GAN-based Anomaly Detection in Imbalance Problems

Junbong Kim[1*], Kwanghee Jeong[2*], Hyomin Choi[2], and Kisung Seo[2†]
* equal contribution, † corresponding author

[1] IDIS, Seoul, Korea
[2] Department of Electronic Engineering, Seokyeong University, Seoul, Korea
jbkim@idis.co.kr,jkh910902@skuniv.ac.kr,hiahiml@skuniv.ac.kr,
ksseo@skuniv.ac.kr

Abstract. Imbalance problems in object detection are one of the key issues that affect the performance greatly. Our focus in this work is to address an imbalance problem arising from defect detection in industrial inspections, including the different number of defect and non-defect dataset, the gap of distribution among defect classes, and various sizes of defects. To this end, we adopt the anomaly detection method that is to identify unusual patterns to address such challenging problems. Especially generative adversarial network (GAN) and autoencoder-based approaches have shown to be effective in this field. In this work, 1) we propose a novel GAN-based anomaly detection model which consists of an autoencoder as the generator and two separate discriminators for each of normal and anomaly input; and 2) we also explore a way to effectively optimize our model by proposing new loss functions: Patch loss and Anomaly adversarial loss, and further combining them to jointly train the model. In our experiment, we evaluate our model on conventional benchmark datasets such as MNIST, Fashion MNIST, CIFAR 10/100 data as well as on real-world industrial dataset – smartphone case defects. Finally, experimental results demonstrate the effectiveness of our approach by showing the results of outperforming the current State-Of-The-Art approaches in terms of the average area under the ROC curve (AUROC).

Keywords: Imbalance problems, Anomaly Detection, GAN, Defects Inspection, Patch Loss, Anomaly Adversarial Loss

1 Introduction

The importance of the imbalance problems in machine learning is investigated widely and many researches have been trying to solve them [12],[20],[23],[28],[34]. For example, class imbalance in the dataset can dramatically skew the performance of classifiers, introducing a prediction bias for the majority class [23]. Not only class imbalance, but various imbalance problems exist in data science. A general overview of imbalance problems is investigated in the literature

[12],[20],[23],[28]. Specifically, the survey of various imbalance problems for object detection subject is described in the review paper [34].

We handle a couple of imbalance problems closely related to industrial defects detection in this paper. Surface defects of metal cases such as scratch, stamped, and stain are very unlikely to happen in the production process, thereby resulting in outstanding class imbalance. Besides, size of defects, loss scale, and discriminator distributional imbalances are covered as well. In order to prevent such imbalance problems, anomaly detection [8] approach is used. This method discards a small portion of the sample data and converts the problem into an anomaly detection framework. Considering the shortage and diversity of anomalous data, anomaly detection is usually modeled as a one-class classification problem, with the training dataset containing only normal data [40].

Reconstruction-based approaches [1],[41],[43] have been paid attention for anomaly detection. The idea behind this is that autoencoders can reconstruct normal data with small errors, while the reconstruction errors of anomalous data are usually much larger. Autoencoder [33] is adopted by most reconstruction-based methods which assume that normal and anomalous samples could lead to significantly different embeddings and thus differences in the corresponding reconstruction errors can be leveraged to differentiate the two types of samples [42]. Adversarial training is introduced by adding a discriminator after autoencoders to judge whether its original or reconstructed image [10],[41]. Schlegl et al. [43] hypothesize that the latent vector of a GAN represents the true distribution of the data and remap to the latent vector by optimizing a pre-trained GAN-based on the latent vector. The limitation is the enormous computational complexity of remapping to this latent vector space. In a follow-up study, Zenati et al. [52] train a BiGAN model [4], which maps from image space to latent space jointly, and report statistically and computationally superior results on the MNIST benchmark dataset. Based on [43],[52], GANomaly [1] proposes a generic anomaly detection architecture comprising an adversarial training framework that employs adversarial autoencoder within an encoder-decoder-encoder pipeline, capturing the training data distribution within both image and latent vector space. However, the studies mentioned above have much room for improvement on performance for benchmark datasets such as Fashion-MNIST, CIFAR-10, and CIFAR-100.

A novel GAN-based anomaly detection model by using a structurally separated framework for normal and anomaly data is proposed to improve the biased learning toward normal data. Also, new definitions of the patch loss and anomaly adversarial loss are introduced to enhance the efficiency for defect detection. First, this paper proves the validity of the proposed method for the benchmark data, and then expands it for the real-world data, the surface defects of the smartphone case. There are two types of data that are used in the experiments – classification benchmark datasets including MNIST, Fashion-MNIST, CIFAR10, CIFAR100, and a real-world dataset with the surface defects of the smartphone. The results of the experiments showed State-Of-The-Art performances in four benchmark dataset, and average accuracy of 99.03% in the real-world dataset of the smartphone case defects. To improve robustness and performance, we select the final model by conducting the ablation study. The result of the ablation study and the visualized images are described.

In summary, our method provides the methodological improvements over the recent competitive researches, GANomaly[1] and ABC[51], and overcome the State-Of-The-Art results from GeoTrans[13] and ARNet[18] with significant gap.

