PANDA - ADAPTING PRETRAINED FEATURES FOR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly detection methods require high-quality features. One way of obtaining strong features is to adapt pre-trained features to anomaly detection on the target distribution. Unfortunately, simple adaptation methods often result in catastrophic collapse (feature deterioration) and reduce performance. DeepSVDD combats collapse by removing biases from architectures, but this limits the adaptation performance gain. In this work, we propose two methods for combating collapse: i) a variant of early stopping that dynamically learns the stopping iteration ii) elastic regularization inspired by continual learning. In addition, we conduct a thorough investigation of Imagenet-pretrained features for one-class anomaly detection. Our method, PANDA, outperforms the state-of-the-art in the one-class and outlier exposure settings (CIFAR10: 96.2% vs. 90.1% and 98.9% vs. 95.6%)
.

1 INTRODUCTION

Detecting anomalous patterns in data is of key importance in science and industry. In the computational anomaly detection task, the learner observes a set of training examples. The learner is then tasked to classify novel test samples as normal or anomalous. There are multiple anomaly detection settings investigated in the literature, corresponding to different training conditions. In this work, we deal with two settings: i) when only normal images are used for training ii) Outlier Exposure (OE) where an external dataset simulating the anomalies is available.

In recent years, deep learning methods have been introduced for anomaly detection, typically extending classical methods with deep neural networks. Different auxiliary tasks (e.g. autoencoders or rotation classification) are used to learn representations of the data, while a great variety of anomaly criteria are then used to determine if a given sample is normal or anomalous. An important issue for current methods is the reliance on limited normal training data for representation learning, which limits the quality of learned representations. A solution, that we will investigate in this work, is to pre-train features on a large external dataset, and use the features for anomaly detection. As there is likely to be some mismatch between the external dataset and the task of anomaly detection on the target distribution, feature adaptation is an attractive option. Unfortunately, feature adaptation for anomaly detection often suffers from *catastrophic collapse* - a form of deterioration of the pre-trained features, where all the samples, including anomalous, are mapped to the same point. DeepSVDD (Ruff et al., 2018) proposed to overcome collapse by removing biases from the model architecture, but this restricts network expressively and limits the pre-trained models that can be borrowed off-the-shelf. Perera & Patel (2019) proposed to jointly train anomaly detection with the original task which has several limitations and achieves only limited adaptation success.

We propose two techniques to overcome catastrophic collapse: i) an adaptive early stopping method that selects the stopping iteration per-sample, using a novel generalization criterion ii) an elastic regularization, motivated by continual learning, that postpones the collapse. We also provide an extensive evaluation of Imagenet-pretrained features on one-class anomaly detection. Thorough experiments demonstrate that we outperform the state-of-the-art by a wide margin: e.g. CIFAR10 results: 96.2% vs. 90.1% without outlier exposure and 98.9% vs. 95.6% with outlier exposure.

We present several insightful critical analyses: i) We show that pre-trained features strictly dominate current self-supervised RotNet-based feature learning methods. We discuss the relative merits of each paradigm and conclude that for most practical purposes, using pre-trained features is preferable.

ii) We analyse the results of the popular method, DeepSVDD. We discover that the feature adaptation of its current architecture, which is designed to prevent collapse, does not improve over simple data whitening. iii) We show that collapse can be avoided using early stopping, and suggest an appropriate unsupervised criterion. We also show it can be mitigated using continual learning.

1.1 Related Work

Classical anomaly detection: The main categories of classical anomaly detection methods are: i) reconstruction-based: compress the training data using a bottleneck, and use a reconstruction loss as an anomaly criterion (e.g. (Candès et al., 2011; Jolliffe, 2011), K nearest neighbors (Eskin et al., 2002) and K-means (Hartigan & Wong, 1979)), ii) probabilistic: modeling the probability density function and labeling unlikely sampled as anomalous (e.g. Ensembles of Gaussian Mixture Models (Glodek et al., 2013), kernel density estimate (Latecki et al., 2007)) iii) one-class classification (OCC): finding a separating manifold between normal data and the rest of input space (e.g. One-class SVM (Scholkopf et al., 2000)).

Deep learning methods: The introduction of deep learning has affected image anomaly detection in two ways: extension of classical methods with deep representations and novel self-supervised deep methods. Reconstruction-based methods have been enhanced by learning deep autoencoder-based bottlenecks (D'Oro et al., 2019) which can provide better models of image data. Deep methods extended classical methods by creating a better representations of the data for parametric assumptions about probabilities, a combination of reconstruction and probabilistic methods (such as DAGMM (Zong et al., 2018)), or in a combination with an OCC (Ruff et al., 2018). Novel deep methods have also been proposed for anomaly detection including GAN-based methods (Zong et al., 2018). Another set of novel deep methods use auxiliary self-supervised learning for anomaly detection. The seminal work by Golan & El-Yaniv (2018) was later extended by Hendrycks et al. (2019b) and Bergman & Hoshen (2020).

