SimpleNet: A Simple Network for Image Anomaly Detection and Localization

Zhikang Liu[1] Yiming Zhou[2] Yuansheng Xu[2] Zilei Wang[1*]
Department of Automation, University of Science and Technology of China[1]
Meka Technology Co.,Ltd[2]
lzk@mail.ustc.edu.cn zhouyiming.donal@gmail.com xys-tc@hotmail.com zlwang@ustc.edu.cn

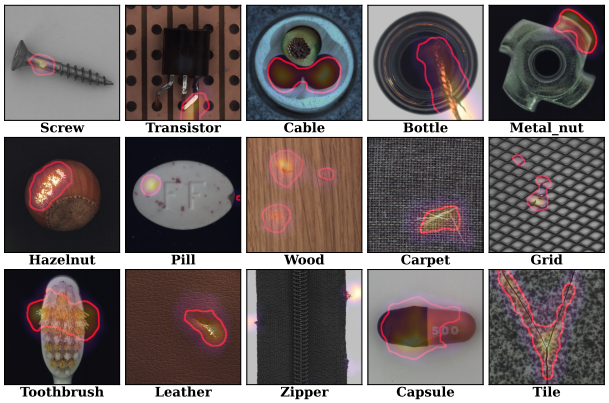


Figure 1. Visualization of samples in MVTec AD. The produced anomaly maps superimposed on the images. Anomaly region of high anomaly score is colored with orange. The red boundary denotes contours of actual segmentation maps for anomalies.

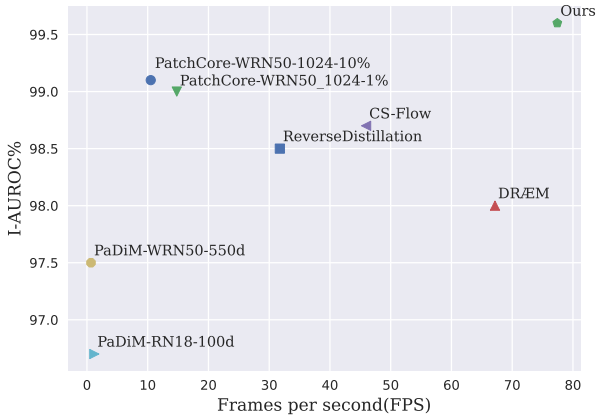


Figure 2. Inference speed (FPS) versus I-AUROC on MVTec AD benchmark. SimpleNet outperforms all previous methods on both accuracy and efficiency by a large margin.

Abstract

We propose a simple and application-friendly network (called SimpleNet) for detecting and localizing anomalies. SimpleNet consists of four components: (1) a pre-trained Feature Extractor that generates local features, (2) a shallow Feature Adapter that transfers local features towards target domain, (3) a simple Anomaly Feature Generator that counterfeits anomaly features by adding Gaussian noise to normal features, and (4) a binary Anomaly Discriminator that distinguishes anomaly features from normal features. During inference, the Anomaly Feature Generator would be discarded. Our approach is based on three intuitions. First, transforming pre-trained features to target-oriented features helps avoid domain bias. Second, generating synthetic anomalies in feature space is more effective, as defects may not have much commonality in the image space. Third, a simple discriminator is much efficient and practical. In spite of simplicity, SimpleNet outperforms previous methods quantitatively and qualitatively. On the MVTec AD benchmark, SimpleNet achieves an anomaly detection AUROC of 99.6%, reducing the error by 55.5% compared to the next best performing model. Furthermore, SimpleNet is faster than existing methods, with a high frame rate of 77 FPS on a 3080ti GPU. Additionally, SimpleNet demonstrates significant improvements in performance on the One-Class Novelty Detection task. Code: https://github.com/DonaldRR/SimpleNet.

1. Introduction

Image anomaly detection and localization task aims to identify abnormal images and locate abnormal subregions. The technique to detect the various anomalies of interest has a broad set of applications in industrial inspection [3, 6]. In industrial scenarios, anomaly detection and localization is especially hard, as abnormal samples are scarce and anomalies can vary from subtle changes such as thin scratches to large structural defects, *e.g.* missing parts. Some examples from the MVTec AD benchmark [3] along with results from our proposed method are shown in Figure 1. This situation

*Corresponding author

prohibits the supervised methods from approaching.

Current approaches address this problem in an unsupervised manner, where only normal samples are used during the training process. The reconstruction-based methods [10, 21, 31], synthesizing-based methods [17, 30], and embedding-based methods [6, 22, 24] are three main trends for tackling this problem. The reconstruction-based methods such as [21,31] assume that a deep network trained with only normal data cannot accurately reconstruct anomalous regions. The pixel-wise reconstruction errors are taken as anomaly scores for anomaly localization. However, this assumption may not always hold, and sometimes a network can "generalize" so well that it can also reconstruct the abnormal inputs well, leading to misdetection [10, 19]. The synthesizing-based methods [17, 30] estimate the decision boundary between the normal and anomalous by training on synthetic anomalies generated on anomaly-free images. However, the synthesized images are not realistic enough. Features from synthetic data might stray far from the normal features, training with such negative samples could result in a loosely bounded normal feature space, meaning indistinct defects could be included in in-distribution feature space.

