Redefining Normal: A Novel Object-Level Approach for Multi-Object Novelty Detection

Mohammadreza Salehi, Nikolaos Apostolikas
Efstratios Gavves, Cees G. M. Snoek, and Yuki M. Asano

University of Amsterdam, The Netherlands
s.salehidehnavi@uva.nl

1 Additional Experiments

Table 1: Pascal VOC per class results in the uni-class setting. This paper sets a new state-of-the-art for the proposed multi-object benchmark by considerably passing the second best-performing method.

		KDAD	RDAD	MSAD	Transformaly	This paper
Normal class	0	96.2	88.7	99.7	98.0	99.7
	1	67.6	42.4	63.4	46.4	75.6
	2	90.8	63.3	96.3	86.5	97.6
	3	81.4	61.1	97.7	83.5	91.9
	4	82.2	55.3	87.1	71.0	92.6
	5	69.4	55.0	93.9	82.6	93.8
	6	71.4	51.7	83.7	61.9	90.8
	7	87.6	76.0	95.4	90.4	98.5
	8	84.4	52.6	92.2	73.7	97.7
	9	91.6	62.6	99.3	94.4	99.3
	10	77.2	44.2	97.1	90.9	98.7
	11	82.7	53.3	96.2	93.9	97.9
	12	83.9	60.7	96.3	94.3	99.2
	13	83.4	44.2	90.3	81.2	97.9
	14	79.2	50.0	83.4	76.9	89.2
	15	86.1	67.9	96.3	86.4	97.9
	16	87.6	66.6	93.6	84.30	97.7
	17	95.7	63.6	98.9	96.2	98.3
	18	72.8	40.4	77.3	58.1	92.7
	19	90.1	71.5	98.8	96.8	99.0
	AVG	83.0	58.9	91.8	82.5	**95.3**

1.1 Pascal VOC per class results

Here, we show the per-class results of both multi-class and single-class settings in Table 1 and Table 6. The results are obtained similar to Table 2 and Table 4 in the main paper. As is shown, this paper gets better average results in both uni-class and multi-class settings for the multi-object dataset. While Transformaly and MSAD get slightly better or comparable results on single-object datasets such as CIFAR10, they show significantly lower performance on multi-object samples. This performance drop supports our postulation that these methods have implicit object-centric assumptions.

Table 2: Ablation studies of architectural changes. All the numbers are AUROC. The performance for one semantic (CIFAR-10) and one pixel-level (MVTecAD) dataset is reported. As it is shown, different distillation and normalization functions get the best results for different datasets. As we are addressing semantic novelty detection, only top1 layer is distilled and L_2 normalization is applied on both networks. Guided masking is also used as it is consistently effective across different tasks.

(a) Ablating the effect of each modification applied to the baseline.

	Backbone	Num layer	Loss	Normalization		Results	
				Teacher	Student	CIFAR-10	MVTecAD
Modified	ViT-S16	top5	smooth L_1	Layer	-	89.4	**90.4**
	ViT-S16	top5	L_2	Layer	-	90.1	90.2
	ViT-S16	top5	L_2	-	-	89.6	86.8
	ViT-S16	top3	L_2	-	-	90.8	86.1
	ViT-S16	top3	L_2	Layer	-	91.1	90.8
	ViT-S16	top1	L_2	Layer	-	91.5	87.1
	ViT-S16	top1	L_2	L_2	L_2	**92.6**	85.3
Baseline	VGG-16	top5	L_2 + cosine	-	-	87.2	87.7

(b) Ablating the effect of masking on the top-performing models from Table 3a.

	Backbone	Num layer	Loss	Mask	Results	
					CIFAR-10	MVTecAD
With Masking	ViT-S16	top5	smooth L_1	Random	89.8	91.0
	ViT-S16	top5	smooth L_1	Guided	90.0	**91.4**
	ViT-S16	top1	L_2	Guided	**93.4**	88.1
Without Masking	ViT-S16	top5	smooth L_1	-	89.4	90.4
	ViT-S16	top1	L_2	-	92.6	85.3

1.2 COCO per class results

Here, we show the per-class results for COCO with a similar evaluations as Table 2 in the main paper. Table 4 shows the results. As is shown, this paper nearly always gets better results compared to MSAD and Transformaly. This particularly shows that this paper is a better fit for real-world applications since most real scenarios are multi-object scenes.

