Logitboost-SO Learning Algorithm for Human Iris Recognition

Wen-Shiung Chen
Dept. of Electrical Engineering,
National Chi Nan University,
Nantou, Taiwan
e-mail: wschen@ncnu.edu.tw*w*

Lili Hsieh
Information Management Dept.
Hsiuping Institute of Technology
Taichung, Taiwan
e-mail: lily@mail.hit.edu.tw

Wei-Chih Tang
Dept. of Electrical Engineering,
National Chi Nan University,
Nantou, Taiwan
e-mail: 97323559@ncnu.edu.tw

***Abstract*—Boosting has been used extensively in the field of machine learning. This work intends to apply boosting method to iris biometrics. The recognition performance of boosting-based classification can be improved greatly by means of different re-weighting rules and different voting regulations based on the assigned weights. This paper proposes a novel Logitboost-SO algorithm which integrates similarity-oriented concepts into additive logistic model. We modify the existing manner of combining classifiers with Logitboost by utilizing multi-weight update rule to refine boosting algorithm. The experimental results show that Logitboost-SO applied to iris recognition is better than existing boosting algorithms.**

***Keywords*–Biometrics; Iris Recognition; Boosting; Adaboost; Logitboost..**

I. INTRODUCTION

Iris is a thin circular organ which lies between the cornea and the sclera in human eyes. This trait has always been used for high security applications since jailers identified criminals by their irises in 18th century Paris prison. Among all the biometric traits, human iris has rich texture information and excellent uniqueness, which has been proved by Daugman in [1]. He presented the results of 200 billion iris pair comparisons to give the conclusion that the iris is fit for national scale deployment of recognition. After that, there are also many other researchers who propose different novel iris recognition systems [2]-[6]. The framework of recognition includes three units: pre-processing, feature extraction and learning/classification. This work aims at the learning and/or classification step. In any biometric recognition system, some known data are fed into the machine learning sub-system for training a classifier, and then the optimal classifier learn well and created after training. However, Logitboost has not been paid much attention on iris. As a superior boosting, Logitboost has only been regarded as a learning tool and its potentials are ignored. Even so, there are many fields including tumor [7], protein [8], and text [9] classification in which the researchers proposed the variants of Logitboost to discriminate the classes. In these papers, the base classifier of Logitboost is a breaking through point and provides Logitboost lots of improvement space. Here we choose a boosting algorithm as machine learning.

Boosting is a learning method which finds a way to combine "weak" classifiers and build up a "strong" classifier. Since Freund and Schapire [10] invented Adaboost in 1996, there are many works about how to improve Adaboost and some variants of Adaboost are proposed. Logiboost [11] applied backfitting to a logistic regression model and gives limited weights to mislabeled samples. SOBoost [12] operated similarity oriented rules. In this paper, we propose a novel boosting algorithm, called Logitboost-SO, which exploits multiple re-weighting rules to integrate the similarity oriented concepts into logistic regression model.

This paper is organized as follows. In Section II, we give the motivation to this work and then propose a new boosting learning algorithm. Section III states the experiment on iris recognition and discusses the results. Section IV makes a conclusion.

II. THE PROPOSED BOOSTING ALGORITHM

A. Motivation

The motivation of adopting SOBoost as the weak learner can be summed up with three points: (i) SOBoost is a novel boosting algorithm which follows the similarity rules and outperforms Adaboost; (ii) the confidence value of decision tree has only two values, 1 and -1. However, the confidence-rated classifier of SOBoost can produce a confidence value between -1 and 1; (iii) Logitboost demands strictly on the component classifier. However, the re-weighting and potential function of SOBoost is similar to Logitboost's (see Fig. 1), hence we infer that there exists a way to blend the two of them.

