Deepening Stereotype Recognition for Detecting Anomalies

PH.Swarna Rekha[1], M.Karuna[2], E.Rupa[3]

[1]Assistant Professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[2]Assistant Professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[3]Assistant Professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

Abstract: ***At present, many anomaly detection researches focus on two problems: one is that the anomaly on pixels cannot be accurately located; the other is that the training data cannot include the anomalies. We introduce the "Stereotype Deepening" algorithm to solve the challenging problems, which uses transitive learning in the process of training the tree-like teacher-student network structure to deepen the "Stereotype". Therefore, in the abnormal area, the descriptors given by the student will deviate from the descriptors given by the teacher. Additionally, peer bias is also taken into account as an abnormal score item. Experiments have been conducted on different types of datasets to prove the effectiveness of this algorithm for anomaly detection and anomaly localization. By comparison, the method proposed in this paper has significant advantages in textures data type.***

Keywords: Stereotype deepening; Transitive learning; Knowledge distillation; Anomaly detection; Anomaly localization

1 Introduction

In the real world, a common requirement is to determine which instances are different from other instances, and such a process is called anomaly detection [1]. Up to now, anomaly detection is still a challenging task. To solve various problems in anomaly detection, different anomaly detection algorithms have appeared one after another. Deep neural networks (DNN) have developed rapidly in recent years, and anomaly detection algorithms based on deep learning have shown excellent performance. Because many data used for anomaly detection are difficult to be labeled, anomaly detection methods based on unsupervised learning have been widely studied. Most of the initial work focuses on image reconstruction, and the common method is to use general models such as generative adversarial networks (GANs) [2] and autoencoders [3, 4]. Some researchers have found that the pre-trained DNN has powerful functions. They used a tiny stack of autoencoders and a convolutional neural network (CNN) to form a cascade classifier to cooperate in cubic-patch anomaly detection [5]. In [6], they applied the student-teacher structure to unsupervised anomaly detection using a pre-trained residual neural network (ResNet) and completed anomaly detection and anomaly localization through multiscale anomaly segmentation. Afterward, Salehi et al. [7] used a visual geometry group (VGG) as a pre-training network to distill the knowledge into a cloning network. They used the distance between activation values and the directional similarity of activation vectors between several key layers to complete anomaly detection and used the gradient of overall loss to find anomaly regions that caused their values to increase to complete anomaly location. Compared with [6], Salehi et al. [7] completed anomaly detection and anomaly localization from different angles.

Inspired by the previous work, this paper proposed an anomaly detection method based on "Stereotype Deepening". In this paper, a tree-like teacher-student network with transitive learning characteristics is introduced to complete regional anomaly detection and localization. Figure 1 shows the detection results represented by anomaly maps. It has been confirmed in [6] that there will be a cognitive bias between students and teachers, which is called "Stereotype". As shown in Figure 2, our intuition is that the network will deepen this cognitive bias in the process of transitive learning, and abnormal areas can be distinguished by deepening "Stereotype".

The main contributions are as follows:

- We presented a tree-like teacher-student anomaly detection structure based on "stereotype deepening", which associating anomalies with pixels and locating anomaly areas.

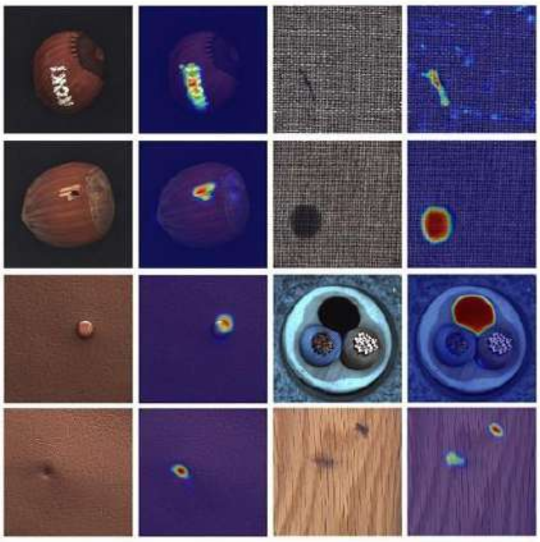

Figure 1: Comprehensive assessment results. The evaluation types include textures and objects. There are apparent color differences between the abnormal area and the surrounding area.

- We proposed two loss functions: one is a new compactness loss, which is not affected by batch size, and the other is the regression error between descriptors.
- We integrated inference bias, delivery bias and peer bias to evaluate the performance of anomaly detection and localization, so that the result was more obvious. Anomaly maps are used to intuitively express the results of anomaly localization.
- Experiments on three datasets have proved the effectiveness of the method proposed in this paper. Our algorithm shows satisfactory results on all datasets, especially in the category of textures.

