Computer Vision and Image Understanding 210 (2021) 103226

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

One-class anomaly detection via novelty normalization

Jhih-Ciang Wu [a,b], Sherman Lu [a], Chiou-Shann Fuh [b], Tyng-Luh Liu [a,*]

[a] *Institute of Information Science, Academia Sinica, Taipei 115, Taiwan*
[b] *Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan*

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
Deep learning
Anomaly detection
Unsupervised learning
Convolutional neural network

ABSTRACT

Anomaly detection is an important task in many real-world applications, such as within cybersecurity and surveillance. As with most data these days, the size and dimensionality of the data within these fields are constantly growing, which makes it essential to develop an approach that can both accurately and efficiently identify anomalies within these datasets. In this paper, we address the problem of one-class anomaly detection, where after training on a singular class, we try to determine whether or not inputs belong to that said class. Most of the currently existing methods have limitations in which the criterion of the novel class relies solely on the reconstruction error term. We attempt to break away from this restriction by proposing the use of an autoencoder network with a normalization term. We pair this with an additive novelty scoring module during the training procedure as a way to determine the difference between a given image and our determined normal class, therefore improving the efficiency of our model. We evaluate our model on MNIST, CIFAR-10, and Fashion-MNIST, three popular datasets for image classification, and compare the results against other various state-of-the-art models to determine the efficacy of our efforts. Our model not only outperforms the existing methods, but it also gives us a narrower range of AUCs for the tested classes, suggesting a stark improvement in both accuracy and precision. Moreover, we discover that introducing this "Novelty Normalization" concept into our model allows us to expand its usage into multiclass scenarios without a steep drop in accuracy.

1. Introduction

One-class anomaly detection (Ruff et al., 2018) is a branch of anomaly detection (Chandola et al., 2009). Its objective is to determine whether or not an instance belongs to a predetermined singular "normal" class. The main challenge this task presents is that samples of the normal class are the only instances the training procedure can see. For example, if the dataset consists of images of digits, the model will only get to train on a single digit. During the inference stage, the model is expected to predict a lower (novelty) score for the normal class and a higher score for the other "anomalous" classes.

A unique characteristic of anomaly detection is the difficulty of modeling the distribution of anomalous data, which often exist in limited quantities or even are unavailable for training. On the other hand, the class of normal data is generally more well-defined and such samples are abundant in practical applications of interest. We thus consider casting the targeted problem as a one-class formulation where only samples of the normal category are provided for training the machine learning model to perform out-of-distribution (OOD) detection.

One-class anomaly detection is commonly seen in applications to cybersecurity (Ten et al., 2011) and surveillance (Maiorano and Petrosino, 2016). Intrusion and fraud detection relies on anomaly detection to ensure network and information security. Anomaly detection is also used to recognize suspicious behavior on cameras, GPS, and even heat sensors. Given these high-risk applications, it is necessary to develop a system that can carry out the detection task as accurately and precisely as possible.

There are many works that tackle one-class anomaly detection. The paper's concept is visualized in Fig. 1. Classical approaches like One-Class SVM (Schölkopf et al., 2001) and Kernel Density Estimation (KDE) (Bishop, 2006) seek the latent representation for the given class to obtain tolerable performances. However, they often fail due to the high dimensionality of the problem. For this reason, Hadsell et al. (2006) proposes a form of dimensionality reduction where the resulting mapping relies on the neighborhood and disregards any of the distance measurements in the input space. Other papers, such as (Pidhorskyi et al., 2018), addresses this representation learning perspective task. In recent years, autoencoder-based methods have become commonplace in one-class anomaly detection problems (Kingma and Welling, 2013; Abati et al., 2019).

We choose to consider this problem from a more intuitive perspective. People have the ability to label seemingly abnormal objects without utilizing object reconstruction techniques. For example, when someone sees the fuselage of a plane, we do not have to revisualize it with the wings attached to know that it is a plane. In this paper, we

[*] Corresponding author.
E-mail address: liutyng@iis.sinica.edu.tw (T.-L. Liu).

https://doi.org/10.1016/j.cviu.2021.103226
Received 22 May 2020; Received in revised form 13 May 2021; Accepted 14 May 2021
Available online 24 May 2021
1077-3142/© 2021 Elsevier Inc. All rights reserved.

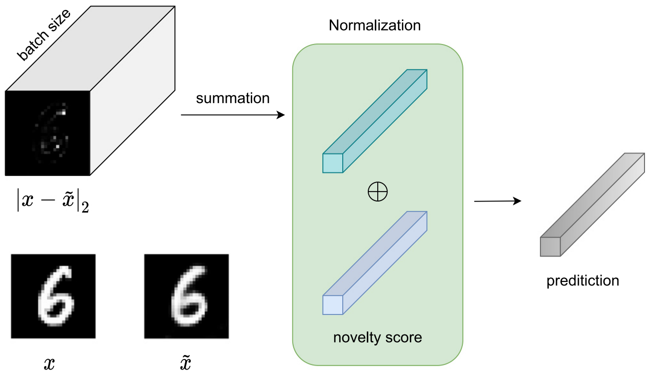

Fig. 1. The overall concept of this paper. Given a batch of images x, we train an autoencoder that learns how to produce the reconstructed images $\tilde{x}$. We not only rely on the reconstruction error, but we also incorporate the novelty score from our proposed module. The normalization module normalizes the input through batch dimension and outputs the prediction. ⊕ indicates element-wise addition.

extend to combine the existing object reconstruction error term with our own architecture when detecting objects as a way to better mimic natural tendencies in the human decision-making process.