 ISBN: 978-1-61208-139-7

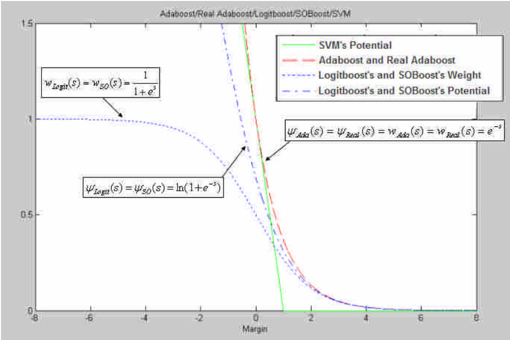


Figure 1. Comparison of the potential and weight function of the machine learning algorithm.

B. The Logitboost-SO Learning Algorithm

Though many people use Adaboost, there is a problem with it. And that is when you have the samples of enormous classification errors, the weights will increase excessively. In the result, this phenomenon will destroy the whole training system. Friedman *et al.* [11] (2000) applied log-likelihood loss function to replace the exponential loss function used in Adaboost and limited the weight distributed over the "miss" samples. Moreover, Logitboost fits the weak learner by a weighted least-square regression, which means not only the miss samples, but also the "too-correct" classifications are punished. The confidence scores of the real Adaboost are generated from the natural logarithm of the ratio of $P_w^+(\phi_t)$ to $P_w^-(\phi_t)$, where ϕ_t is the best feature of t iteration, $P_w^+(\phi_t)$ and $P_w^-(\phi_t)$ are the probability distribution of positive and negative samples. Hence the samples which are more similar may have lower confidence score, and this is a little contradictive. Thus in 2008, He *et al.* [12] proposed a similarity-oriented boosting (SOBoost) algorithm. In SOBoost they use the ratio of the bidirectional cumulative probability distribution to construct the confidence function. And therefore they ensure the monotonous of the hypotheses.

The main idea of Logitboost-SO machine learning algorithm is to integrate SOBoost concept into the Logitboost scheme. In the Logitboost-SO algorithm, the confidence function of SOBoost is used to fit an additive logisitic regression model. Besides, the weights used for calculating the confidence function are separated from the weights used for fitting logistic model and update independently according to the re-weighting rule of the SOBoost. The flowchart of Logitboost-SO is shown in Fig. 5. The pseudo-code is listed and described briefly as follows:

- **Training set** : $(x_1, y_1),\dots,(x_m, y_m)$ where $x_i \in X$,

  $y_i = \{-1,+1\}$, feature $\phi_k \in \Phi$.

- **Initialize** the weights

$$w_{same}(i) = D_{same}(i) = \frac{1}{2m_{same}},$$

$$w_{same}(i) = D_{same}(i) = \frac{1}{2m_{same}},$$

$$w_{different}(i) = D_{different}(i) = \frac{1}{2m_{different}};$$

committee function $F(x_i) = 0$;

probabilites $p_{same}(i) = p_{different}(i) = \frac{1}{2}$.

- **For** $t = 1,2,\dots,T$

1. For $k = 1,2,\dots,q$

a. Divide X into J parts $X_1,\dots,X_J$.

b. Compute

the probability

$$P_w^l(j) = P(x_i \in X_j, y_i = l) = \sum_{i:x_i \in X_j, y_i = l} D_t(i), l = \pm 1;$$

cumulative probability

$$C_w^+(\phi_t) = \int_{-\infty}^{\phi_t} P_w^+(\phi_t) d\phi_t, \quad C_w^-(\phi_t) = \int_{\phi_t}^{\infty} P_w^-(\phi_t) d\phi_t$$

the weights

$$w_{same}(i) = p_{same}(i)(1 - p_{same}(i)),$$
$$w_{different}(i) = p_{different}(i)(1 - p_{different}(i));$$

working responses

$$z_{same}(i) = \frac{1 - p_{same}(i)}{p_{same}(i)(1 - p_{same}(i))},$$

$$z_{different}(i) = \frac{-p_{different}(i)}{p_{different}(i)(1 - p_{different}(i))};$$

the confidence function

$$\begin{aligned} h(\phi_k) &= 2\,\text{sigmf}(C_w^+(\phi_k) - C_w^-(\phi_k), \alpha, c) - 1 \\ &= \frac{2}{1 + \exp(-\alpha(C_w^+(\phi_k) - C_w^-(\phi_k)) - c)} - 1 \end{aligned}.$$