2 Related Works

Imbalance problems
A general review of Imbalance problems in deep learning is provided in [5]. There are lots of class imbalance examples in various areas such as computer vision [3],[19],[25],[48], medical diagnosis [16],[30] and others [6],[17],[36],[38] where this issue is highly significant and the frequency of one class can be much larger than another class. It has been well known that class imbalance can have a significant deleterious effect on deep learning [5]. The most straightforward and common approach is the use of sampling methods. Those methods operate on the data itself to increase its balance. Widely used and proven to be robust is oversampling [29]. The issue of class imbalance can be also tackled on the level of the classifier. In such a case, the learning algorithms are modified by introducing different weights to misclassification of examples from different classes [54] or explicitly adjusting prior class probabilities [26]. A systematic review on imbalance problems in object detection is presented in [34]. In here, total of eight different imbalance problems are identified and grouped four main types: class imbalance, scale imbalance, spatial imbalance, and objective imbalance. Problem based categorization of the methods used for imbalance problems is well organized also.

Anomaly detection
For anomaly detection on images and videos, a large variety of methods have been developed in recent years [7],[9],[22],[32],[37],[49],[50],[55]. In this paper, we focus on anomaly detection in still images. Reconstruction-based anomaly detection [2],[10],[43],[44],[46] is the most popular approach. The method compress normal samples into a lower-dimensional latent space and then reconstruct them to approximate the original input data. It assume that anomalous samples will be distinguished through relatively high reconstruction errors compared with normal samples.

Autoencoder and GAN-based anomaly detection
Autoencoder is an unsupervised learning technique for neural networks that learns efficient data encoding by training the network to ignore signal noise [46]. Generative adversarial network (GAN) proposed by Goodfellow et al. [15] is the approach co-training a pair networks, generator and discriminator, to compete with each other to become more accurate in their predictions. As reviewed in

[34], adversarial training has also been adopted by recent work within anomaly detection. More recent attention in the literature has been focused on the provision of adversarial training. Sabokrou et al. [41] employs adversarial training to optimize the autoencoder and leveraged its discriminator to further enlarge the reconstruction error gap between normal and anomalous data. Furthermore, Akcay et al. [1] adds an extra encoder after autoencoders and leverages an extra MSE loss between the two different embeddings. Similarly, Wang et al. [45] employs adversarial training under a variational autoencoder framework with the assumption that normal and anomalous data follow different Gaussian distribution. Gong et al. [14] augments the autoencoder with a memory module and developed an improved autoencoder called memory-augmented autoencoder to strengthen reconstructed errors on anomalies. Perera et al. [35] applies two adversarial discriminators and a classifier on a denoising autoencoder. By adding constraint and forcing each randomly drawn latent code to reconstruct examples like the normal data, it obtained high reconstruction errors for the anomalous data.

3 Method

3.1 Model Structures

In order to implement anomaly detection, we propose a GAN-based generative model. The pipeline of the proposed architecture of training phase is shown in the Figure 1. The network structure of the Generator follows that of an autoencoder, and the Discriminator consists of two identical structures to separately process the input data when it is normal or anomaly. In the training phase, the model learns to minimize reconstruction error when normal data is entered to the generator, and to maximize reconstruction error when anomaly data is entered. The loss used to minimize reconstruction error with normal image input is marked in blue color in four ways. Also, the loss used for maximizing the error with anomaly image input is marked in red color in two ways. In the inference phase, reconstruction error is used to detect anomalies as a criteria standard.The matrix maps in the right part of Figure 1 show that each value of the output matrix represents the probability of whether the corresponding image patch is real or fake. The way is used in PatchGAN [11] and it is totally different from Patch Loss we proposed in this paper.

3.2 Imbalance Problems in Reconstruction-based Anomaly Detection

In order to handle anomaly detection for defects inspection, the required imbalance characteristics are described. We define imbalance problems for defects as class imbalance, loss function scale imbalance, distributional bias on the learning model, and imbalance in image and object (anomaly area) sizes. Table 1 summarizes the types of imbalance problems and solutions.

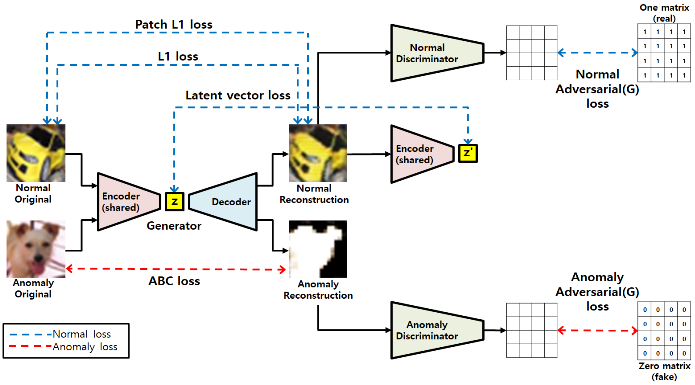

Fig. 1: Pipeline of the proposed approach for anomaly detection.

Table 1: Imbalance problems and solutions of the proposed method

Imbalance Problems	Solutions
class imbalance	data sampling with k-means clustering (section 3.5)
loss scale imbalance	loss function weight search (section 3.4)
discriminator distributional Bias	two discriminator (section 3.3)
Size imbalance between object(defect) and image	reconstruction-based methods

Class imbalance

Class imbalance is well known, and surface defects of metal cases such as scratch, stamped, and stain are very unlikely to happen in the production process, therefore resulting in outstanding class imbalance problems between normal and anomaly. Not only the number of normal and defective data is imbalanced, but also the frequency of occurrence differ among the types of defects such as scratch, stamped, and stain, so imbalance within each class exists in anomaly data. To resolve such class imbalance, data is partially sampled and used in training. Here, if the data is sampled randomly without considering its distribution, the entire data and the sampled data might not be balanced in their distribution. Therefore, in this paper,we use the method of dividing the entire data into several groups by k-means clustering, and then sample the same number of data within each group.