To accomplish this, we begin by arguing that only utilizing the reconstruction error to measure the novelty is insufficient. As such, we propose the use of an additive novelty scoring (NS) module that is designed to work in conjunction with the preexisting reconstruction-based methods.

We then move on to address one-class anomaly detection. Many former works utilize a min–max scaling strategy that takes the scores obtained from the model and maps it onto a $[0, 1]$ scale. The scaled score, as well as the ground truth label, are used to compute the AUC during the testing stage. We adopt this strategy but choose to implement it within the model that is located inside the normalization module. The effectiveness of this module appears in our ablation study below.

Lastly, we evaluate our model using the MNIST, CIFAR-10 and Fashion-MNIST. We conclude that it outperforms the existing state-of-the-art models, and this improvement is especially significant when comparing test results on the CIFAR-10. The main contributions of our method are characterized as follows.

- We propose an additive module for novelty scoring that aims to *complement* reconstruction-based methods. We argue that using reconstruction error alone to measure the anomaly is not enough. Contrarily, the model is intuitively expected to have the capability to determine the normality directly like human perception.
- We introduce a normalization term to accomplish conclusive predictions that associate the novelty score with the reconstruction error. Particularly, we adopt min–max scaling strategy through the batch dimension and map the output scores to be within the target range.
- We present comprehensive experiments and compare our approach to several state-of-the-art methods for one-class anomaly detection. Furthermore, we extend the method into a multi-class setting to demonstrate model versatility.

2. Related work

2.1. Unsupervised learning and anomaly detection

Unsupervised learning (Barlow, 1989) is a type of machine learning that does not use labels as a part of the training process. Some well-known methods, such as Principal Component Analysis (PCA) and clustering, are used in unsupervised learning to determine the a priori probability distribution. The relation between unsupervised learning and anomaly detection is inseparable. For example, KDE (Bishop, 2006) utilizes PCA for one-class anomaly detection. Amer et al. (2013), Li et al. (2003), Wang et al. (2016) proposed some modifications to make one-class SVM (OCSVM) (Schölkopf et al., 2001) more appropriate for anomaly detection. Other works like (Eskin et al., 2002) have proposed ways to tackle unsupervised anomaly detection by using a geometric framework. There are two general types of input data for anomaly detection: sequential (*e.g.*audio, video and protein sequences) and non-sequential (*e.g.*image and signal) (Chalapathy and Chawla, 2019). We particularly address non-sequential data in this paper.

2.2. Kernel-based one-class anomaly detection

The overall objective of one-class anomaly detection is to differentiate between samples that belong to a defined normal class and the anomalous class (Ruff et al., 2018). In the one-class situation, the normal class contains all of the samples of a singular label, with the anomalous class containing the rest of the samples. We use two kernel-based one-class anomaly detection models, OCSVM and KDE, to help compare our results. KDE is one of the earliest implementations of one-class novelty detection. The model uses PCA to determine which subspace best represents the provided image. OCSVM, on the other hand, models the behavior of a given class in the latent space. Although these are different approaches, they ultimately share similar objectives. Of these two models, OCSVM performs significantly better on the MNIST, while KDE does slightly better on the CIFAR-10.

2.3. Deep learning-based approaches

We also compared our model against several others that use a deep learning framework to perform anomaly detection. These include models like denoising autoencoder (DAE) (Vincent et al., 2008), variational autoencoder (VAE) (Kingma and Welling, 2013), Autoregressive Novelty Detectors (AND) (Abati et al., 2019), and Deep Support Vector Data Description (DSVDD) (Ruff et al., 2018). In particular, AND focuses on sequential data, such as video data, using an autoregressive network for the latent representation space. These models utilize variations of an autoencoder network to complete the task at hand. Similar to the kernel-based models, the best performing deep learning based models depend upon the dataset. Although VAE and AND perform better on the MNIST, DSVDD significantly outperforms the rest of the models for the CIFAR-10. Most recently, Goodfellow et al. (2014) proposed generative adversarial networks (GAN) that have achieved impressive results on many tasks like image generation (Shaham et al., 2019), face recognition (Tran et al., 2017), image classification (Deng et al., 2019) and style transformation (Karras et al., 2019). Several works have proposed the implementation of GAN as an alternative approach to one-class anomaly detection. This includes AnoGAN (Schlegl et al., 2017), MinLGAN (Wang et al., 2018) and OCAGN (Perera et al., 2019). AnoGAN is the first work that uses GAN for anomaly detection. It trains a generative model and uses a discriminator that is designed to differentiate between the real and falsely generated data. MinLGAN proceeds the existing work laid out by AnoGAN, proposing minimum likelihood regularization to enhance the anomalous sample from the generator and except these samples far away from the normal distribution. OCGAN approaches this problem differently, taking an encoder and decoder as generators and introduces two discriminators into the adversarial training. In doing so, it forces the latent space representation to strictly decode the images that are similar to the normal class. This enhances the reconstruction error for the anomalous classes. Another recent work, MemAE (Gong et al., 2019), proposes the adaptation of an autoencoder with a memory module. Much like OCGAN, this algorithm also forces the latent representation to decode a normal-like image. The memory module records the principal patterns during training and uses attention-based weights to combine it with a fusion latent for the normal class.

