PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation Supplementary Material

Tal Reiss *, Niv Cohen *, Liron Bergman & Yedid Hoshen
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel

Contents

1. 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 image. Each test image is checked for its rotation prediction accuracy, which is assumed to be worse for an anomalous images than for a 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.5 (in the main text), when initialized with pretrained features, the RotNet network 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 of normal and anomalous images of the auxiliary tasks, will therefore be smaller than with randomly-initialized networks - leading to degraded anomaly detection performance. For example, consider the illustrative case described above. A RotNet netwrok 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 on 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 capabilities.

The above argument can be formulated using mutual information: In cases where the additional traits whice are 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 (conditional on the information already given by traits common to the entire datasets). When the conditional mutual information between the predicted rotation and the class traits decreases, we expect the predicted rotation to be less discriminative for anomaly detection, as we indeed see in Tab.1.

It is interesting to note that using features learned with RotNet 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 perfor-

*Equal contribution

mance. 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 the test set images and the train set image transformed by the same transformation. When ensembling the 36 transformations - and using the average kNN distance, yields 88.7%. Another metric we examined is computing the average kNN distance between test data transformed under a specfic transformation and the training set transformed by another transformation. Using the average same-transformation kNN distance minus the average different transformation kNN distance, achieves 89.8% - a little better than the RotNet performance.

2. Implementation Details

2.1. 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 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 [1], as explained in Sec.3 in the main text. 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 (including repetitions).

Anomaly scoring: Unless specified otherwise, we score the anomalies according to the kNN method with $k = 2$ nearest neighbours.

SES distance normalization: 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 an issue: 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 of the main text. For synthetic outlier images, we used the first 48k images of 80 Million Tiny Images [2] with CIFAR10 and 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*: 0.1 with *gradient clipping*: 1e-3, *momentum*: 0.9, and no weight decay.

2.2. Anomaly Detection 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 ROCAUC.

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

MHRot [3]: 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 [3].

2.3. SPADE

Architecture: In all experiments, we use a Wide-ResNet50×2 feature extractor, which was pre-trained on ImageNet.

Resolution: MVTec images were resized to 256×256 and cropped to 224×224. All metrics were calculated at 256×256 image resolution, and we used cv2.INTERAREA for resizing when needed.

Layers: Unless otherwise specified, we used features from the ResNet at the end of the first block (56×56), second block (28×28) and third block (14×14), all with equal weights. In Tab. 2 we compare different level of the feature pyramid as feature descriptor. We experienced that using activations of too high resolution (56×56) significantly hurts performance due to limited context, while using the higher levels on their own, results in diminished performance (due to lower resolution). Using a combination of all three upstream layers in the pyramid results in the best performance.

Combining features from different layers: We evaluated two ways of combining per-pixel features extracted from different layers. Concatenation - resampling the activation to the same resolutions and concatenating all per-pixel features to form a combined feature. Ensembling - computing the per-pixel anomaly score using the per-pixel feature of each layer, and adding the per-pixel per-layer scores of all layers to form a combined score. We found the ensemble approach was more robust and yielded a bit better results. Therefore, we report it.

Postprocessing: After computing the pixel-wise anomaly score for each image, we smoothed the results with a Gaussian filter ($\sigma = 5$).

For fast nearest neighbour implementation, we used the *FAISS* library [4].

Table 1: 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

Table 2: Anomaly segmentation accuracy on MVTec with different ResNet layers (PRO %)

Summed layers	Layers 0,1,2	Layer 0	Layer 1	Layer 2
Carpet	98.9	86.7	97.9	98.8
Grid	97.6	98.9	98.9	96.3
Leather	99.1	98.3	99.3	99.0
Tile	94.4	80.5	91.1	94.2
Wood	93.7	92.3	94.5	92.5
Bottle	98.1	89.6	98.0	97.9
Cable	96.4	70.8	93.5	96.9
Capsule	99.0	97.3	98.8	98.7
Hazelnut	98.6	96.6	97.6	98.6
Metal nut	97.4	88.4	97.0	96.7
Pill	96.4	95.9	95.8	95.9
Screw	99.2	98.8	99.4	98.6
Toothbrush	98.8	96.9	98.8	98.6
Transistor	94.2	71.0	84.0	95.9
Zipper	98.1	96.4	97.9	97.7
Average	**97.3**	90.6	96.2	97.1

3. Datasets

Standard datasets: We evaluate our method on a set of commonly used datasets: *CIFAR10* [5]: Consists of RGB images of 10 object classes. *Fashion MNIST* [6]: Consists of grayscale images of 10 fashion item classes. *CIFAR100* [5]: 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 dataset[7], we split each class to the first 10,000 images as train and the last 2,500 as test.