2 Related Work

2.1 Supervised Anomaly Detection

Many supervised anomaly detection methods are in the form of binary classification. Because of the use of labels, they can produce highly accurate results. Some studies [8, 9, 10] try to use the method based on active learning. Gaddam et al. [11] proposed a novel anomaly detection method. Based on K-Means and ID3, which first obtained k different clusters and then constructed an ID3 decision tree in each cluster. This approach avoids both the forced assignment and class dominance. Jumutc and Suykens [12] extended the supervised novelty detection. They introduced a new coupling term between classes which took advantage of finding a reasonable decision boundary.

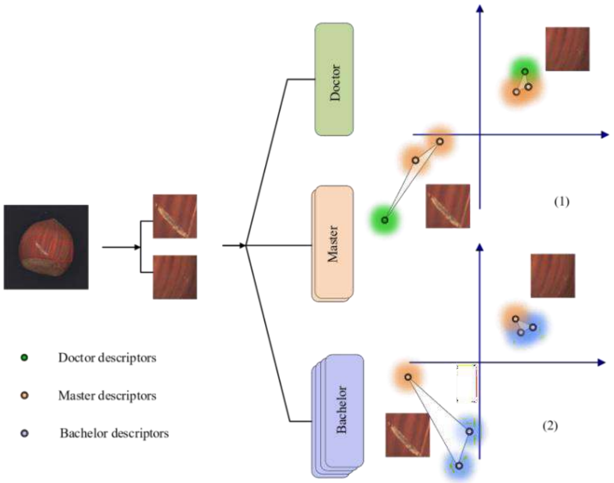

Figure 2: Prior distribution. Coordinate system (1) shows the cognitive deviation between one doctor and two masters, and coordinate system (2) shows the cognitive deviation between one master and two bachelors. Their distributions are relatively similar in normal areas, but their distributions will be quite different in abnormal areas due to deepened cognitive bias.

Although supervised anomaly detection has high accuracy, it has poor generality due to the uncertainty of anomalies and the lack of data labels. Some previous works have tried to solve these problems from various aspects, but supervised anomaly detection still has limitations.

2.2 Semi-supervised Anomaly Detection

Labels for normal data are more accessible to obtain than labels for abnormal data. Therefore, many researchers chose to use semi-supervised methods to complete anomaly detection. Gu et al. [13] proposed a corrupted GAN (CorGAN) for outlier detection. Assuming that the generator generates outliers of negative classes, the discriminator was trained to distinguish the training dataset from the data generated by the generator. To avoid reaching Nash equilibrium in the training process, they also proposed several techniques to break the fusion and establish robust outlier identifiers. Similarly, influenced by GANs, Sabokrou et al. [14] took the lead in adding one-class classification to the end-to-end architecture and introduced an anomaly detection network structure of *R*+*D*, in which R consists of encoder and decoder, while *D* is a CNN network, which was used to classify the data regenerated by *R*. Perera and Patel [15] proposed deep one-class classification (DOC) to solve the one-class classification problem and introduced the joint loss based on compactness loss and descriptive loss to train the network. Finally, it was verified by experiments on anomaly detection, novelty detection, and mobile active authentication datasets. Unsupervised anomaly detection has also been widely studied in many application areas. In detecting abnormal climate, Racah et al. [16] proposed a multi-channel spatial-temporal CNN architecture for semi-supervised bounding box prediction and exploratory data analysis to address the challenge of incomplete extreme weather labeling data. This method can apply time information and unlabeled data to improve the positioning of extreme weather events. In addition, in remote sensing applications, there is also a challenge of collecting labeled data. Wu and Prasad [17] provide a semi-supervised anomaly detection method for hyperspectral image classification, which used unlabeled data with pseudo-labels generated by the C-DPMM-based clustering algorithm to train the neural network.

Semi-supervised anomaly detection can carry out end-to-end learning and improve the situation of insufficient data labels. However, it takes a long time in the training process, and the effect of feature extraction is not good.