Loss function scale imbalance

The proposed method uses the weighted sum of 6 types of loss functions to train the generator. The scale of the loss function used here is different, and even if the scale is the same, the effect on the learning is different. In addition, GAN contains a min-max problem that the generator and the discriminator learn by competing against each other, making the learning difficult and unstable. The loss scales of the generator and the discriminator should be sought at a similar rate to each other so that GAN is effectively trained. To handle such loss function scale imbalance problems, weights used in loss combination are explored by a grid search.

Discriminator distributional bias
The loss will be used to update the generator differently for normal and anomaly data. When the reconstruction data is given to the discriminator, the generator is trained to output 1 from normal data and 0 from anomaly data. Thus, when training from both normal and anomaly data, using a single discriminator results in training the model to classify only normal images well. Separating discriminator for normal data and for anomaly data is necessary to solve this problem. This method only increases the parameters or computations of the model in the training phase, but not those in inference phase. As a result, there is no overall increase in memory usage or latency at the final inferences.

Size imbalance between object(defect) and image
Industrial defect data exhibits smaller size of defect compared to the size of the entire image. Objects in such data occupy very small portion of the image, making it closer to object detection rather than classification, so it is difficult to expect fair performance with classification methods. To solve this, we propose a method generating images to make the total reconstruction error bigger not affected by the size of the defect and the size of the entire image which contains the defect.

3.3 Network Architecture

The proposed model is a GAN-based network structure consisting of a generator and a discriminator. The generator is in the form of an autoencoder to perform image to image translation. And a modified U-Net[39] structure is adopted, which has an effective delivery of features using a pyramid architecture. The discriminator is a general CNN network, and two discriminators are used only in the training phase.

The generator is a symmetric network that consists of four 4 x 4 convolutions with stride 2 followed by four transposed convolutions. The total parameters of generator is composed of a sum of 0.38K, 2.08K, 8.26K. 32.9K, 32.83K, 16.42K, 4.11K, and 0.77K, that is 97.75K totally. The discriminator is a general network that consists of three 4 x 4 convolutions with stride 2 followed by two 4 x 4 convolutions with stride 1. The total parameters of discriminator is composed of a sum of 0.38K, 2.08K, 8.26K, 32.9K, and 32.77K, that is 76.39K.

3.4 Loss Function

Total number of loss functions used in the proposed model is eight. Six losses for training of generator, one for normal discriminator and another for anomaly discriminator. The loss function for training of each discriminator is adopted from LSGAN [31] as shown in Eq. (1). It uses the a-b coding scheme for the discriminator, where a and b are the labels for fake data and real data, respectively.

$$\min_D V_{\text{LSGAN}}(D) = [(D(x)-b)^2] + [(D(G(x))-a)^2] \tag{1}$$

Six kinds of loss functions, as shown in from Eq. (2) to (8) are employed to train the generator. Among them, four losses are for normal images. First, L1 reconstruction error of generator for normal image is provided as shown in Eq. (2). It penalizes by measuring the L1 distance between the original x and the generated images ($\hat{x} = G(x)$) as defined in [1]:

$$\mathcal{L}_{\text{recon}} = \|x - G(x)\|_1 \tag{2}$$

Second, the patch loss is newly proposed in this paper as shown in Eq. (3). Divide a normal image and a generated image separately into M patches and select the average of the biggest n reconstruction errors among all the patches.

$$\mathcal{L}_{\text{patch}} = f_{avg}(n)(\left\|x_{patch(i)} - G(x_{patch(i)})\right\|_1), i = 1, 2, ..., m \tag{3}$$

Third, latent vector loss [1] is calculated as the difference between latent vectors of generator for normal image and latent vectors of cascaded encoder for reconstruction image as shown in Eq. (4)

$$\mathcal{L}_{\text{enc}} = \|G_E(x) - G_E(G(x))\|_1 \tag{4}$$

Eq. (5) defines the proposed adversarial loss for the generator update use in LSGAN[31], where y denotes the value that G wants D to believe for fake data.

$$\min_G V_{\text{LSGAN}}(G) = [(D(G(x))-y)^2] \tag{5}$$

Fourth, the adversarial loss used to update the generator is as shown in Eq. (6). The loss function intends to output a real label of 1 when a reconstruction image (fake) is into the discriminator.

$$\min_G V_{\text{LSGAN}}(G) = [(D(G(x))-1)^2] \tag{6}$$

Two remaining losses for anomaly images are as follows. One is anomaly adversarial loss for updating generator and the other is ABC [51] loss. Unlike a general adversarial loss of Eq (6), anomaly reconstruction image should be generated differently from real one to classify anomaly easily, the anomaly adversarial loss newly adopted in our work is as shown in Eq. (7).

