IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 24, 2020, accepted June 11, 2020, date of publication June 23, 2020, date of current version July 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004473

An SVM-Based AdaBoost Cascade Classifier for Sonar Image

HUIPU XU AND HAIYAN YUAN
College of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China
Corresponding author: Huipu Xu (hpx1212@dlmu.edu.cn)

ABSTRACT This paper proposes an improved AdaBoost classifier for sonar images with low resolution ratio and noise. First, Histogram of oriented gradient (HOG) is used to perform feature extraction, and a weak classifier is obtained by Support vector machine (SVM) at the same time. Then, multiple SVM models are constructed for target classification based on the AdaBoost cascade classification framework. A new function for updating sample weights has been designed in this paper to improve the accuracy of the classifier. And new iteration rules of classifier have been made to reduce the training time of the proposed method. The experimental results on the sonar dataset which are proposed for improving the generalization ability in this paper show that the classification accuracy of the proposed algorithm is about 92%, and the accuracy on Cifar-10 dataset is also higher than general methods.

INDEX TERMS AdaBoost, sonar image classification, SVM.

I. INTRODUCTION

The sonar image classification is one of the most important research topics with great significance. Up to now, related works have involved the tracking and protection of endangered aquatic organisms [1], classification and tracking of sea-surface obstacles [2], [3], water environment sounding and modeling [4], seabed modeling and mapping [5], [6], salvage and rescue [7], submarine pipeline detection [8]–[11], submarine target location and identification [12]–[14], submarine environment description [15], and clinical Medical disease detection [16]. Generally, the color features of the images obtained by underwater equipment are usually lost, and the scattering of light and non-target also cause disturbance. As a result, the texture characteristics of the target are weak. The shape and size of the targets are the main features. And two aspects generally need to be considered: one is the diversity of the feature in the incomplete target, and another is the feature difference between the target and background. Thus, the optical image is usually processed for noise reduction, and sonar images have serious features missing. Therefore, it is more effective to construct a robust classifier to distinguish target information. Due to the special imaging environment, sonar images have strict requirements for classifiers, it is necessary to design different classifiers according to the situation. Unlike optical images, the complex and diverse underwater environment will cause more interference to sonar images. These interferences will greatly affect the image's feature distribution, and make the task of underwater sonar image target classification more difficult.

At present, there are many methods for image classification. The support vector machine (SVM) could construct a hyperplane to achieve binary classification of input images. The combination of bayesian and dictionary learning can reduce the impact of noise effectively. In addition, target classification could also be achieved by adding effective features and highlighting the edge characteristics. However, most of these methods are based on optical images. Due to the difference between the sonar environment and optical environment and the complexity of the sonar environment, we need to design a classification method for sonar image. At the same time, there is an urgent need to study and solve problems for improving accuracy and speed of sonar image classification, and for reducing the model complexity. For this reason, this paper designs an AdaBoost cascade classifier for sonar image, we expect that the target framework can be applied to sonar images in different noise environments.

The AdaBoost cascade framework is an adaptive boosting algorithm based on boosting proposed by Freund and Schapire [17]. The main idea is to form a new strong classifier by combining different basic classifiers for achieving the higher accuracy based on current samples. And the weights

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

of the current base classifiers can be adjusted adaptively based on the classifier accuracy after each training. At the same time, the weights of the current samples are changed to obtain the ''attention'' of the subsequent classifier, and finally the construction of the classifier framework is finished. Although this idea has a boosting effect, it is still limited by the accuracy of the selected base classifiers, and the combination of classifiers also increases the training time. Many researchers used various methods to solve these problems. Y Honghui [18] proposed the SVM ensemble based on weighted reduced nearest neighbor (SVME-WRNN) and the SVM ensemble based on weighted immune clonal instance selection algorithm (SVME-WICISA). The method of sample selection was more accurate and reduced the impact of the number of training samples with the same classification accuracy. Wu and Nagahashi [19] added a parameter term to the loss function by analyzing the marginality of the samples, so in each training epoch, the samples near the edges had a greater chance of having positive edges. Chen and Chen [20] employed a novel cascade structure, where staged classification information and inter-stage cross reference information were used to enhance the detection performance of the cascade classifier instead of standard enhanced cascade. Experimental results showed that this method achieved higher detection accuracy and efficiency. Jie and Gongjian [21] proposed to design thresholds based on sample distribution to improve classification accuracy. Cao *et al.* [22] designed an ND-AdaBoost method, which reduced sensitivity of AdaBoost to noise and improved performance through a detection mechanism. Wanpeng *et al.* [23] proposed a robust AdaBoost classifier construction method against external point interference, introduced the ransac algorithm, and finally, selected the best classifier from all AdaBoost models to eliminate classifier degradation caused by external points. Pang *et al.* [24] proposed to iteratively divide a strong classifier into two parts until a predefined number of stages was generated. By directly minimizing the computational cost of the cascade, it searched for the best partition point of each stage and provided theoretical support to ensure the existence of a unique optimal solution. Zhang and Yang [25] introduced a single-layer neural network to optimize the threshold based on the adaptive ELM with S-shaped activation function, and also employed a grid search strategy to select the regularization parameter C from a wide range. Hu *et al.* [26] used AdaBoost method to find samples corresponding with larger weights, and removed them as possible outliers, and then retrained and redesigned the classifier model. When studying the classification of underwater target images, most of them focused on the elimination of noise points, and most of them performed the filtering of noise points during feature extraction or detection. This can easily cause useful feature information to be filtered as noise. The method proposed in this paper eliminates noise during the model training stage, and focuses on those difficult samples, which can effectively reduce the training time and improve the classification accuracy.

