



A fast button surface defect detection method based on Siamese network with imbalanced samples

Songlin Wu¹ · Yubin Wu¹ · Danhua Cao¹ · Caiyun Zheng¹

Received: 12 October 2018 / Revised: 22 April 2019 / Accepted: 26 July 2019 /
Published online: 6 September 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Surface defect detection for button is a tough task because of complex surface texture, variety of defects, and limited defect samples which often leads to an imbalanced issue. Aiming at solving these problems, a similarity metric method based on Siamese network is proposed for detecting defects of button and applied in a practical machine-vision-based system. In our system, the Siamese network with a specifically designed loss function is used for automatic feature extraction and similarity metric of samples. The learning process minimizes the specific loss function, which drives intra-class distance among positives to be smaller and inter-class distance to be larger in the feature space, so that after training, defect-free samples are clustered while defect samples are mapped to outliers. The proposed method is evaluated on button datasets of multiple kinds of defects including dent, crack, stain, hole, uneven etc. Experimental results show 98% detection precision for the proposed method, and 95% detection precision when dealing with imbalance issue, indicating its advantage over conventional methods. Comparison experiments show that for our task, the proposed loss function outperforms other recent published loss functions in face recognition or ReID field. Moreover, we optimize our method with different strategies. Our method reaches 6 fps detection speed on an embedded DSP platform, indicating its potential in providing an effective approach for online detection on production.

Keywords Surface defect · Imbalanced issue · Machine-vision-based system · Siamese network

1 Introduction

Defect detection for industrial products plays a significant part in industrial manufacture. Nowadays surface inspection in practical production relies mainly on manual inspection,

✉ Danhua Cao
dhcao@hust.edu.cn

¹ School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China

which is generally subjective and inefficient. Past studies have shown the great potential of machine vision based method in taking the place of manual inspection and improving detection efficiency. In the field of automated optical inspection (AOI), vision based methods have been successfully applied to metal [13, 17, 28], textiles, fabric [1, 3], caps [32], LCD [4], components [15, 29] and other applications.

Buttons contain various defects, such as dents, holes, cracks, stains, wrong painting, fading, and uneven surface. Meanwhile, the various patterns and reflective properties of the surface pose challenges to automated detection. Therefore, an AOI system for button detection and general method for online detection are urgently needed.

Surface defects detection methods have made significant progress over the past decades. Conventional methods can be divided into segmentation based approaches [1, 13], template matching approaches [5, 9, 12], and feature-based approaches [18, 28]. However, the former two methods are mostly applied to targets with simple or regular surfaces, suffering from lack of versatility. Feature-based approaches including histogram [32], co-occurrence matrix [2], LBP [18], require manual design of feature extraction, algorithms, and selection of suitable classifiers. In recent years there has been an increasing body of researches on deep learning based approaches [13, 19, 20, 23, 25, 31]. Deep learning methods usually use end-to-end training, combining feature extraction and classification together, and are reported to have strong versatility [19]. [3, 22] proposed convolution neural network (CNN) based methods and obtained appealing results. The automatic identification of defects is the advantage of deep learning method. [14, 30] proposed deep learning methods for casting and laser melting parts, showing the advantage of deep learning in automatic defect extraction. However, deep learning methods are limited by extensive calculation and demand of expensive hardware platform. Therefore, how to implement deep learning methods on the embedded system is a tough problem, Lizhe Liu et al. [16] proposed a fast detection method for button surface defects based on CNN and reached a good performance at speed of 5 fps on DSP based smart camera, however this method required sufficient defective samples for training.

Unfortunately, negative samples are much fewer than positives in practical applications. Imbalanced samples lead to an unsatisfied performance in machine learning based methods, that is over-fitting. In this situation, the predictive model developed by conventional machine learning algorithms could be biased and inaccurate. Traditional resampling techniques increase the likelihood of over-fitting, and thus may be not very effective for high dimensional data. In Chuanxia Jian's research [11], a new sampling method is proposed to address the imbalanced defect example classification for mobile phone screen glass. This method improves the classification accuracy of the minority class [11]. However, this method suffers from a complex detection process. Fortunately, we find that similarity metric is a potential solution to the imbalanced problem, considering the similarity between positive samples and various appearance of negative samples. Siamese network is a deep learning based method used for measuring similarity among samples and extracting features automatically [7]. The main idea of Siamese network is to find a mapping function which maps input patterns into a target space such that after trained with a specific loss function, similar images are mapped close to each other in the feature space while dissimilar image pairs are mapped far from each other [26]. The similarity metric based Siamese network is effective to distinguish small differences between the image pairs and can be extended to our task.

The remainder of this paper is organized as follows. Section 2 will detail our detection process. Section 3 will show the detection method based on Siamese network. Section 4 will show results of the experiments on button dataset. Finally, section 5 is a conclusion.

2 Detection process

Our approach is to build a trainable model that nonlinearly maps the input sample images to a certain feature space. The mapping function is achieved after training the model. Figure 1 depicts the general pipeline of our method. A Siamese network architecture is used for extracting features by learning similarity. The loss layer is the key to our network, specifically designed to minimize the squared Euclidean distance between positive samples and maximize the distance between positives and negatives. The positive feature points are all clustered into a specific space, while negatives are far away from positive feature points. In order to classify the features, we use SVDD model to construct a hypersphere so that we can consider the defect point as an outlier according to the distance to the center of hyperspheres.

It is worth mentioning that we normalize the object region by converting the circular button foreground into a rectangle. To realize it, we take the center of the circle area as the base point, and convert the pixels from Cartesian coordinate (x, y) to polar coordinate (r, θ) , respectively shown as (1) and (2) in Fig. 2.

$$r_0 = \sqrt{x_0^2 + y_0^2} \tag{1}$$

$$\theta_0 = \arctan\left(\frac{y_0}{x_0}\right) \tag{2}$$

Where (x_0, y_0) is a pixel in the circle area, and (r_0, θ_0) is the corresponding pixel in the rectangle area.

The input of our network is cropped blocks of the original image. Considering that button defects vary in size, and tiny defects (less than twenty pixels) could be far smaller than the whole button image, we adopt blocking detection method to increase the ratio of the defect area to the input image