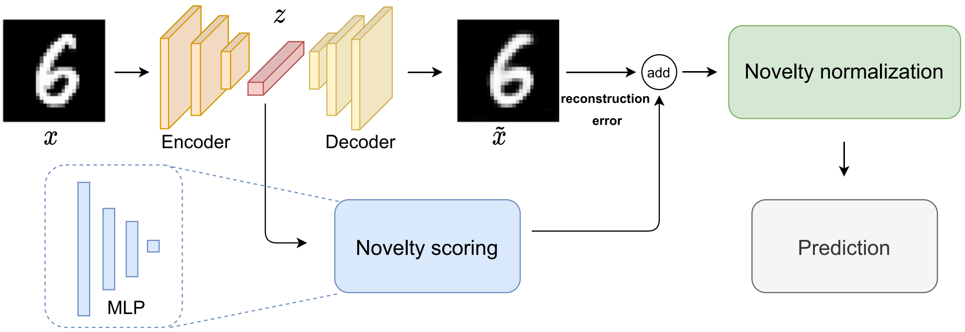

Fig. 2. Our proposed architecture for anomaly detection. The network is composed of an autoencoder, a novelty scoring module and a novelty normalization module. The optimization criterion incorporates the reconstruction error from the autoencoder and a novelty score from the novelty scoring module. The combined score then goes through the module of novelty normalization to obtain the final prediction.

3. Proposed method

3.1. Motivation

As stated previously, our model architecture is born out of necessity stemming from the current nature of data. There is no guarantee that every dataset will be labeled. Therefore, we have to design a system that can handle those types of situations in an unsupervised learning manner. The existing methods predict the anomaly score based on either the reconstruction error or discriminator output (Sabokrou et al., 2018). We choose to approach the problem of one-class anomaly detection from a perception standpoint, basing it on an autoencoder that incorporates additive modules.

On the other hand, due to the setup of the simulated dataset (which contains about 6,000 and 5,000 training examples for the MNIST and CIFAR-10, respectively), we are only left with approximately one-tenth of the original training data. Therefore, the model size is strictly constrained. For this reason, we design a simple and slight network with robust modules which can be trained end-to-end.

3.2. Proposed strategy

The full architecture is shown in Fig. 2. Our model consists of three main parts: an autoencoder model, a novelty scoring module, and normalization. We describe the details of each section below.

Autoencoder. An autoencoder is comprised of an encoder and a decoder. We include it as a way to minimize the distance between the model's inputs and outputs. The aim of an encoder is to learn a latent representation z for the input image x. The decoder then takes z as input and uses it to reconstruct the input image x. Following most practices, the network is optimized by the reconstruction error, which in this case is defined as the distance between the input and output images, or numerically as

$$\mathcal{L}_r = \|x - \tilde{x}\|_2 \,, \tag{1}$$

where $\tilde{x} = De(z)$ and $z = En(x)$. We choose to use mean square error as our measurement for this distance. Although there are several various autoencoders available, such as a denoising autoencoder (Vincent et al., 2008), we opt for a vanilla autoencoder in our experiment.

In addition to the reconstructed image $\tilde{x}$, the latent representation z is also necessary. For example, DSVDD uses the encoder as a transformation that maps the input image onto a high-dimensional point and deems it anomalous if the distance from the center is greater than the given radius R. We reuse the latent representation z as the input to our novelty scoring module since it is an appropriate feature vector.

Novelty scoring. The purpose of Novelty Scoring (NS) is to quantify the difference between the input image and the normal class. We design the NS module to predict a lower score of novelty for an arbitrary sample from the normal class and a higher score for a sample from any of the anomalous classes. To this end, we implement the NS with a two-layer MLP and apply the sigmoid function to the scalar output so that the novelty scoring will be constrained in $[0, 1]$. Specifically, we define the loss for learning the NS module as

$$\mathcal{L}_{ns} = \mathrm{MLP}(z; W) \,, \tag{2}$$

where the MLP transforms the latent representation z into a novelty score. Since (2) is expected to output a near 0 value for each sample from the normal class and the model learning has access to only those normal-class training data, the optimization would quickly run into a trivial solution that all parameters in W are reduced to 0. To prevent the scenario, we regulate W in L_0-norm inspired by Louizos et al. (2017), namely,