Transferring pretrained representations: Learning deep features requires extensive datasets, preferably with labels. An attractive property of deep neural networks, is that representations learned on very extensive datasets, can be transferred to data-poor tasks. Specifically deep neural representations trained on the ImageNet dataset have been shown by Huh et al. (2016) to significantly boost performance on other datasets that are only vaguely related to some of the ImageNet classes. This can be performed with and without finetuning. Although much recent progress has been performed on self-supervised feature learning (Gidaris et al., 2018; Chen et al., 2020), such methods are typically outperformed by transferred pretrained features. Transferring ImageNet pre-trained features for out-of-distribution detection has been proposed by Hendrycks et al. (2019a). Similar pre-training has been proposed for one-class classification has been proposed by Perera & Patel (2019), however they require joint optimization with the original task.

2 Background: Feature Adaptation for Anomaly Detection

2.1 A Three-stage Framework

We present our general framework in which we examine several adaptation-based anomaly detection methods, including our method. Let us assume that we are given a set $\mathcal{D}_{train}$ of normal training samples: $x_1, x_2..x_N$. The framework consists of three steps:

Feature extractor pretraining: A pre-trained feature extractor ψ_0 is typically learned using self-supervised learning (auto-encoding, rotation or jigsaw prediction). We denote the loss function of the auxiliary task $L_{pretrain}$. The auxiliary task can be learned either on the training set $\mathcal{D}_{train}$ or on an external dataset $\mathcal{D}_{pretrain}$ (such as ImageNet). In the latter case, the pretrained extractor can be obtained off-the-shelf. We will investigate and analyse the merits of each choice in Sec. 4.2.

Feature adaptation: Features trained on auxiliary tasks or datasets may require adaptation before being used for anomaly scoring on the target data. This can be seen as a finetuning stage of the pre-trained features on the target training data. We denote the feature extractor after adaptation ψ.

Anomaly scoring: Having adapted the features for anomaly detection, we extract the features $\psi(x_1), \psi(x_2)..\psi(x_N)$ of the training set samples, we proceed to learn a scoring function, which

describes how anomalous a sample is. Typically, the scoring function seeks to measure the density of normal data around the test sample $\psi(x)$ (either by direct estimation or via some auxiliary task) and assign a high anomaly score to low density regions.

2.2 EXISTING FEATURE-ADAPTATION METHODS

In this section, we review two seminal methods that use feature adaptation for anomaly detection:

DeepSVDD: Ruff et al. (2018) suggest to first train an autoencoder E on the normal-only train images. The encoder is then used as the initial feature extractor $\psi_0(x) = E(x)$. As the features of the encoder are not specifically adapted to anomaly detection, DeepSVDD adapts ψ on the training data. The adaptation takes place by minimizing the compactness loss:

$$L_{compact} = \sum_{x \in \mathcal{D}_{train}} \|\psi(x) - c\|^2 \tag{1}$$

Where c is a constant vector, typically the average of $\psi_0(x)$ on the training set. However, the authors were concerned of the trivial solution $\psi = c$, and suggested architectural restrictions to mitigate it, most importantly removing the biases from all layers. We empirically show that the effect of adaptation of the features in DeepSVDD does not outperform simple feature whitening (see Sec. 4.2.2).

Joint optimization (JO): Perera & Patel (2019) proposed to use a deep feature extractor trained for object classification on the ImageNet dataset. Due to fear of "learning a trivial solution due to the absence of a penalty for miss-classification", the method do not adapt by finetuning on the compactness loss only. Instead, they relaxed the task setting, by assuming that a number ($\sim 50k$) of labelled original ImageNet images, $\mathcal{D}_{pretrain}$, are still available at adaptation time. They proposed to train the features ψ under the compactness loss jointly with the original ImageNet classification linear layer W and its classification loss, here the CE loss with the true label $\ell_{pretrain}(p, y) = -\log(p_y)$:

$$L_{Joint} = \sum_{(x,y) \in \mathcal{D}_{pretrain}} \ell_{pretrain}(\text{softmax}(W\psi(x)), y) + \alpha \sum_{x \in \mathcal{D}_{train}} \|\psi(x) - c\|^2 \tag{2}$$

Where W is the final linear classification layer and α is a hyper-parameter weighting the two losses. We note that the method has two main weaknesses: i) it requires retaining a significant number of the original training images which can be storage intensive ii) jointly training the two tasks may reduce the anomaly detection task accuracy, which is the only task of interest in this context. Our proposed method, PANDA, is able to sidestep these issues.

3 PANDA: FEATURE ADAPTATION FOR ANOMALY DETECTION

We present PANDA (Pre-trained Anomaly Detection Adaptation), a new method for anomaly detection in images. The core of our method lies in adapting general pre-trained features to anomaly detection on the target distribution.

Pre-trained feature extractor: Our method is agnostic to the specific pretrained feature extractor. We investigated different choices of the initial pre-trained feature extractor ψ_0 and found that ImageNet pretrained features achieve better results. The assumption of the availability of the ImageNet trained feature extractor and its merits will be discussed at length in Sec. 4.2.