2. Fit the function $h(x_i)$ by a weighted least-squares regression of $z(x_i)$ to x_i with weights $w(x_i)$ and get

$$h_t(\phi_t) = \arg\min_{h:\phi \in \Phi} \sum_{i=1}^{m} w(x_i)\left(z(x_i) - h(x_i)\right)^2 \cdot$$

3. Update

$$D_{t+1}(i) = \frac{D_t(i)\,\text{sigmf}(-y_i h_t(\phi_t(x_i)), \beta, 0)}{Z_t}$$

$$= \frac{D_t(i)}{Z_t(1+\exp(-\beta(-y_i h_t(\phi_t(x_i)))))}$$

where Z_t is a normalization factor

$$Z_t = \sum_{i=1}^{N} D_t(i)\,\text{sigmf}(-y_i h_t(\phi_t(x_i)), \beta, 0);$$

and

$$F_{t+1}(x_i) = F_t(x_i) + \frac{h_t(\phi_t(x_i))}{2},$$

$$p_{t+1}(x_i) = \frac{1}{1+e^{-2F_{t+1}(x_i)}}.$$

- **Output** the final classifier $H(x) = \text{sign}\sum_{t=1}^{T} h_t(\phi_t(x))$

The framework can be constructed by the following steps. First of all, we divide the dataset into J parts, and J depends on how sensitively you want about this machine. Second, we calculate the probability of each decided level. Based on decided levels we can get the weights which were generated from Logitboost idea. These probabilities are used to link the cumulative probabilities and confident scores which are applied in the SOBoost idea. Finally, these parameters work as the input of weak classifiers and the reweighting functions of Logitboost, and then we can obtain this machine. Concisely speaking, the flowchart of Logitboost-SO algorithm (Fig. 5) may expound this integrated concept.

III. EXPERIMENTAL RESULTS AND DISCUSSION

Verification experiments are carried out on UBIRIS.v1, to obtain a threshold separating false rejection rate (FRR) and false acceptance rate (FAR). For the case of FRR, we obtain the distribution of matching distance between the unknown classes and the registered classes. For the case of FAR, we also obtain the distribution of matching between the unknown classes for impostors and the registered classes. We use the equal error rate (EER) to evaluate feature extraction performance.

Embedded Based on the proposed learning algorithm, an iris recognition system, as shown in Fig. 2, is designed. The experiments of verification and identification are performed on UBIRIS.v1 Session 1, which has at least 5 pictures in each class. Moreover, a four-fold cross-validation is carried out to confirm the performance. In each iteration of the cross-validation, three of four partitions are used as the training set, and the other one partition is used as testing set.

As mentioned previously, we divide the iris region and feed in the boosting algorithm as different feature candidates. In Fig. 3, we apply colors in order of importance (sum of the weights) to illustrate the weights distributed by Discrete Adaboost on 8×4 non-overlapping blocks. The brighter color means that the selected region is more important.

Furthermore, in the sake of comparing the number of divisions, we also partition the iris region into 32×13 overlapping blocks with size 64×32 and feed them as different "features" in Discrete Adaboost. Since the experimental result of 32×13 divisions is much better than 8×4 divisions when using Discrete Adaboost, all the other boosting algorithms are only conducted under 32×13 divisions. Besides, since the weights of confidence-rated predictor cannot be summed, we only present the first four regions selected by Logitboost-SO in Fig. 4. From Figs. 3 and 4 we can see obviously that the lower half part of the iris region is more discriminative.