Recently, the embedding-based methods [6, 7, 22, 24] achieve state-of-the-art performance. These methods use ImageNet pre-trained convolutional neural networks (CNN) to extract generalized normal features. Then a statistical algorithm such as multivariate Gaussian distribution [6], normalizing flow [24], and memory bank [22] is adopted to embed normal feature distribution. Anomalies are detected by comparing the input features with the learned distribution or the memorized features. However, industrial images generally have a different distribution from ImageNet. Directly using these biased features may cause mismatch problems. Moreover, the statistical algorithms always suffer from high computational complexity or high memory consumption.

To mitigate the aforementioned issues, we propose a novel anomaly detection and localization network, called SimpleNet. SimpleNet takes advantage of the synthesizing-based and the embedding-based manners, and makes several improvements. First, instead of directly using pre-trained features, we propose to use a feature adaptor to produce target-oriented features which reduce domain bias. Second, instead of directly synthesizing anomalies on the images, we propose to generate anomalous features by posing noise to normal features in feature space. We argue that with a properly calibrated scale of the noise, a closely bounded normal feature space can be obtained. Third, we simplify the anomalous detection procedure by training a simple discriminator, which is much more computational efficient than the complex statistical algorithms adopted by the aforementioned embedding-based methods. Specifically, SimpleNet makes use of a pre-trained backbone for normal feature extraction followed by a feature adapter to transfer the feature into the target domain. Then, anomaly features are simply generated by adding Gaussian noise to the adapted normal features. A simple discriminator consisting of a few layers of MLP is trained on these features to discriminate anomalies.

SimpleNet is easy to train and apply, with outstanding performance and inference speed. The proposed SimpleNet, based on a widely used WideResnet50 backbone, achieves 99.6 % AUROC on MVTec AD while running at 77 fps, surpassing the previous best-published anomaly detection methods on both accuracy and efficiency, see Figure 2. We further introduce SimpleNet to the task of One-Class Novelty Detection to show its generality. These advantages make SimpleNet bridge the gap between academic research and industrial application. Code will be publicly available.

2. Related Work

Anomaly detection and localization methods can be mainly categorized into three types, *i.e.*, the reconstruction-based methods, the synthesizing-based methods, and the embedding-based methods.

Reconstruction-based methods hold the insight that anomalous image regions should not be able to be properly reconstructed since they do not exist in the training samples. Some methods [10] utilize generative models such as auto-encoders and generative adversarial networks [11] to encode and reconstruct normal data. Other methods [13,21,31] frame anomaly detection as an inpainting problem, where patches from images are masked randomly. Then, neural networks are utilized to predict the erased information. Integrating structural similarity index (SSIM) [29] loss function is widely used in training. An anomaly map is generated as pixel-wise difference between the input image and its reconstructed image. However, if anomalies share common compositional patterns (e.g. local edges) with the normal training data or the decoder is "too strong" for decoding some abnormal encodings well, the anomalies in images are likely to be reconstructed well [31].

Synthesizing-based methods typically synthesize anomalies on anomaly-free images. DRÆM [30] proposes a network that is discriminatively trained in an end-to-end manner on synthetically generated just-out-of-distribution patterns. CutPaste [17] proposes a simple strategy to generate synthetic anomalies for anomaly detection that cuts an image patch and pastes at a random location of a large image. A CNN is trained to distinguish images from normal and augmented data distributions. However, the appearance of the synthetic anomalies does not closely match the real anomalies'. In practice, as defects are various and unpredictable, generating an anomaly set that includes all outliers is impossible. Instead of synthesizing anomalies on images, with the proposed SimpleNet, negative samples

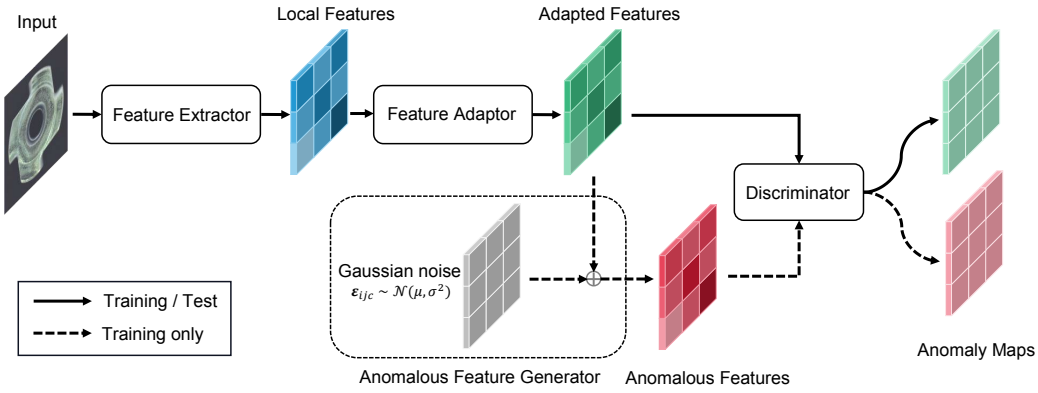


Figure 3. Overview of the proposed SimpleNet. In the training phase, nominal samples are fed into a pre-trained *Feature Extractor* to get local features. Then, a *Feature Adaptor* is utilized to adapt pre-trained features into the target domain. Anomalous features are synthesized by adding Gaussian noise to the adapted features. The adapted features and the anomalous features and used as positive and negative samples respectively to train the final *Discriminator*. The *Anomalous Feature Generator* is removed at inference.

are synthesized in the feature space.