2.3 Unsupervised Anomaly Detection

Compared with supervised and semi-supervised anomaly detection, an obvious advantage of unsupervised anomaly detection is that it can distinguish normal from abnormality by learning unlabeled dataset. Zong et al. [18] proposed a deep auto-encoded gaussian mixture model (DAGMM), which is easy to carry out end-to-end training for anomaly detection. DAGMM model consists of a compression network and an estimation network. A deep automatic encoder was used to generate low-dimensional representation and reconstruction errors for each input data. Further, low-dimensional representation and reconstruction errors have been fed into the gaussian mixture model. The problem of continuous anomaly detection in application fields such as image analysis and video surveillance is a challenge that needs to be solved. Lu et al. [19] used an autoencoder model to capture the inherent difference in density between outliers and normal instances and integrated the model into a recurrent neural network (RNN). It was convenient to capture the context information and finally updated the network through hierarchical training. Unlike [19], Leveau and Joly [20] used an adversarial autoencoder for anomaly detection and further improved its performance by introducing explicit rejection classes in the prior distribution and adding random input images to the autoencoder. Some scholars have proposed a deep structured energy-based model (DSEBM), which extended the energy-based model to a deep architecture with three types of structures and solved the anomaly detection problem by directly modeling the data distribution using the deep architecture [21]. Moreover, they also provided two decision criteria for training, namely energy score and reconstruction error. Mishra et al. [22] used CVAEs to solve the anomaly detection problem under the zero-shot learning. They treat it as a missing data problem, generate samples from a given attribute, and use the generated samples to classify invisible classes. Some people introduced an anomaly detection method for a mobile autonomous robot based on GAN, which builds a GAN to collect images by remotely operating the robot in a given environment [23]. The shifted grid divides all images into patches for training GAN. It compared the bottleneck feature of the generated patch with that of the actual patch.

Although many unsupervised anomaly detection methods have been proposed, many existing methods are based on data reconstruction, and the results of anomaly detection will be affected by reconstruction errors. Considering this problem, this paper utilized a tree-like teacher-student structure to deepen the "Stereotype" generated in the process of transitive learning and used the compactness loss with irrelevant batches and regression error to optimize the network. Finally, inference bias, delivery bias, and peer bias were used as anomaly evaluation indicators.

3 Preliminary Work

3.1 Knowledge Distillation

For better training effect, many models were trained from one or more large neural networks. However, this method consumes lots of computing resources and is difficult to deploy. To solve this problem, Hinton et al.[24] tried to use knowledge distillation to transfer knowledge from bulky models to small models, which is more suitable for deployment to a large number of users. The knowledge distillation model consists of two parts: teacher network and student network. The teacher network has a complicated structure and numerous parameters, while the student network has a simple structure and few parameters. During training, the student network learns the knowledge extracted by the teacher network.

The teacher network generated classification results through the softmax layer. The results contain probability information of each category, but only one category belongs to positive labels, and the rest belongs to negative labels. The probability of each negative label is usually much smaller than that of the positive label, so the information carried by the negative labels is ignored frequently. To avoid this problem, the distillation method changed outputs of the softmax by controlling the temperature T so that the output probability distribution was smoother, and the result was recorded as soft targets. The smoother probability distribution of outputs can amplify information carried by the negative labels. The softmax function is defined as follows:

$$q_i = exp^{a_i/T} / \Sigma^n_{i=1} exp^{a_i/T} \tag{1}$$

where T represents the distillation temperature. When training the teacher network, the temperature T was set to 1, and the training was achieved by minimizing the cross-entropy between the softmax layer output and the target. After the teacher network was trained, a higher temperature T greater than 1 was set, and it is used to train the student network. The difference between the output of the student network and the soft target was regarded as the distillation loss. When temperature T in the student network was set to 1, the difference between output and ground truth was taken as another loss. Both losses are used to evaluate the performance of student network. Its results showed that the distilled student network had comparable performance to the teacher network, which was easier to deploy.

3.2 Descriptor Compactness

The neural network model is prone to over-fitting due to excessive sample noise interference, high model complexity, and too much iteration. Over-fitting can easily lead to deviations in the results, so it is also essential to solve the model over-fitting. In addition to the common causes of overfitting, Tian et al. [25] found that the severity of overfitting is directly related to the correlation between the descriptor dimensions. Therefore, in their experiment, an error term was introduced to illustrate the compactness of descriptors, and the redundancy between descriptors of different dimensions was reduced through training so that each dimension carried as much information as possible. The correlation coefficients of different dimensions are expressed as:

$$r_{ij} = (b_i - \bar{b}_i)^T (b_j - \bar{b}_j) / \sqrt{(b_i - \bar{b}_i)^T (b_i - \bar{b}_i)} \sqrt{(b_j - \bar{b}_j)^T (b_j - \bar{b}_j)} \tag{2}$$

Among them, $\bar{b}_i$ and $\bar{b}_j$ respectively represent the mean value of the i_{th} column and the j_{th} column.

The correlation matrix $[r_{ij}]$ is denoted as R.

4 Algorithm

In this section, the proposed "Stereotype Deepening" algorithm is described in detail. In process of the training network, transitive learning was used to deepen the "Stereotype", so that the discrepancy between the descriptors was enlarged. The student networks were updated by minimizing the mixed loss. In the evaluation, inference bias, delivery bias, and peer bias were used to measure the effect of anomaly detection and localization.