$$\min_G V_{\text{LSGAN}}(G) = [(D(G(x))-0)^2] \tag{7}$$

ABC loss as shown Eq. (8) is used here to maximize L1 reconstruction error $\mathcal{L}_\theta(\cdot)$ for anomaly data. Because the difference between the reconstruction errors for normal and anomaly data is large, the equation is modified by adding the exponetial and log function to solve the scale imbalance.

$$\mathcal{L}_{\text{ABC}} = -\log(1 - e^{-\mathcal{L}_\theta(x_i)}) \tag{8}$$

Total loss function consists of weighted sum of each loss. All losses for normal images are grouped together and same as for anomaly images. Those two group of losses are applied to update the weights of learning process randomly. The scale imbalances exist among the loss functions. Although the scale could be adjusted in same range, the effect might be different, so we explore the weight of each loss using the grid search. Because ABC loss can have the largest scale, the weighted sum of normal data is set more than twice as large as the weighted sum of anomaly data. In order to avoid huge and unnecessary search space, each weight of the loss functions is limited from 0.5∼1.5 range. Then the grid search is executed the each weight adjusting by 0.5. Total possible cases for the grid search is 314. The final explored weights of loss function are shown in Table 2.

Table 2: Weight combination of loss functions obtained by Grid search

Normal reconstruction L1 loss	ABC loss	Normal adversarial loss	Normal reconstruction patch L1 loss	Normal latent vector loss	Anomaly adversarial loss
1.5	0.5	0.5	1.5	0.5	1.0

3.5 Data Sampling

As mentioned in section 3.2, the experimental datasets include imbalance problems. For benchmark datasets such as MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100, have a class imbalance problems presenting imbalance of data sampling. The real-world dataset, surface defects of smartphone case is not only the number of normal and defective data are imbalanced, but also the frequency of occurrences differs among the types of defects. Also the size of image and object (defect) is imbalanced too. To solve those imbalance problem, k-means clustering-based data sampling is performed to make balanced distribution of data. In learning stage for benchmark datasets, all data is used for normal case. In case of anomaly, the same number is sampled for each class so that the total number of data is similar to normal. At this time, k-means clustering is performed on each class, and data is sampled from each cluster in a distribution similar to the entire dataset. For anomaly case of defect dataset, data is sampled using the same method as the benchmark, and a number of normal data is sampled, equal to the number of data combined with the three kinds of defects - scratch, stamped and stain. Detail number of data is described in section 4.1

4 Experiments

In this section, we perform substantial experiments to validate the proposed method for anomaly detection. We first evaluate our method on commonly used benchmark datasets - MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100. Next, we conduct experiments on real-world anomaly detection dataset - smartphone case defect dataset. Then we present the respective effects of different designs of loss functions through ablation study.

4.1 Datasets

Datasets used in the experiments include of four standard image datasets: MNIST [27], Fashion-MNIST[47], CIFAR-10[24], and CIFAR-100[24]. Additionaly smartphone case defect data is added to evaluate the performance in real-world environments.

For MNIST, Fashion-MNIST, CIFAR-10 data set, each class is defined as normal and the rest of nine classes are defined as anomaly. Total 10 experiments are performed by defining each 10 class once as normal. With CIFAR-100 dataset, one class is defined as normal and the remaining 19 classes are defined as anomaly among 20 superclasses. Each superclass is defined as normal one by one, so in total 20 experiments were conducted. Also, in order to resolve imbalance in the number of normal data and anomaly data, and in distribution of sampled data, the method proposed in 3.5 is applied when sampling training data. 6000 normal and anomaly data images were used for training MNIST and Fashion-MNIST, and 5000 were used for CIFAR-10. Additionally, for MNIST and Fashion-MNIST, the images were resized into 32x32 so that the size of the feature can be the same when concatenating them in the network structure.

Smartphone case defect dataset consists of normal, scratch, stamped and stain classes. And there are two main types of data set. The first dataset contains patch images that are cropped into 100x100 from their original size of 2192x1000 . The defective class is sampled in the same number as the class with the least number of data among defects by deploying the method from section 3.5, and the normal data is sampled in a similar number to that of the detective data. 900 images of normal data and 906 images of anomaly data were used for training, and 600 images of normal data and 150 images of anomaly data were used for testing. In the experiments, images were resized from 100x100 to 128x128. The second dataset consists of patch images cropped into 548x500. The same method of sampling as the 100x100 patch images is used. 1800 images of normal data and 1814 images of anomaly data are used for training, and 2045 images of normal data and 453 images of normal data are used for testing.

4.2 Experimental Setups

Experimentation is performed using Intel Core i7-9700K @ 3.60GHz and NVIDIA geforce GTX 1080ti with Tensorflow 1.14 deep learning framework. For augmentation on MNIST, the method of randomly cropping 0∼2 pixels from the boundary and resizing again was used, while for Fashion-MNIST, images were vertically and horizontally flipped, randomly cropped 0∼2 pixels from their boundary, and resized again. Then, the images were rotated 90, 180, or 270 degrees. For CIFAR-10 and CIFAR-100, on top of the augmentation method utilized for Fashion-MNIST, hue, saturation, brightness, and contrast are varied as additional augmentation. For defective dataset, vertical flipping and horizontal flipping are used along with rotation of 90, 180, or 270 degrees.

Hyperparameters and details on augmentation are as follows Table 3.