Embedding-based methods achieve state-of-the-art performance recently. These methods embed normal features into a compressed space. The anomalous features are far from the normal clusters in the embedding space. Typical methods [6,7,22,24] utilize networks that are pre-trained on ImageNet for feature extraction. With a pre-trained model, PaDiM [6] embeds the extracted anomaly patch features by multivariate Gaussian distribution. PatchCore [22] uses a maximally representative memory bank of nominal patch features. Mahalanobis distance or maximum feature distance is adopted to score the input features in testing. However, industrial images generally have a different distribution from ImageNet. Directly using pre-trained features may cause a mismatch problem. Moreover, either computing the inverse of covariance [6] or searching through the nearest neighbor in the memory bank [22] limits the real-time performance, especially for edge devices.

CS-Flow [24], CFLOW-AD [12], and DifferNet [23] propose to transform the normal feature distribution into Gaussian distribution via normalizing flow (NF) [20]. As normalizing flow can only process full-sized feature maps, i.e., down sample is not allowed and the coupling layer [9] consumes a few times of memory than the normal convolutional layer, these methods are memory consuming. Distillation methods [4, 7] train a student network to match the outputs of a fixed pre-trained teacher network with only normal samples. A discrepancy between student and teacher output should be detected given an anomalous query. The computational complexity is doubled as an input image should pass through both the teacher and the student.

SimpleNet overcomes the aforementioned problems. SimpleNet uses a feature adaptor that performs transfer learning on the target dataset to alleviate the bias of pre-trained CNNs. SimpleNet proposes to synthesize anomalous in the feature space rather than directly on the images. SimpleNet follows a single-stream manner at inference and is totally constructed by conventional CNN blocks which facilitate fast training, inference, and industrial application.

3. Method

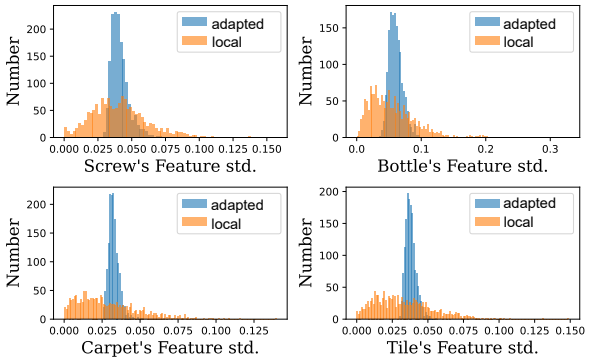
The proposed SimpleNet is elaborately introduced in this section. As illustrated in Figure 3, SimpleNet consist of a *Feature Extractor*, a *Feature Adaptor*, an *Anomalous Feature Generator* and a *Discriminator*. The *Anomalous Feature Generator* is only used during training, thus SimpleNet follows a single stream manner at inference. These modules will be described below in sequence.

3.1. Feature Extractor

Feature Extractor acquires local feature as in [22]. We reformulate the process as follows. We denote the training set and test set as $\mathcal{X}_{train}$ and $\mathcal{X}_{test}$. For any image $x_i \in \mathbb{R}^{H\times W\times 3}$ in $\mathcal{X}_{train} \bigcup \mathcal{X}_{test}$, the pre-trained network ϕ extracts features from different hierarchies, as normally done with ResNet-like backbone. Since pre-trained network is biased towards the dataset in which it is trained, it is reasonable to choose only a subset of levels for the target dataset. Formally, we define L the subset including the indexes of hierarchies for use. The feature map from level $l \in L$ is denoted as $\phi^{l,i} \sim \phi^l(x_i) \in \mathbb{R}^{H_l\times W_l\times C_l}$, where H_l, W_l and C_l are the height, width and channel size of the feature map. For an entry $\phi^{l,i}_{h,w} \in \mathbb{R}^{C_l}$ at location (h, w), its neighborhood with patchsize p is defined as

$$\mathcal{N}_p^{(h,w)} = \{(h', y') | h' \in [h - \lfloor p/2 \rfloor, ..., h + \lfloor p/2 \rfloor], \\ y' \in [w - \lfloor p/2 \rfloor, ..., w + \lfloor p/2 \rfloor]\} \quad (1)$$

Aggregating the features within the neighborhood $\mathcal{N}_p^{h,w}$ with aggregation function f_{agg} (use adaptive average pool-



dimension = channel

Figure 4. Histogram of standard deviation along each dimension of local feature and adapted feature. The adapted feature space becomes more compact when training with anomalous features.

ing here) results in the local feature $z^{l,i}_{h,w}$, as

$$z^{l,i}_{h,w} = f_{agg}(\{\phi^{l,i}_{h',y'} | (h', y') \in \mathcal{N}^{h,w}_p\}) \tag{2}$$

To combine features $z^{l,i}_{h,w}$ from different hierarchies, all feature maps are linearly resized to the same size (H_0, W_0), i.e. the size of the largest one. Simply concatenating the feature maps channel-wise gives the feature map $o^i \in \mathbb{R}^{H_0 \times H_0 \times C}$. The process is defined as

$$o^i = f_{cat}(resize(z^{l',i}, (H_0, W_0)) | l' \in L \tag{3}$$

we define $o^i_{h,w} \in \mathbb{R}^C$ as the entry of o^i at location (h, w).

We simplify the above expressions as

$$o^i = F_\phi(x^i) \tag{4}$$

where F_ϕ is the Feature Extractor.