This paper aims to build a robust classifier suitable for sonar images and solve the problem of small sonar image classification. The contributions are as follows:

1. For the situation that there is few sonar images and a single sample, the dataset enhancement algorithm is used to create an artificial sonar dataset with a small number of images.
2. Instead of using the original decision tree, SVM is designed as a meta classifier in the AdaBoost cascade framework, while a stochastic gradient descent algorithm is used to optimize the model.
3. A new iterative rule is proposed to improve the upper limit of the classifier iteration, and the best classification accuracy is selected as the termination point, which effectively preserves and improves the decision-making degree of high-quality classifier. The training time is reduced, and the classification accuracy is improved.
4. A new updating strategy of sample weights is proposed, which adopts the method of segment processing innovatively. It can restrain the phenomenon that it is difficult to separate the hard-to-separate samples by continuously obtaining too many weights. The cascade framework with a new updating strategy of sample weights can reduce the attention of the classifier to the hard-to-separate samples, and improve the classification accuracy.

The main content of this article is arranged as follows: Section II introduces the algorithms designed in this article, the feature extraction method of HOG, support vector machine and AdaBoost classifier model, and the improvements made in this article. And in Section III, experiments are conducted to compare our method with others. The first is the comparison experiment of different classifier architectures, the second is the comparison experiment before and after the improvements, and the last is the comparison of the same type of algorithm. Finally, conclusions are drawn in Section IV.

II. CLASSIFIER DESIGN

This paper uses AdaBoost as the main framework and selects the linear SVM as the meta-classifier to replace the traditional decision tree algorithm. The HOG feature extraction method can obtain the pixel matrix distribution of the image, and can determine whether the target belongs to the same category according to the similarity of different matrices. Considering that the model may be overfitting due to a small amount of data, a stochastic gradient descent (SGD) algorithm is introduced to optimize the objective function to avoid local optimization. Figure 1 shows the classifier model of this paper.

A. LINEAR SVM

SVM is a strong classifier. We use SVM instead of weak classifier, because the cascade framework that adopts weak classifier for sonar images has low accuracy. At the same time, the new iteration rules will discard some of the better classifiers. The discard has little effect on strong classifiers. SVM aims to build a hyperplane and realizes that the

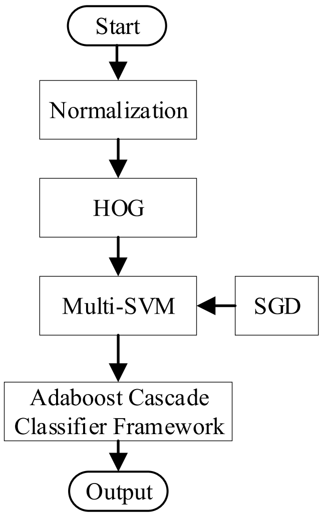

FIGURE 1. **The robust classification takes AdaBoost algorithm as the main framework. Firstly, HOG feature extraction algorithm is used to extract the pixel matrix distribution of the image, and the processed features are used for subsequent classifier training. Multiple linear SVM is selected as the meta-classifier to replace the decision tree algorithm in the cascade framework. Stochastic gradient descent algorithm is introduced to optimize the loss function to avoid local optimization of the model.**

data changes from high-dimensional non-linear indivisible to low-dimensional linearly separable. In high-dimensional space, H1 and H2 are the samples that are closest to the classification plane and are parallel to the classification plane. As shown by the blue line in the figure 2, a straight line corresponds to the blue line. The distance between the plane H1 and H2 is called the classification interval. It corresponds to the red line in the figure 2. There are many such the classification interval, we call it the optimal classification hyperplane.

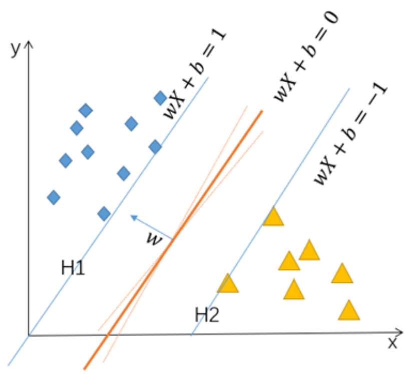

FIGURE 2. **SVM performs binary classification in two dimensions. As shown by the blue line in the figure 2, a straight line corresponds to the blue line. The distance between the plane H1 and H2 is called the classification interval. It corresponds to the red line in the figure 2. There are many such the classification interval, we call it the optimal classification hyperplane.**

Given an input data $X = \{X_1, \ldots, X_N\}$, $X_i = [x_1 \ldots x_n]$, the learning targets $y = \{y_1 \ldots y_N\}$, $y \in \{-1, +1\}$, represent positive and negative examples, where each sample of the input data contains multiple feature space, and the classification plane can be expressed as $w\ X + b = 0$, where w,b are the normal vector and intercept of the hyperplane. When the classification plane can completely predict the class of the input sample, and when separating and maximizing them, respectively. When the minimum of $\|w\|$ is satisfied, this plane is the optimal hyperplane.