4.1 Network Structure

As shown in Figure 3, masters and bachelors were set up as students in the tree-like teacher-student structure, and it completed the training of the network one by one through transitive learning. For convenience, we gave students different names for each layer. The students who were obtained after completing the first transitive learning are called masters, and then the network carried out the second round of transfer study, at which time bachelors were got. Bachelors were obtained by studying the knowledge of masters. We divided the bachelors into different classes according to which master is its teacher. Masters act as teachers and students. We trained all the students on the given training data $U = \{u_1,u_2,...,u_n\}$ that only contains anomaly-free images. Each network except doctor takes the descriptor of the previous network as the regression target. For example, the regression targets for bachelors are the feature descriptors output by masters. After training, we used both abnormal and non-abnormal images as test data, and inference bias, delivery bias, and peer bias caused by "Stereotype" were used as indicators for abnormal evaluation.

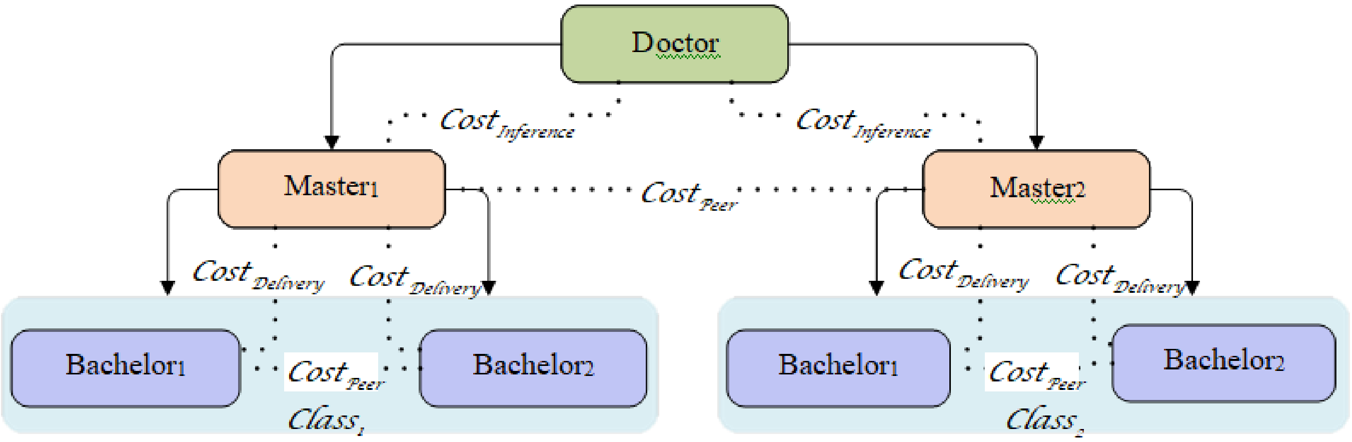

Figure 3: *Bachelor*$_1$ and *Bachelor*$_2$ belong to *Class*$_1$, *Bachelor*$_3$ and *Bachelor*$_4$ belong to *Class*$_2$. Among them, $Cost_{Inference}$ represents the inference bias, $Cost_{Delivery}$ represents the delivery bias, and $Cost_{Peer}$ represents the peer bias

4.2 The Process of Training

This section will introduce network training in detail. The process is divided into three stages. The training structure is shown in Figure 4.

4.2.1 Training of Doctor Network

The input image *G* was randomly cut into patch-sized image regions *I*, and the doctor network D outputs a d-dimensional descriptor for each patch *I*. Because the pre-trained deep neural network has a strong representation ability, it performs well in classification. Therefore, we used the pre-trained network *D* as the basic network of the classification network T, and the loss of the classification network can be expressed as:

$$L_k = -\Sigma y log(T(I)) \tag{3}$$

where *T* is the classification network, and *y* is the classification label.

4.2.2 Training of Masters and Bachelors

In this part, we used the mean square error and the improved descriptor compactness loss as the mixed loss. The network trained to get masters first and then to get bachelors through masters. We always let the current network fit the description of the previous network. Specifically, *D* first extracted patch-based descriptors for each image on the dataset *U*, and masters were trained by regressing the descriptors output by *D*. After masters trained on the dataset *U*, using the same method to train bachelors with outputs of masters. In particular, before training the next layer of students, we all Normalized the output provided by this layer. Masters have dual identities in the whole network structure. They are students for the front layer and teachers for the back layer.