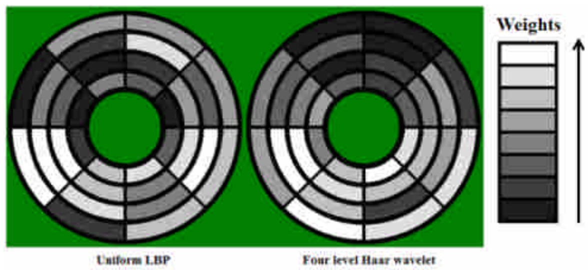
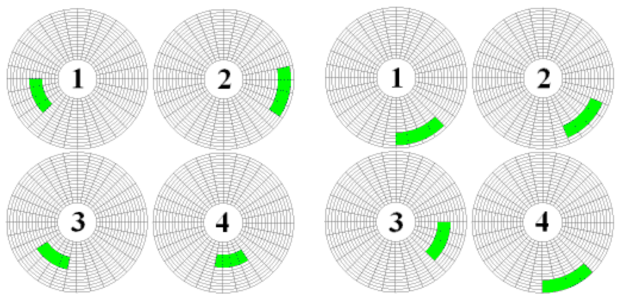


Figure 3. Weight distribution obtained from Discrete Adaboost after 500 iterations. The brighter color means that the selected region is given more weight.



(a) 4-level Haar wavelet feature (b) ULBP feature

Figure 4. The first four sub-regions selected by Logitboost-SO with different feature types.

With the levels of Haar wavelet getting higher more global information is extracted and more local information is lost. According to the characteristics of the input images, the suitable feature type for recognition is different. In Table 1, we compare FAR and FRR of different levels of Haar wavelets with discrete Adaboost, and discover that the use of four-level Haar wavelet is the best choice for our experiments.

From Tables 2 and 3 it is clear that the results of 32×13 candidates are much better than those of 8×4 candidates when using discrete Adaboost. Hence in the experiments of other boosting algorithms, the iris images are all divided into 32×13 candidates.

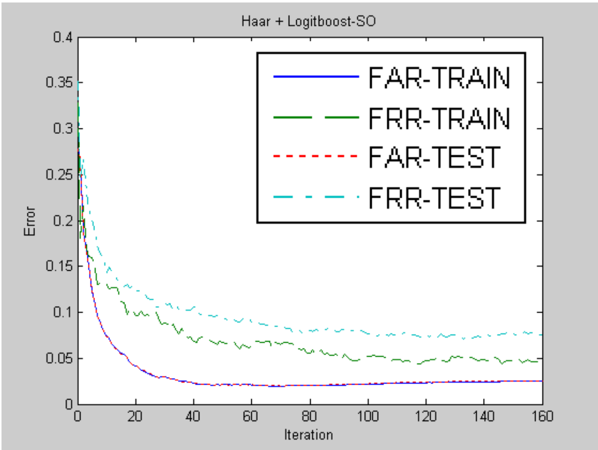
Moreover, real Adaboost performs incredibly well on training set but worse on testing set. This reveals that real Adaboost tends to be overfitting. However, as mentioned in [5], SOBoost has no such a problem. Besides, since Logitboost-SO learns the SOBoost rules, we can also avoid this problem. The results in Tables 2 and 3 confirm this argument.

Experimental results also show that the proposed boosting is better than existing boosting algorithms, whether for verification or identification applications. The

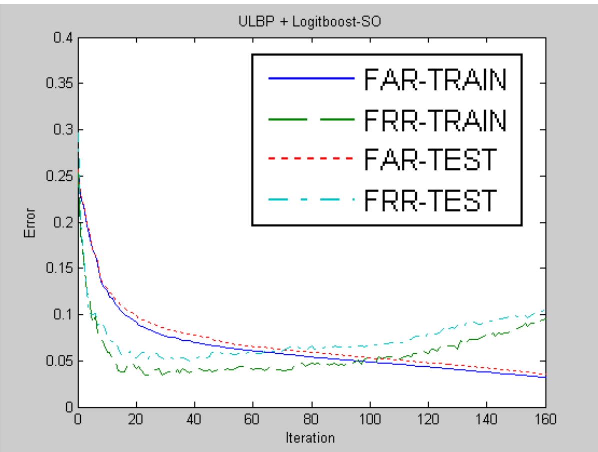
 ISBN: 978-1-61208-139-7

FAR and the FRR curves of Logitboost-SO on training and testing set are shown in Fig. 6.