Table 3: Hyperparameters used for model training

Hyperparameters					Parameters of patch reconstruction error loss (Eq. 3)		
Epoch	Batch size	Learning rate init	Learning rate decay epoch	Learning rate decay factor	Patch size	Stride	Number of selected patch
300	1	0.0001	50	0.5	16	8	3

4.3 Benchmarks Results

In order to evaluate the performance of our proposed method, we conducted experiments on MNIST, Fashion-MNIST, CIFAR-10 and CIFAR-100. For estimating recognition rates of the trained model, AUROC(Area Under the Receiver Operating Characteristic) is used. Table 5 contains State-Of-The-Art studies and their recognition rates, and Figure 2 compares recognition rates, model parameters, and FPS on CIFAR-10 among the representative studies.

According to Table 5, for the MNIST, the paper with highest performance among previous studies is ARNet, which presented average AUROC value of 98.3 for 10 classes, however our method obtains average AUROC of 99.7 with the proposed method. Also, ARNet have standard deviation of 1.78 while ours shows 0.16, therefore significantly reducing deviations among classes. For the Fashion-MNIST, ARNet previously performs the best with average AUROC of 93.9 and standard deviation of 4.7, but our method accomplishes much better results of average AUROC of 98.6 and standard deviation 1.20. For the CIFAR-10, the average AUROC ARNet, so far as the best, with is 86.6 with standard deviation 5.35, but our method achieves quite better result of average AUROC of 90.6 with standard deviation 3.14. Lastly, for CIFAR-100, compared to the results from ARNet, which are average AUROC of 78.8 and standard deviation 8.82, the proposed method shows outstanding improvement of average AUROC 87.4 and standard deviation 4.80. In summary, four different datasets were used for evaluating the performance, and we achieve highly improved results from the previous State-Of-The-Arts studies in terms of recognition rate and learning stability on all the tested datasets.

Figure 2 shows a graph that compares the performance of State-Of-The-Arts researches regarding for average AUROC, FPS, and the number of model parameters used in inference. FPS is calculated as averaging total 10 estimations of processing time for 10,000 CIFAR-10 images. In the graph, the x-axis stands for FPS, i.e. how many images are inferred per second, and the y-axis represents AUROC(%), the recognition rate. The area of the circle of each method indicates the number of model parameters used in inference. The graph shows that our method has AUROC of 90.6%, which is 4% higher than that of ARNet, 86.6%, known to be the highest in the field. Among the papers mentioned, GeoTrans has the least number of model parameters, 1,216K. However, the model proposed in this paper require 98K parameters, which only 8% of those of GeoTrans therefore resulting in huge reduction. Finally in terms of FPS, ARNet was known to be the fastest with 270FPS, the proposed method was able to process with 532FPS, almost twice as fast.

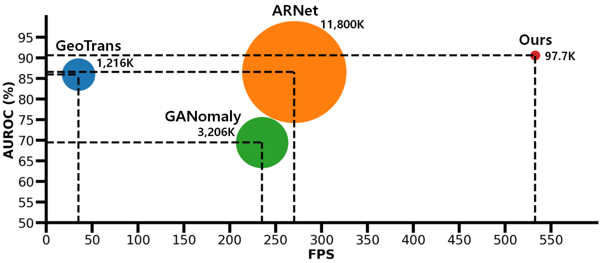

Fig. 2: Comparison of AUROC, FPS, and model parameters for CIFAR10 dataset.

Table 4: AUROC results for defect dataset (%)

Dataset	Train	Test
100 x 100 patch defect	99.73	99.23
548 x 500 patch defect	99.74	98.84

4.4 Defect Dataset Results

Table 4 shows the experiment results with smartphone detective dataset. Testing for 100x100 patch images, the performance is absolutely high with AUROC(%) of 99.23. The bigger size of image data with 548x500 patch, much more difficult than of 100x100 patch data, is tested and shows very similar results of 98.84 still. Therefore, the superiority of our method is proved even in real-world smartphone detective dataset.

Table 5: Comparison with the state-of-the-art literature in AUROC(%) for benchmark datasets