Mean Square Error Similar to the training of the doctor network, it also needs to extract the knowledge of the previous network into the current network when training the student network. Meanwhile, the distance is used to measure the difference between *M*(*I*) and *D*(*I*). *M*(*I*) is the d-dimension descriptor given by the master network:

$$L_d = ||M(I) - D(I)||^2 \tag{4}$$

Descriptor Compactness Error For a set of *I* inputs, to eliminate redundancy and minimize correlation between descriptors, we have made some improvements based on the method used in [25]. The improved method can ensure the accuracy of the calculation. Each patch *I* passing through the network will be transformed into a d-dimensional descriptor in a batch, and we have calculated the correlation between any two descriptors. After calculation, it was found that simply summing these correlation coefficients cannot accurately express the overall correlation, and it would be affected by the batch size when minimizing the descriptor correlation. The batch size was determined by the number of patches, which could reflect the number of random combinations of different descriptors,

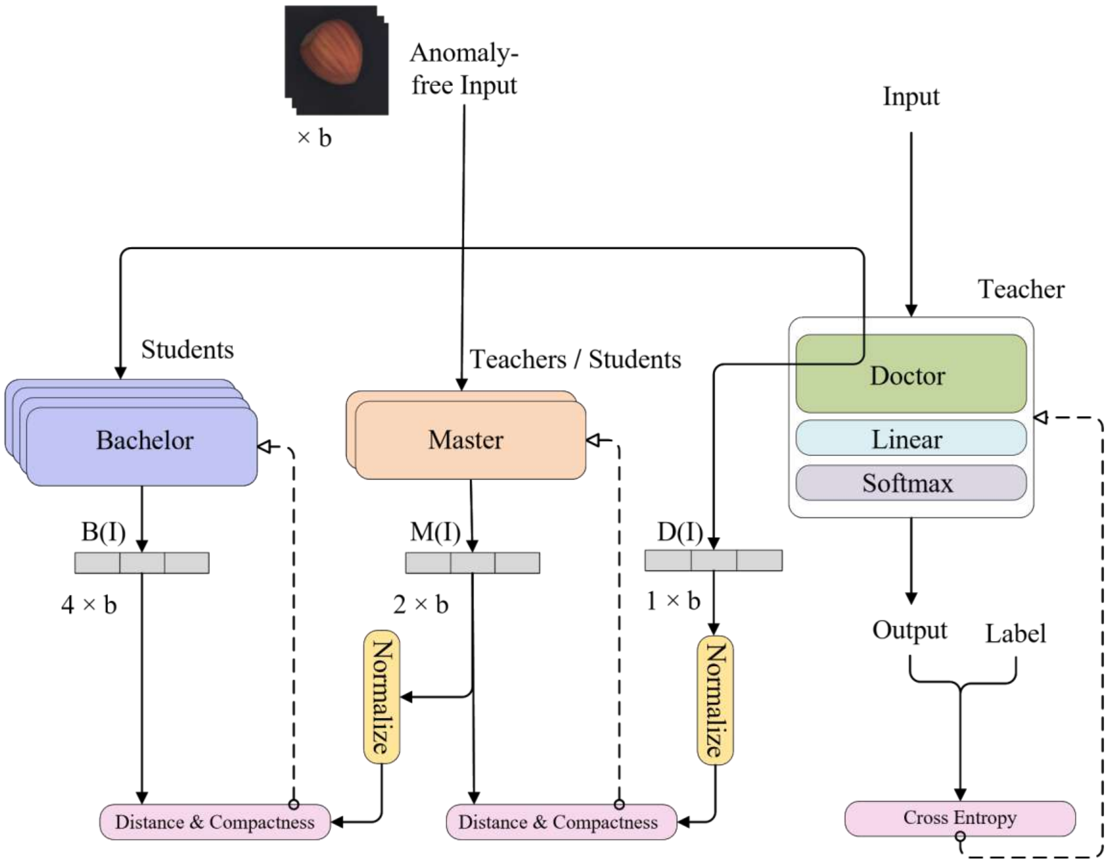

Figure 4: *b* denotes batch size, and *D*(*I*), *M*(*I*) and *B*(*I*) are descriptors. From right to left are the three stages of training. The masters are trained according to the doctor, and then the masters are used to train the bachelors.

and further affected the sum of correlation coefficients. Therefore, to eliminate the influence of batch size on descriptor compactness, the improved method can be expressed as:

$$L_p = n(n-1)/2 \tag{5}$$

n is the batch size.

Mixed Loss The training losses of masters and bachelors are obtained by summing these two weights and are finally expressed as follows:

$$L_t = \mu L_d + (1-\mu)L_p \tag{6}$$

in this formula, μ represents the weighting factor.

4.3 Anomaly Evaluation

For the test set $W = \{w_1, w_2, \dots, w_n\}$ that contains both normal data and abnormal data, it determines the abnormal area by the degree of difference between the descriptors output by each network. Since doctor module has been trained with abnormal data, when abnormal data was input to doctor, the descriptor output by doctor would conform to the features' distribution of the abnormal area. However, masters only learned the distribution of normal data during training, so their descriptors would deviate from the description of doctor when they encounter abnormal areas, resulting in inference bias. Because it only used normal data to train masters, the weight obtained after completing masters training will lack the induction of comprehensive information. Bachelors would be affected by this one-sided factor when training according to the masters, and there would be delivery bias when bachelors learned descriptors of masters. Additionally, there would be peer biases between masters and peer biases among bachelors. Figure 3 clearly identifies three types of biases.