In our identification experiments, the *k*-nearest neighbor algorithm is cascaded after the boosting methods. From Table 3 and Fig. 7 we observe that 3NN performs worse than 1NN. The reason is that the boosting has trained the classification system completely, and hence the larger value of *k* only makes the boundary less distinct.



(a) 4-level Haar wavelet feature



(b) ULBP feature

Figure 6. FAR and FRR curves of Logitboost-SO on training and testing set.

The performances of using ULBP feature are worse than those of four-level Haar wavelet features. This result agrees with most experiments of related iris recognition literature in which the local features always perform better than global features. That is because, although the global features are invariant to rotation, scale and translation, global features are statistical information. Hence the extracted statistical features can only give indistinct information.

TABLE 1. COMPARISON OF DIFFERENT LEVELS OF HAAR WAVELET

Haar wavelet		
Training set	***FAR***	***FRR***
2-level	0.273151	0.313725
3-level	0.154558	0.184874
4-level	0.084045	0.088235
Testing set	***FAR***	***FRR***
2-level	0.272485	0.443969
3-level	0.155489	0.233405
4-level	0.083933	0.115632

TABLE 2. VERIFICATION PERFORMANCE COMPARISON OF BOOSTING

4-level Haar wavelet		
Training set	***FAR***	***FRR***
Discrete Adaboost (8×4)	0.084045	0.088235
Discrete Adaboost (32×13)	0.023358	0.039216
Real Adaboost	0.000000	0.000000
Logitboost	0.000000	0.077031
SOBoost	0.104189	0.061625
Logitboost-SO	**0.024844**	**0.037815**
Testing set	***FAR***	***FRR***
Discrete Adaboost (8×4)	0.083933	0.115632
Discrete Adaboost (32×13)	0.023368	0.113490
Real Adaboost	0.000891	0.251249
Logitboost	0.000322	0.194147
SOBoost	0.104293	0.054961
Logitboost-SO	**0.024880**	**0.070664**
ULBP		
Training set	***FAR***	***FRR***
Discrete Adaboost (8×4)	0.056420	0.064426
Discrete Adaboost (32×13)	0.034094	0.054622
Real Adaboost	0.010614	0.000000
Logitboost	0.000051	0.152661
SOBoost	0.078475	0.047619
Logitboost-SO	**0.049455**	**0.046218**
Testing set	***FAR***	***FRR***
Discrete Adaboost (8×4)	0.059885	0.103498
Discrete Adaboost (32×13)	0.033032	0.126338
Real Adaboost	0.029831	0.121342
Logitboost	0.001057	0.360457
SOBoost	0.083388	0.053533
Logitboost-SO	**0.053907**	**0.064954**

TABLE 3. IDENTIFICATION PERFORMANCE COMPARISON OF BOOSTING

Recognition Rate (%)		
Algorithms	***1NN***	***3NN***
4-Haar + Discrete Adaboost (8×4)	77.3019	77.5161
4-Haar + Discrete Adaboost (32×13)	90.1499	89.2934
4-Haar + Real Adaboost	91.8630	92.0771
4-Haar + Logitboost	93.3619	93.3619
4-Haar + SOBoost	91.8630	91.2206
4-Haar + Logitboost-SO	**94.4325**	**94.2184**
Recognition Rate (%)		
Algorithms	***1NN***	***3NN***
ULBP + Discrete Adaboost (8×4)	75.8030	75.8030
ULBP + Discrete Adaboost (32×13)	83.0835	81.5846
ULBP + Real Adaboost	84.3683	83.2976
ULBP + Logitboost	84.1542	84.3683
ULBP + SOBoost	88.2227	87.1520
ULBP + Logitboost-SO	**89.7216**	**87.7944**