3.2. Feature Adaptor

As industrial images generally have a different distribution from the dataset used in backbone pre-training, we adopt a Feature Adaptor G_θ to transfer the training features to the target domain. The Feature Adaptor G_θ projects local feature $o_{h,w}$ to adapted feature $q_{h,w}$ as

$$q^i_{h,w} = G_\theta(o^i_{h,w}) \tag{5}$$

The Feature Adaptor can be made up of simple neural blocks such as a fully-connected layer or multi-layer perceptron (MLP). We experimentally find that a single fully-connected layer yields good performance.

3.3. Anomalous Feature Generator

To train the Discriminator to estimate the likelihood of samples being normal, the easiest way is sampling negative samples, i.e. defect features, and optimizing it together with normal samples. The lack of defects makes the sampling distribution estimation intractable. While [17, 18, 30] relying on extra data to synthesize defect images, we add simple noise on normal samples in the feature space, claiming that it outperforms those manipulated methods.

The anomalous features are generated by adding Gaussian noise on the normal features $q^i_{h,w} \in \mathbb{R}^C$. Formally, a noise vector $\epsilon \in \mathbb{R}^C$ is sampled, with each entry following an i.i.d. Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. The anomalous feature $q^{i-}_{h,w}$ is fused as

$$q^{i-}_{h,w} = q^i_{h,w} + \epsilon \tag{6}$$

Figure 4 illustrates the influence of anomalous features on four classes of MVTec AD. We can see that the standard deviation along each dimension of the adapted features tends to be consistent. Thus, the feature space tends to be compact when distinguishing anomalous features from normal features.

3.4. Discriminator

The Discriminator D_ψ works as a normality scorer, estimating the normality at each location (h, w) directly. Since negative samples are generated along with normal features $\{q^i | x^i \in \mathcal{X}_{train}\}$, they are both fed to the Discriminator during training. The Discriminator expects positive output for normal features while negative for anomalous features. We simply use a 2-layer multi-layer perceptron (MLP) structure as common classifiers do, estimating normality as $D_\psi(q_{h,w}) \in \mathbb{R}$.

3.5. Loss function and Training

A simple truncated $l1$ loss is derived as

$$l^i_{h,w} = \max(0, th^+ - D_\psi(q^i_{h,w})) + \max(0, -th^- + D_\psi(q^{i-}_{h,w})) \tag{7}$$

Zielwert für Normal --> >0.5

Zielwert für Anomale --> <-0.5

th^+ and th^- are truncation terms preventing overfitting. They are set to 0.5 and -0.5 by default. The training objective is

$$\mathcal{L} = \min_{\theta,\psi} \sum_{x^i \in \mathcal{X}_{train}} \sum_{h,w} \frac{l^i_{h,w}}{H_0 * W_0} \tag{8}$$

Warum die Normierung?

We will experimentally evaluate the proposed truncated $l1$ loss function with the widely used cross-entropy loss in the experiments section. The pseudo-code of the training procedure is shown in Algorithm 1.

3.6. Inference and Scoring function

The Anomalous Feature Generator is discarded at inference. Note that the remaining modules can be stacked into an end-to-end network. We feed each $x_i \in \mathcal{X}_{test}$ into the aforementioned Feature Extractor F_ϕ and the Feature Adaptor G_θ sequentially to get adapted features $q^i_{h,w}$ as in Equation 5. The anomaly score is provided by the Discriminator

phi: Feature Extractor / Backbone

theta: Discriminator

D_ψ as

$$s^i_{h,w} = -D_\psi(q^i_{h,w}) \tag{9}$$

The anomaly map for anomaly localization during inference is defined as

$$S_{AL}(x_i) := \{s^i_{h,w} | (h,w) \in W_0 \times H_0\} \tag{10}$$

Upscaling nicht notwendig für img lvl classification

Then $S_{AL}(x_i)$ is interpolated to have the spatial resolution of the input sample and Gaussian filtered with $\sigma = 4$ for smooth boundaries. As the most responsive point exists for any size of the anomalous region, the maximum score of the anomaly map is taken as the anomaly detection score of each image

auch hier könnte eigenes adaptiertes Verfahren verwendet werden

$$S_{AD}(x_i) := \max_{(h,w)\in W_0 \times H_0} s^i_{h,w} \tag{11}$$

4. Experiments

4.1. Datasets.

We conduct most of the experiments on the MVTec Anomaly Detection benchmark [3], that is, a famous dataset in the anomaly detection and localization field. MVTec AD contains 5 texture and 10 object categories stemming from manufacturing with a total of 5354 images. The dataset is composed of normal images for training and both normal and anomaly images with various types of defect for test. It also provides pixel-level annotations for defective test images. Typical images are illustrated in Figure 1. As in [6, 22], images are resized and center cropped to 256×256 and 224×224, respectively. No data augmentation is applied. We follow the one-class classification protocol, also known as cold-start anomaly detection, where we train a one-class classifier for each category on its respective normal training samples.

We conduct one-class novelty detection on CIFAR10 [16], which contains 50K training images and 10K test images with scale of 32×32 in 10 categories. Under the setting of one-class novelty detection, one category is regarded as normal data and other categories are used as novelty.

CIFAR 10 nicht relevant für unsere Arbeit

4.2. Evaluation Metrics.

Image-level anomaly detection performance is measured via the standard Area Under the Receiver Operator Curve, which we denote as I-AUROC, using produced anomaly detection scores S_{AD} (Equation 11). For anomaly localization, the anomaly map S_{AL} (Equation 10) is used for an evaluation of pixel-wise AUROC (denoted as P-AUROC). In accordance with prior works [6, 22], we compute on MVTec AD the class-average AUROC and mean AUROC overall categories for detection and localization. The comparison baselines includes AE-SSIM [3], RIAD [31], DRÆM [30], CutPaste [17], CS-Flow [24], PaDiM [6], RevDist [7] and PatchCore [22].