Feature Adaptation: Similarly to SVDD and Joint Optimization, we also use the compactness loss (Eq. 1) to adapt the general pre-trained features to the task of anomaly detection on the target distribution. Instead of constraining the architecture or introducing external data into the adaptation procedure we tackle catastrophic collapse directly. The main issue is that the optimal solution of the compactness loss can result in "collapse", where all possible input values are mapped to the same point ($\psi(x) = c, \quad \forall x$). Learning such features will not be useful for anomaly detection, as both normal and anomalous images will be mapped to the same output, preventing separability. The issue is broader than the trivial "collapsed" solution after full convergence, but rather the more general issue of feature deterioration, where the original good properties of the pretrained features are lost. Even a non-trivial solution might not require the full discriminative ability of the original features which are none-the-less important for anomaly detection.

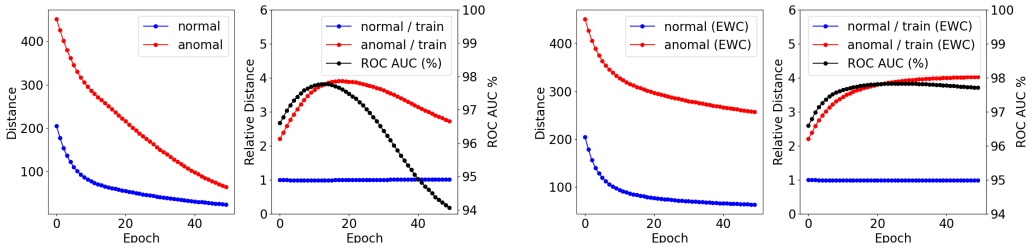

Figure 1: CIFAR100 Class 17 (right to left): (1) - During training all samples approach the center of train set features (2) - When normalized by the train average distance s_t, the normal samples stay dense, while the anomalous ones initially move further away and then "collapse". The ROC AUC performance behaves similarly to the anomalous samples' normalized distance. (3),(4) - when training with EWC the collapse is mitigated.

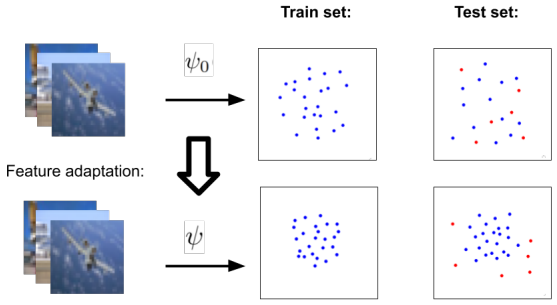

Figure 2: An illustration of our feature adaptation procedure, the pre-trained feature extractor ψ_0 is adapted to make the normal features more compact resulting in feature extractor ψ. After adaptation anomalous test images lie in a less dense region of the feature space.

To avoid this collapse, we suggest two options: (i) finetuning the pretrained extractor with compactness loss (Eq.1) and using sample-wise early stopping (ii) when collapse happens prematurely, before any significant adaptation happens, we suggest mitigating it using a Continual Learning-inspired adaptive regularization.

Sample-wise early stopping (PANDA-SES): Early stopping is one of the simplest methods used to regularize neural network. While stopping the training process after constant number of iterations (we use $2.3k$ minibatches) helps to control the collapse of the original features in most examined datasets (Sec. 4.2), in other cases, collapse occurs earlier in the training process - the best number of early stopping iterations may vary between datasets. We thus propose "samplewise early stopping" (SES). The intuition for the method can be obtained from Fig. 1. We can see that anomaly detection accuracy is correlated to the ratio between the average compactness loss of test set anomalies and the average compactness loss of training set normal images. We thus propose to save checkpoints of our network at fixed intervals during the training process - corresponding to different early stopping iterations ($\psi_1, \psi_2..\psi_T$), for each network ψ_t we compute the average loss on the training set images s_t. During inference, we score a target image x using each model $\psi_t(x) = f_t$, and normalize the score by the relevant average score s_t. We set the maximal normalized score, as the anomaly score of this sample, as this roughly estimates the model that achieves the best separation between normal and anomalous samples. Note that each sample is scored using only its features f_t, and the normal train set average score s_t, without seeing the labels of any other test set samples.

Continual Learning (PANDA-EWC): We propose a new solution for overcoming premature feature collapse that draws inspiration from the field of continual learning. The task of continual learning tackles learning new tasks without forgetting the previously learned ones. We note however that our task is not identical to standard continual learning as: i) we deal with the one-class classification setting whereas continual-learning typically deals with multi-class classification ii) we aim to avoid forgetting the expressivity of the features but do not particularly care if the actual classification

performance on the old task is degraded. A simple solution for preventing feature collapse is by regularization of the change in value of the weights of the feature extractor ψ from those of the pre-trained extractor ψ_0. However, this solution is lacking as the features are more sensitive to some weights than others and this can be "exploited" by the adaptation method.