$$\|W\|_0 = \lim_{p \to 0} \sum_{i=1}^{n} |w_i|^p \,. \tag{3}$$

In practice, we realize (3) by counting the total number of non-zero entries in W instead. We count the number of parameters that are less than 1e-10 in numerical. We then obtain our total loss before joining it into the normalization module with

$$\mathcal{L}_{total} = \underbrace{(1 - \lambda_1) \times \mathcal{L}_r}_{\text{reconstruction error}} + \underbrace{\lambda_1 \times \mathcal{L}_{ns}}_{\text{novelty score}} - \underbrace{\lambda_2 \times \|W\|_0}_{\text{regularization term}} \,, \tag{4}$$

where $0 < \lambda_1, \lambda_2 < 1$ are hyperparameters to respectively control the loss terms and the effect of regularization.

During the backward pass on $\mathcal{L}_{total}$, $\mathcal{L}_r$ updates both the encoder and decoder parameters, and $\mathcal{L}_{ns}$ updates the encoder and MLP parameters. We want to minimize on (4) during the training stage because this points to optimal reconstruction and a low novelty score. Based on this, we can formulate the prediction with the inferred fixed model as follows:

$$p(x) = (1 - \lambda) \times \|x - \tilde{x}\|_2 + \lambda \times \mathrm{MLP}(En(x)), \tag{5}$$

where we set $\lambda = \lambda_1$ (as in (4)) in testing stage.

Novelty normalization. Using min–max scaling to map input ranges onto a target domain has been a popular strategy as of late. This strategy can be adopted during either the data pre-processing (Ruff et al., 2018) or post-processing phases (Abati et al., 2019). We diverge from this concept by implementing this technique within the loss term (4) instead. We incorporate this mapping into our normalization module in order to use the batch dimension to obtain the rescaling loss term. This is performed before backpropagation and normalizes the reconstruction error and novelty score at the same time. The normalization term is defined as

$$Reg(S_i) = \frac{S_i - S_{min}}{S_{max} - S_{min}} \,, \tag{6}$$

Table 1
One-class anomaly detection average AUCs in percentage on MNIST (digits from 0–9). All results except *Ours* are from Perera et al. (2019).

Method	0	1	2	3	4	5	6	7	8	9	MEAN
OCSVM (Schölkopf et al., 2001)	98.8	**99.9**	90.2	95.0	95.5	96.8	97.8	96.5	85.3	95.5	95.54
KDE (Bishop, 2006)	88.5	99.6	71.0	69.3	84.4	77.6	86.1	88.4	66.9	82.5	81.43
DAE (Vincent et al., 2008)	89.4	**99.9**	79.2	85.1	88.8	81.9	94.4	92.2	74.0	91.7	87.66
VAE (Kingma and Welling, 2013)	99.7	**99.9**	93.6	95.9	97.3	96.4	99.3	97.6	92.3	97.6	96.96
Pix CNN (Van den Oord et al., 2016)	53.1	99.5	47.6	51.7	73.9	54.2	59.2	78.9	34.0	66.2	61.83
GAN (Schlegl et al., 2017)	92.6	99.5	80.5	81.8	82.3	80.3	89.0	89.8	81.7	88.7	86.62
AnoGAN (Schlegl et al., 2017)	96.6	99.2	85.0	88.7	89.4	88.3	94.7	93.5	84.9	92.4	91.27
DSVDD (Ruff et al., 2018)	98.0	99.7	91.7	91.9	94.9	88.5	98.3	94.6	93.9	96.5	94.80
AND (Abati et al., 2019)	98.4	99.5	94.7	95.2	96.0	97.1	**99.1**	97.0	92.2	97.9	96.71
OCGAN (Perera et al., 2019)	**99.8**	**99.9**	94.2	96.3	**97.5**	**98.0**	**99.1**	98.1	93.9	**98.1**	97.50
Ours	**99.8**	**99.9**	**95.5**	**96.9**	97.3	97.5	98.9	**98.4**	**94.7**	97.6	**97.65**

Table 2
One-class anomaly detection average AUCs in percentage for each class on CIFAR-10. The Car class is annotated as Automobile in the original dataset. All experimental results except *Ours* are from Perera et al. (2019).