In practice, it is difficult to obtain a complete dataset with many samples. This situation may cause the model to overfit, to prevent the model from overfitting, this paper introduces SGD to optimize the objective function. When both conditions are met $y_i w^T x_i \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, l$, the objective function of the linear SVM is shown as follows:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i. \tag{1}$$

In equation (1), ξ represents a relaxation variable. As a constraint, when the model is large enough, the model can meet the above conditions. And at the same time, $\sum_{i=1}^{l} \xi_i$ is a penalty term, $C(C > 0)$ is the penalty term coefficient. It could be seen that the main goal of SVM also included minimizing constraints. The loss function is a function that measures the fit between the actual and predicted values of the model. The robustness of the model increases as the loss function decreases. The loss function is the key to model convergence. After considering the constraints, the objective function of the SVM is:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{l} \max[0, 1 - y_i w^T x_i], \tag{2}$$

when a stochastic gradient optimization algorithm is used to solve the local optimum of SVM, a sample is randomly selected to update the parameters. So, the objective function of the above formula is:

$$\min \frac{1}{2} w^T w + \frac{C}{l} \max[0, 1 - y_i w^T x_i]. \tag{3}$$

Dataset learning will be very fast by using a sample gradient to update the gradient, which avoids learning the entire dataset. Gradient of the above formula can be obtained:

$$\nabla_t = w - I[y_i w^T x_i > 1] \frac{C}{l} y_i x_i, \tag{4}$$

in this formula, the second term is an indicator function, which is 1 when $y_i \omega^T x_i > 1$ is satisfied, and 0 otherwise. So, the iteration formula is:

$$w^{(t+1)} = w^t - \eta_t \nabla_t = w^t - \eta_t w + \eta I[y_i w^T x_i > 1] \frac{C}{l} y_i x_i, \tag{5}$$

in

$$\eta_t = \frac{\eta_0}{(1 + \lambda \eta_0 t)}, \ \eta_0 = 1. \tag{6}$$

where η is the stride, which determined the time of the model reach the optimal value.

The SVM method can effectively reduce the dimension of the feature matrix while maintaining high classification accuracy. Multi-classification using multiple SVMs into AdaBoost can eliminate the disadvantages of weak classifiers with low classification accuracy.

B. IMPROVED ADABOOST CASCADING FRAMEWORK

The proposed AdaBoost is to build a classifier based on the classification results of each classifier. AdaBoost generally uses the decision tree as the weak classifier. In each decision tree, different thresholds can be selected as decision points. The advantage of this framework is that it can be applied to almost all current machine learning algorithms and has improved the original accuracy. At the same time, no prior conditions are needed. The decision of different classifiers can reduce the lower limit of classifiers. The principle of AdaBoost algorithm is as follows:

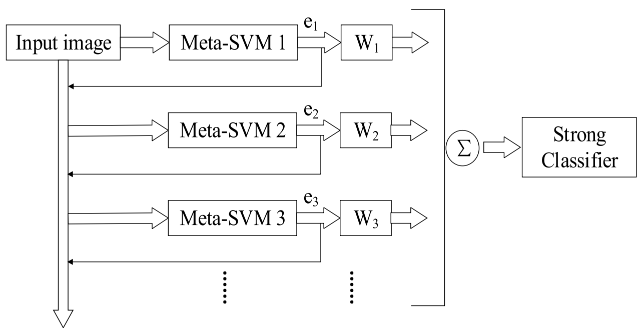

FIGURE 3. AdaBoost classification framework based on SVM. SVM is selected as the meta-classifier. The current meta-classifier can obtain errors through training, according to the weights of the classifier can be calculated and the sample distribution can be adjusted for successive training.

Firstly, the weight distribution of training samples is initialized. Each training sample is initialized with the same weight, D_1 is the weight distribution of the first training sample, representing the sample weight, and ω_i is the weight of the training sample i.

$$\begin{aligned} D_1 &= (\omega_{11}, \ldots, \omega_{1i}, \ldots, \omega_{1n}) \\ &= ({}^1/_N, \ldots, {}^1/_N), \quad i = 1, 2, \ldots, N. \end{aligned} \tag{7}$$

After that, the meta-classifier is trained repeatedly. The following step is a complete training iteration process, which is recorded as the m-th iteration, where $m = 1, 2, \ldots, M$. Using the data with weight distribution to train the meta-classifier, the trained meta-classifier is obtained as follows:

$$G_m(x) : x \to \{-1, +1\}. \tag{8}$$

Then the classification error rate e_m of the meta-classifier $G_m(x)$ on the training data is calculated as follows:

$$e_m = P(G_m(x_i) \neq y_i = \sum_{i=1}^{N} \omega_{mi} I(G_m(x_i) \neq y_i). \tag{9}$$

From equation (9) above, the error rate of classifier in the training set is equal to the sum of sample weights that is misclassified by classifier, so, the weights of the current samples increase as the accuracy of the current meta classifiers decrease, y is predictive value, the error rate in each layer is getting lower, and this reduction will become smaller in the later layers.