Following ideas from continual learning, we use elastic weight consolidation (EWC) (Kirkpatrick et al., 2017). Using a number of mini-batches (we use 100) of pretraining on the auxiliary task, we compute the diagonal of the Fisher information matrix F for all weight parameters of the network. Note that this only needs to happen once at the end of the pretraining stage and does not need to be repeated. The value of the Fisher matrix for diagonal element θ' is given by:

$$F_{\theta'} = \mathbb{E}_{(x,y)\in\mathcal{D}_{pretrain}}\left[\left(\frac{\partial}{\partial\theta}L_{pretrain}(x,y);\theta'\right)^2|\theta\right] \tag{3}$$

We follow (Kirkpatrick et al., 2017) in using the diagonal of the Fisher information matrix F_{θ_i}, to weight the Euclidean distance of the change of each network parameter $\theta_i \in \psi_0$ and its corresponding parameter $\theta_i^* \in \psi$. This weighted distance can be interpreted as a measure of the curvature of the loss landscape as function of the parameters - larger values imply high curvature, inelastic weights. We use this regularization in combination with the compactness loss, the losses are weighted by the factor λ, which is a hyperparameter of the method (we always use $\lambda = 10^4$):

$$L_\theta = L_{compact}(\theta) + \frac{\lambda}{2}\cdot\sum_i F_{\theta_i}(\theta_i - \theta_i^*)^2 \tag{4}$$

Network ψ is initialized with the parameters of the pretrained extractor ψ_0 and trained with SGD.

Anomaly scoring: Given strong features and appropriate adaptation, our transformed data typically follows the standard anomaly detection assumption i.e. high-density in regions of normal data. As in classical anomaly detection, scoring can be done by density estimation. Our method performs better with strong non-parametric anomaly scoring methods. We evaluate several anomaly scoring methods: i) Euclidean Distance to the mean of the training features ii) the K nearest-neighbor distance between the target (test set) features and the features of the training set images iii) Computing the K-means of the training set features, and computing the distance between the target sample features to the nearest mean. See Sec. 4.2.3 for comparison results.

Outlier Exposure: An extension of the typical image anomaly detection task (Hendrycks et al., 2018), assumes the existence of an auxiliary dataset of images $\mathcal{D}_{OE}$, which are more similar to the anomalies than normal data. In case such information is available, we simply train a linear classification w layer together with the features ψ under a logistic regression loss (Eq. 5). As before, ψ is initialized with the weights from ψ_0. After training ψ and w, we use $w \cdot \psi(x)$ as the anomaly score. Results and critical analysis of this setting are presented in Sec. 4.2.

$$L_{OE} = \sum_{x\in\mathcal{D}_{train}} log(\sigma(1 - w\cdot\psi(x))) + \sum_{x\in\mathcal{D}_{OE}} log(\sigma(w\cdot\psi(x))) \tag{5}$$

4 Image Anomaly Detection

4.1 High-Level Results

In this section, we present high-level results of our method PANDA-EWC, (PANDA-SES can be found in Sec.4.2) compared to the state-of-the-art: One-class SVM (Scholkopf et al., 2000), DeepSVDD (Ruff et al., 2018), Multi-Head RotNet (Hendrycks et al., 2019b). We also compare our method to raw (unadapted) pretrained features. As Joint Optimization requires extra data, we did not add it to this table, but compare and outperform it in Tab. 4. We compare our PANDA-OE to the OE baseline in Hendrycks et al. (2019b) on CIFAR10, as the code or results for other classes were unavailable. To investigate performance in domains significantly different from the dataset used to pretrain the features, we evaluated our method across a large range of datasets: standard datasets (CIFAR10/100, CatsVsDogs), Black-and-white dataset (Fashion MNIST), Small fine-grained datasets (Birds200/Oxford Flowers), Medical dataset (WBC), Very finegrained anomalies (MVTec), and aerieal images (DIOR). A detailed description of the datasets is found in the

Table 1: Anomaly detection performance (Average ROC AUC %)

Dataset	Self-Supervised			Pretrained		OE	
	OC-SVM	DeepSVDD	MHRot	Unadapted	PANDA	MHRot	PANDA-OE
CIFAR10	64.7	64.8	90.1	92.5	**96.2**	95.6	**98.9**
CIFAR100	62.6	67.0	80.1	**94.1**	**94.1**	-	**97.3**
FMNIST	92.8	84.8	93.2	94.5	**95.6**	-	91.8
CatsVsDogs	51.7	50.5	86.0	96.0	**97.3**	-	94.5
DIOR	70.7	70.0	73.3	93.0	**94.3**	-	**95.9**

Table 2: Pretrained feature performance on various small datasets (Average ROC AUC %)

Dataset	Self-Supervised			Pretrained
	OC-SVM	DeepSVDD	MHRot	Unadapted
Birds 200	62.0	60.8	64.4	**95.3**
Flowers	74.5	78.1	65.9	**94.1**
MvTec	70.8	77.9	65.5	**86.5**
WBC	75.4	71.2	57.7	**87.4**

appendix Sec. C, and represenative frames are shown in Fig. 3. For outlier exposure (OE), we followed Hendrycks et al. (2018) and used 50k randomly sampled images from 80M Tiny Images. Implementation details are reported in Appendix D.

The main results are i) pre-trained features achieve significantly better results than self-supervised features on all datasets. ii) Feature adaptation significantly improves the performance on larger datasets iii) Outlier exposure can further improve performance in the case where the given outliers are more similar to the anomalies than the normal data. OE achieves near perfect performance on CIFAR10/100 but hurts performance for Fashion MNIST/CatsVsDogs which are less similar to the 80M Tiny images dataset. A detailed analysis of the reason for better performance for each of these methods and an examination of its appropriateness will be presented in Sec. 4.2.