Small datasets: We report results on several small datasets from different domains: 102 *Category Flowers & Caltech-UCSD Birds* 200 [8] [9]: For each of those datasets we evaluated the methods using only the first 20 classes as normal train set, and using the entire test set for evaluation. *MVTec* [10]: This datasets contain 15 different industrial products, with normal images of proper products for train and $1-9$ 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. We also use the MVTec dataset for the anomaly segmentation results.

Symmetric datasets: We evaluated our method on datasets that contain symmetries, such as images that have no preferred angle (microscopy, aerial images.): *WBC* [11]: We used the 4 big classes in *"Dataset 1"* of microscopy images of white blood cells, and a $80\%/20\%$ train-test split. *DIOR* [12]: 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 can see that RotNet-type methods perform particularly poorly on such datasets.

4. Choosing the Layers to Finetune

Fine-tuning all layers is prone to feature collapse, even with continual learning (see Tab.3). Finetuning Blocks 3 & 4, or 2, 3 & 4, results in similar performance. Finetuning only block 4 results in 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 Tab.4). We therefore recommend finetuning Blocks 3&4.

5. SPADE: Detailed Results

In this section, we report the full results for SPADE and its relevant baselines. We evaluate our method using two established metrics. The first is per-pixel ROCAUC. The ROC

Table 3: Performance of finetuning different ResNet blocks (CIFAR10 w. EWC, ROC AUC %)

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

curve is calculated by first computing the anomaly score of each pixel and then scanning over the range of thresholds, on pixels from all test images together. The anomalous category is designated as positive. It was noted by several previous works that ROCAUC is biased in favor of large anomalies. In order to reduce this bias, Bergmann et al [13] propose the PRO (per-region overlap) curve metric. They first separate anomaly masks into their connected components, therefore dividing them into individual anomaly regions. By changing the detection threshold, they scan over false positive rates (FPR), for each FPR they compute PRO i.e. the proportion of the pixels of each region that are detected as anomalous. The PRO score at this FPR is the average coverage across all anomalous regions. The PRO curve metric computes the integral across FPR rates from 0 to 0.3. The PRO score is the normalized value of this integral. We can see from Tab. 6 and Tab. 5 that our method significantly outperforms the baselines in terms of both metrics. Qualitative results of our method are presented in Fig. 1.

Table 4: Deep SVDD vs. PCA Whitening Anomaly Detection Performance (ROC AUC %)

	with std										
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 5: Sub-Image anomaly detection accuracy on MVTec (ROCAUC %)

	AE_{SSIM}	AE_{L2}	AnoGAN	CNN Dict	TI	VM	CAVGA-R_u	SPADE
Carpet	87	59	54	72	88	-	-	98.6
Grid	94	90	58	59	72	-	-	99.0
Leather	78	75	64	87	97	-	-	99.5
Tile	59	51	50	93	41	-	-	89.8
Wood	73	73	62	91	78	-	-	95.8
Bottle	93	86	86	78	-	82	-	98.1
Cable	82	86	78	79	-	-	-	93.2
Capsule	94	88	84	84	-	76	-	98.6
Hazelnut	97	95	87	72	-	-	-	98.9
Metal nut	89	86	76	82	-	60	-	96.9
Pill	91	85	87	68	-	83	-	96.5
Screw	96	96	80	87	-	94	-	99.5
Toothbrush	92	93	90	77		68	-	98.9
Transistor	90	86	80	66	-	-	-	81.0
Zipper	88	77	78	76	-	-	-	98.8
Average	87	82	74	78	75	77	89	**96.2**

Table 6: Sub-Image anomaly detection accuracy on MVTec (PRO %)