Algorithm 1 SimpleNet training pseudo-code, Pytorch-like

```
# F: Feature Extractor
# G: Feature Adaptor
# N: i.i.d Gaussian noise
# D: Discriminator
pretrain_init(F)
random_init(G, D)
for x in data_loader:
    o = F(x)  # normal features
    q = G(o)  # adapted features
    q_ = q + random(N) # anomalous features

    loss = loss_func(D(q), D(q_)).mean()
    loss.backward() # back-propagate

    F = F.detach() # stop gradient
    update(G, D) # Adam

# loss function
def loss_func(s, s_):
    th_ = -th = 0.5
    return max(0, th-s) + max(0, th_+s_)
```

Feauture Extractor fixed

4.3. Implementation Details

This section describes the configuration implementation details of the experiments in this paper. All backbones used in the experiments were pre-trained with ImageNet [8]. The 2nd and 3rd intermediate layers of the backbone e.g. $l' \in [2, 3]$ in Equation 3 are used in the feature extractor as in [22] when the backbone is ResNet-like architecture. By default, our implementation uses WideResnet50 as backbone, and the feature dimension from the feature extractor is set to 1536. The later feature adaptor is essentially a fully connected layer without bias. The dimensions of the input and output features for the FC layer in the adaptor are the same. The anomaly feature generator adds i.i.d. Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each entry of normal features. σ is set to 0.015 by default. The subsequent discriminator composes of a linear layer, a batch normalization layer, a leaky relu(0.2 slope), and a linear layer. th^+ and th^- are both set to 0.5 in Equation 7. The Adam optimizer is used, setting the learning rate for the feature adaptor and discriminator to 0.0001 and 0.0002 respectively, and weight decay to 0.00001. Training epochs is set to 160 for each dataset and batchsize is 4.

1024 + 512

Müsste nicht ein Mapping in einen Niedrigdimensionaler en Raum finktioneren? Redundazen sollten so herausgenommen werden können

Optimierungspotenzial! mMn

4.4. Anomaly detection on MVTec AD

Anomaly detection results on MVTec AD are shown in Table 1. Image-level anomaly score is given by the maximum score of the anomaly map as in Equation 11. SimpleNet achieves the highest score for 9 out of 15 classes. For textures and objects, SimpleNet reaches new SOTA of 99.8% and 99.5% of I-AUROC, respectively. SimpleNet achieves significantly higher mean image anomaly detection performance i.e. I-AUROC score of 99.6%. Please note

Warum nur gaussian noise? Simplex Noise zB könnte bessere Ergebnisse liefern

Table 1. Comparison of SimpleNet with state-of-the-arts works on MVTec AD. Image-wise AUROC (I-AUROC) and pixel-wise AUROC (P-AUROC) are displayed in each entry as I-AUROC%/P-AUROC%. P-AUROC for CS-Flow is not recorded in [24]

Type	Reconstruction-based		Synthesizing-based		Embedding-based				Ours
Model	AE-SSIM	RIAD	DRÆM	CutPaste	CS-Flow	PaDiM	RevDist	PatchCore	SimpleNet
Carpet	87/64.7	84.2/96.3	97.0/95.5	93.9/98.3	**100**/-	99.8/99.1	98.9/98.9	98.7/**99.0**	99.7/98.2
Grid	94/84.9	99.6/98.8	99.9/99.7	**100**/97.5	99.0/-	96.7/97.3	**100**/**99.3**	98.2/98.7	99.7/98.8
Leather	78/56.1	**100**/99.4	**100**/98.6	**100**/**99.5**	**100**/-	**100**/99.2	**100**/99.4	**100**/99.3	**100**/99.2
Tile	59/17.5	98.7/89.1	99.6/**99.2**	94.6/90.5	**100**/-	98.1/94.1	99.3/95.6	98.7/95.6	99.8/97.0
Wood	73/60.3	93.0/85.8	99.1/**96.4**	99.1/95.5	**100**/-	99.2/94.9	99.2/95.3	99.2/95.0	**100**/94.5
Avg. Text.	78/56.7	95.1/93.9	99.1/**97.9**	97.5/96.3	**99.8**/-	95.5/96.9	99.5/97.7	99.0/97.5	**99.8**/97.5
Bottle	93/83.4	99.9/98.4	99.2/**99.1**	98.2/97.6	99.8/-	99.1/98.3	**100**/98.7	**100**/98.6	**100**/98.0
Cable	82/47.8	81.9/84.2	91.8/94.7	81.2/90.0	99.1/-	97.1/96.7	95.0/97.4	99.5/**98.4**	**99.9**/97.6
Capsule	94/86.0	88.4/92.8	**98.5**/94.3	98.2/97.4	97.1/-	87.5/98.5	96.3/98.7	98.1/98.8	97.7/**98.9**
Hazelhut	97/91.6	83.3/96.1	**100**/**99.7**	98.3/97.3	99.6/-	99.4/98.2	99.9/98.9	**100**/98.7	**100**/97.9
Metal Nut	89/60.3	88.5/92.5	98.7/**99.5**	99.9/93.1	99.1/-	96.2/97.2	**100**/97.3	**100**/98.4	**100**/98.8
Pill	91/83.0	83.8/95.7	98.9/97.6	94.9/95.7	98.6/-	90.1/95.7	96.6/98.2	96.6/97.4	**99.0**/**98.6**
Screw	96/88.7	84.5/98.8	93.9/97.6	88.7/96.7	97.6/-	97.5/98.5	97.0/**99.6**	98.1/99.4	**98.2**/99.3
Toothbrush	92/78.4	**100**/98.9	**100**/98.1	99.4/98.1	91.9/-	**100**/98.8	99.5/**99.1**	**100**/98.7	99.7/98.5
Transistor	90/72.5	90.9/87.7	93.1/90.9	96.1/93.0	99.3/-	94.4/97.5	96.7/92.5	**100**/96.3	**100**/**97.6**
Zipper	88/66.5	98.1/97.8	**100**/98.8	99.9/99.3	99.7/-	98.6/98.5	98.5/98.2	99.4/98.8	99.9/**98.9**
Avg. Obj.	91/75.8	89.9/94.3	97.4/97.0	95.5/95.8	98.2/-	96.0/97.8	98/97.9	99.2/**98.4**	**99.5**/**98.4**
Average	87/69.4	91.7/94.2	98.0/97.3	96.1/96.0	98.7/-	95.8/97.5	98.5/97.8	99.1/**98.1**	**99.6**/**98.1**