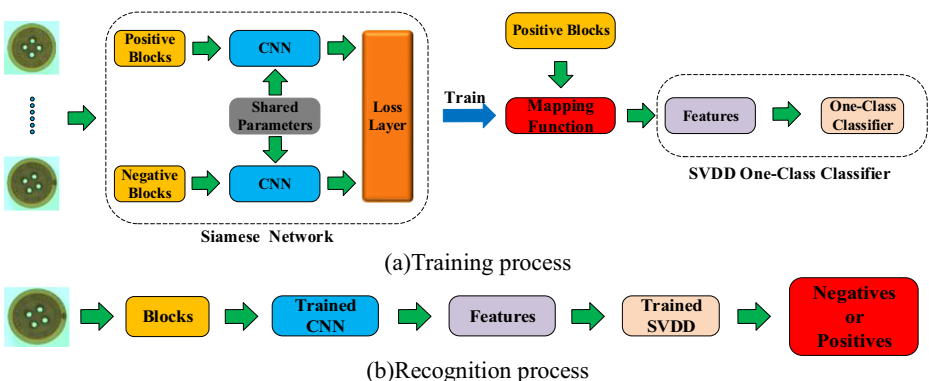


Fig. 1 Pipeline of our method. (a) Training process (b) Recognition process

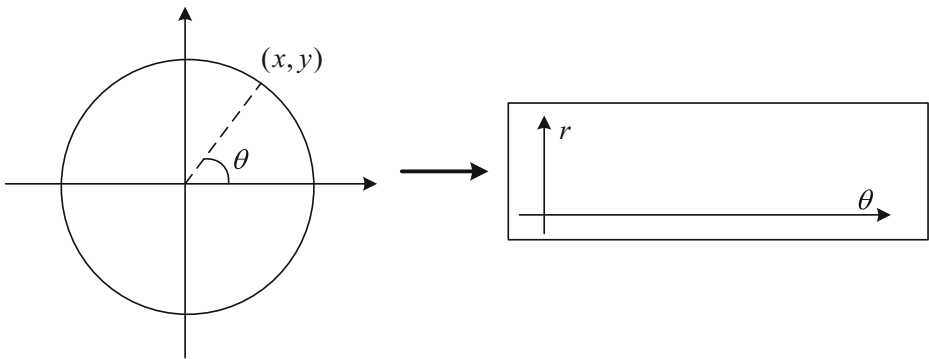


Fig. 2 The normalization of a circle area

area. We crop small blocks of size 32×32 from the normalized region image and adjacent areas have an overlap, as is shown in Fig. 3(b). After cropping the normalized button region, we will get datasets of defect-free blocks and defect blocks, as shown in Fig. 3(c) and Fig. 3(d).

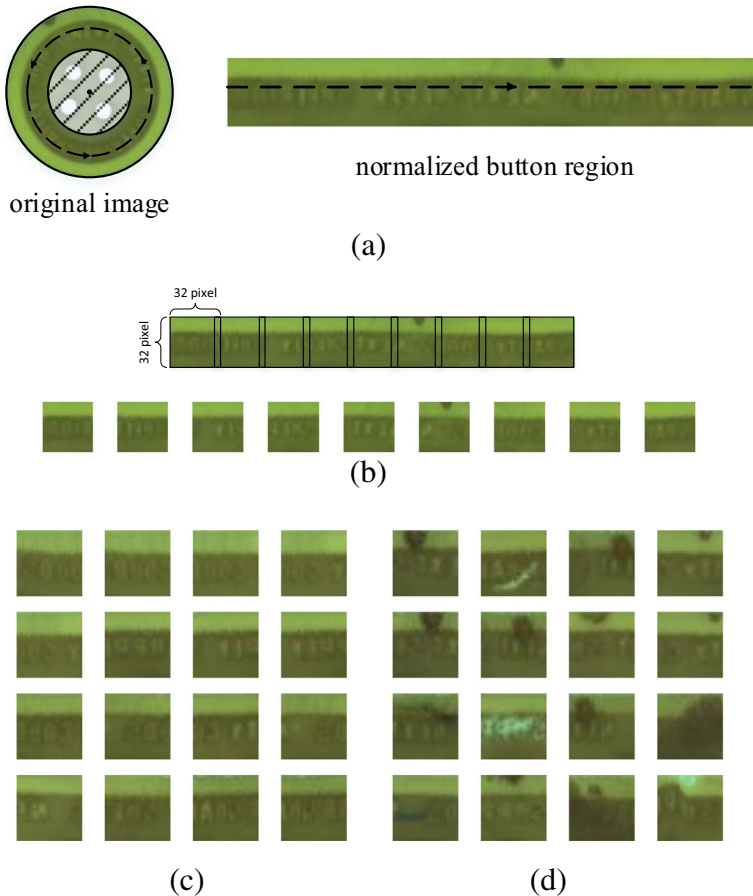


Fig. 3 Steps of cropping images. (a) Regional normalization through polar coordinate transformation. (b) Overlap cropping strategy. (c) Positive cropped images. (d) Negative cropped images

3 Detection method based on Siamese network

3.1 Feature extraction with Siamese network

The architecture of the proposed network is given in Fig. 4. The Siamese network for feature extraction consists of two identical branches with shared parameters. Each branch poses a CNN architecture. In general, images of different labels go through two branches during training and the outputs features are fed to the loss layer. The loss layer consists of the new loss function and optimizers. The loss function is represented by inter-class loss and intra-class loss. The two optimizers work for minimizing the inter-class loss and intra-class loss respectively. After training the network, we achieve the mapping function $f(x)$, which is the forward propagation of CNN. The output of CNN is the extracted features of an input image.

In order to achieve online detection, we propose a small network to ensure the detection speed. CNN has a strong ability for classification and feature extraction. Unlike CNN used for classification, the output of proposed CNN is not a label or probability but a feature vector. The proposed architecture consists of six layers, including three convolutional layers with size of 3×3 kernel, three max-pool layers and a fully connected layers. Each layer contains trainable parameters and consists of a linear transformation followed by a nonlinear mapping, which is implemented by rectified linear units (ReLU) [8] to accelerate the training process. The network takes an image blocks of size $32 \times 32 \times 3$ as input, and outputs a 120-dimensional feature vector. The detailed architecture of the network is shown in Fig. 5.

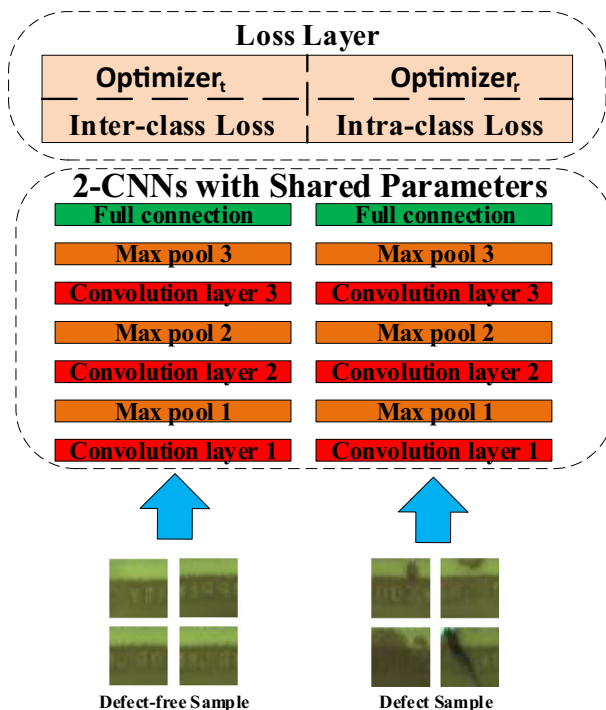


Fig. 4 Siamese network Architecture

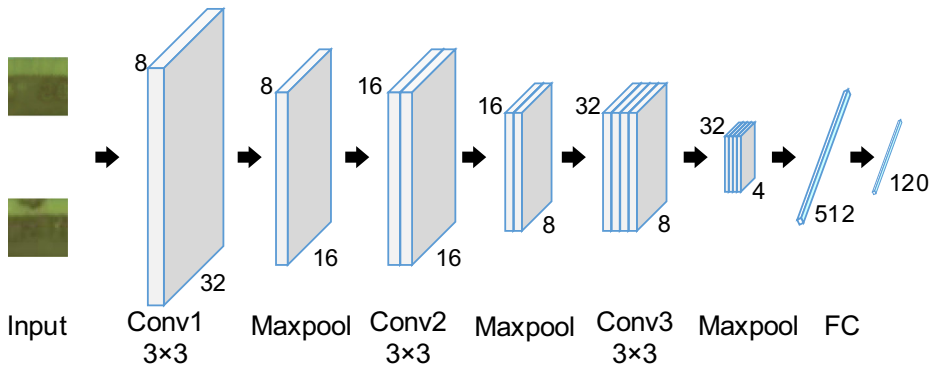


Fig. 5 CNN Architecture

3.2 Loss function

3.2.1 Related loss functions

Similarity metric has been successfully used in face recognition and ReID field, and can be extended to our task with some small changes. Since we are more concerned with cluster of positive samples, the proposed loss function is specifically designed to encourage defect-free features to be close and push defect ones far away in feature space during learning stage. We compare our loss function with other loss functions described below in defect detection task.