Dataset	Method	0	1	2	3	4	5	6	7	8	9	**avg**	SD
MNIST	AE	98.8	99.3	91.7	88.5	86.2	85.8	95.4	94.0	82.3	96.5	91.9	5.90
	VAE[21]	92.1	**99.9**	81.5	81.4	87.9	81.1	94.3	88.6	78.0	92.0	87.7	7.05
	AnoGAN[43]	99.0	99.8	88.8	91.3	94.4	91.2	92.5	96.4	88.3	95.8	93.7	4.00
	ADGAN[10]	99.5	**99.9**	93.6	92.1	94.9	93.6	96.7	96.8	85.4	95.7	94.7	4.15
	GANomaly[1]	97.2	99.6	85.1	90.6	94.5	94.9	97.1	93.9	79.7	95.4	92.8	6.12
	OCGAN[35]	99.8	**99.9**	94.2	96.3	97.5	98.0	99.1	98.1	93.9	98.1	97.5	2.10
	GeoTrans[13]	98.2	91.6	99.4	99.0	99.1	99.6	**99.9**	96.3	97.2	99.2	98.0	2.50
	ARNet[18]	98.6	**99.9**	99.0	99.1	98.1	98.1	99.7	99.0	93.6	97.8	98.3	1.78
	OURS	**99.9**	**99.9**	**99.7**	**99.8**	**99.7**	**99.8**	99.7	**99.7**	**99.5**	**99.4**	**99.7**	**0.16**
Fashion-MNIST	AE	71.6	96.9	72.9	78.5	82.9	93.1	66.7	95.4	70.0	80.7	80.9	11.03
	DAGMM[53]	42.1	55.1	50.4	57.0	26.9	70.5	48.3	83.5	49.9	34.0	51.8	16.47
	DSEBM[55]	91.6	71.8	88.3	87.3	85.2	87.1	73.4	98.1	86.0	97.1	86.6	8.61
	ADGAN[10]	89.9	81.9	87.6	91.2	86.5	89.6	74.3	97.2	89.0	97.1	88.4	6.75
	GANomaly[1]	80.3	83.0	75.9	87.2	71.4	92.7	81.0	88.3	69.3	80.3	80.9	7.37
	GeoTrans[13]	99.4	97.6	91.1	89.9	92.1	93.4	83.3	98.9	90.8	99.2	93.5	5.22
	ARNet[18]	92.7	99.3	89.1	93.6	90.8	93.1	85.0	98.4	**97.8**	98.4	93.9	4.70
	OURS	**99.5**	**99.6**	**98.2**	**98.6**	**98.1**	**99.5**	**95.9**	**99.4**	97.6	**99.6**	**98.6**	**1.20**
CIFAR-10	AE	57.1	54.9	59.9	62.3	63.9	57.0	68.1	53.8	64.4	48.6	59.0	5.84
	VAE[21]	62.0	66.4	38.2	58.6	38.6	58.6	56.5	62.2	66.3	73.7	58.1	11.50
	DAGMM[53]	41.4	57.1	53.8	51.2	52.2	49.3	64.9	55.3	51.9	54.2	53.1	5.95
	DSEBM[55]	56.0	48.3	61.9	50.1	73.3	60.5	68.4	53.3	73.9	63.6	60.9	9.10
	AnoGAN[43]	61.0	56.5	64.8	52.8	67.0	59.2	62.5	57.6	72.3	58.2	61.2	5.68
	ADGAN[10]	63.2	52.9	58.0	60.6	60.7	65.9	61.1	63.0	74.4	64.4	62.4	5.56
	GANomaly[1]	**93.5**	60.8	59.1	58.2	72.4	62.2	88.6	56.0	76.0	68.1	69.5	13.08
	OCGAN[35]	75.7	53.1	64.0	62.0	72.3	62.0	72.3	57.5	82.0	55.4	65.6	9.52
	GeoTrans[13]	74.7	**95.7**	78.1	72.4	87.8	**87.8**	83.4	**95.5**	93.3	91.3	86.0	8.52
	ARNet[18]	78.5	89.8	86.1	77.4	**90.5**	84.5	89.2	92.9	92.0	85.5	86.6	5.35
	OURS	92.6	93.6	**86.9**	**85.4**	89.5	**87.8**	**93.5**	91.0	**94.6**	**91.7**	**90.6**	**3.14**

Dataset	Method	0	1	2	3	4	5	6	7	8	9	10
CIFAR-100	AE	66.7	55.4	41.4	49.2	44.9	40.6	50.2	48.1	66.1	63.0	52.7
	DAGMM[53]	43.4	49.5	66.1	52.6	56.9	52.4	55.0	52.8	53.2	42.5	52.7
	DSEBM[55]	64.0	47.9	53.7	48.4	59.7	46.6	51.7	54.8	66.7	71.2	78.3
	ADGAN[10]	63.1	54.9	41.3	50.0	40.6	42.8	51.1	55.4	59.2	62.7	79.8
	GANomaly[1]	57.9	51.9	36.0	46.5	46.6	42.9	53.7	59.4	63.7	68.0	75.6
	GeoTrans[13]	74.7	68.5	74.0	81.0	78.4	59.1	81.8	65.0	**85.5**	90.6	87.6
	ARNet[18]	77.5	70.0	62.4	76.2	77.7	64.0	86.9	65.6	82.7	90.2	85.9
	OURS	**85.5**	**86.1**	**94.4**	**87.3**	**91.7**	**85.1**	**89.9**	**88.3**	83.8	**92.4**	**94.2**
	Method	11	12	13	14	15	16	17	18	19	**avg**	SD
	AE	62.1	59.6	49.8	48.1	56.4	57.6	47.2	47.1	41.5	52.4	8.11
	DAGMM[53]	46.4	42.7	45.4	57.2	48.8	54.4	36.4	52.4	50.3	50.5	6.55
	DSEBM[55]	62.7	66.8	52.6	44.0	56.8	63.1	73.0	57.7	55.5	58.8	9.36
	ADGAN[10]	53.7	58.9	57.4	39.4	55.6	63.3	66.7	44.3	53.0	54.7	10.08
	GANomaly[1]	57.6	58.7	59.9	43.9	59.9	64.4	71.8	54.9	56.8	56.5	9.94
	GeoTrans[13]	83.9	83.2	58.0	**92.1**	68.3	73.5	93.8	**90.7**	85.0	78.7	10.76
	ARNet[18]	83.5	**84.6**	67.6	84.2	74.1	**80.3**	91.0	85.3	85.4	78.8	8.82
	OURS	**86.0**	83.7	**76.8**	89.5	**80.6**	80.2	**94.9**	89.7	**87.1**	**87.4**	**4.93**