We take the deviation degree between the descriptors given by the master and the descriptors given by the doctor as the first score. *D*(*x*) represents the descriptor of the doctor, and $M_i(x)$ is the descriptor given by the i_{th} master. The first anomaly score is expressed as:

$$Cost_{Inference} \quad \sum_i \sqrt{(D(x) - M_i(x)^{2}} \tag{7}$$

As mentioned before, it can be known that bachelors are distilled by masters, so the difference between bachelors and masters is taken as the second score, which is expressed as:

$$Cost_{Transitivity} = \sum_i \sum_j \sqrt{(M_i(x) - B_j(x))^2} \tag{8}$$

$B_j(x)$ is the descriptor given by the j_{th} bachelor.

The deviations between students of the same level are combined as the third abnormal score, which is represented by $Cost_{Peer}$.

$$Cost_{Peer} = \sum_{i}\sum_{k_M}\sqrt{(M_i(x) - M_{k_M}(x))^2} + \sum_{j}\sum_{k_B}\sqrt{(B_j(x) - B_{k_B}(x))^2} \tag{9}$$

M_{kM} represents any other masters except the i_{th} master. $B_{kB}(x)$ represents the descriptor given by any other bachelors except the j_{th} bachelors. The total anomaly score is expressed as:

$$CostTotal= CostInference+ CostTransitivity+ CostPeer \tag{10}$$

5 Experiment

In this part, the "Stereotype Deepening" algorithm proposed in this paper will be verified from two aspects: anomaly detection and anomaly localization. All experiments were conducted under the environment of Intel(R) Core(TM) i7-8700 CPU and NVIDIA GeForce GTX 1660. The code has been released at: https://github.com/zhmhbest/StudentTeacherAnomalyDetection.

5.1 Datasets

We tested the proposed method on three datasets: MNIST, CIFAR-10, and MVTec.

MNIST: It contains 70,000 handwritten digits, of which 60,000 belong to the training set, and the rest belong to the test set [26].

CIFAR-10: This data set consists of 10 categories of color images. Each category contains 6,000 images. Among them, 50,000 images are used for training, and the remaining 10,000 are used as the test set [27].

MVTec: It consists of more than 5,000 high-resolution images, including 10 different object categories and 5 different texture categories. The images in the training set are non-anomalous, and the testing set contain part of the abnormal images [28].

5.2 Experimental Results

The network structure used to train doctor, master and bachelor is given in Table 1.

5.2.1 Comparisons based on the MNIST and CIFAR-10 Datasets

The experiment first verifies the performance of the entire network. Only one category in the dataset is regarded as normal data, and all other categories are regarded as abnormal. For example, if 0 is regarded as normal data on the MNIST dataset, the remaining numbers are considered abnormal for testing network performance. We use the area under the AUROC curve to evaluate the performance of our method and other related works.

The "Stereotype Deepening" algorithm shows high accuracy on both MNIST and CIFAR-10 data sets, especially on CIFAR-10 our average accuracy rate is 0.1805 higher than the LSA algorithm. Table 2 shows the comparison results.

Table 1: The network structure when the patch size is 64. Leaky rectified linear units with slope 5 × 10^{-3} are applied as activation functions after each convolution layer

		Parameters	
Layer	Output Size	Kernel	Stride
Input	64×64×3	-	-
Conv2d	61×61×64	4×4	1
MaxPool2d	30×30×64	2×2	2
Conv2d	27×27×32	4×4	1
MaxPool2d	13×13×32	2×2	2
Conv2d	10×10×16	4×4	1
MaxPool2d	5×5×16	2×2	2
Conv2d	2×2×8	4×4	1
Conv2d	1×1×4	2×2	1
Linear	1×1×1	-	-
Flatten	1×1	-	-
Linear	1×512	-	-

Table 2: Anomaly detection results. The table gives the anomaly detection accuracy of different algorithms in one-class classification, and the mean values are reflected in their overall performance