1.3 Analyzing computational efficiency

The training and inference times for both Transformaly and our method are shown in Table 5 using an A6000 GPU. Our method includes two training stages, requiring 3 and 10 epochs respectively, leading to a slightly longer training time. Despite this, our method achieves approximately 50% faster inference time, which is more critical in practical applications since anomaly detection models are usually trained once but used repeatedly.

2 Additional Ablations

Here, we provide ablation studies on the effect of different normalization and distillation functions applied to the features of different layers. As shown in Table 3b, the smooth L_1 [1] loss is the most effective distillation function for pixel-level tasks. Also, applying normalization is beneficial on both pixel-level and semantic tasks. When only the teacher network is normalized, the student's features are projected to the teacher's representation space by a shared linear head. Moreover, the results demonstrate that increasing the number of distillation layers is mainly beneficial for pixel-level and not semantic tasks. In this paper, we focus more on distilling the final block with L_2 normalization applied to all the network's features and L_2 distillation loss, as our task is semantic novelty detection.

In Table 3b, we choose the top-performing models from Table 3a and evaluate them in situations with different input masking procedures. As is shown, different kinds of masking are beneficial for both semantic and pixel-level tasks. Masking helps improve the results for CIFAR-10 by roughly 1.2% and MVTecAD by 1% , which shows the generality of the proposed approach. Also, guided masking shows consistently better performance compared to random masking, which supports the effectiveness of giving more attention to informative areas instead of random.

3 Qualitative Results

We are comparing our method with KDAD [2] in terms of the regions each method focuses on to calculate the anomaly score for each input image. In this comparison, the normal class is 'Airplane', and we expect the anomaly score to be lower for the areas containing airplanes and higher for regions containing abnormal objects. As shown in Figure 1, our method improves upon the upgraded KDAD baseline by producing more semantically meaningful results, with a focus on abnormal objects rather than the entire image.

Fig. 1: Qualitative comparison of the proposed method and KDAD. As shown, the proposed method focuses more on the abnormal object to produce the anomaly score for each image.

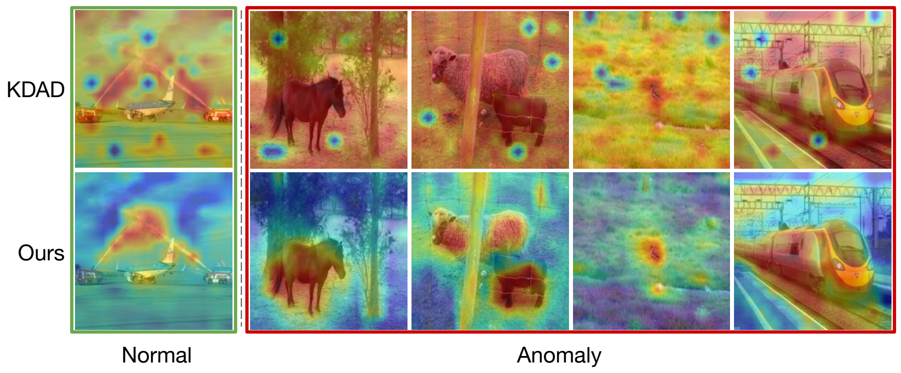

Table 4: COCO per class results. ID specifies the class ID in the dataset. The performance is only reported for valid class IDs.