After a training iteration for the meta-classifiers, the weight coefficient is calculated. The original AdaBoost updates the classifier weights when the classifier error is less than 50%, and the number of iterations is preset.

Compared with the machine learning algorithm without cascade frame, multiple iterations will increase the training time, so we propose a new iterative rule. We stipulate that when the training error is zero, the given classifier weight is 1, and the training is terminated. This improvement can reduce the training time and the impact of other meta- classifiers on the final accuracy. Therefore, the improved iteration rules of classifier are expressed as follows:

$$\begin{cases} a_m = \dfrac{1}{2} \ln \dfrac{1 - e_m}{e_m}, & \dfrac{1}{e^2 + 1} < e_m \leq \dfrac{1}{2} \\ a_m = 1, & 0 < e_m \leq \dfrac{1}{e^2 + 1}, \end{cases} \tag{10}$$

where the weight coefficient a_m represents the importance of the meta-classifier $G_m(x_i)$ in the final strong classifier. The importance of the current meta-classifier increases as the accuracy of the current meta-classifiers increases in the final strong classifier. At that time, when $e_m \leq 1/2$, e is a constant, the a_m weight is constantly updated following the training process.

Finally, we need to update the weight distribution of the dataset. The purpose of this step is to get a new sample weight distribution for the next iteration. The sample weight update function before improvement is as follows:

$$\begin{cases} \omega_{m+1,i} = \dfrac{\omega_{mi}}{2Z_m} (-a_m y_i G_m(x_i))^2, & i = 1, 2, \ldots, N, |x| < 0.1 \\ \omega_{m+1,i} = \dfrac{\omega_{mi}}{Z_m} (-a_m y_i G_m(x_i)), & others, \end{cases} \tag{11}$$

where ω_{mi} is the weight of the i-th sample in the m round, and $\sum_{i=1}^{N} \omega_{mi} = 1$, and Z_m is the normalization factor:

$$Z_m = \sum_{i=1}^{N} \omega_{mi} \exp(-a_m y_i G_m(x_i)). \tag{12}$$

The samples correctly classified by the weak classifiers get a lower weight, while the samples wrongly classified get a higher weight. Because the hard-to-separate samples are continuously wrongly classified, they get a high weight. AdaBoost can focus on the hard-to-separate samples, without changing the training data, but it constantly changes the weight distribution of the training data. The final sample weight distribution DM is as follows:

$$D_{m+1} = (\omega_{m+1,1}, \omega_{m+1,2} \ldots \omega_{m+1,N}). \tag{13}$$

However, we found during the experiment that in the early stage of training, the classifier cannot find the hard-to-separate samples, and some samples may be misclassified as hard-to-separate and get the same weights as hard-to-separate samples. With the increase of the number of iterations, the hard-to-separate samples continue to be misclassified and get higher weight and get too much attention of the classifiers.

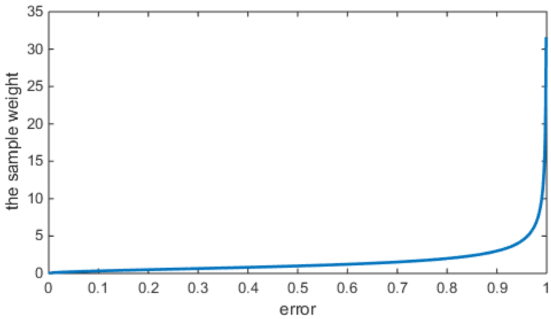

FIGURE 4. Curve of sample weight coefficient with error shows the growth curve of sample weights. The sample weight coefficient changes slowly when the classification error (misclassification rate) of the sample is between 0-0.8, and after the error reaches 0.8, the sample weights tend to be infinite, which means that the sample gets too large weight and increases the training difficulty.