Triplet loss Triplet loss drives the similarity metric to be small for pairs of images from the same class, and large for pairs from different class [21]. In a standard triplet loss network, the inputs are a batch of triplet units $\{x^a, x^p, x^n\}$, where x^a is the anchor image, x^a and x^p are the same class., the loss function can be defined as:

$$L(x^a, x^p, x^n, W) = \sum \max \left\{ \|f(x^a) - f(x^p)\|_2^2 + m - \|f(x^a) - f(x^n)\|_2^2, 0 \right\} \quad (3)$$

However, how to choose proper triplet unit is a complex problem. If triplet units are easy to distinguish, the triplet loss will rapidly decrease to 0, leading to the ineffectiveness of network training. Generating all possible triplets would result in numerous triplets. Meanwhile, the triplet loss function is quite sensitive to the selection of anchor point, which means improper anchors can result in great interference in the training stage and lead to a slow convergence.

Batch hard triplet loss(BHTL) Batch hard triplet loss was proposed for mining hard triplets, first applied in person ReID subfield [10]. As the dataset gets larger, the possible number of triplets grows cubically, however most of possible triplets are easy triplets which are less useful for training. Thus mining hard triplets is crucial for learning. The hardest positive and the hardest negative samples within the batch are selected when forming the triplets for computing

the loss. In our task, we choose a certain set of N triplets units, their images are stacked into a batch of size $3N$. Batch hard triplet loss is defined as:

$$L_{BH} = \sum_{a=1}^N \left[m + \max_{p=1\dots N} D(f(x_a), f(x_p)) - \min_{n=1\dots N} D(f(x_a), f(x_n)) \right] \tag{4}$$

where x_a corresponds to the anchor sample. x_p and x_n correspond to the farthest positive and the nearest negative samples of x_a , respectively, $\langle x_a, x_p, x_n \rangle$ is the hardest triplets for the anchor a . There are N hardest triplets of a mini batch. They can be considered moderate triplets, since they are the hardest within a small subset of the data, which is exactly best for learning with the triplet loss [10].

Margin sample mining loss(MSML) Triplet loss and batch hard triplet loss only consider relative distances between positive and negative pairs. Margin sample mining loss(MSML) is a metric learning loss for ReID [27], which not only considers the relative distances but also considers absolute distances between positive and negative pairs. It picks the most dissimilar positive pairs and the most similar negative pair in the whole batch. In our task, MSML is defined as:

$$L_{msml} = \max \left(\max_{A,A'} \left(\|f_A - f_{A'}\|_2 - \min_{C,B} (\|f_C - f_B\|_2) + m, 0 \right) \right) \tag{5}$$

where A and A' are the positive samples, B and C belong to different identities. Subscript A and A' denote the hardest positive pairs, and subscript B and C denote the most difficult pairs to be distinguished.

3.2.2 Proposed loss function

Aiming at defect detection task, we design a new loss function to cluster positive samples. The purpose of proposed loss function is to guide the learning process so that after training, positive points should locate in a hypersphere in the d -dimensional feature space. Our network adopts mini batch training.

It is noted to mention that we use the cluster center to take place of the anchor point in triplet loss. Given a mini batch $X^P = \{x_1^P, x_2^P, \dots, x_n^P\}$ containing n images of positive block images, the cluster center is defined as

$$C^p = \frac{1}{n} \sum_{i=1}^n f(x_i^P) \tag{6}$$

Where $\{f(x_1^P), f(x_2^P), \dots, f(x_n^P)\}$ is feature vector acquired through CNN.

We use Euclidean distance to measure similarity of feature points. In order to improve the ability to predict unknown defect samples, we consider not only relative distance but also absolute distance. We proposed a concentric-circles model with two margin parameters predefined as α_1 and α_2 . α_1 represents the minimum margin between negative points and positive points and α_2 represents the maximum margin between positive points and negative points.

Batch hard mining is used in our model. Different with triplet loss, BHTL, and MSML, we constrain the positive points near the center point instead of anchor leading to a smaller intra-variance. Given mini batches $X^P = \{x_1^P, x_2^P, \dots, x_n^P\}$ and $X^N = \{x_1^N, x_2^N, \dots, x_n^N\}$, the inter-

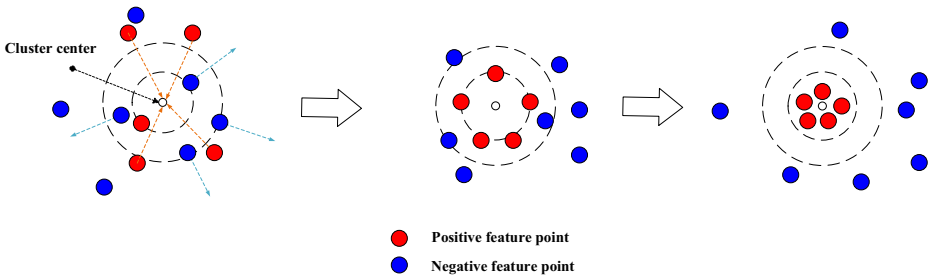


Fig. 6 The illustration of clustering at different phase of training

class loss $L_t(W)$ and intra-class loss $L_r(W)$ are defined as

$$L_t(W) = \sum_{i=1}^n \frac{1}{2} \max\{0, \|f(x_*^P) - C^P\|_2^2 - \|f(x_i^N) - C^P\|_2^2 + \alpha_1\} \tag{7}$$

$$L_r(W) = \sum_{i=1}^n \frac{1}{2} \max\{0, \|f(x_i^P) - C^P\|_2^2 - \alpha_2\} \tag{8}$$

where x_*^P is the farthest positive point to the center.

The total loss is defined as

$$L_s(W) = L_t(W) + L_r(W) \tag{9}$$

For every mini batch, the training process consists of two steps: minimization of the inter-class loss and minimization of the intra-class loss.

For the first step, if $L_t(W) \leq 0$, which means that the distance of negative point set is far enough to positive point set. The partial derivative of both the positive and negative features are 0. Otherwise the partial derivative of the farthest positive features is defined as

$$\frac{\partial L_t(W)}{\partial f(x_*^P)} = f(x_*^P) - C^P \tag{10}$$

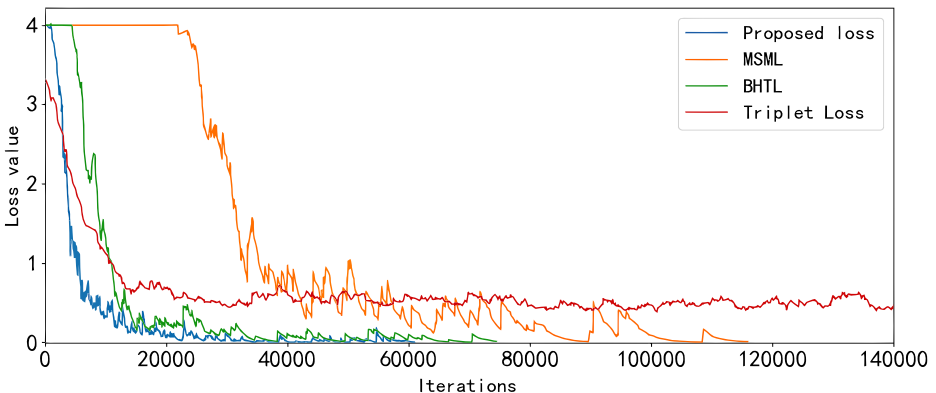


Fig. 7 The loss curve of four methods

The partial derivative of the negative features is defined as

$$\frac{\partial L_r(W)}{\partial f(x_i^N)} = C^P - f(x_i^N) \quad (11)$$

For the second steps, if $L_r(W) \leq 0$, the distance of positive point set is close enough to the cluster center. The partial derivative of positive features is 0. Other-wise, the partial derivative of the positive features is defined as