4.2 Analysis and further Evaluation

In this section we analyze the factors of variation in performance between different methods:

4.2.1 An analysis of the choice of feature representation

A comparison of self-supervised and pre-trained features: In Tab. 1 and Tab. 2, we present a comparison between methods that use self-supervised and pre-trained feature representations. We see that the autoencoder used by DeepSVDD is particularly poor. The results of the MHRotNet as a feature extractor are better, but still underperform PANDA methods (see App. A for more details). The performance of the raw deep ResNet features without adaptation significantly outperforms all methods, including Fashion MNIST and DIOR which have significant differences from the ImageNet dataset. We can therefore conclude that ImageNet-pretrained features typically have significant advantages over self-supervised features. Tab. 2 shows that self-supervised methods do not perform well on small datasets as such methods require large numbers of normal samples in order to learn strong features. On the other hand ImageNet-pretrained features obtain very strong results.

Do pretrained features generalize to anomaly detection on domains far from the pretraining dataset? The results in Tab. 2 on FMNIST, DIOR, WBC, MVTec suggest that it does. We evaluated the ImageNet-pretrained features on datasets of various sizes, domains, resolutions and symmetries. On all those datasets pretrained features outperformed the SOTA. These datasets include significantly different objects from those of ImageNet, but also fine-grained intra-object anomalies, and represent a spectrum of data types: aerial images, microscopy, industrial images. This shows that one of the main concerns of using pre-trained features, namely, generalizing to distant domains is not an issue in practice.

Table 3: Comparison of average transformation prediction accuracy (%)

Method	Normal			Anomalous		
	Horizontal	Vertical	Rotation	Horizontal	Vertical	Rotation
Self-supervised	94.0	91.4	94.0	67.9	67.5	51.6
Pretrained	94.4	94.4	92.3	71.4	69.9	61.3

On the different supervision settings for one-class anomaly detection: Anomaly detection methods employ different levels of supervision. Within the one-class classification task, one may use outlier exposure (OE) - an external dataset (e.g. ImageNet), pretrained features, or no external supervision at all. The most extensive supervision is used by OE, which requires a large external dataset at training time, and performs well only when such a dataset is from a similar domain to the anomalies (see Tab. 1). In cases where the dataset used for OE has significantly different properties, the network may not learn to distinguish between normal and anomalous data, as the normal and anomalous data may have more in common than the OE dataset. E.g. both normal and anomalous classes of Fashion MNIST are greyscale, OE using 80M Tiny Images will not be helpful. Pretrained features further improve OE, in cases where is suitable e.g. CIFAR10.

Pretraining, like Outlier Exposure, is also achieved through an external labelled dataset, but differently from OE, the external dataset is only required once - at the pretraining stage and is not used again. Additionally, the same features are applicable for very different image domains from that of the pretraining dataset (e.g. Fashion MNIST - greyscale images, DIOR - aerial images, WBC-medical images, MVTec - industrial images). Self supervised feature learning requires no external dataset at all, which can potentially be an advantage. While there might be image anomaly detection tasks where ImageNet-pretrained weights are not applicable, we saw no evidence for such cases after examining a broad spectrum of domains and datasets (Tab. 8). This indicates that the extra supervision of the ImageNet-pretrained weights comes at virtually no cost.

Can pretrained features boost the performance of RotNet-based methods? We did not find evidence that pretrained features improve the performance of RotNet-based AD methods such as Hendrycks et al. (2019b) (CIFAR10: 90.1% vs. 86.6% without and with pretraining). As can be seen in Tab. 3, pretrained features improve the auxiliary task performance on the normal data, but also on the anomalous samples. As such methods rely on a generalization gap between normal and anomalous samples, deep features actually reduce this gap, as a solution to the auxiliary task becomes feasible for both types of images. For a more detailed analysis see Appendix A.

4.2.2 Feature adaptation methods

Benefits of feature adaptation: Feature adaptation aims to make the distribution of the normal samples more compact, w.r.t. the anomalous samples. Our approach of finetuning pretrained features for compactness under EWC regularization, significantly improves the performance over "raw" pretrained features (see Tab.1). While the distance from the normal train samples center, of both normal and anomalous test samples is reduced (see Fig.1), the average distance from the center of anomalous test samples is typically further than that of normal samples, in relative terms. This makes anomalies easier to detect by standard classifiers such as kNN.

While PANDA-EWC may train more than $7.8k$ minibatches without catastrophic collapse on CIFAR10, performance of training without regularization usually peaks higher but collapse earlier. We therefore set our constant early stopping epoch such that the net trains with to $2.3k$ minibatches on all datasets for comparison. Our PANDA-SES method usually achieves an anomaly score not far from the unregularized early stopping peak performance, but is most important in cases where unregularized training fails completely.