	Student	1-NN	OC-SVM	ℓ_2-AE	VAE	SSIM-AE	CNN-Dict	SPADE
Carpet	69.5	51.2	35.5	45.6	50.1	64.7	46.9	96.1
Grid	81.9	22.8	12.5	58.2	22.4	84.9	18.3	97.0
Leather	81.9	44.6	30.6	81.9	63.5	56.1	64.1	98.8
Tile	91.2	82.2	72.2	89.7	87.0	17.5	79.7	77.1
Wood	72.5	50.2	33.6	72.7	62.8	60.5	62.1	93.8
Bottle	91.8	89.8	85.0	91.0	89.7	83.4	74.2	95.6
Cable	86.5	80.6	43.1	82.5	65.4	47.8	55.8	85.3
Capsule	91.6	63.1	55.4	86.2	52.6	86.0	30.6	95.5
Hazelnut	93.7	86.1	61.6	91.7	87.8	91.6	84.4	94.8
Metal nut	89.5	70.5	31.9	83.0	57.6	60.3	35.8	94.1
Pill	93.5	72.5	54.4	89.3	76.9	83.0	46.0	96.2
Screw	92.8	60.4	64.4	75.4	55.9	88.7	27.7	97.4
Toothbrush	86.3	67.5	53.8	82.2	69.3	78.4	15.1	94.4
Transistor	70.1	68.0	49.6	72.8	62.6	72.5	62.8	67.8
Zipper	93.3	51.2	35.5	83.9	54.9	66.5	70.3	96.9
Average	85.7	64	47.9	79	63.9	69.4	51.5	**92.1**

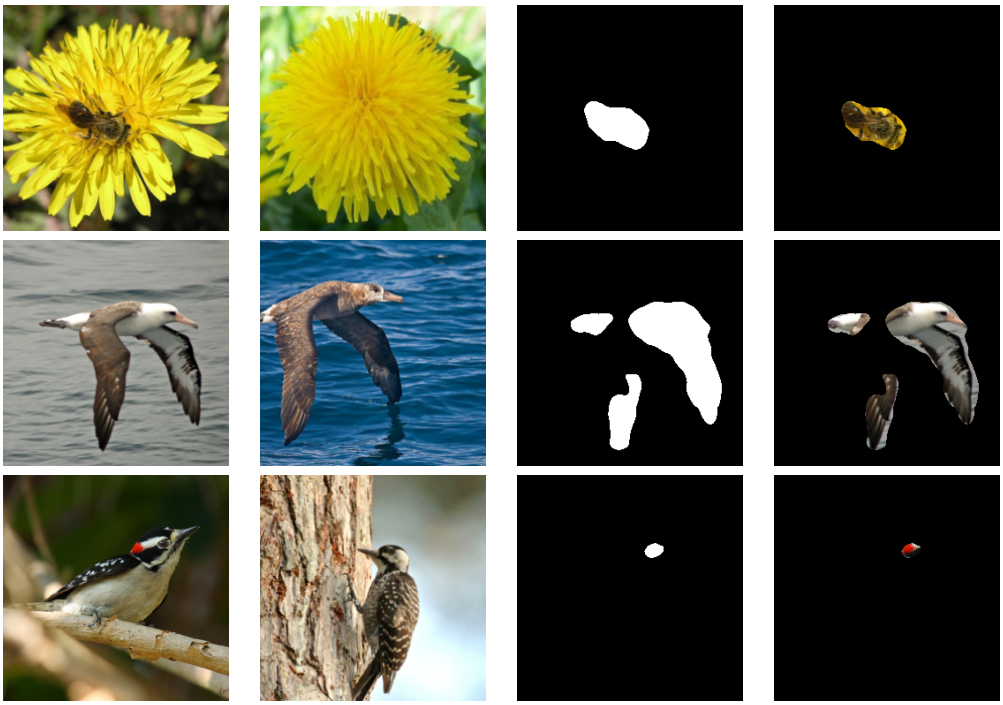

Figure 1: An evaluation of SPADE on detecting anomalies between flowers with or without insects (taken from one category of 102 Category Flower Dataset [8]) and bird varieties (taken from Caltech-UCSD Birds 200) [14]. (left to right) i) An anomalous image ii) A normal train set image iii) The mask detected by SPADE iv) The predicted anomalous image pixels. SPADE was able to detect the insect on the anomalous flower (top), the white colors of the anomalous albatross (center) and the red spot on the anomalous bird (bottom).

References

[1] 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. 2

[2] 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. 2

[3] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019. 2

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 2

[5] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[6] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 3

[7] 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, pages 366–374, 2007. 3

[8] 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*, pages 722–729. IEEE, 2008. 3, 6

[9] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 3

[10] 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*, pages 9592–9600, 2019. 3

[11] 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. 3

[12] 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. 3

[13] 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. 4

[14] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 6