$$\frac{\partial L_r(W)}{\partial f(x_i^P)} = f(x_i^P) - C^P \quad (12)$$

Figure 6 illustrates the ideal situation. The dashed circle outside represents the margin of negative samples and the dashed circle inside represents the margin of positives. The red dots and blue dots correspond to high dimensional features of positive blocks and negative blocks. At the beginning of the training phase, the value of loss function will be large because of the random initialization of weights. The positive points approach the center point gradually and the negative points are given penalty if it is mapped inside the hypersphere. After training, we will obtain the mapping function through the forward propagation of CNN.

In this paper, the anchor is replaced by center of positive features, more samples involved in the calculation. Comparison experiment shows that our method can accelerate training convergence. Figure 7 shows the loss curve of proposed network and other three loss functions mentioned above in the training phase. Obviously, BHTL, MSML and proposed loss function converge faster than the triplet loss. The network trained with BHTL sometimes cannot effectively separate the negative samples from positives at the beginning, in that the distance from positive samples to the anchor may be farther than that from negatives to the anchor, as a result of random initialization of weights. Therefore, the loss of BHTL fails to drop in the beginning until the network pushes the negative points far enough away. Similarly, MSML also exists this phenomenon. For our loss function, we use cluster center to take place of anchor, which makes the training phase faster and more stable. On the intel i7 7700k and GTX1050ti platforms, the training time of our method is about 20 min, outperforming MSML(45mins), BHTL(30 min) and triplet loss(120 min).

3.3 SVDD one-class classifier

After training the Siamese network, we obtain the mapping function $f(x_i^l)$ from the input image to the feature space. Positive samples are clustered in the feature space, while negative samples can be seen as outliers. Therefore, the detection task can be taken as outlier detection. In fact, we can construct a hypersphere based on a center point and radius, where the center is the mean of positive points and the radius is a threshold, such that the point outside the hypersphere is considered as an outlier. However, the threshold is hard to set, and some positive outliers will influence the calculation of the center. SVDD is a method for outlier detection, widely applied to various fields for its strength in learning without any a priori knowledge on the distribution of dataset. SVDD attempts to find a hypersphere that best describes the region of the feature space in which a set of data points lie [24]. The hypersphere is only affected by support vectors. SVDD introduces kernel function, which makes the classifier more flexible. The objective function of SVDD is

$$\max_{R, \alpha, \zeta} R^2 + C \sum_{i=1}^l \xi_i \quad \text{subject to } \|\phi(x_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i, i = 1, 2, 3, \dots, l, \quad \xi_i \geq 0, i = 1, 2, 3, \dots, l \quad (13)$$

Where ϕ is kernel function, x_i is the training data, C is the user-specified parameter. After (8) is solved, a hypersphere is characterized by the center \mathbf{a} and the radius R , ξ is the distance between instance i to center.

The problem (8), as is often referred to as the primal problem, is difficult to solve. we can change the primal problem to dual problem using Lagrange multipliers.

$$\max_{\alpha} \sum_{i=1}^l \alpha_i \phi_{i,i} - \mathbf{a}^T \phi \mathbf{a} \quad \text{subject to } \sum_{i=1}^l \alpha_i = 1, 0 \leq \alpha_i \leq C \quad (14)$$

The dual problem (9) takes a similar form as the SVM dual problem. Therefore, existing optimization methods such as decomposition methods for SVM dual problems can be easily applied to SVDD problem.

For any test instance \mathbf{x} , we must check the value

$$\|\phi(\mathbf{x}) - \mathbf{a}\|^2 - R^2 = K(\mathbf{x}, \mathbf{x}) - 2 \sum_{i: \alpha_i > 0} \alpha_i K(\mathbf{x}, \mathbf{x}_i) - R^2 + C \quad (15)$$

Where the samples \mathbf{x}_i for which $\alpha_i \neq 0$ are called support vectors. K is the kernel function. C is independent from the test instances, and can be stored after solving the dual problem.

If (10) is positive, \mathbf{x} is considered as an outlier, the corresponding block is defective.

4 Experiments and discussion

In order to verify the effectiveness of the proposed method, an experimental physical setup has been built. The physical setup shown in Fig. 8 is used for capturing the images of button surface. The vision module consists of a camera based on DSP and an integrating sphere light source. The button images are captured by the DSP based camera triggered by the signal come from the trigger sensor. Then, the camera runs the detection algorithm and controls the sorting

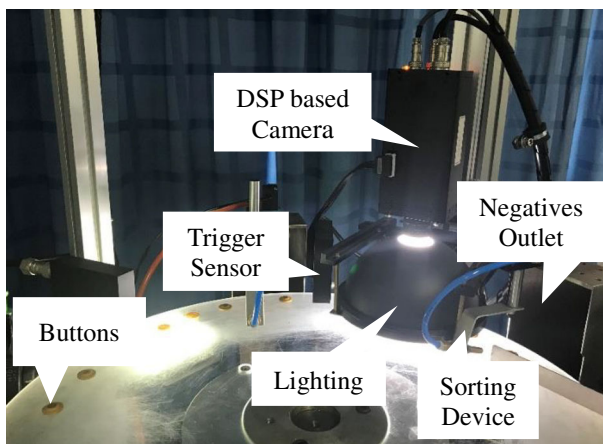



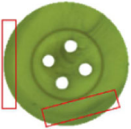






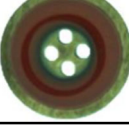
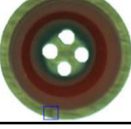


Fig. 8 System for detection

Table 1 Typical image of datasets and defect type

Sample	Defect-free	Defect	Defect type	Button features
1			stain, hole, wrong paint	character, random patterns
2			stain, uneven, crack	random patterns
3			scratch, crack	mixed colors
4			stain, scratch, dent	pure color, reflective
5			stain	reflective, transparent
6			bubble, dent, color defect, scratch	strong reflection

device according to the detection results. The sorting device is a solenoid valve with the conduit connected to an air compressor. The negative samples will be blown to outlet for negatives, while positive samples will leave the transfer platform through the outlet for positives and move on to next detection link.

4.1 Button datasets

We evaluate the proposed method on images of six kind of samples photo-graphed by ourselves and not published. The original image is $640 \times 480 \times 3$ acquired by a CCD camera.

Table 2 Numbers of test buttons

sample	1	2	3	4	5	6
defect-free number	213	211	201	396	477	174
defective number	142	196	131	134	276	188

Table 3 Cooperation methods

Methods	Batch size	Margin parameters	Feature classifier
Triplet loss	1	2	SVDD
BHTL	16	2	SVDD
MSML	16	2	SVDD
Proposed method	16	Inter-class margin: 2 Intra-class margin: 1	SVDD

Table 1 gives the details of the datasets including typical defect types that may occur in production. The numbers of each kind of test samples are listed in Table 2.