Dataset	Method	0	1	2	3	4	5	6	7	8	9	Mean
	DSVDD[29]	0.98	0.997	0.917	0.919	0.949	0.885	0.983	0.946	0.939	0.965	0.948
	OCGAN[30]	**0.998**	**0.999**	0.942	0.963	0.975	0.98	0.991	0.981	0.939	0.981	0.975
	CAVGA Du[31]	0.994	0.997	0.989	0.983	**0.997**	0.968	0.988	0.986	0.988	**0.991**	0.986
MINST	LSA[32]	0.993	0.999	0.959	0.966	0.956	0.964	0.994	0.98	0.953	0.981	0.975
	Ours	0.991	0.995	**0.994**	**0.996**	0.995	**0.995**	**0.994**	**0.991**	**0.992**	0.989	**0.9932**
	DSVDD[29]	0.617	0.659	0.508	0.591	0.609	0.657	0.677	0.673	0.759	0.731	0.648
	OCGAN[30]	0.757	0.531	0.64	0.62	0.723	0.62	0.723	0.575	0.82	0.554	0.6566
	CAVGA Du[31]	0.653	0.784	**0.761**	0.747	0.775	0.552	0.813	0.745	0.801	0.741	0.737
CIFAR-10	LSA[32]	0.735	0.58	0.69	0.542	0.761	0.546	0.751	0.535	0.717	0.548	0.641
	Ours	**0.834**	**0.852**	0.748	**0.761**	**0.801**	**0.762**	**0.901**	**0.841**	**0.887**	**0.828**	**0.8215**

5.2.2 Comparisons based on MVTec Dataset

The MVTec dataset provides anomalies based on different entities. In addition to verifying the anomaly detection capabilities of the network, we also verified the effect of anomaly localization through experiments.

Table 3: Anomaly localization results in terms of AUROC. The table shows the average AUROC on *Textures* and *Objects* expressed as *Textures mean* and *Objects mean*, respectively

		Textures										Objects						
	Carpet	Grid	Leather	Tile	Wood	Textures mean	Bottle	Cable	Capsule	Hazelnut	Metal nut	Pill	Screw	Toothbrush	Transistor	Zipper	Objects mean	Mean
STAD[6]	0.695	0.819	0.819	0.921	0.725	0.7958	0.918	0.865	**0.916**	0.937	**0.895**	**0.935**	**0.928**	0.863	0.701	0.933	**0.889**	0.858
CAVGA Du[31]	0.73	0.75	0.71	0.7	0.85	0.748	0.89	0.63	0.83	0.84	0.67	0.88	0.77	0.91	0.73	0.87	0.802	0.784
CAVGA Ru[31]	0.78	0.78	0.75	0.72	0.88	0.782	0.91	0.67	0.87	0.87	0.71	0.91	0.78	**0.97**	0.75	0.94	0.838	0.819
CAVGA Dw[31]	0.8	0.79	0.8	0.81	0.89	0.818	0.93	0.86	0.89	0.9	0.81	0.93	0.79	0.96	**0.8**	**0.95**	0.882	0.861
Ours	**0.958**	**0.955**	**0.9633**	**0.922**	**0.925**	**0.9447**	**0.9533**	**0.876**	0.906	**0.972**	0.842	0.8086	0.906	0.74	0.67	0.8714	0.8731	**0.885**

We considered the inferred bias of each module in the anomaly region. Excepting the inference bias, we also considered the delivery bias between masters and bachelors, which would make the descriptors given by the masters and bachelors different in the same abnormal area. Students of the same grade got different initial settings of the networks and did not use abnormal images for training. Therefore, different students' expressions in abnormal areas would also have significant differences when encountering abnormal areas. Figure 5 shows anomaly maps, in which anomaly maps given by comprehensive evaluation are included, and the difference of color reflects anomaly degree. According to Figure 5, the performance of the three biases in abnormal areas is different, and the comprehensive evaluation result is more obvious. As shown in Figure 5, the result of anomaly detection and localization using only the delivery bias is more effective than that of inference bias, which shows that the "Stereotype" is reasonable and available for anomaly detection and localization.