To achieve high accuracy, requirements for accuracy is often above 90% in practical classification. The weights of samples before the misclassification rate of 0.8 is small, which makes it impossible to determine whether they are difficult samples in time, increases the training cost (time-consuming), and may cause the misclassification of samples by the classifier to affect the overall classification accuracy. Therefore, we are inspired by the segmentation idea of $\text{smooth}_{\text{L1}}$ function shown in (14):

$$Smooth_{L1}(x) = \begin{cases} x^2, & |x| < 1 \\ |x| - 1, & otherwise. \end{cases} \tag{14}$$

So, we design a new update function of sample weight. Equation (15) is the updated function of sample weight after improvement:

$$\begin{cases} \omega_{m+1,i} = \dfrac{\omega_{mi}}{2Z_m}(-a_m y_i G_m(x_i))^2, & i=1,2,\ldots,N, |x| \leq 0.8 \\ \omega_{m+1,i} = 2e_m + 0.4, & |x| > 0.8. \end{cases} \tag{15}$$

After the above steps, after obtaining the new sample weights, these meta-classifiers are combined linearly to achieve weighted voting, where the sum of all absolute values is not 1. The resulting strong classifier is as follows:

$$G(x) = \sum_{m=1}^{M} a_m G_m(x_i). \tag{16}$$

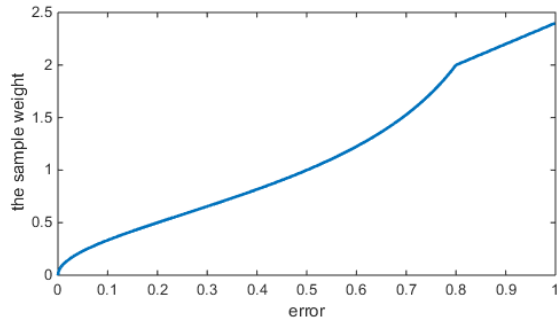

FIGURE 5. Curve of updated sample weight coefficient with error shows the change curve of sample weight coefficient after improvement. We let the weight increase slowly between 0-0.8 and change linearly between 0.8-1, so that most of the samples in the early stage can get the same attention, while in the later stage the hard-to-separate samples can be distinguished effectively.

III. EXPERIMENTS

A. DATASETS

The classification method designed in this paper uses a simple MNIST dataset to verify the model's rationality, and then uses the Cifar-10 dataset and the artificial sonar dataset to verify its effectiveness and compare it with other methods. The experimental results show that the proposed method can effectively improve the classification accuracy and have less training time. This paper uses the Cifar-10 dataset, the MNIST handwritten dataset, and the artificial sonar dataset produced by us for the comprehensive verification. The Cifar-10 dataset includes 60,000 color images of 32*32 pixels. It contains 50,000 images for training and 10,000 images for testing. Cifar-10 has a total of 10 types of labels, and each of them has 6000 pictures. These 10 types of labels are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. There is no overlap in each picture, that is, airplane only includes airplane, automobile only includes small car, and two different types of objects will not appear in the same picture.

In view of the particularity of sonar image, it is often impossible to obtain a complete and effective dataset for model verification. Therefore, in research work, it is difficult to obtain a large amount of sonar image dataset, so this paper uses a dataset enhancement algorithm to make our large number of dataset by using a small number of sonar target images. The sonar image dataset is specifically selected based on the characteristics of the sonar image: 1. We use affine, rotation and scaling to simulate different sonar imaging orientations; 2. We use two channel transformation methods to simulate the underwater environment; 3. According to the characteristics of the sonar images, different types of noise and filtering methods are added. Manually produced sonar dataset is used in conjunction with existing optical datasets to verify the effectiveness of the proposed method.

Usually the classification of sonar images is mainly aimed at the large seabed environment and relatively small artificial objects and fish, etc. Here we mainly deal with artificial objects. According to the principle of the MNIST dataset,

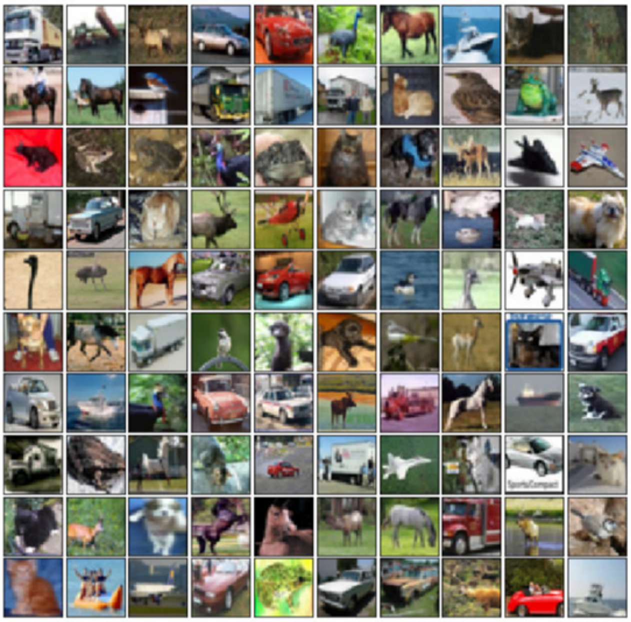

FIGURE 6. Example of 10 categories of Cifar-10 dataset. Cifar-10 dataset includes 60,000 32∗32 color images, including 50,000 images of training set and 10,000 images of test set. The CIfar-10 dataset consisted of airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