4.2 Performance evaluation with enough samples

In this section, we evaluate the performance of our method based on true positive recognition (TPR) rate, true negative recognition (TNR) rate, and recognition (R) rate.

$$TPR = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{16}$$

$$TNR = \frac{N_{TN}}{N_{TN} + N_{FN}} \tag{17}$$

$$R = \frac{TPR + TNR}{2} \tag{18}$$

where N_{TP} is the number of correctly detected positive samples, N_{FP} is the number of falsely detected positive samples, N_{TN} is the number of correctly detected negative samples, N_{FN} is the number of falsely detected negative samples.

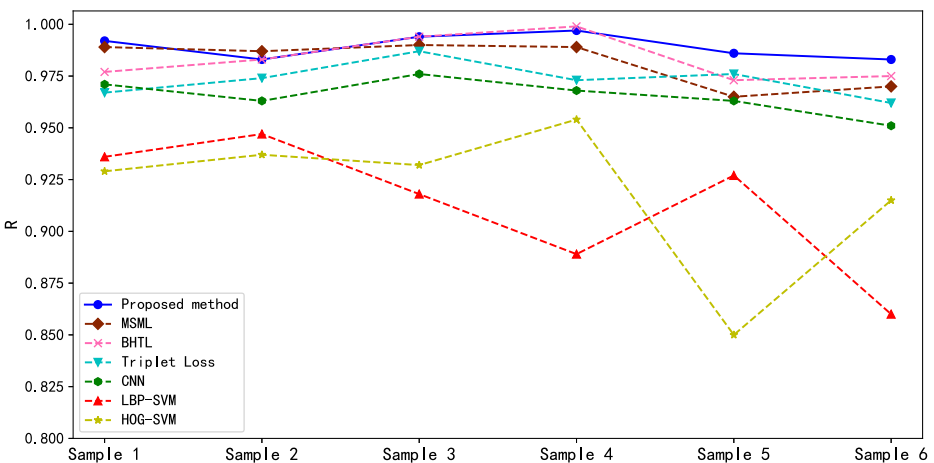


Fig. 9 Results of 7 methods on button datasets

Table 4 Details of the test result

sample	HOG-SVM			LBP-SVM			CNN			Triple Loss			BHTL			MSML			Proposed Method			
	TPR	TNR	R	TPR	TNR	R	TPR	TNR	R	TPR	TNR	R	TPR	TNR	R	TPR	TNR	R	TPR	TNR	R	
1	0.943	0.915	0.929	0.934	0.937	0.936	0.976	0.965	0.971	0.972	0.965	0.967	0.968	0.986	0.986	0.977	0.991	0.986	0.989	0.991	0.993	0.992
2	0.930	0.944	0.937	0.955	0.939	0.947	0.962	0.964	0.963	0.978	0.969	0.974	0.986	0.980	0.983	0.983	0.986	0.987	0.987	0.986	0.980	0.983
3	0.917	0.946	0.932	0.935	0.900	0.918	0.975	0.977	0.976	0.985	0.989	0.987	0.987	1.00	0.994	0.987	1.00	0.990	0.990	0.995	0.992	0.994
4	0.976	0.931	0.954	0.929	0.828	0.889	0.972	0.963	0.968	0.975	0.970	0.973	0.998	1.00	0.999	0.992	0.985	0.989	0.989	0.994	1.00	0.997
5	0.750	0.949	0.850	0.912	0.942	0.927	0.952	0.973	0.963	0.989	0.964	0.976	0.971	0.975	0.973	0.977	0.953	0.965	0.965	0.979	0.993	0.986
6	0.925	0.904	0.915	0.799	0.920	0.860	0.943	0.958	0.951	0.972	0.952	0.962	0.966	0.984	0.975	0.972	0.968	0.970	0.982	0.984	0.984	0.983

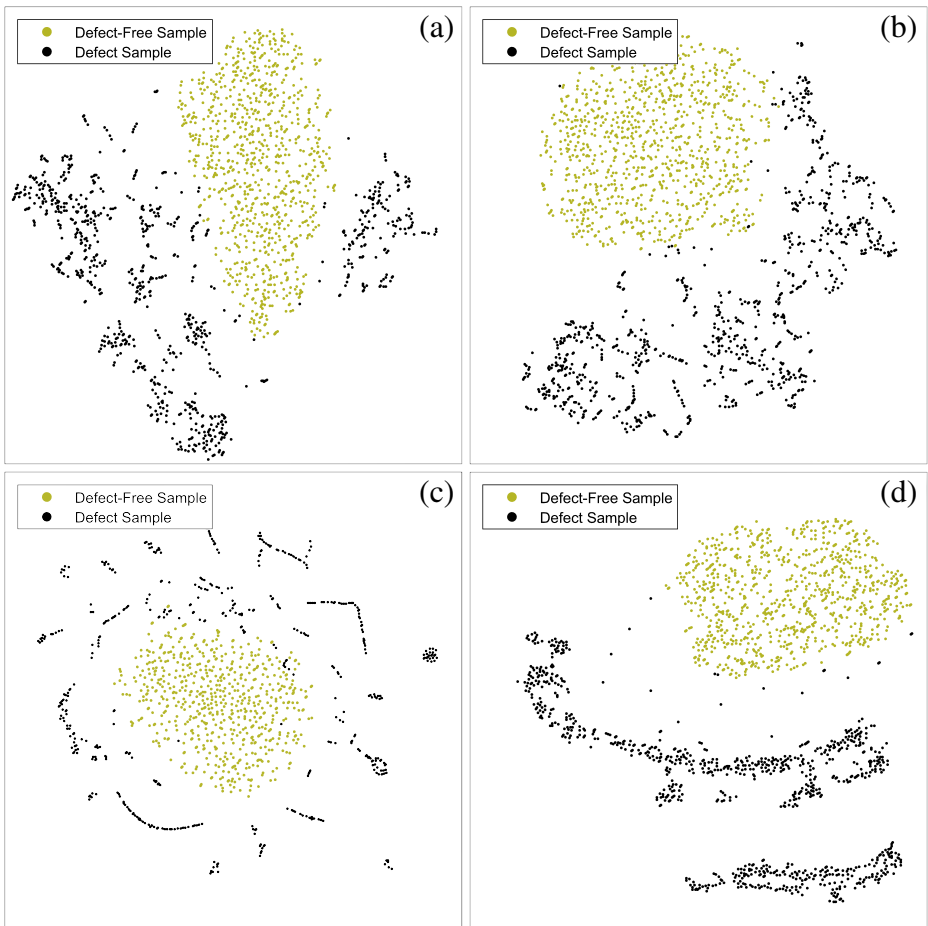


Fig. 10 Visualization of the samples 6. (a) Triplet loss. (b) Batch hard triplet loss. (c) Margin sample mining loss. (d) Proposed loss

The training set for each kind of samples consists of about 8000 positive blocks and 6000 negative blocks cropped from 200 positive samples and 200 negative samples using the method mentioned in section 2.

We compare our method with other methods as follows.

- SVM with HOG features.
- SVM with LBP features.
- CNN for classification. The CNN model is similar to the architecture used in the Siamese network above except that the CNN has an additional two neurons output layer fully connected to the last layer.
- Similarity metric methods including triplet loss, BHTL, MSML and proposed method. Table 3 shows the details of four methods.