Method	PLANE	CAR	BIRD	CAT	DEER	DOG	FROG	HORSE	SHIP	TRUCK	MEAN
OCSVM	63.0	44.0	64.9	48.7	73.5	50.0	72.5	53.3	64.9	50.8	58.56
KDE	65.8	52.0	65.7	49.7	72.7	49.6	75.8	56.4	68.0	54.0	60.97
DAE	41.1	47.8	61.6	56.2	72.8	51.3	68.8	49.7	48.7	37.8	53.58
VAE	70.0	38.6	**67.9**	53.5	74.8	52.3	68.7	49.3	69.6	38.6	58.33
Pix CNN	78.8	42.8	61.7	57.4	51.1	57.1	42.2	45.4	71.5	42.6	55.06
GAN	70.8	45.8	66.4	51.0	72.2	50.5	70.7	47.1	71.3	45.8	59.16
AnoGAN	67.1	54.7	52.9	54.5	65.1	60.3	58.5	62.5	75.8	66.5	61.79
DSVDD	61.7	65.9	50.8	59.1	60.9	**65.7**	67.7	**67.3**	75.9	73.1	64.81
AND	71.7	49.4	66.2	52.7	73.6	50.4	72.6	56.0	68.0	56.6	61.72
OCGAN	75.7	53.1	64.0	62.0	72.3	62.0	72.3	57.5	**82.0**	55.4	65.63
Ours	**79.6**	**69.9**	66.1	**62.5**	**75.6**	65.0	**77.1**	66.3	78.4	**74.3**	**71.48**

$$S_i = (1-\lambda_1) \times \mathcal{L}_r(x_i) + \lambda_1 \times \mathcal{L}_{ns}(x_i)\,, \tag{7}$$

where S_{max} and S_{min} are the respective maximum and minimum S_i for each batch in the training stage. We keep each S_{max} and S_{min} and take average to obtain $\bar{S}_{max}$ and $\bar{S}_{min}$. With the learned $\bar{S}_{max}$ and $\bar{S}_{min}$, we can readily normalize each testing example using (6) for novelty normalization during inference.

4. Experiments

In this section, we evaluate our model on the MNIST (LeCun et al., 1998b), CIFAR-10 (Krizhevsky et al., 2009) and Fashion-MNIST (Xiao et al., 2017), which are standard benchmarks for image classification. We follow the practice in previous work to create simulated data for one-class anomaly detection. To quantify the effectiveness, we compare our result using the Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC) curve performance measure.

Note that although MemAE (Gong et al., 2019) is also evaluated on MNIST and CIFAR-10, the evaluations are carried out with distinct training and testing splits from those in the other methods included in our comparison. For the sake of consistency to the majority, MemAE is not included in our experiments.

4.1. Datasets

4.1.1. Datasets and setup

MNIST and CIFAR-10 are the most commonly used datasets for image classification. More recently, a variant dataset of MNIST named Fashion-MNIST is released. We show some training examples in Fig. 3. We perform target transform to define the normal and anomalous class. In other words, we adapt training and testing splits given by the dataset and select one of the classes as the normal class. The remaining classes are treated as anomalies. For the training stage, we strictly use the instances of the normal class from the training split without any data augmentation to train our model. For the testing stage, we use the entire testing split to evaluate the performance.

4.1.2. MNIST

The MNIST dataset contains images of handwritten digits from 0 to 9. The images all have a resolution of 28 × 28 pixels. This dataset is one of the most widely used datasets for image classification. The training set has an original size of 60,000 images with approximately 6,000 images per class. We isolate one class as the normal class and treat the rest as a part of the anomalous class. The testing set contains 10,000 images with 10% of them representing the normal class and the other 90% representing the anomalous class.

4.1.3. CIFAR-10

The CIFAR-10 dataset contains images from 10 different classes, each of which has a resolution of 32 × 32 × 3 pixels. Although this dataset is more complex due to the diversity and general nature of images displaying everyday objects and animals, it is also widely used to train and test image detection models. The dataset has a training set size of 50,000 images, with 5,000 images for every class. The test set has 10,000 examples, with the same splits between the normal and anomalous classes as seen in the MNIST.

4.1.4. Fashion-MNIST

The Fashion-MNIST is a similar dataset to MNIST that replaces digits with fashion objects. These two datasets contain the same resolution for each instance in train/test splits. By the resemblance, we use the same network configuration and hyper-parameters. That is, we adopt our previous setting for MNIST to tackle one-class anomaly detection on Fashion-MNIST.

4.2. Network architecture

Our autoencoder follows the LeNet-based (LeCun et al., 1998a) architecture, where we construct the encoder (and thus the decoder) by using a similar network as in Ruff et al. (2018). The encoder is made up of 2D-convolutional layers and followed by a dense layer. The decoder is symmetric to the encoder and composed of transposed 2D-convolutional layers. Each convolutional layer is followed by a leaky ReLU activation and a 2 × 2 pixel pooling. We use two convolutional

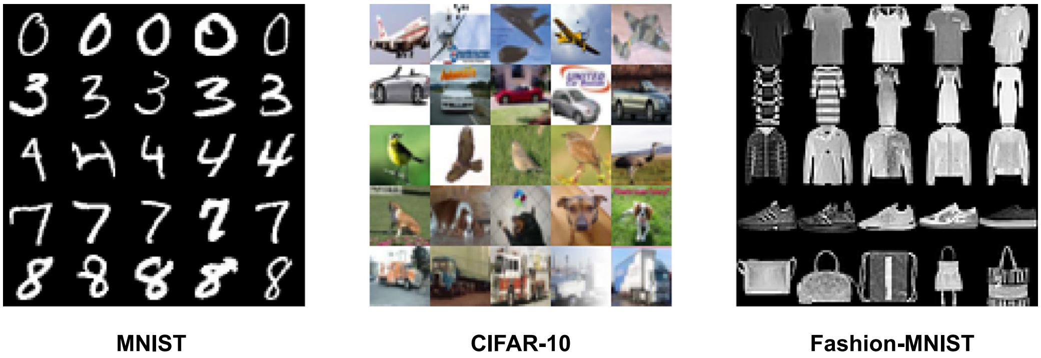

Fig. 3. Examples of training images from the three datasets, namely, MNIST, CIFAR-10 and Fashion-MNIST. Each row indicates the same object class. It can be observed that images of CIFAR-10 are more complex than those of MNIST and Fashion-MNIST.