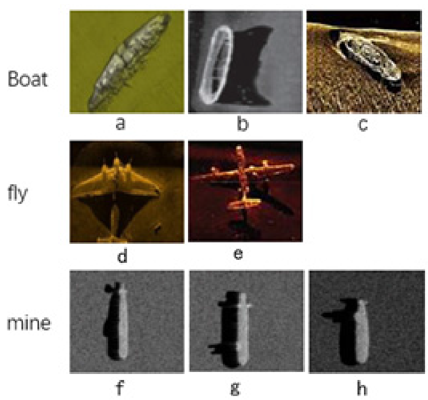

FIGURE 7. An example of an artificial sonar image dataset category. According to the production and case of MNIST dataset, we use numbers to represent different categories and add labels for samples respectively. Boat1 is number 1, boat2 is number 2, boat3 is number 3, fly1 is number 4, fly2 is number 5, mine1 is number 6, mine2 is number 7, and mine3 is number 8.

we manually produced sonar dataset which is a total of 600 images with 90∗90 pixels, including 400 training images and 200 test images. And it includes eight categories, which are labeled as boat1 (Fig.7 a), boat2 (Fig.7 b), boat3 (Fig.7 c), fly1 (Fig.7 d), fly2 (Fig.7 e), mine1 (Fig.7 f), mine2 (Fig.7 g) and mine3 (Fig.7 h).

The computational resources employed in this paper is on an Intel(R) Xeon(R) W-2133 CPU with an NVIDIA GTX1080Ti GPU in the Windows Server 2016 Standard.

B. ANALYSIS OF RESULTS

In this section, we mainly discuss two parts: one is the classification comparison between before and after the

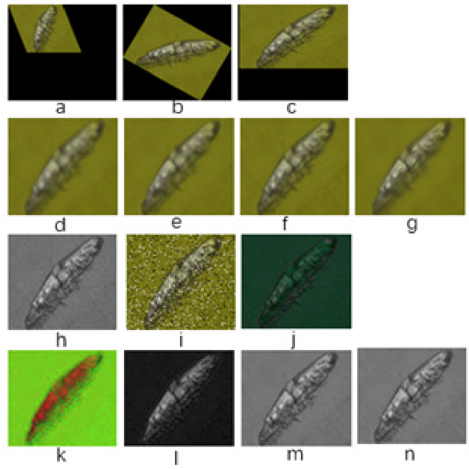

FIGURE 8. An example of the dataset enhancement method for boat1. boat1 was taken as an example to compare the processing effects of different dataset enhancement algorithms. The renderings of 4 kinds of dataset enhancement algorithms were described in fig.8. a ∼c are images obtained by the conversion algorithm, namely affine, rotation and scaling. d∼g are images with median filtering, bilateral filtering, gaussian filtering and mean filtering respectively. h∼j are the images after adding gaussian noise, additive noise and salt and pepper noise. k∼n are the images after the channel separation algorithm.

improvement of the AdaBoost algorithm; the other is the comparison of the proposed method with other similar methods on public dataset Cifar-10.

In the experiment, we select the linear kernel as the kernel function of the support vector machine. The kernel function is selected from RBF, Linear Poly and Sigmoid. Under the same conditions of comparing, the linear kernel function is optimal (the kernel function can map the original features to another high-dimensional feature space), and parameter gamma is the coefficient of the kernel function and C is the coefficient of penalty term. When the weight of the AdaBoost weak learner is SAMME.R (Real AdaBoost), the base_estimator (a parameter of weak learner) must use the classifiers that support probabilistic prediction. The SAMME (Discrete AdaBoost) algorithm does not have this limitation. The return value of Discrete AdaBoost is not a discrete type, but a real value which indicates the confidence (a probability value) of such a prediction, and the former will cause a reduction in classification accuracy by 0.046.

TABLE 1. Effect of setting different C and gamma values on classification accuracy.

	C=0.1	C=0.5	C=1	C=2
Gamma=0.1	0.844	0.813	0.875	0.859
Gamma=0.5	0.828	0.797	0.875	0.844
Gamma =1	0.828	0.844	0.875	0.844
Gamma =2	0.813	0.828	0.875	0.828
Gamma =5	0.828	0.844	0.875	0.828

Experiment is done on the artificial sonar dataset, and the results show that different C and gamma have different effects on classification accuracy under the combined conditions (Table 1). When the number of iterations is 50, the module could converges. When C is 1 and the value of the gamma coefficient is between 1-10, the best results can be obtained.

The selection of a suitable classifier for sonar data is the main content of this article. The histogram of oriented gradient algorithm which can maintain good invariance to optical and geometric, and it can well describe the edge of the image features. Unit operations can also be finished in the local binary mode, and it has significant advantages such as rotation invariance and gray invariance and can describe the information of edge and direction about the images. Therefore, experiments are performed to verify the impact of these two features on the classification results. The experimental results are shown in the following table 2.