The R value for all methods are shown in Fig. 9 and details of results are shown in Table 4.

Table 5 Result with different ratio between defect-free and defect

Sample 1	CNN		Triplet		BHTL		MSML		Proposed	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
4:1	0.986	0.943	0.986	0.965	0.977	0.971	0.986	0.979	0.986	0.986
10:1	0.991	0.915	0.991	0.951	0.986	0.944	0.986	0.958	0.971	0.981
20:1	0.991	0.880	0.991	0.930	0.986	0.923	0.977	0.951	0.986	0.971
40:1	0.995	0.845	0.981	0.916	0.981	0.894	0.977	0.930	0.986	0.958
Sample 2	CNN		Triplet		BHTL		MSML		Proposed	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
4:1	0.962	0.939	0.986	0.964	0.986	0.974	0.978	0.987	0.981	0.980
10:1	0.976	0.872	0.986	0.959	0.986	0.969	0.978	0.979	0.976	0.979
20:1	0.981	0.855	0.986	0.944	0.986	0.964	0.978	0.979	0.976	0.974
40:1	0.986	0.786	0.978	0.929	0.986	0.954	0.971	0.969	0.981	0.964
Sample 3	CNN		Triplet		BHTL		MSML		Proposed	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
4:1	0.970	0.954	0.985	0.985	0.985	0.992	0.985	0.992	0.985	0.992
10:1	0.985	0.924	0.985	0.969	0.980	0.985	0.975	0.985	0.985	0.977
20:1	0.985	0.908	0.980	0.954	0.975	0.969	0.975	0.977	0.975	0.969
40:1	0.995	0.893	0.980	0.939	0.975	0.962	0.970	0.969	0.970	0.962
Sample 4	CNN		Triplet		BHTL		MSML		Proposed	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
4:1	0.972	0.954	0.990	0.978	0.990	0.993	0.990	0.985	0.985	0.993
10:1	0.979	0.918	0.980	0.970	0.985	0.978	0.985	0.978	0.985	0.985
20:1	0.985	0.903	0.985	0.955	0.980	0.970	0.989	0.963	0.989	0.978
40:1	1.000	0.866	0.985	0.940	0.980	0.955	0.980	0.948	0.985	0.963
Sample 5	CNN		Triplet		BHTL		MSML		Proposed	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
4:1	0.960	0.953	0.981	0.953	0.979	0.971	0.964	0.975	0.985	0.985
10:1	0.966	0.928	0.986	0.942	0.971	0.964	0.979	0.971	0.986	0.978
20:1	0.984	0.891	0.981	0.927	0.971	0.946	0.971	0.957	0.979	0.960
40:1	0.986	0.851	0.986	0.902	0.971	0.921	0.971	0.931	0.979	0.953
Sample 6	CNN		Triplet		BHTL		MSML		Proposed	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
4:1	0.981	0.947	0.981	0.941	0.989	0.979	0.989	0.957	0.989	0.973
10:1	0.977	0.894	0.986	0.931	0.971	0.973	0.977	0.947	0.989	0.968
20:1	0.989	0.856	0.981	0.915	0.971	0.957	0.983	0.931	0.977	0.968
40:1	0.989	0.824	0.989	0.894	0.971	0.926	0.983	0.904	0.971	0.952

From Fig. 9 we can see that deep learning based methods are the top performers. In a contrast, conventional methods suffer from limitation in generalization, with a clearly weaker ability in dealing with buttons with random pattern, strong reflection or transparency.

From Table 4 we can see that the proposed method achieves better performances on almost all the experimental sample sets, except sample 2 and sample 4 where our method is slightly lower in accuracy than MSML and BHTL. More specifically, our loss function improves accuracies for button samples characterized by random patterns (sample 1) and strong reflection (sample 5 and 6). Our method focusses on intra-class distance, therefore for sample 1,5 and 6, the network will cluster the positive samples regardless of the various distribution of illumination or random patterns. In general, the proposed loss function is more effective, especially for the transparent and reflective samples.

We visualize distribution of sample 6 with features extracted by different loss functions in Fig. 10. Sample 6 is a typical example because of strong reflective surface and a variety of defects. The visualization of high dimensional data is realized by t-SNE method.

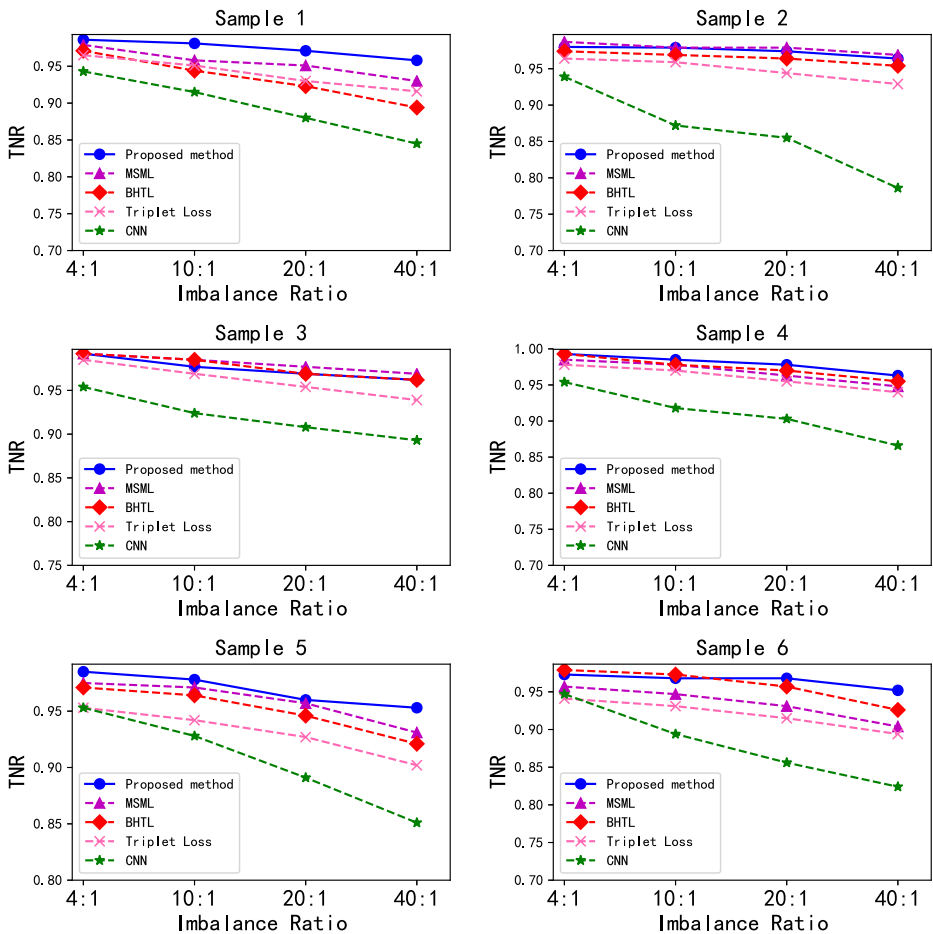


Fig. 11 Result of TNR with reduction of defect samples

From Fig. 10 we can see that all the loss functions above work in consistence with our assumption that positive samples cluster within a certain space and the negatives scatter somewhere. Moreover, Fig. 10 demonstrates that the features extracted by our method have higher cohesion and lower coupling distribution. This is because in our work, we constrain the positive points near the center point leading to a small intra-variance.