4.5 Ablation Studies

To understand how each loss affect GAN-based anomaly learning, we defined CIFAR-10 bird class as normal and the rest of the classes as anomaly, and conducted an ablation study. The network structure and learning parameters are the same as the experiments with the benchmarks, and training data is also sampled based on k-means clustering to 5000 normal and 5000 anomaly images. All of the given testing data from the dataset is used for testing. The results shows that the performance improves as each loss gets added to the basic autoencoder. In the Table 6, No.2(only ABC added) and No.7(all losses added) don't show much difference in AUROC, 85.44% and 86.76%. However, from Figure 3 it can be found that reconstructed images exhibits significant difference. It visualizes the result of experiment No.2 and No.7 with normal data. In the case of ABC, the normal images could not be reconstructed similarly, so the center was made 0 due to loss from anomaly. On the other hand, experiment No.7 which exploited combination of the proposed losses reconstructed the image similarly to the original. The average error regarding reconstruction error map also shows difference of 2∼3 times.

Table 6: Ablation study experiment results to confirm the impact of losses used in generator update (AUROC(%))

No.	Auto Encoder	ABC	Generator Normal	Generator Anomaly	Normal patch	Normal latent	AUROC
1	✓						**64.68**
2	✓	✓					**85.44**
3	✓		✓				**65.74**
4	✓			✓			**65.93**
5	✓				✓		**64.87**
6	✓					✓	**64.89**
7	✓	✓	✓	✓	✓	✓	**86.76**

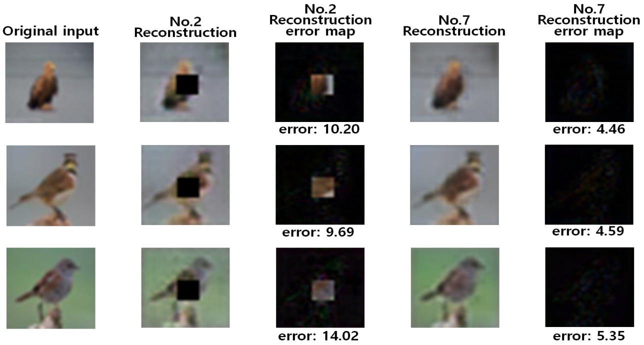

Fig. 3: Visualized results for No.2 and No.7 experiments in ablation study

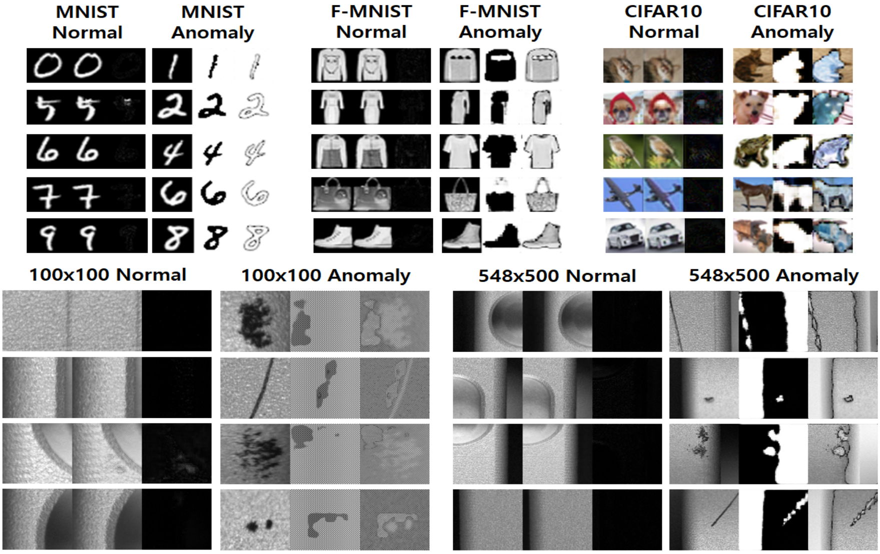

Fig. 4: Visualization of experimental results on the Benchmark dataset & defect data (Original, reconstruction, reconstruction error map order from the left of each image)

5 Conclusion

We proposed a novel GAN-based anomaly detection model by using a new framework, newly defined loss functions, and optimizing their combinations. The discriminators for normal and anomaly are structurally separated to improve the learning that has been biased toward normal data. Also, a new definition of patch loss and anomaly adversarial loss, which are effective for fault detection was introduced and combined with the major losses proposed from previous studies to perform joint learning. In order to systemize the proportion of each loss in the combination, the weight of each loss was explored using the grid search. The main results of our experiments successfully demonstrated that the proposed method much further improves the AUROC for CIFAR-10 data compared to the results of State-Of-The-Art including GANomaly[1], GeoTrans[13] and ARNet[18]. Especially, we applied our method to real-world data set with surface defects of smartphone case and validated outstanding superiority of anomaly detection of the defects. In the future, we will try to extend our approach to a hierarchical anomaly detection scheme from pixel level to video level.