ID	Transformaly	MSAD	This paper	ID	Transformaly	MSAD	This paper
0	-	-	-	46	67.2	70.8	94.4
1	46.1	58.2	77.5	47	57.0	89.3	88.1
2	56.5	71.5	88.3	48	79.5	75.0	96.2
3	57.9	68.2	86.7	49	73.6	93.9	93.6
4	82.6	90.4	96.5	50	72.0	91.4	94.5
5	93.4	97.8	98.6	51	64.5	90.9	91.6
6	79.7	86.1	95.6	52	81.2	83.3	94.8
7	91.9	97.0	98.4	53	78.9	90.3	92.9
8	63.9	73.2	87.2	54	81.0	90.6	97.3
9	74.0	83.8	95.9	55	80.3	95.3	95.7
10	79.9	88.8	96.3	56	92.9	91.3	98.3
11	59.6	86.6	95.4	57	82.3	97.3	95.2
12	-	-	-	58	81.7	93.3	97.8
13	76.7	88.2	95.0	59	88.9	96.6	98.5
14	83.1	93.8	97.5	60	76.0	98.1	96.1
15	56.8	66.3	79.6	61	67.2	94.6	95.9
16	62.5	70.5	82.0	62	49.2	92.4	81.0
17	65.1	87.2	97.1	63	59.8	65.8	95.9
18	44.0	57.6	82.1	64	53.0	89.3	84.9
19	75.8	92.8	95.1	65	73.4	68	97.4
20	85.1	96.6	98.5	66	-	-	-
21	79.5	94.3	98.2	67	59.1	91.0	91.1
22	95.7	98.3	98.7	68	-	-	-
23	94.9	98.9	98.7	69	-	-	-
24	98.0	99.3	99.4	70	94.7	79.0	99.5
25	95.0	98.7	99.4	71	-	-	-
26	-	-	-	72	74.0	98.9	96.2
27	51.8	64.4	78.6	73	77.3	90.1	97.3
28	62.1	70.6	91.6	74	91.4	91.8	98.9
29	-	-	-	75	56.3	97.3	95.3
30	-	-	-	76	90.4	91.7	98.8
31	44.8	58.1	78.8	77	43.9	96.6	85.6
32	54.8	75.4	91.9	78	85.7	63.7	98.5
33	48.1	75.1	91.0	79	83.8	95.4	98.4
34	60.5	92.7	94.3	80	89.4	95.2	98.7
35	97.7	99.3	99.4	81	87.8	97.0	98.0
36	93.7	97.8	98.3	82	79.6	96.4	97.1
37	90.0	94.6	92.8	83	-	-	-
38	72.2	96.1	97.7	84	58.0	93.8	90.1
39	93.4	98.8	98.2	85	68.6	78.4	90.0
40	96.5	99.2	99.0	86	72.0	78.5	94.7
41	75.1	95.5	98.7	87	56.0	88.0	93.7
42	89.4	97.7	98.4	88	65.8	84.6	95.0
43	97.2	99.6	99.7	89	81.3	87.6	94.8
44	53.4	70.8	84.3	90	74.9	94.0	95.4
45	-	-	-				

Table 5: Computational efficiency on CIFAR-10.

Method	**Batch size**	**Training epochs**	**Training time (min)**	**Inference FPS**	**Performance**
Transformaly	32	10	63	100	94.9
Ours	32	13 (3 + 10)	75 (45 + 30)	164	98.6

Table 6: Pascal VOC per class results in the multi-class setting. This paper pushes the state-of-the-art by roughly 5% in the proposed multi-object benchmark.

		MSAD	Transformaly	RDAD	KDAD	DSVDD	**This paper**
Abnormal class	0	33.3	21.3	48.4	36.1	45.4	40.9
	1	44.1	91.1	60.1	62.4	48.3	68.0
	2	56.9	39.4	51.6	46.6	47.8	79.6
	3	41.5	67.9	53.6	71.9	46.1	85.4
	4	39.1	70.6	52.2	40.3	34.3	48.4
	5	51.7	73.9	56.1	83.9	43.6	70.2
	6	34.3	70.9	53.0	84.4	56.7	91.9
	7	48.2	51.0	38.7	49.6	41.2	60.7
	8	16.8	62.4	42.5	62.9	46.6	33.4
	9	51.9	23.9	42.5	53.6	47.30	61.7
	10	66.6	24.4	44.3	59.5	44.4	58.2
	11	50.7	51.0	43.2	55.0	51.0	60.1
	12	60.3	56.9	43.2	55.1	51.7	53.4
	13	59.0	55.7	44.2	63.8	58.5	71.5
	14	55.4	53.2	41.5	57.6	53.5	69.3
	15	43.8	64.2	39.9	72.4	42.1	83.0
	16	45.9	61.5	39.9	35.9	40.2	54.8
	17	58.1	25.8	39.8	38.2	44.8	49.0
	18	51.5	65.7	42.6	76.0	59.4	82.0
	19	51.4	51.4	44.6	56.8	41.9	56.2
	AVG	48.0	54.1	46.2	58.1	47.3	**63.0**

References

1. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015) 3
2. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14902–14912 (June 2021) 3