Table 2. Performance on MVTec AD under different combinations of hierarchy levels of WideResNet50 to use.

level1	level2	level3	I-AUROC%	P-AUROC%
✓			93.0	94.2
	✓		98.4	96.7
		✓	99.2	97.5
✓	✓		96.7	96.7
	✓	✓	**99.6**	**98.1**
✓	✓	✓	99.1	**98.1**

that, a reduction from an error of 0.9% for PatchCore [22] (next best competitor, under the same WideResnet50 backbone) to 0.4% for SimpleNet means a reduction of the error by 55.5%. In industrial inspection settings, this is a relevant and significant reduction.

4.5. Anomaly localization on MVTec AD

The anomaly localization performance is measured by pixel-wise AUROC, which we note as P-AUROC. Comparisons with the state-of-the-art methods are shown in Table 1. SimpleNet achieves the best anomaly detection performance of 98.1% P-AUROC on MVTec AD as well as the new SOTA of 98.4% P-AUROC for objects. SimpleNet achieves the highest score for 4 out of 15 classes. We visualize representative samples for anomaly localization in Figure 8.

4.6. Inference time

Alongside the detection and localization performance, inference time is the most important concern for industrial model deployment. The comparison with the state-of-the-art methods on inference time is shown in Figure 2. All the methods are measured on the same hardware containing a Nvidia GeForce GTX 3080ti GPU and an Intel(R) Xeon(R) CPU E5-2680 v3@2.5GHZ. It clearly shows that our method achieves the best performance as well as the fastest speed at the same time. SimpleNet is nearly $8\times$ faster than PatchCore [22].

Mit LayerCut oder ohne?

4.7. Ablation study

Neighborhood size and hierarchies. We investigate the influence of neighborhood size p in Equation 1. Results in Figure 6 show a clear optimum between locality and global context for anomaly predictions, thus motivating the neighborhood size $p = 3$. We design a group of experiments to test the influence of hierarchies subset L on model performance and the results are shown in Table 2. We index the first three WideResNet50 blocks with $1 - 3$. As can be seen, features from hierarchy level 3 can already achieve state-of-the-art performance but benefit from additional hierarchy level 2. We chose $2 + 3$ as the default setting.

Im Wesentlichen qualitativ die gleichen Ergebnisse, wie bei uns

Adaptor configuration. Adaptor provides a transformation (projection) on the pre-trained features. Our default feature adaptor is a single FC layer without bias, with equal input and output channels. A comparison of different feature adaptors is shown in Table 3, the first row "Ours" implementation follows the same configuration as in Table 1. "Ours-complex-FA" replaces the simple feature adaptor with a nonlinear one (i.e. 1 layer MLPs with nonlinearity). The row "Ours-w/o-FA" drops the feature adaptor. The results indicate that a single FC layer yields the best performance. Intuitively, the feature adaptor finds a projection such that the faked abnormal features and projected pre-trained features are easily severed, meaning a simple solution to the discriminator. This is also indicated by the phenomenon that using a feature adaptor helps the network con-

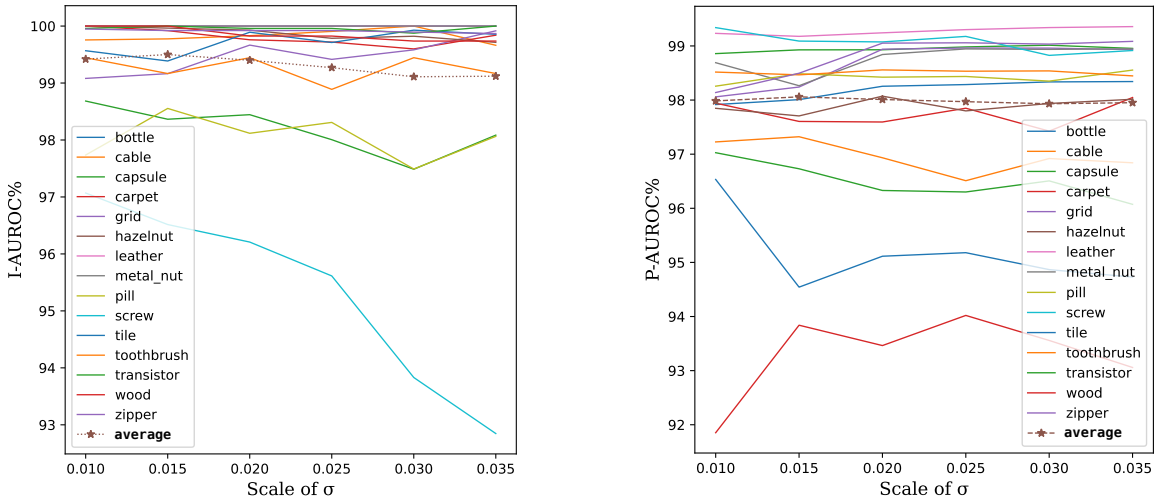
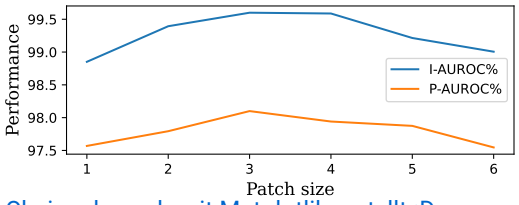