Since we have known that Logitboost-SO performs better than the existing boosting algorithms, the next step of our experiment is to optimize the performance of Logitboost-SO. Since Logitboost-SO uses the confidence rule of SOBoost, we can also improve the performance by adjusting the gradient of the confidence function α. From Fig. 8 we can see clearly that when α is getting smaller the speed of convergence is getting slower. There is a trade-off between growth rate and recognition accuracy.

After the most suitable parameter α has been found, and is applied to replace the initial default setting α = 1, we use a two-stage classification scheme to make a decision. In practice, if the confidence score obtained from one of the feature type is too close to 0 and the magnitude is lower than the threshold, another feature type is introduced to provide confidence score instead. Tables 4 and 5 provide comparisons of individual and combined feature type for verification and identification applications respectively. It is clear that cascading system outperforms the system of single feature type.

TABLE 4. VERIFICATION PERFORMANCE COMPARISON OF INDIVIDUAL AND COMBINED FEATURE TYPES

Testing Error		
Algorithms	***FAR***	***FRR***
4-Haar + Logitboost-SO	**0.018772**	**0.073519**
ULBP + Logitboost-SO	**0.063165**	**0.052106**
(4-Haar + Logitboost-SO) →(ULBP + Logitboost-SO)	**0.021419**	**0.030692**
(ULBP + Logitboost-SO) →(4-Haar + Logitboost-SO)	**0.020296**	**0.034975**

TABLE 5. IDENTIFICATION PERFORMANCE COMPARISON OF INDIVIDUAL AND COMBINED FEATURE TYPES

Recognition Rate (%)	
Algorithms	***Accuracy***
4-Haar + Logitboost-SO	**94.6467**
ULBP + Logitboost-SO	**92.5054**
(4-Haar + Logitboost-SO) →(ULBP + Logitboost-SO)	**97.0021**
(ULBP + Logitboost-SO) →(4-Haar + Logitboost-SO)	**94.8608**

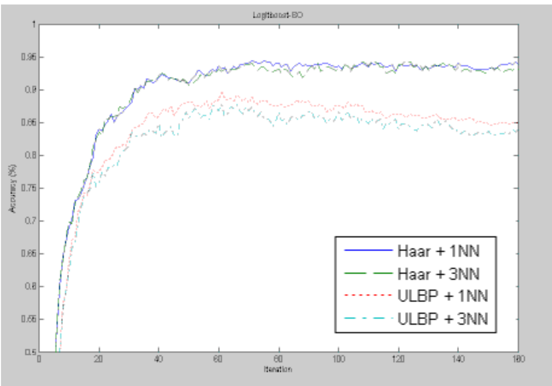


Figure 7. Recognition accuracy of Logitboost-SO on training and testing set.

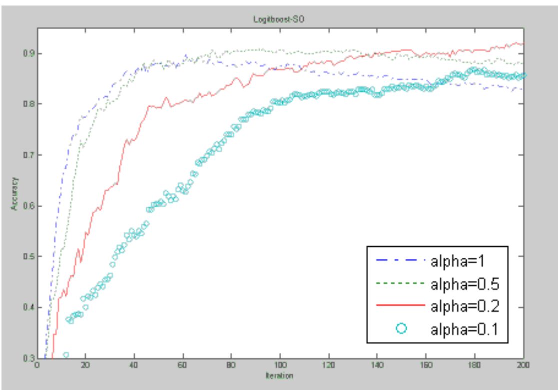


Figure 8. Recognition accuracy of Logitboost-SO using different α.