A comparison of PANDA against other adaptation methods: In Tab. 4 we compare PANDA against (i) JO (Perera & Patel, 2019) - co-training compactness with ImageNet classification which requires ImageNet data at training time. We can see that PANDA - EWC always outperforms JO feature adaptation. (ii) PANDA early stopping (ImageNet pretraining + adaptation, with early stopping after constant iterations number), generally has higher performance than PANDA-EWC, but has severe collapse issues on some classes. (iii) PANDA-SES is similar to early stopping, but PANDA-

Table 4: A comparison of different feature adaptation methods (Avg. ROC AUC %)

Dataset	Baseline	PANDA		
	JO	Early stopping	SES	EWC
CIFAR10	93.2	**96.2**	95.9	**96.2**
CIFAR100	91.1	**94.8**	94.6	94.2
FMNIST	94.9	95.4	95.5	**95.6**
CatsVsDogs	96.1	91.9	95.7	**96.4**
DIOR	93.1	95.4	**95.6**	95.5

Table 5: Performance of finetuning different ResNet blocks (CIFAR10 w. EWC, ROC AUC %)

Trained Blocks	1,2,3,4	2,3,4	3,4	4
Avg	94.9	95.9	**96.2**	94.8

SES does not collapse as badly on CatsVsDogs dataset. We note that weighting equally the changes in all parameters ($\sum_i (\theta_i - \theta_i^*)^2$) achieves similar results to early stopping.

Which are the best layers to finetune? Fine-tuning all the layers, is prone to feature collapse, even with continual learning (see Tab.5). Finetuning Blocks 3 & 4, or 2, 3 & 4, results in similar performance. Finetuning only block 4 results in a very similar performance to linear whitening of the features according to the train samples (94.6 with whitening vs. 94.8 with finetuning only the last block). Similar effect as can be seen in the original DeepSVDD architecture (see also Tab.7, Appendix B). We therefore recommend finetuning Blocks 3 & 4.

DeepSVDD architectural changes: DeepSVDD (Ruff et al., 2018) proposes various architectural changes, such as removing the bias parameters from the network, to prevent collapse to trivial features. We found empirically that the results obtained by the constrained architecture were about the same as those achieved with simple whitening of the data (64.8% vs. 64.6%, see Tab.7). We also ablated DeepSVDD by (re-)adding the biases into its LeNet architecture did not deteriorate its anomaly detection performance. Architectural modifications are not the focus of this work, further investigation into architectures less prone to feature collapse is left for future work.

4.2.3 Anomaly scoring functions

Does kNN improve over distance to the center? kNN achieves an improvement of around 2% on average w.r.t. to distance to the center (CIFAR10: 94.2% vs 96.2%).

Can we improve over the linear complexity of kNN? A naive implementation of kNN has linear runtime complexity in the number of training samples. K-means with a small number of clusters gives $\sim$1% decrease (CIFAR10: 94.9% vs 96.2%, with 10 means). We note that even for very large datasets, or many thousands of means, both kNN and K-means can run faster than real-time.

5 Conclusion and Outlook

We proposed an anomaly detection method that adapts pretrained features and mitigates or avoids catastrophic collapse. We showed that our results significantly outperform current methods while addressing their limitations. We analysed the reasons for the strong performance of our method and related popular methods to the different stages of our framework.

The main limitation of this work is the requirement for strong pretrained feature extractors. Much work was done on transferable image and text features and it is likely that current extractors can be effective to obtain features for time series and audio as well. Generic feature extractors are not currently available for tabular data, their development is an exciting direction for future work.

REFERENCES

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *ICLR*, 2020.

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 Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *JACM*, 2011.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Pierluca D'Oro, Ennio Nasca, Jonathan Masci, and Matteo Matteucci. Group anomaly detection via graph autoencoders. 2019.

Jeremy Elson, John R Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *ACM Conference on Computer and Communications Security*, volume 7, pp. 366–374, 2007.

Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pp. 77–101. Springer, 2002.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Michael Glodek, Martin Schels, and Friedhelm Schwenker. Ensemble gaussian mixture models for probability density estimation. *Computational Statistics*, 28(1):127–138, 2013.

Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *NeurIPS*, 2018.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1979.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *arXiv preprint arXiv:1901.09960*, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019b.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Ian Jolliffe. *Principal component analysis*. Springer, 2011.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 61–75. Springer, 2007.

Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:296–307, 2020.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

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

Lukas Ruff, Nico Gornitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, 2018.

Bernhard Scholkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *NIPS*, 2000.

Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xin Zheng, Yong Wang, Guoyou Wang, and Jianguo Liu. Fast and robust segmentation of white blood cell images by self-supervised learning. *Micron*, 107:55–71, 2018.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018.