TABLE 2. Comparative experiments on classifiers built on sonar datasets.

classifier	Accuracy	Time/s
LBP + linear SVM	0.143	17
HOG+ linear SVM	0.714	11
HOG + linear SVM + AdaBoost	0.875	19
HOG + LBP + linear SVM + AdaBoost	0.750	6.5
Algorithm proposed in this paper （Improve iteration rules）	0.890	14
Algorithm proposed in this paper (Improved iteration rules and weight functions)	0.932	13

It can be seen from table 2, when building a meta classifier, combination of LBP and SVM shows low accuracy. AdaBoost is a cascade framework based on the basic classifier, so selecting a classifier that can reliably classify data is a prerequisite. The combination of HOG and SVM shows a good advantage, and after embedding the cascade framework, the accuracy has been significantly improved. The accuracy of using the above two feature extraction methods at the same time decreased unusually. This is because different features are used for feature fusion. Different feature dimensions are obtained by different feature method. Low-dimensional feature vectors need to be zero-filled to form a new feature matrix. In this process, the influence of noise points may be increased, resulting in a decrease in accuracy.

According to the improved algorithm proposed in this paper, we have a comparison test before and after the improvement and verify it on the sonar dataset. We verify in two steps. The first step is the impact of the sample weight update function on the classification accuracy. The second step is to verify the impact of the classifier iteration rule on the training time. The training time is also a major aspect of the model. The original AdaBoost iteration rules were generated based on experience.

We set the same number of initial iterations. Before the improvement, the classification accuracy is 0.875. The time is 19 seconds. After the improvement, the time is reduced 5 seconds, which is a reduction of 26.3%. Our iterative rule can greatly reduce the training time, and at the same time, the training accuracy has also been improved, reaching 0.890. This is because after stopping the iteration, there will be fewer basic classifiers to make decision, and the accuracies of these classifiers is less than the maximum value.

The accuracy before the update function of sample weight is 0.890, and the training time is 14 seconds. After the new sample weight function, the accuracy is significantly improved, which is 4.2%. This is because too large weights of hard-to-separate samples are controlled and at the same time, the positive samples get more ‘‘attention’’ than the original algorithm. This improvement not only improves the accuracy but can shorten training time.

C. COMPARISON EXPERIMENTS WITH OTHER ALGORITHMS

In the previous section, we conduct two comparative experiments on the artificial sonar dataset to verify the effectiveness of our proposed improved algorithm. In this section, we use the public Cifar-10 dataset to compare our algorithm with other similar algorithms. The type algorithm is shown in Table 3 below.

TABLE 3. Comparison of recognition rates of common classification algorithms on Cifar-10.

classifier	Accuracy
Log. Reg+ 200 PCA Comp	0.410
KNN+30PCA	0.418
GRB + 3,072 Features	0.478
RFC/1024 + 200 PCA [27]	0.495
SVM	0.499
Algorithm proposed in this paper	0.532

As can be seen from table 3, benefited from the two improvements proposed in this article, our algorithm has certain advantages in classification accuracy compared with classic algorithms in the field of machine learning, such as logistic regression algorithms, principal component analysis, and other algorithms. This shows that the algorithm proposed in this paper is feasible for sonar image classification.

IV. CONCLUSION

This paper proposes a classifier construction method for sonar image and improves the AdaBoost cascade framework. First, a new classifier iteration rule is designed, which increases the upper limit of classifier iterations, reduces training time and improves classification accuracy; then a new sample weight update strategy is put forward, which sectional processing is used innovatively, and sample weights are adjusted correspondingly in the corresponding period by using different loss functions. The attention of hard-to-separate samples can be effectively reduced, and classification accuracy is improved. It shows our method has higher accuracy and less training time compared with other methods. Our next job is to look for more convenient methods for sonar images that needn't select different features according to different sonar images with keeping the same high accuracy and fast training time.

REFERENCES

[1] M.-C. Chuang, J.-N. Hwang, J.-H. Ye, S.-C. Huang, and K. Williams, ‘‘Underwater fish tracking for moving cameras based on deformable multiple kernels,’’ *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 9, pp. 2467–2477, Sep. 2017.