Table 3
One-class anomaly detection average AUCs in percentage on Fashion-MNIST. GEOM — indicates the approach without Dirichlet post-processing. All experimental results except *Ours* are from Golan and El-Yaniv (2018) and Bergman and Hoshen (2020).

Method	mAUC
DSVDD (Ruff et al., 2018)	92.8
DAGMM (Zong et al., 2018)	51.8
DSEBM (Zhai et al., 2016)	86.6
GEOM- (Golan and El-Yaniv, 2018)	79.8
GEOM (Golan and El-Yaniv, 2018)	93.5
GOAD (Bergman and Hoshen, 2020)	94.1
Ours	**94.7**

layers for MNIST/Fashion-MNIST and three convolutional layers for CIFAR-10. We set the dimension of latent representation z equal to 32 and 64 for MNIST/Fashion-MNIST and CIFAR-10, respectively. The kernel size of the convolutional layers is 5 × 5 pixels with a zero-padding size of 2 pixels. In the NS module, we use two and three dense layers, the former for MNIST and the latter for CIFAR-10, both of which are followed by a sigmoid activation in the last layer. We set the loss parameter $\lambda_1 = 0.5$ and the weights regularization parameter $\lambda_2 =$ 1e-5 for our experiment. We use the Adam optimizer (Kingma and Ba, 2014) that decays at a learning rate of 0.1 for every 30 steps. All of our networks are trained from scratch.[1]

4.3. Results

As seen in Tables 1 and 2, our model either matches or outperforms the other state-of-the-art models in six classes, as well as the overall mean AUC, for each of the two datasets. This improvement in the AUC of Table 1, however, is less noticeable when testing on the MNIST. Our model is comparable with other models in two classes, and the other four classes where we outperform only does so by less than one percentage point.

This trend continues in Table 2. Our model outperforms the other models in six classes as well as the overall mean AUC. We surpass OCGAN's top mean AUC by over 5%, obtaining a mean AUC of 71.48%, compared to DSVDD's mean AUC of 61.72% and OCGAN's of 65.63%. Unlike the results from the MNIST test, where the range of top AUCs is fairly narrow, our model tops the next best model's AUC by upwards of 4%. This could be attributed to the difference in the complexity of the two datasets. Since the CIFAR-10 is more complex, the improvement in anomaly detection would be more apparent.

We also observe that our range of the AUCs is narrower across all classes when testing on CIFAR-10. The range of AUCs generated by our model is 17.1%, compared to DSVDD's range of 25.1% and OCGAN's range of 28.9%. This is notable because of the complexity of the CIFAR-10. We were not only able to increase the accuracy of the results, but also the precision of it as well.

Table 3 shows the experimental results on Fashion-MNIST. We compare our method with several techniques of anomaly detection reported in Golan and El-Yaniv (2018), Bergman and Hoshen (2020). Notice that both GEOM and GOAD require Dirichlet post-processing to achieve good performance, while our approach alone yields state-of-the-art result.

We only analyze the design principle of GEOM in that the subsequent technique GOAD indeed shares the same core idea. While their results on CIFAR-10 are not included in Table 2, the two classification-based anomaly detection approaches, GEOM and GOAD, achieve mean AUC of 86% and 88.2%, respectively. Compared with evaluating on Fashion-MNIST, their performance gains on CIFAR-10 versus ours are manifest. We explain our observations as follows. Different from reconstruction-based methods (including ours) for anomaly detection, GEOM considers geometry-based pretext tasks and carries out self-supervised learning to extensively exploit the characteristics of CIFAR-10 dataset. For each image of a specific object class, GEOM applies standard image processing methods such as rotation, flipping, and translation to yield totally 72 different transformations. The strategy would result in a pretext task to predict the underlying transformation of each transformed sample. In inference, GEOM applies all the 72 geometric transformations to each testing image and estimates the "normality score" by accumulating all the pretext predictions from the 72 transformed images. Such a self-supervised scheme has been proposed for unsupervised feature learning. However, it essentially learns to memorize the geometric characteristics of each object category, including object's primary location and orientation, rather than to detect the local abnormalities of interest. In other words, the reconstruction-based techniques have the potential to not only classify but also localize the anomaly regions, which is critical for practical applications. In the case of CIFAR-10, GEOM and GOAD are shown to be effective in distinguishing each particular class from the others, but it is not clear conceptually why their use of geometry-based pretext can solve the real-life problem of anomaly detection. Besides the generalization issue, the network architecture for GEOM and GOAD is the Wide Residual Network (Zagoruyko and Komodakis, 2016), a ResNet variant that includes tunable depth and width hyperparameters to seek a refined model for the particular dataset. In comparison, our method adopts a relatively shallow network similar to DSVDD to emphasize the usefulness of the proposed reconstruction-based approach to anomaly detection.