4.3 Performance evaluation with limited defect samples

In practical applications, defect samples are much fewer than defect-free samples. To explore the potential robustness of the methods and simulate the case of varying degrees of scarcity of defect samples, we select four ratios between defect-free samples and defect samples in the training set as 4:1, 10:1, 20:1 and 40:1 by reducing the number of defective samples to train our model. We evaluate the results by TPR and TNR and the details are shown in Table 4.

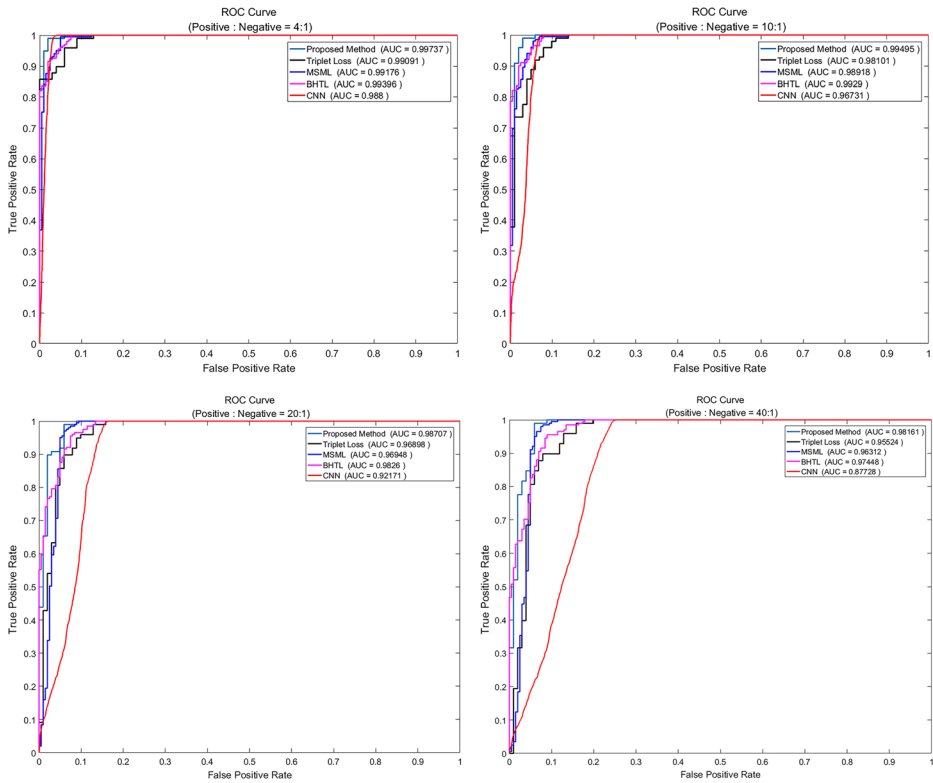


Fig. 12 ROC curve of sample 6

As we can see in Table 5 and Fig. 11, the reduction of defect sample has small effect on TPR, while the TNR value has dropped to varying degrees as the imbalance ratio increases. Therefore, in the following discussion we mainly focus on the TNR value.

The performances of similarity metric based methods are relatively stable compared to that of CNN with cross entropy loss, which drops rapidly even to the extent of losing efficacy on detection when dealing with various defect types. In terms of the overall ability in resistance to the change of imbalance ratio, our method performs better for the consideration of both inter-class distance and intra-class distance, particularly for sample 5 and 6 with complicated surface for their strong reflective characteristic and various defects. While the TNR of Triplet, BHTL, MSML decreases rapidly as the imbalance ratio changes, our method remains a high precision over 95% throughout.

Table 6 Details of time consuming

Operation	Strategies	Time consuming	Speed enhance
op1		2693 ms	1.0
op2	Strategy 1	408 ms	6.6
op3	Strategies 1 + 2	301 ms	8.9
op4	Strategies 1 + 2 + 3 + 4	258 ms	10.4
op5	Strategies 1 + 2 + 3 + 4 + 5 + 6	156 ms	17.3

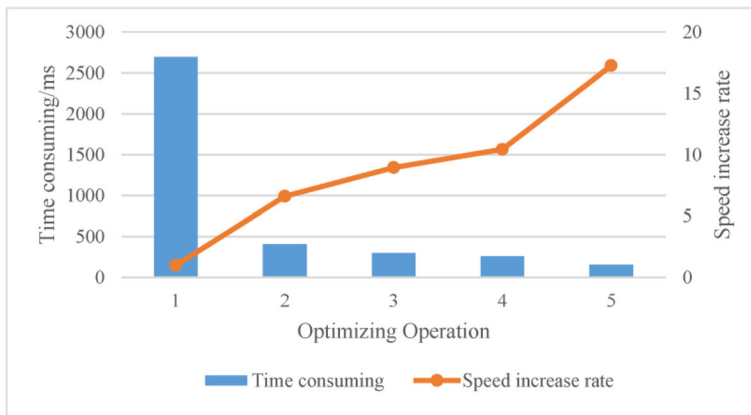


Fig. 13 The influence of different optimization operations on the running time

Taking sample 6 as an example, we evaluate the results by ROC and AUC, as is shown in Fig. 12. Our method has a larger AUC value than other methods, and with the reduction of defect samples, the ROC curve changes within a small extent.

We conclude that our method can deal with the detection task with fewer defect samples, and it is more robust in our task. However, defect samples are still necessary. If the defect samples are too rare, it is still difficult to detect defects of complicated surface. In this case, the defect detection task remains to be researched on in the future.

4.4 Online detection in DSP

CNN requires a huge amount of calculation, and is usually implemented on GPU platform. For online detection on production, detection based on the embedded machine vision system has been widely used. However, it is hard to run a CNN model on embedded platform without optimizing, because of a large sum of floating-point arithmetic, which has become the bottleneck of online detection. Therefore, we optimize the calculation in the embedded system. In this paper, the embedded platform is TMS320DM6437, which is a fixed-point DSP with the performance of up to 5600 million instructions per second (MIPS) at a clock rate of 594 MHz.

To accelerate computation speed, several strategies are used for optimizing, which are as follows.

- Strategy 1: Converting floating-point data into fixed-point data, enabling SIMD (Single instruction multiple data) instructions.
- Strategy 2: Convert convolution to matrix multiplication [6].
- Strategy 3: Using proper compiler options.

Table 7 Time consuming of each processing phase

	Preprocess	Feature extraction	SVDD classifier	Total
Before optimizing	353 ms	2232 ms	108 ms	2693 ms
After optimizing.	60 ms	72 ms	27 ms	159 ms

- Strategy 4: Using some keywords like restrict and const.
- Strategy 5: Using RAM and Cache.
- Strategy 6: Using intrinsic operations and optimized libraries.

Table 6 and Fig. 13 show the impact of different optimization strategies on the speed of the algorithm. Strategy 1 has the greatest increase in speed. DM6437 is a fix-point processor so that fixed-point operations are much faster than floating-point operations. Converting floating-point data to fixed-point data is conducive to the software pipeline and SIMD instruction.

After optimizing, the time consuming of each computing unit is as shown in Table 7.

Our final detection speed is 6 fps, 17.3 times faster than that before optimization, which is applicable for online case.

5 Conclusion

In this paper, we propose a Siamese Network based method to deal with button surface defect detection problem with imbalance issue caused by limited defect samples. In the system, we design a new loss function for the network and use SVDD method for classification. This method is implemented in a practical machine-vision-based system. Our method achieves appealing performance in detection of multiple types of defects, even with limited defect samples. On an embedded DSP platform with 594 MHz frequency, this method can reach the detection speed of 6 fps, which guarantees online detection.