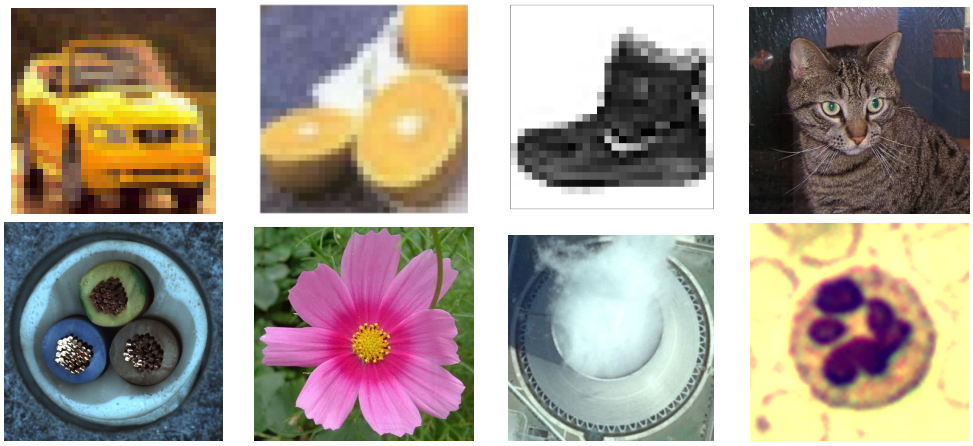

Figure 3: Representative images of the different datasets, from the left clockwise: CIFAR10, CIFAR100, Fashion MNIST, DogsVsCats, MVTec, Oxford Flowers, DIOR and WBC.

A Pretrained features, RotNet auxiliary tasks and generalization

Let us take a closer look at the application of RotNet-based methods for image anomaly detection. We will venture to understand why initializing RotNets with pretrained features may actually impair their anomaly detection performance. In such cases, a network for rotation classification is trained on normal samples, and used to classify the rotation (and translations) applied to a test rotated image.

Table 6: Pretrained vs. Raw Initialization Anomaly Detection Performance (ROC AUC %)

CIFAR10 class	0	1	2	3	4	5	6	7	8	9	Avg
Pretrained MHRot	70.1	93.7	84.4	76.1	89.7	87.3	91.1	94.4	86.8	90.8	86.4
MHRot	77.5	96.9	87.3	80.9	92.7	90.2	90.9	96.5	95.2	93.3	90.1

To score an anomaly, the image is deemed anomalous if its rotation prediction accuracy is worse than that of a typical normal image.

To correctly classify a rotation of a new image, the network may use traits within the image that are associated with its correct alignment. Such features may be associated with the normal class, or with the entire dataset (common to both the anomalous classes together). For illustrative purposes, let us consider a normal class with images containing a deer, and the anomalous class with images containing a horse. The horns of the deer may indicate the "upward" direction, but so does the position of the sky in the image, which is often sufficient to classify the rotation correctly. As shown in Tab.3, when initialized with pretrained features, the RotNet achieves very good performance on the auxiliary tasks, both within and outside the normal class, indicating the use the more general traits that are common to more classes.

Although at first sight it may appear that the improved auxiliary task performance should improve the performance on anomaly detection, this is in fact not the case! The reason is that features that generalize better, achieve better performance on the auxiliary task for anomalous data. The gap between the performance on the auxiliary tasks will therefore be smaller than with randomly-initialized networks - leading to degraded anomaly detection performance. For example, consider the illustrative example described above. A RotNet that "overfits" to work only on the normal class deer, relying on the horns of the deer would classify rotations more accurately on deer images than horse images (as its main feature is horns). On the other hand, a RotNet that also uses more general traits can use the sky position for rotation angle prediction. In this case, it will achieve higher accuracy for both deer and horse images. The gap in performance is likely to be reduced, leading to lower anomaly detection success.

The above argument can be formulated using mutual information: In cases where the additional traits unique to the class do not add much information regarding the correct rotation over the general features common to many classes, the class will have limited mutual information with the predicted rotation as well (conditional on the information already given traits common to the entire datasets). When the conditional mutual information between the predicted rotation and the class traits is decreases, we expect the predicted rotation to be less discriminative for anomaly detection, as we indeed see in Tab.6.

It is interesting to note that using RotNet features for our transfer learning approach achieves inferior results to both MHRot and our method. Only through an ensemble of all rotations, as MHRot does, it achieves strong performance comparable to the MHRot performance. MHRot achieved 89.7% in our re-implementation. Using the MHRot features as ψ_0, we compute the kNN distance of the unadapted features between test set images and train set image transformed by the same transformation. By ensembling the 36 transformations - using the average kNN distance, yields 88.7%. Another metric is computing the average kNN distance between test data transformed under a specfic transformation and the training set transformed by another transformation. By using the average same-transformation kNN distance minus the average different transformation kNN distance, achieves 89.8% - a little better than the RotNet performance.

B Feature adaptation, DeepSVDD and feature collapse

To understand whether DeepSVDD gains its significant performance from its pretrained features or from its feature adaptation, we tried to replace its feature adaptation by closed-form linear data whitening. For both pretrained features and anomaly scoring, we used the DeepSVDD original code (Ruff et al., 2018). We can see that a linear method such as data whitening achieves comparable results (Tab.7). We believe that large architectures are required for meaningful feature adaptation.