4.4. Ablation study

Previous work (Perera et al., 2019) states that the resulting the AUC from the autoencoder is already fairly high when testing on the MNIST. They eventually boost the AUC by 1.8% after introducing

[1] We implement our method using Pytorch.

Table 4
Our ablation study on the MNIST and CIFAR-10 testing splits. The first row shows the mean AUC of the autoencoder only model and the following rows show the results from adding the additional modules step by step.

Model	MNIST	CIFAR-10
Autoencoder only	95.54	59.74
+ Normalization	96.67	65.61
+ NS	97.65	71.48

Table 5
Our Multiclass anomaly detection results. We test this using the three worst classes from our single-class test: bird, cat, and dog.

Normal class	BIRD + CAT + DOG
Autoencoder only	59.81
+ Normalization	60.64
+ NS	62.77

the discriminators and the classifier. We present our experiment that compares variants of the proposed modules for assessing the relative effectiveness of each module in Table 4.

We notice that the result of our ablation study on the MNIST is extremely similar to the one done in Perera et al. (2019). We see a minimal increase in the AUC when adding on novelty normalization and novelty scoring (NS). Comparatively, the improvement in AUCs are significant when testing on the CIFAR-10. We see an increase in the AUC of nearly 6% just by adding novelty normalization. The AUC increases by almost another 6% when adding on NS.

This trend of a minimal increase in performance with the MNIST might be due to the simplicity of the dataset. Since the autoencoder is already generating such high AUCs, although adding on additional parts to the model might still increase performance, that increase is minuscule when compared to testing on CIFAR-10. The CIFAR-10 is complex, so adding the additional parts to the model will result in a more perceivable increase in performance. It is also worth noting that the increase in the AUC after adding normalization and then NS is symmetric for both datasets. In MNIST, each addition results in an AUC increase of about 1%, and in CIFAR-10 every extra addition to the model results in an increase of roughly 6%.

4.5. Multiclass anomaly detection

To evaluate a more challenging setting for anomaly detection, we attempt to generalize one-class anomaly detection to a multiclass anomaly detection scenario. In this situation, multiple classes from the dataset are defined to be the normal class (*e.g.*, 0 indicates the normal classes and 1 is for the remaining classes). The original labels are stripped and defined as either normal or an anomaly. We test this scenario on the CIFAR-10 and use the three classes with the worst AUC results to populate the normal class.

DSVDD and OCGAN intuitively does not work under the multiclass anomaly detection setting. The DSVDD criterion for anomaly detection essentially relies on the distance to the center. OCGAN forces the latent representation into a space to generate the normal class-like image through adversarial training. Both strategies necessitate the usage of a singular class during training.

Our results from this experiment are shown in Table 5. Although they do not appear to increase the AUC for any of the three classes from our original model, the results are comparable with the other methods that we used to compare the one-class situation with. Furthermore, it shows that our method improves upon the basic autoencoder model, regardless of if it is in the one-class or multiclass scenario. We see improvements to the model after adding on novelty normalization and NS in both instances. This suggests that generalizing our model to fit a multiclass situation works just as well as the state-of-the-art models that are already out there.

5. Conclusion

In this paper, we presented an alternative approach to one-class anomaly detection. We built upon existing architectures that incorporate an autoencoder by building an additive novelty scoring module and novelty normalization into our model. The purpose of using novelty scoring was to figure out the difference between any input image and the normal class. We also made a point to utilize one-class learning with novelty normalization in our model, since in practice only the samples of the normal class are provided in training the neural network model.

We also showed that our results outperformed other state-of-the-art methods. Not only were our results more accurate, but they were also more precise. Furthermore, our ablation study justified the importance of novelty scoring and normalization to our model.

Most notably, however, we demonstrated that our model can be used in a generalized setting. Although our main focus was on one-class situations, further testing proved how it could be utilized in multiclass circumstances.

Our experiment proved that the proposed model is effective in anomaly detection. In future work, we hope to improve upon our current model to better handle more complex datasets for single and multiclass scenarios.

CRediT authorship contribution statement

Jhih-Ciang Wu: Surveying the related work, Carrying out the coding and experiments, Contribute to the discussion of developing, Proposed method, Writing of this manuscript. **Sherman Lu:** Surveying the related work, Carrying out the coding and experiments, Contribute to the discussion of developing, Proposed method, Writing of this manuscript. **Chiou-Shann Fuh:** Co-adviser of first author, Contribute to the discussion of developing, Proposed method, Writing of this manuscript. **Tyng-Luh Liu:** Co-adviser of first author, Contribute to the discussion of developing, Proposed method, Writing of this manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the MOST, Taiwan under Grant 110-2634-F-001-009. We are also grateful to the *National Center for High-performance Computing* for providing computational resources and facilities.