Acknowledgement

This work was supported by National Research Foundation of Korea Grant funded by the Korea government (NRF-2019R1F1A1056135)

References

1. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian conference on computer vision. pp. 622–637. Springer (2018)
2. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**(1), 1–18 (2015)
3. Beijbom, O., Edmunds, P.J., Kline, D.I., Mitchell, B.G., Kriegman, D.: Automated annotation of coral reef survey images. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 1170–1177. IEEE (2012)
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4183–4192 (2020)
5. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018)
6. Cardie, C., Howe, N.: Improving minority class prediction using case-specific feature weights (1997)
7. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
9. Chu, W., Xue, H., Yao, C., Cai, D.: Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos. IEEE Transactions on Multimedia **21**(1), 246–255 (2018)
10. Deecke, L., Vandermeulen, R., Ruff, L., Mandt, S., Kloft, M.: Image anomaly detection with generative adversarial networks. In: Joint european conference on machine learning and knowledge discovery in databases. pp. 3–17. Springer (2018)
11. Demir, U., Unal, G.: Patch-based image inpainting with generative adversarial networks. arXiv preprint arXiv:1803.07422 (2018)
12. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data sets. Springer (2018)
13. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. In: Advances in Neural Information Processing Systems. pp. 9758–9769 (2018)
14. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1705–1714 (2019)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
16. Grzymala-Busse, J.W., Goodwin, L.K., Grzymala-Busse, W.J., Zheng, X.: An approach to imbalanced data sets based on changing rule strength. In: Rough-neural computing, pp. 543–553. Springer (2004)
17. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73**, 220–239 (2017)
18. Huang, C., Cao, J., Ye, F., Li, M., Zhang, Y., Lu, C.: Inverse-transform autoencoder for anomaly detection. arXiv preprint arXiv:1911.10676 (2019)

19. Johnson, B.A., Tateishi, R., Hoan, N.T.: A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. International journal of remote sensing **34**(20), 6969–6982 (2013)
20. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 27 (2019)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
22. Kiran, B.R., Thomas, D.M., Parakkal, R.: An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. Journal of Imaging **4**(2), 36 (2018)
23. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**(4), 221–232 (2016)
24. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
25. Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine learning **30**(2-3), 195–215 (1998)
26. Lawrence, S., Burns, I., Back, A., Tsoi, A.C., Giles, C.L.: Neural network classification and prior class probabilities. In: Neural networks: tricks of the trade, pp. 299–313. Springer (1998)
27. LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ (1998)
28. Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A., Seliya, N.: A survey on addressing high-class imbalance in big data. Journal of Big Data **5**(1), 42 (2018)
29. Ling, C.X., Li, C.: Data mining for direct marketing: Problems and solutions. In: Kdd. vol. 98, pp. 73–79 (1998)
30. Mac Namee, B., Cunningham, P., Byrne, S., Corrigan, O.I.: The problem of bias in training data in regression problems in medical decision support. Artificial intelligence in medicine **24**(1), 51–70 (2002)
31. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017)
32. Markou, M., Singh, S.: Novelty detection: a review—part 2:: neural network based approaches. Signal processing **83**(12), 2499–2521 (2003)
33. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional autoencoders for hierarchical feature extraction. In: International conference on artificial neural networks. pp. 52–59. Springer (2011)
34. Oksuz, K., Cam, B.C., Kalkan, S., Akbas, E.: Imbalance problems in object detection: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
35. Perera, P., Nallapati, R., Xiang, B.: Ocgan: One-class novelty detection using gans with constrained latent representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2898–2906 (2019)
36. Philip, K., Chan, S.: Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: Proceeding of the Fourth International Conference on Knowledge Discovery and Data Mining. pp. 164–168 (1998)
37. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. Signal Processing **99**, 215–249 (2014)
38. Radivojac, P., Chawla, N.V., Dunker, A.K., Obradovic, Z.: Classification and knowledge discovery in protein databases. Journal of Biomedical Informatics **37**(4), 224–239 (2004)

39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
40. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International conference on machine learning. pp. 4393–4402 (2018)
41. Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E.: Adversarially learned one-class classifier for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3379–3388 (2018)
42. Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. pp. 4–11 (2014)
43. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International conference on information processing in medical imaging. pp. 146–157. Springer (2017)
44. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
45. Wang, X., Du, Y., Lin, S., Cui, P., Shen, Y., Yang, Y.: advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection. Knowledge-Based Systems **190**, 105187 (2020)
46. Xia, Y., Cao, X., Wen, F., Hua, G., Sun, J.: Learning discriminative reconstructions for unsupervised outlier removal. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1511–1519 (2015)
47. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
48. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3485–3492. IEEE (2010)
49. Xu, K., Jiang, X., Sun, T.: Anomaly detection based on stacked sparse coding with intraframe classification strategy. IEEE Transactions on Multimedia **20**(5), 1062–1074 (2018)
50. Xu, K., Sun, T., Jiang, X.: Video anomaly detection and localization based on an adaptive intra-frame classification network. IEEE Transactions on Multimedia **22**(2), 394–406 (2019)
51. Yamanaka, Y., Iwata, T., Takahashi, H., Yamada, M., Kanai, S.: Autoencoding binary classifiers for supervised anomaly detection. In: Pacific Rim International Conference on Artificial Intelligence. pp. 647–659. Springer (2019)
52. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222 (2018)
53. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. arXiv preprint arXiv:1605.07717 (2016)
54. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on knowledge and data engineering **18**(1), 63–77 (2005)
55. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations (2018)