[2] D. Kim, D. Lee, H. Myung, and H.-T. Choi, "Object detection and tracking for autonomous underwater robots using weighted template matching," in *Proc. Oceans*, Yeosu, South Korea, May 2012, pp. 1–5.
[3] I. Karoui, I. Quidu, and M. Legris, "Automatic sea-surface obstacle detection and tracking in forward-looking sonar image sequences," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4661–4669, Aug. 2015.
[4] A. Renga, G. Rufino, M. D'Errico, A. Moccia, V. Boccia, M. D. Graziano, C. Aragno, and S. Zoffoli, "SAR bathymetry in the tyrrhenian sea by COSMO-SkyMed data: A novel approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 7, pp. 2834–2847, Jul. 2014.
[5] T. S. Brandes and B. Ballard, "Adaptive seabed characterization with hierarchical Bayesian modeling of SAS imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1278–1290, Mar. 2019.
[6] Y. Jiang, Y. Luo, and J. Shi, "The research on simulation of seabed reverberation model based on multi-beam sounding system," in *Proc. IEEE/OES China Ocean Acoust. (COA)*, Harbin, China, Jan. 2016, pp. 1–5.
[7] Y.-H. Chien, "The effect of a pro-rata rebate warranty on the age replacement policy with salvage value consideration," *IEEE Trans. Rel.*, vol. 59, no. 2, pp. 383–392, Jun. 2010.
[8] W.-M. Tian, "Integrated method for the detection and location of underwater pipelines," *Appl. Acoust.*, vol. 69, no. 5, pp. 387–398, May 2008.
[9] M. Jacobi and D. Karimanzira, "Multi sensor underwater pipeline tracking with AUVs," in *Proc. Oceans*, Sep. 2014, pp. 1–6.
[10] M. Guan, Y. Cheng, Q. Li, C. Wang, X. Fang, and J. Yu, "An effective method for submarine buried pipeline detection via multi-sensor data fusion," *IEEE Access*, vol. 7, pp. 125300–125309, 2019.
[11] A. Jasper, "Oil/Gas pipeline leak inspection and repair in underwater poor visibility conditions: Challenges and perspectives," *J. Environ. Protection*, vol. 3, no. 5, pp. 394–399, 2012.
[12] H. Cho, J. Gu, and S.-C. Yu, "Robust sonar-based underwater object recognition against angle-of-view variation," *IEEE Sensors J.*, vol. 16, no. 4, pp. 1013–1025, Feb. 2016.
[13] G. Marani and S. Choi, "Underwater target localization," *IEEE Robot. Autom. Mag.*, vol. 17, no. 1, pp. 64–70, Mar. 2010.
[14] Q. Zhihong and W. Xue, "An overview of anti-collision protocols for radio frequency identification devices," *China Commun.*, vol. 11, no. 11, pp. 44–59, Nov. 2014.
[15] S. Dos, "Object classification in semi structured environment using forward-looking sonar," *Sensors*, vol. 17, no. 10, pp. 22–35, Sep. 2017.
[16] Y. Guo, "Automatic classification of intracardiac tumor and thrombi in echocardiography based on sparse representation," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 2, pp. 601–611, Mar. 2015.
[17] Y. Freund and R. E. Schapire, "A decision—Theoretic generalization of on -line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*. Berlin, Germany: Springer, 1999, pp. 119–139.
[18] Y. Honghui, W. Yun, and D. Jian, "Instance selection and SVM ensembles for underwater acoustic target recognition," *J. Northwestern Polytechnical Univ.*, vol. 32, no. 3, pp. 362–367, Jun. 2014.
[19] S. Wu and H. Nagahashi, "Parameterized AdaBoost: Introducing a parameter to speed up the training of real AdaBoost," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 687–691, Jun. 2014.
[20] Y.-T. Chen and C.-S. Chen, "Fast human detection using a novel boosted cascading structure with meta stages," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1452–1464, Aug. 2008.
[21] T. Jie and W. Gongjian, "An algorithm based on improved AdaBoost classifer for object recognition," *Wireless Internet Technol.*, vol. 20, pp. 100–104, Oct. 2016.
[22] J. Cao, S. Kwong, and R. Wang, "A noise-detection based AdaBoost algorithm for mislabeled data," *Pattern Recognit.*, vol. 45, no. 12, pp. 4451–4465, Dec. 2012.
[23] C. Wanpeng, L. Yunbin, and S. Hui, "Robust AdaBoost classifier construction method against outlier interference," *Comput. Eng. Appl.*, vol. 54, no. 7, pp. 132–137, Apr. 2018.
[24] Y. Pang, J. Cao, and X. Li, "Cascade learning by optimally partitioning," 2015, *arXiv:1508.04326*. [Online]. Available: http://arxiv.org/abs/1508.04326
[25] P.-B. Zhang and Z.-X. Yang, "A novel AdaBoost framework with robust threshold and structural optimization," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 64–76, Jan. 2018.
[26] W. Hu, W. Hu, and S. Maybank, "AdaBoost-based algorithm for network intrusion detection," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 38, no. 2, pp. 577–583, Apr. 2008.
[27] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, "CIFAR-10: KNN-based ensemble of classifiers," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, New York, NY, USA, Dec. 2016, pp. 1192–1195.

HUIPU XU was born in Zhengzhou, Henan, in 1975. She received the B.S. degree in measurement and control technology and instrument from Northeastern University at Qinhuangdao, Qinhuangdao, Hebei, China, in 2000, and the M.S. degree in power electronics and power transmission, and the Ph.D. degree in control theory and control engineering from Dalian Maritime University, Dalian, Liaoning, China, in 2003 and 2009, respectively. She is currently an Assistant Professor with the College of Marine Electrical Engineering, Dalian Maritime University. Her research interests include intelligent vehicle and cooperative control of multiagent, underwater target orientation, and tracking.

HAIYAN YUAN was born in Kaifeng, Henan, in 1993. She received the B.S. degree in electrical engineering and automation from Luoyang Normal University at Luoyang, Henan, China, in 2017. She studies at Dalian Maritime University, Dalian, Liaoning, China, where she is committed to the research of underwater sonar image target classification method. Her research interests include machine learning algorithms for sonar image classification and the application of convolutional neural networks to small underwater data processing.