References

Abati, D., Porrello, A., Calderara, S., Cucchiara, R., 2019. Latent space autoregression for novelty detection. In: Computer Vision and Pattern Recognition. pp. 481–490.

Amer, M., Goldstein, M., Abdennadher, S., 2013. Enhancing one-class support vector machines for unsupervised anomaly detection. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. ACM, pp. 8–15.

Barlow, H.B., 1989. Unsupervised learning. Neural Comput. 1 (3), 295–311.

Bergman, L., Hoshen, Y., 2020. Classification-Based Anomaly Detection for General Data, In: International Conference on Learning Representations.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Chalapathy, R., Chawla, S., 2019. Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Comput. Surv. (CSUR) 41 (3), 15.

Deng, J., Guo, J., Xue, N., Zafeiriou, S., 2019. Arcface: Additive angular margin loss for deep face recognition, In: Conference on Computer Vision and Pattern Recognition, pp. 4690–4699.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S., 2002. A geometric framework for unsupervised anomaly detection. In: Applications of Data Mining in Computer Security. Springer, pp. 77–101.

Golan, I., El-Yaniv, R., 2018. Deep anomaly detection using geometric transformations. Adv. Neural Inf. Process. Syst. 31, 9758–9769.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d., 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, In: International Conference on Computer Vision, pp. 1705–1714.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680.

Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In: Computer Vision and Pattern Recognition. 2, IEEE, pp. 1735–1742.

Karras, T., Laine, S., Aila, T., 2019. A style-based generator architecture for generative adversarial networks, In: Conference on Computer Vision and Pattern Recognition, pp. 4401–4410.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images. Technical Report, Citeseer.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998a. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

LeCun, Y., Cortes, C., Burges, C., 1998b. MNIST Handwritten digit database, 1998. URL http://www.research. att. com/~yann/ocr/mnist.

Li, K.-L., Huang, H.-K., Tian, S.-F., Xu, W., 2003. Improving one-class SVM for anomaly detection. In: International Conference on Machine Learning and Cybernetics. 5, IEEE, pp. 3077–3081.

Louizos, C., Welling, M., Kingma, D.P., 2017. Learning sparse neural networks through L_0 regularization. arXiv preprint arXiv:1712.01312.

Maiorano, F., Petrosino, A., 2016. Granular trajectory based anomaly detection for surveillance. In: International Conference on Pattern Recognition. IEEE, pp. 2066–2072.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al., 2016. Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems. pp. 4790–4798.

Perera, P., Nallapati, R., Xiang, B., 2019. Ocgan: One-class novelty detection using gans with constrained latent representations. In: Computer Vision and Pattern Recognition. pp. 2898–2906.

Pidhorskyi, S., Almohsen, R., Doretto, G., 2018. Generative probabilistic novelty detection with adversarial autoencoders. In: Advances in Neural Information Processing Systems. pp. 6822–6833.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M., 2018. Deep one-class classification, In: International Conference on Machine Learning, pp. 4393–4402.

Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E., 2018. Adversarially learned one-class classifier for novelty detection. In: Computer Vision and Pattern Recognition. pp. 3379–3388.

Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G., 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging. Springer, pp. 146–157.

Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. Neural Comput. 13 (7), 1443–1471.

Shaham, T.R., Dekel, T., Michaeli, T., 2019. Singan: Learning a generative model from a single natural image, In: International Conference on Computer Vision, pp. 4570–4580.

Ten, C.-W., Hong, J., Liu, C.-C., 2011. Anomaly detection for cybersecurity of the substations. IEEE Trans. Smart Grid 2 (4), 865–873.

Tran, L., Yin, X., Liu, X., 2017. Disentangled representation learning gan for pose-invariant face recognition, In: Conference on Computer Vision and Pattern Recognition, pp. 1415–1424.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: International Conference on Machine Learning. ACM, pp. 1096–1103.

Wang, C., Zhang, Y.-M., Liu, C.-L., 2018. Anomaly detection via minimum likelihood generative adversarial networks. In: International Conference on Pattern Recognition. IEEE, pp. 1121–1126.

Wang, S., Zhu, E., Yin, J., Porikli, F., 2016. Anomaly detection in crowded scenes by SL-HOF descriptor and foreground classification. In: International Conference on Pattern Recognition. IEEE, pp. 3398–3403.

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. In: Wilson, R.C., Hancock, E.R., Smith, W.A.P. (Eds.), British Machine Vision Conference 2016, York, UK, September 19-22, 2016. BMVA Press.

Zhai, S., Cheng, Y., Lu, W., Zhang, Z., 2016. Deep structured energy based models for anomaly detection. arXiv preprint arXiv:1605.07717.

Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H., 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection, In: International Conference on Learning Representations.