Figure 5. I-AUROC% and P-AUROC% for each class of MVTec AD dataset with varied σ. (Best viewed in color.)



Obviously auch mit Matplotlib erstellt :D

Figure 6. Performance with varied patch sizes on MVTec AD.

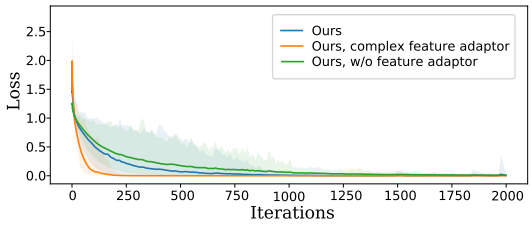


Figure 7. Visualization of loss during training. The plotted lines show the mean loss for all classes in the MVTec AD dataset. The transparent color shows the range of loss fluctuation.

verge fast (Figure 7). We observe a significant performance drop with a complex feature adaptor. One possible reason is that a complex adaptor may lead to overfitting, reducing the generalization ability for various defects in test. Figure 4 compares the histogram of standard deviation along each dimension of the features before and after the feature adaptor. We can see that, when training with anomalous features, adapted feature space becomes compact.

Scale of noise. The scale of noise in the anomaly feature generator controls how far away the synthesized abnormal features are from the normal ones. To be specific, high σ results in abnormal features keeping a high Euclidean distance towards normal features. Training on a large σ will result in a loose decision bound, leading to a high false negative. Conversely, the training procedure will become unstable if σ is tiny, and the discriminator cannot generalize to normal features well. Figure 5 details the effect of σ for each class in MVTec AD. As can be seen, $\sigma = 0.015$ reaches the balance and yield the best performance.

Loss function. We compared the proposed loss function

Table 3. Comparison of different feature adaptors. "Ours" implementation follows the same configuration as in Table 1. "Ours-complex-FA" replaces the simple feature adaptor with a nonlinear one. "Ours-w/o-FA" drops the feature adaptor, equivalent to using an identity fully-connected layer. "Ours-CE" uses cross-entropy loss. I-AUROC% and P-AUROC% of MVTec AD are shown.

Model	I-AUROC%	P-AUROC%
Ours	**99.6**	**98.1**
Ours-complex-FA	98.3	97.2
Ours-w/o-FA	99.2	97.9
Ours-CE	99.4	97.8

Table 4. Performance under different backbones on MVTec AD.

Model	I-AUROC%	P-AUROC%
ResNet18	98.3	95.7
ResNet50	99.6	98.0
ResNet101	99.2	97.6
WideResNet50	**99.6**	**98.1**

Table 5. One-Class Novelty Detection I-AUROC(%) results on CIFAR-10 dataset.

Method	LSA	DSVDD	OCGAN	HRN	DAAD
AUROC	64.1	64.8	65.6	71.3	75.3
Method	DisAug CLR	IGD	MKD	RevDist	**SimpleNet**
AUROC	80.0	83.68	84.5	86.5	**86.5**