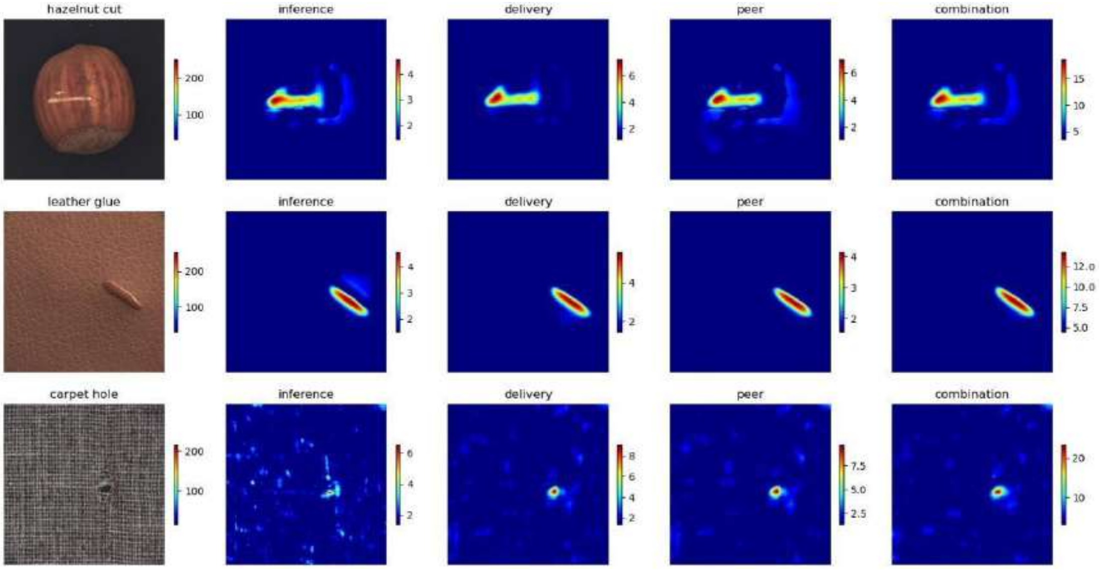

Figure 5: Anomaly map and anomaly score at all levels. The various bias produced in the process of transitive learning is evident in the abnormal areas

Table 3 shows the anomaly localization results on the MVTec dataset. Experiments showed that our method has outstanding advantages in the category of textures. The single AUROC on textures is above 0.92. For *Textures*, "Stereotype Deepening" is superior to other baselines. The average results in the category of *Objects* can reach a level comparable to other methods. In general, this method is better than most of the algorithms we compared.

6 Conclusion

In this paper, an algorithm based on a tree-like teacher-student structure was proposed for anomaly detection and location, called "Stereotype Deepening". A descriptor compactness loss that is irrelevant to batch size was used in the training of the teacher network, and transitive learning is applied to train the student networks. During the experiment, the anomaly detection performance of the network was tested on the MNIST and CIFAR-10 datasets. Afterward, the "Stereotype" generated by the network during the training process was used to complete the anomaly localization. Peer bias and delivery bias verified the effectiveness of "Stereotype" from two dimensions. The experiment proved that our method has a significant effect on the textures type.

References

[1] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. abs/1901.03407, 2019.

[2] Thomas Schlegl, Philipp Seeb¨ock, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. abs/1703.05921:146–157, 2017.

[3] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. abs/1804.04488:161–169, 2018.

[4] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and HonglinQiao. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 187–196, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[5] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017.

[6] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4182–4191, June 2020.

[7] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad Hossein Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. pages 14902–14912, June 2021.

[8] M. Almgren and E. Jonsson. Using active learning in intrusion detection. In *Proceedings. 17th IEEE Computer Security Foundations Workshop*, pages 88–98, Los Alamitos, CA, USA, jun 2004. IEEE Computer Society.

[9] Yang Li and Li Guo. An active learning based tcm-knn algorithm for supervised network intrusion detection. *Computers Security*, 26(7):459–467, 2007.

[10] Jay Stokes, John Platt, Joseph Kravis, and Michael Shilman. Aladin: Active learning of anomalies to detect intrusions. Technical Report MSR-TR-2008-24, March 2008.

[11] Shekhar R. Gaddam, Vir V. Phoha, and Kiran S. Balagani. K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):345–354, 2007.

[12] VilenJumutc and Johan A.K. Suykens. Multi-class supervised novelty detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2510–2523, 2014.

[13] Jindong Gu, Matthias Schubert, and Volker Tresp. Semi-supervised outlier detection using generative and adversary framework, 2018.

[14] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, June 2018.

[15] Pramuditha Perera and Vishal M. Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019.

[16] Evan Racah, Christopher Beckham, Samira Maharaj, Teganand Ebrahimi Kahou, Mr. Prabhat, and Chris Pal. Extreme weather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[17] Hao Wu and Saurabh Prasad. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Transactions on Image Processing*, 27(3):1259–1270, 2018.

[18] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

[19] Weining Lu, Yu Cheng, Cao Xiao, Shiyu Chang, Shuai Huang, Bin Liang, and Thomas Huang. Unsupervised sequential outlier detection with deep architectures. *IEEE Transactions on Image Processing*, 26(9):4321–4330, 2017.

[20] Valentin Leveau and Alexis Joly. Adversarial autoencoders for novelty detection. Research report, Inria - Sophia Antipolis, 2017.

[21] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1100–1109. JMLR.org, 2016.

[22] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2269–22698, June 2018.

[23] Wallace Lawson, Esube Bekele, and Keith Sullivan. Finding anomalies with generative adversarial networks for a patrolbot. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 484–485, 2017.

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[25] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6128–6136, 2017.

[26] Yann Le Cun and Corinna Cortes. Mnist handwritten digit database. 2010.

[27] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[28] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* June 2019.

[29] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Mu¨ller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402, 10–15 Jul 2018.

[30] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2893–2901, 2019.

[31] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *Computer Vision – ECCV 2020*, pages 485– 503. Springer International Publishing, 2020.

[32] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pages 481–490, 2019.