IV. CONCLUSION

In this paper, we proposed a novel iris recognition method based on Logitboost-SO and the cascading strategy. Unlike existing variants of Logitboost, Logitboost-SO combines Logitboost and SOBoost by using multi-weight update rule. To optimize the performance further, we adjust the gradient of the confidence function. Moreover, the local and global features are cascaded to provide complementary information. Our experimental results present evidence that Logitboost-SO outperforms former boosting algorithms and the cascading system can improve the performance further.

REFERENCES

[1] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 1148-1161, 1993.

[2] W. W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," IEEE Trans. on Signal Processing, vol. 46, no. 4, pp. 1185-1188, Apr. 1998.

[3] L. Ma, T. Tan, Y. Wang and D. Zhang, "Personal identification based on iris texture analysis," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1519-1533, Dec. 2003.

[4] Z. Sun, Y. Wang, T. Tan and J. Cui, "Improving iris recognition accuracy via cascaded classifiers," IEEE Trans. on System, Man and Cybernetics, vol. 35, no. 3, pp. 435-441, Aug. 2005.

[5] J. G. Daugman, "Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons," Proceedings of the IEEE, vol. 94, no. 11, pp. 1927-1935, Nov. 2006.

[6] C.-T. Chou, S.-W. Shih, W.-S. Chen V. W. Cheng and D.-Y. Chen, "Non-orthogonal view iris recognition system," IEEE Trans. on Circuits and Systems for Video Technology, pp. 417-430, vol. 20, no. 3, Mar. 2010.

[7] M. Dettling and P. Buhlmann, "Boosting for tumor classification with gene expression data," Bioinformatics, vol. 19, no. 9, pp. 1061-1069, 2003.

[8] Y. D. Cai, K. Y. Feng, W. C. Lu and K. C. Chou, "Using Logitboost classifier to predict protein structural classes," Journal of Theoretical Biology, vol. 238, no. 1, pp. 172-176, Jan. 2006.

[9] S. Kotsiantis, E. Athanasopoulou and P. Pintelas, "Logitboost of multinomial Bayesian classifier for text classification," International Review on Computers and Software, vol. 1, no. 3, 2006.

[10] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148-156, 1996.

[11] J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," The Annals of Statistics, vol. 38, no. 2, pp. 337-407, Nov. 2000.

[12] Z. He, Z. Sun, T. Tan, X. Qiu, C. Zhong and W. Dong, "Boosting ordinal features for accurate and fast iris recognition," Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Jun. 2000.

