five fuzzy classes: Very Small (VS), Small (S), sTandard(T), Big (B) and Very Big(VB). These 25 fuzzy rules were directly applied to the fuzzy lookup table. Also, the membership functions of two inputs and one output were chosen as shown in Fig. 5. Each function identically was same with an individual fuzzy class. We simulated using the fuzzy lookup method with the lookup tables, which were assigned by the pre-test. Fig. 6 shows the effect on the applied fuzzy algorithm. The error rates were similar to the exponential function. This means the adjusted boundary sizes significantly affect the entire performance of the boosting strong classifier. Also, the error rates were more reduced than the Normal AdaBoost when the number of the combined features was increased. This result verified that the boundary classifiers are reasonable for the ensemble learning methods, such as the boosting.

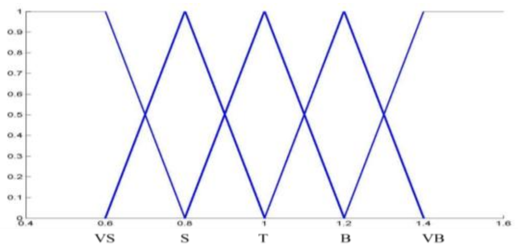


Fig. 5. Input membership function for the fuzzy lookup method.

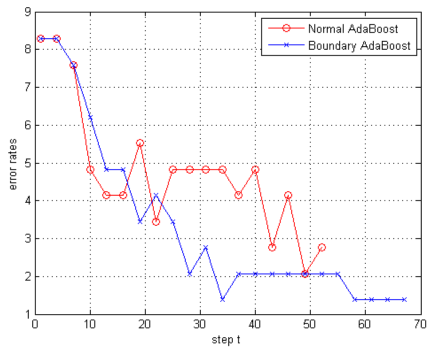


Fig. 6. Comparison of error rates using the fuzzy lookup method and 10 fold cross validation

5. Conclusions

In this paper, we have examined the accuracy of AdaBoost with a boundary. We have improved the AdaBoost performance using by including a boundary classifier. The Boundary AdaBoost can adjust the weights of the features close to the threshold. As expected, it is important to determine the boundary size, which is accomplished by the fuzzy lookup method. Thus, we can adjust the reasonable boundary size to increase the accuracy. Furthermore, for higher speed calculations, we will use linear discriminant analysis to combine features obtained from multiple sensors, and we will study the combined features at higher steps of classification and the voting method, which excludes the histogram bins by using the confidence rate.

References

[1] [1] 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.

[2] R. E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions," *Machine Learning*, Vol. 37, No. 3. pp. 297-335, 1999.

[3] J. Friedman, T. Hastie and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *The Annals of Statistics*, Vol. 28, No. 2, pp. 337-407, 2000.

[4] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost - Teaching AdaBoost to Generalize Better," *Graphicon*, 2005.

[5] R. E. Schapire, "Theoretical Views of Boosting and Applications," *In Proceedings of Algorithmic Learning Theory*, Vol. 1720, pp. 13-25, 1999.

[6] Y. Freund and R. 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.

[7] R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-Based System for Text Categorization," *Machine Learning*, Vol. 39, No. 2/3, pp. 135-168, 2000.

[8] Y. Freund, "Boosting a Weak Learning Algorithm by Majority," *Information and Computation*, Vol. 121, No. 2, pp. 256-285, 1995.

[9] W. W. Cohen, "Fast Effective Rule Induction," *In Proceedings of Machine Learning*, pp. 115-123, 1995.

[10] W. W. Cohen and Y. Singer, "A Simple, Fast, and Effective Rule Learner," *In Proceedings of Artificial Intelligence*, pp. 335-342, 1999.

[11] H. Drucker and C. Cortes, "Boosting Decision Trees," *In Advances in Neural Information Processing Systems*, Vol. 8, pp. 479-485, 1996.

[12] Y. Freund, "An Adaptive Version of the Boost by Majority Algorithm," *In Proceedings of Computational Learning Theory*, pp. 102-113, 1999.

[13] R. Maclin and D. Opitz, "An Empirical Evaluation of Bagging and Boosting," *In Proceedings of Artificial Intelligence*, pp. 546-551, 1997.

[14] Takagi and M. Sugeno, "Fuzzy Identification of Systems and its Applications to Modeling and Control," *On Systems, Man, Cybernetics*, Vol. 15, No. 1, pp. 116-132, 1985.

[15] D. Singer and P. G. Singer, "System Identification Based on Linguistic Variables," *Fuzzy Sets and Systems*, Vol. 47, No. 2, pp. 141-149, 1992.

[16] K. Shahida, Ibraheem, Moinuddin and M. Farooq, "A Table Lookup Scheme for Fuzzy Logic Based Model Identification Applied to Time Series Prediction," *In Proceedings of Information Fusion*, Vol. 2, pp. 1449-1456, 2003.

[17] Q. Zhu, S. Avidan, Y. Mei-Chen and C. Kwang-Ting, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," *In Proceedings of Computer Vision and Pattern Recognition*, Vol. 2, pp. 1491-1498, 2006.

[18] F. Suard, A. Rakotomamonjy, A. Bensrhair and A. Broggi, "Pedestrian Detection using Infrared images and Histograms of Oriented Gradients," *In Proceedings of Intelligent Vehicles Symposium*, pp. 206-212, 2006.

[19] Y. Yamauchi, H. Fujiyoshi, B. Hwang and T. Kanade, "People Detection Based on Co-Occurrence of Appearance and Spatiotemporal Features," *In Proceedings of Pattern Recognition*, pp. 1-4, 2008.

저 자 소 개

이원주 (Wonju Lee)
2005 년 : 협성대학교 컴퓨터공학과 공학사
2007 년 : 연세대학교 전기전자공학과 공학석사
2007 년 ~ 현재 : 연세대학교 전기전자공학과 박사과정

관심분야 : 머신학습, 센서네트워크, 영상처리
Phone : +82-2-2123-2868
E-mail: delicado@empas.com

천민규 (Minkyu Cheon)
2006 년 : 연세대학교 전기전자공학과 공학사
2013 년 : 동대학원 공학박사
2013 년 ~ 현재: 에스원 책임연구원

관심분야 : 머신학습, 패턴인식, 컴퓨터비젼
Phone : +82-2-2123-2868
E-mail: 1000minkyu@yonsei.ac.kr

현창호 (Chang-Ho Hyun)
1999 년 : 광운대학교 제어계측공학과 공학사
2002 년 : 연세대학교 전기전자공학과 공학사
2008 년 : 동대학원 공학박사
2008 년 ~ 2009 년 : 삼성전자 책임연구원
2009 년 ~ 현재 : 국립공주대학교 전기전자제어공학부 조교수

관심분야 : 지능제어, 비선형제어, 로봇제어, 로봇공학
Phone : +82-41-521-9168
E-mail: hyunch@kongju.ac.kr

박민용 (Mignon Park)
1973 년 : 연세대학교 전자공학과 공학사
1977 년 : 연세대학교 전자공학과 공학석사
1982 년 : 일본 동경대학교 공학박사
1982 년 ~ 현재 : 연세대학교
전기전자공학과 정교수

관심분야 : 퍼지제어,로보틱스, 바이오시스템
Phone : +82-2-2123-2868
E-mail: mignpark@yonsei.ac.kr