Table 7: Deep SVDD vs. PCA Whitening Anomaly Detection Performance (ROC AUC %)

CIFAR10 class	0	1	2	3	4	5	6	7	8	9	Avg
PCA whitening	62.0	63.6	49.7	59.9	59.8	65.8	68.3	68.0	75.5	71.2	64.8
Deep SVDD	59.7	64.3	48.4	61.5	61.3	65.5	70.1	68.9	75.3	72.5	64.6

Table 8: Details of Datasets Used for Evaluation

Dataset	No. of classes	No. of train images (Avg.)	No. of test images
CIFAR10	10	5,000	10,000
Fashion MNIST	10	6,000	10,000
CIFAR100	20	2,500	10,000
102 Category Flowers	102	10	7,169
Caltech-UCSD Birds 200	200	30	5,794
CatsVsDogs	2	10,000	5,000
MVTec	15	242	1,725
WBC	4	59	62
DIOR	19	649	9,243

C Detailed Description of Datasets

Standard datasets: We evaluate our method on a set of commonly used datasets: *CIFAR10* (Krizhevsky et al., 2009): Consists of RGB images of 10 object classes. *Fashion MNIST* (Xiao et al., 2017): Consists of grayscale images of 10 fashion item classes. *CIFAR100* (Krizhevsky et al., 2009): We use the coarse-grained version that consists of 20 classes. *DogsVsCats*: High resolution color images of two classes: cats and dogs. The data were extracted from the ASIRRA datasetElson et al. (2007), we split each class to the first 10,000 images as train and the last 2,500 as test.

Small datasets: To further extend our results, we compared the methods on a number of small datasets from different domains: 102 *Category Flowers & Caltech-UCSD Birds* 200 (Nilsback & Zisserman, 2008) Wah et al. (2011): For each of those datasets we evaluated the methods using only each of the first 20 classes as normal, and using the entire test set for evaluation. *MVTec* (Bergmann et al., 2019): This datasets contain 15 different industrial products, with normal images of proper products for train and $2-9$ types of manufacturing errors as anomalies. The anomalies in MVTec are in-class i.e. the anomalous images come from the same class of normal images with subtle variations. As can be seen in the results in Tab.2, self-supervised methods performed quite poorly on these datasets as they require many images to learn strong features. Simply using pretrained features was sufficient to obtain high accuracy (Tab.2).

Symmetric datasets: We evaluated our method on datasets that contain symmetries, such as images that have no preferred angle (microscopy, aerial images. See Fig.3): *WBC* (Zheng et al., 2018): We used the 4 big classes in *"Dataset 1"* of microscopy images of white blood cells, and a $80\%/20\%$ train-test split. *DIOR* (Li et al., 2020): We preprocessed the DIOR aerial image dataset by taking the segmented object in classes that have more than 50 images with size larger than 120×120 pixels. We see in Tab. 12 that for both symmetric datasets our method outperformed MHRot even more significantly. This experiment illustrates a weakness in self-supervised methods that need to exploit specific properties of the data e.g. rotational symmetry. When such properties do not exist in the data, the performance of self-supervised methods is reduced. In this case, rotation prediction conveys no information on rotationally invariant images, and presumably all the prediction performance of MHRot comes from the translation prediction task, which can be less accurate.

D Implementation details

PANDA

Optimization: We finetune the two last blocks of an ImageNet pretrained ResNet152 using SGD optimizer with weight decay of $w = 5 \cdot 10^{-5}$, and momentum of $m = 0.9$. We use $G = 10^{-3}$

gradient clipping. To have a comparable amount of training in the different dataset, we use define the duration of each of our train using a constant number of minibatches, 32 samples each.

EWC: We use the fisher information matrix as obtained by (Kirkpatrick et al., 2017), as explained in Sec.3. We weight the EWC loss with $\lambda = 10^4$. After obtaining EWC regularization, we train our net training on $7.8k$ minibatches.

Early stopping/Sample-wise early stopping: We save a copy of the net every 5 epochs. For early stopping we used the copy trained on $2.3k$ minibatches. For sample-wise early stopping we try all copies trained on up to $150k$ image samples.

Anomaly scoring: Unless specified otherwise, we score the anomalies according to the kNN method with $k = 2$ nearest neighbours. When comparing different networks as in PANDA-SES method, we normalize each set of features by the typical kNN distance of its normal train features. To obtain the typical normal distance we would like to compute the average on the normal samples. However, computing the distance between normal training data has that issue that each point is its own nearest neighbour. Instead, we split the train set features (90% vs. 10%), and compute the kNN between the 10% validation images and the gallery 90% images.

PANDA Outlier Exposure: The method was described in Sec.3. For synthetic outlier images, we used the first 48k images of 80 Million Tiny Images (Torralba et al., 2008) with CIFAR10 & CIFAR100 images removed. We finetune the last block of an ImageNet pretrained ResNet152 with SGD optimizer using 75 epochs and the following parameters: learning rate is 0.1 with gradient clipping, momentum is 0.9, and no weight decay.

Baselines We compare to the following methods:

OC-SVM: One-class SVM with the RBF kernel. The hyper-parameters ($\nu \in \{0.1, ..., 0.9\}, \gamma \in \{2^{-7}, ..., 2^2\}$) were optimized to maximize AUROC.

DeepSVDD: We resize all the images to 32×32 pixels and use the official pyTorch implementation with the CIFAR10 configuration.

MHRot (Hendrycks et al., 2019b): An improved version of the original RotNet approach. For high-resolution images we used the current GitHub implementation. For low resolution images, we modified the code to the architecture described in the paper, replicating the numbers in the paper on CIFAR10.

Outlier Exposure (MHRot): We use the outlier exposure performance as reported in Hendrycks et al. (2019b).