 ISBN: 978-1-61208-139-7

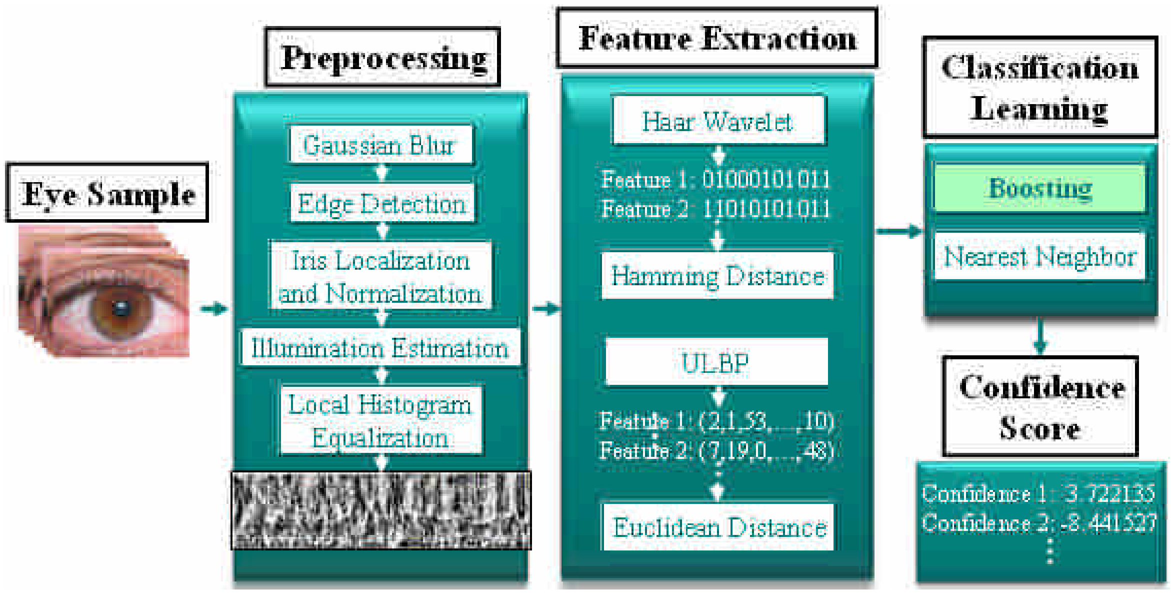


Figure 2. The system architecture.

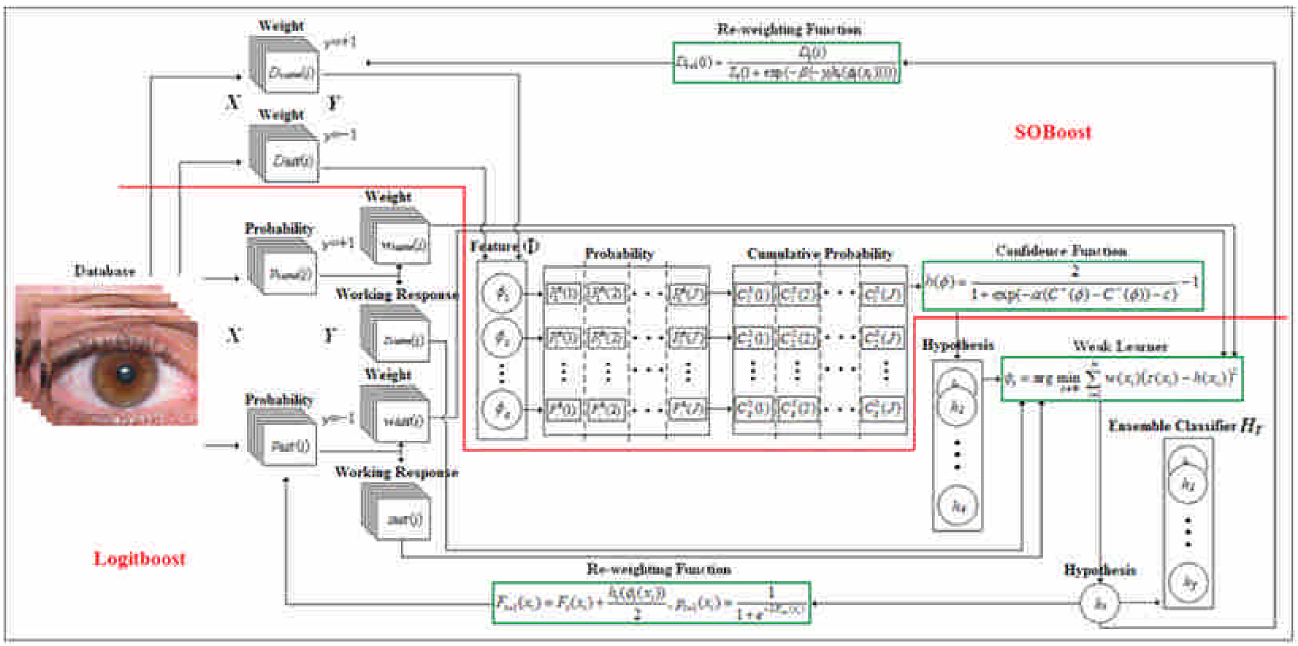


Figure 5. The flowchart of Logitboost-SO machine learning algorithm.