Although the results of the proposed method are only demonstrated for the button, the method can potentially be used for other products.

References

1. Cao J, Zhang J, Wen Z (2017) Fabric defect inspection using prior knowledge guided least squares regression. *Multimed Tools Appl* 76(3):4141–4157
2. Capizzi G, Sciuto GL, Napoli C, Tramontana E, Woźniak M (2015) Automatic classification of fruit defects based on co-occurrence matrix and neural networks. In: *Computer science and information systems (FedCSIS), 2015 federated conference on*. IEEE, pp 861–867
3. Çelik H, Dülger L, Topalbekiroğlu M (2014) Development of a machine vision system: real-time fabric defect detection and classification with neural networks. *The Journal of The Textile Institute* 105(6):575–585
4. Cen Y-G, Zhao R-Z, Cen L-H, Cui L-H, Miao Z-J, Wei Z (2015) Defect inspection for TFT-LCD images based on the low-rank matrix reconstruction. *Neurocomputing* 149:1206–1215
5. Chauhan APS, Bhardwaj SC (2011) Detection of bare PCB defects by image subtraction method using machine vision. In: *Proceedings of the world congress on engineering*. pp 6–8
6. Chellapilla K, Puri S, Simard P (2006) High performance convolutional neural networks for document processing. In: *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft
7. Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: *computer vision and pattern recognition. CVPR 2005. IEEE computer society conference on*, 2005. IEEE, pp 539–546
8. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. pp 315–323
9. Han Y, Wu Y, Cao D, Yun P (2017) Defect detection on button surfaces with the weighted least-squares model. *Frontiers of Optoelectronics* 10(2):151–159
10. Hermans A, Beyer L, Leibe B (2017) In defense of the triplet loss for person re-identification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*

11. Jian C, Gao J, Ao Y (2017) Imbalanced defect classification for mobile phone screen glass using multifractal features and a new sampling method. *Multimed Tools Appl* 76(22):24413–24434
12. Jiang C, Quan Y, Lin X (2016) Defect detection of capacitive touch panel using a nonnegative matrix factorization and tolerance model. *Appl Opt* 55:2331–2338
13. Li W-b, Lu C-h, J-c Z (2013) A lower envelope weber contrast detection algorithm for steel bar surface pit defects. *Opt Laser Technol* 45:654–659
14. Lin J, Yao Y, Ma L, Wang Y (2018) Detection of a casting defect tracked by deep convolution neural network. *Int J Adv Manuf Technol*:1–9
15. Liu Y, Yu F (2014) Automatic inspection system of surface defects on optical IR-CUT filter based on machine vision. *Opt Lasers Eng* 55:243–257
16. Liu L, Cao D, Wu S, Wu Y, Wei T (2018) A fast button surface defects detection method based on convolutional neural network. In: 2017 international conference on optical instruments and technology: optoelectronic measurement technology and systems. International Society for Optics and Photonics, p 1062107
17. Natarajan V, Hung T-Y, Vaikundam S, Chia L-T (2017) Convolutional networks for vot-ing-based anomaly classification in metal surface inspection. In: Industrial Technology (ICIT), 2017 IEEE International Conference on IEEE, pp 986–991
18. Park Y, Kweon IS (2016) Ambiguous surface defect image classification of AMOLED displays in smartphones. *IEEE Transactions on Industrial Informatics* 12(2):597–607
19. Ren R, Hung T, Tan KC (2018) A generic deep-learning-based approach for automated surface inspection. *IEEE transactions on cybernetics* 48(3):929–940
20. Saimurugan M, Ramachandran K, Sugumaran V, Sakthivel N (2011) Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine. *Expert Syst Appl* 38(4): 3819–3826
21. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: *Proc IEEE Conf Comput Vis Pattern Recognit* pp 815–823
22. Shanmugamani R, Sadique M, Ramamoorthy B (2015) Detection and classification of surface defects of gun barrels using computer vision and machine learning. *Measurement* 60:222–230
23. Tajeripour F, Kabir E, Sheikhi A Fabric defect detection using modified local binary patterns. *EURASIP Journal on Advances in Signal Processing* 2008, 2008:60
24. Tax DM, Duin RP (1999) Data domain description using support vectors. In: *ESANN*, pp 251–256
25. Wang T, Chen Y, Qiao M, Snoussi H (2018) A fast and robust convolutional neural network-based defect detection model in product quality control. *Int J Adv Manuf Technol* 94(9–12):3465–3471
26. Wen Y, Zhang K, Li Z, Qiao Y (2016) A discriminative feature learning approach for deep face recognition. In: *European conference on computer vision*. Springer, pp 499–515
27. Xiao Q, Luo H, Zhang C (2017) Margin sample mining loss: a deep learning based method for person re-identification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*
28. Xue-Wu Z, Yan-Qiong D, Yan-Yun L, Ai-Ye S, Rui-Yu L (2011) A vision inspection system for the surface defects of strongly reflected metal based on multi-class SVM. *Expert Syst Appl* 38(5):5930–5939
29. Yang Y, Zha Z-J, Gao M, He Z (2016) A robust vision inspection system for detecting surface defects of film capacitors. *Signal Process* 124:54–62
30. Ye D, Hong GS, Zhang Y, Zhu K, Fuh JYH (2018) Defect detection in selective laser melting technology by acoustic signals with deep belief networks. *Int J Adv Manuf Technol*:1–11
31. Yi L, Li G, Jiang M (2017) An end-to-end steel strip surface defects recognition system based on convolutional neural networks. *steel research international* 88(2):1600068
32. Zhou W, Fei M, Zhou H, Li K (2014) A sparse representation based fast detection method for surface defect detection of bottle caps. *Neurocomputing* 123:406–414



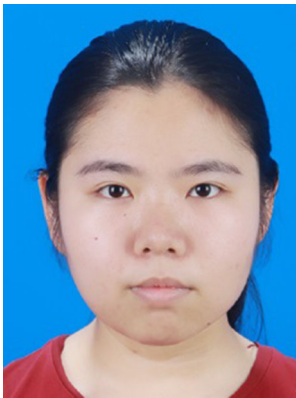
Songlin Wu received the B.S. degree in School of Optical and Electronic Information from Huazhong University of Science and Technology. He is currently studying for a master's degree at the Huazhong University of Science and Technology. His research interests include machine vision, machine learning and automated surface inspection.



Yubin Wu is an associate professor in the School of Optical and Electronic Information, Huazhong University of Science and Technology. He received his M.E. degree in optical engineering from Institute of Optics and Electronics of the Chinese Academy of Sciences in 1987. He received his B.E. degree in optical instruments from Huazhong University of Science and Technology in 1984. His research interests include optoelectronic sensing and signal processing, machine vision, and development of high-tech products.



Danhua Cao is a professor in the School of Optical and Electronic Information, Huazhong University of Science and Technology. She received her Ph.D. degree in electronic physics and devices from Huazhong University of Science and Technology in 1993. She is a permanent member of the Professional Committee of Opto-electronic Technology in the Chinese Optical Society. Her research interests include optoelectronic sensing and signal processing as well as machine vision algorithms and systems.



Caiyun Zheng received the bachelor's degree in School of Optical and Electronic Information from Huazhong University of Science and Technology. She is currently studying for a master's degree at the Huazhong University of Science and Technology. Her research interests include machine vision, machine learning and automated surface inspection.