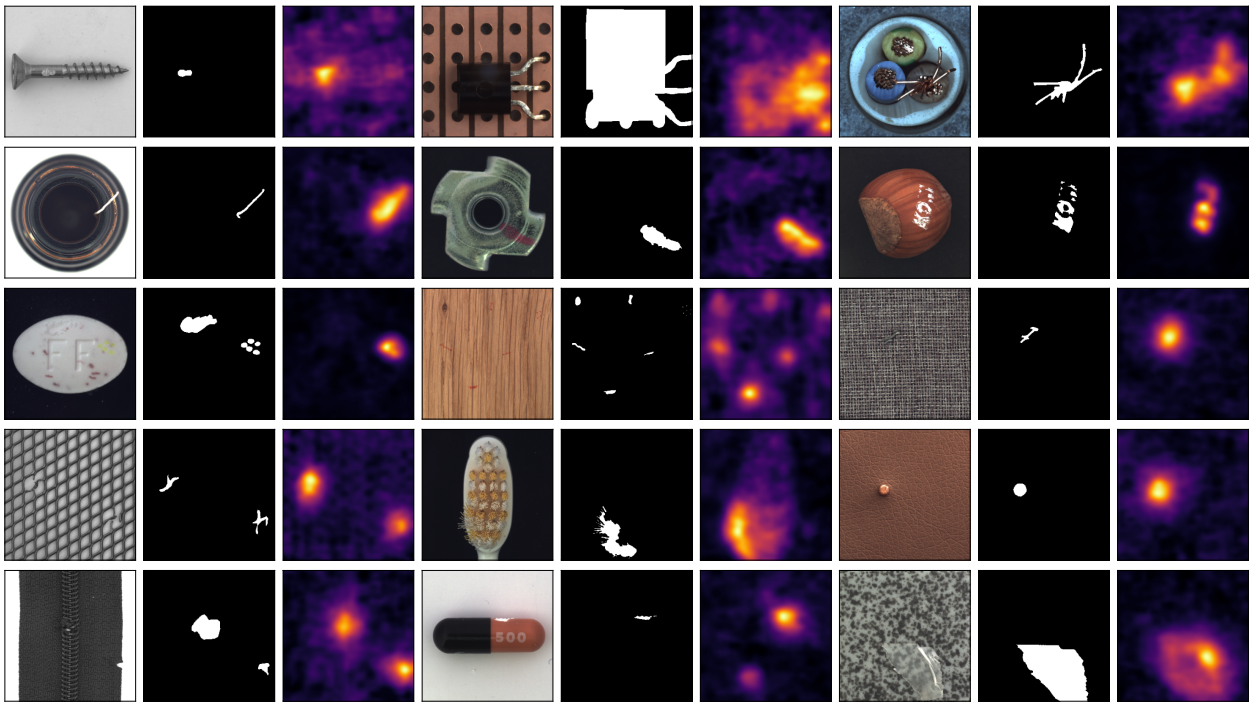


Figure 8. Qualitative results, where sampled image, ground truth, and anomaly map are shown for each class in MVTec AD.

in Section 3.5 with the widely used cross-entropy loss (as show in row "Ours-CE" in Table 3). We found the improvements, 0.2% I-AUROC and 0.3% P-AUROC, over cross-entropy loss.

Dependency on backbone. We test SimpleNet with different backbones, the results are shown in Table 4. We find that results are mostly stable over the choice of different backbones. The choice of WideResNet50 is made to be comparable with PaDiM [6] and PatchCore [22].

Qualitative Results Figure 8 shows results of anomaly localization that indicate the abnormal areas. The threshold for segmentation results is obtained by calculating the F1-score for all anomaly scores of each sub-class. Experimental results prove that the proposed method can localize abnormal areas well even in rather difficult cases. In addition, we can find that the proposed method has consistent performance in both object and texture classes.

4.8. One-Class Novelty Detection

To evaluate the generality of the proposed SimpleNet, we conduct a one-class novelty detection experiment on CIFAR-10 [16]. Following [19], we train the model with samples from a single class and detect novel samples from other categories. We train the corresponding model for each class respectively. Note that the novelty score is defined as the max score in the similarity map. Table 5 reports the I-AUROC scores of our method and other methods. For fair comparison, all the methods are pre-trained on ImageNet. The baselines include VAE [2], LSA [1], DSVDD [25], OCGAN [19], HRN [15], AnoGAN [27], DAAD [14], MKD [26], DisAug CLR [28], IGD [5] and RevDist [7]. Our method outperforms these comparison methods. Note that, IGD [5] and DisAug CLR [28] achieve 91.25% and 92.4% respectively when boosted by self-supervised learning.

5. Conclusion

In this paper, we propose a simple but efficient approach named SimpleNet for unsupervised anomaly detection and localization. SimpleNet consists of several simple neural network modules which are easy to train and apply in industrial scenarios. Though simple, SimpleNet achieves the highest performance as well as the fastest inference speed compared to the previous state-of-the-art methods on the MVtec AD benchmark. SimpleNet provides a new perspective to bridge the gap between academic research and industrial application in anomaly detection and localization.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 62176246 and Grant 61836008. This work is also supported by Anhui Provincial Natural Science Foundation 2208085UD17 and the Fundamental Research Funds for the Central Universities (WK3490000006).

References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 481–490, 2019. 8

[2] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015. 8

[3] 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*, pages 9592–9600, 2019. 1, 5

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4183–4192, 2020. 3

[5] Yuanhong Chen, Yu Tian, Guansong Pang, and Gustavo Carneiro. Deep one-class classification via interpolated gaussian descriptor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 383–392, 2022. 8

[6] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 1, 2, 3, 5, 8

[7] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 5, 8

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 3

[10] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[12] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022. 3

[13] Matthias Haselmann, Dieter P Gruber, and ul Tabatabai. Anomaly detection using deep learning based image completion. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1237–1242. IEEE, 2018. 2

[14] Jinlei Hou, Yingying Zhang, Qiaoyong Zhong, Di Xie, Shiliang Pu, and Hong Zhou. Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8791–8800, 2021. 8

[15] Wenpeng Hu, Mengyu Wang, Qi Qin, Jinwen Ma, and Bing Liu. Hrn: A holistic approach to one class learning. *Advances in Neural Information Processing Systems*, 33:19111–19124, 2020. 8

[16] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 5, 8

[17] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021. 2, 4, 5

[18] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus Robert Muller. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2020. 4

[19] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019. 2, 8

[20] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 3

[21] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13576–13586, 2022. 2

[22] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 2, 3, 5, 6, 8

[23] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021. 3

[24] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1088–1097, 2022. 2, 3, 5, 6

[25] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. 8

[26] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021. 8

[27] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017. 8

[28] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*, 2021. 8

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[30] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 2, 4, 5

[31] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021. 2, 5