Table 1. Anomaly Detection Performance (AUROC) on MVTec AD [5]. PaDiM* denotes a result from [14] with problem-specific backbone selection. The total count of misclassifications was determined as the sum of false-positive and false-negative predictions given a F1-optimal threshold. We did not have individual anomaly scores for competing methods so could compute this number only for *PatchCore*.

Method	SPADE [10]	PatchSVDD [56]	DifferNet [42]	PaDiM [14]	Mah.AD [40]	PaDiM* [14]	PatchCore−25%	PatchCore−10%	PatchCore−1%
AUROC ↑	85.5	92.1	94.9	95.3	95.8	97.9	**99.1**	99.0	99.0
Error ↓	14.5	7.9	5.1	4.7	4.2	2.1	**0.9**	1.0	1.0
Misclassifications ↓	-	-	-	-	-	-	**42**	47	49

Table 2. Anomaly Segmentation Performance (pixelwise AUROC) on MVTec AD [5].

Method	AE_{SSIM} [5]	γ-VAE + grad. [15]	CAVGA-R_w [52]	PatchSVDD [56]	SPADE [10]	PaDiM [14]	PatchCore−25%	PatchCore−10%	PatchCore−1%
AUROC ↑	87	88.8	89	95.7	96.0	97.5	**98.1**	**98.1**	98.0
Error ↓	13	11.2	11	4.3	4.0	2.5	**1.9**	**1.9**	2.0

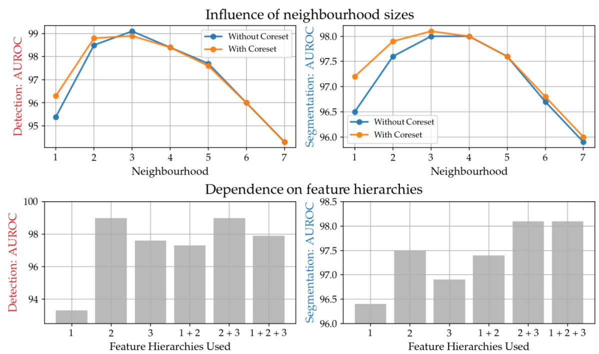

Figure 4. Local awareness and network feature depths vs. detection performance. PRO score results in the supplementary.

4.2. Anomaly Detection on MVTec AD

The results for image-level anomaly detection on MVTec are shown in Table 1. For *PatchCore* we report on various levels of memory bank subsampling (25%, 10% and 1%). For all cases, *PatchCore* achieves significantly higher mean image anomaly detection performance with consistently high performance on all sub-datasets (see supplementary B for detailed comparison). Please note, that a reduction from an error of 2.1% (PaDiM) to 0.9% for *PatchCore*−25% means a reduction of the error by 57%. In industrial inspection settings this is a relevant and significant reduction. For MVTec at optimal F1 threshold, there are only 42 out of 1725 images classified incorrectly and a third of all classes are solved perfectly. In the supplementary material B we also show that both with F1-optimal working point and at full recall, classification errors are also lower when compared to both SPADE and PaDiM. With *PatchCore*, less than 50 images remain misclassified. In addition, *PatchCore* achieves state-of-the-art anomaly segmentation, both measured by pixelwise AUROC (Table 2, 98.1 versus 97.5 for PaDiM) and PRO metric (Table 3, 93.5 versus 92.1). Sample segmentations in Figure 1 offer qualitative impressions of the accurate anomaly localization.

In addition, due to the effectiveness of our coreset memory subsampling, we can apply PatchCore−1% on images of higher resolution (e.g. 280/320 instead of 224) and ensemble systems while retaining inferences times less than PatchCore−10% on the default resolution. This allows us to further push image- and pixel-level anomaly detection as highlighted in Tab. 4 (detailed results in supplementary), in parts more than halving the error again (e.g. 1% → 0.4% for image-level AUROC).

4.3. Inference Time

The other dimension we are interested in is inference time. We report results in Table 5 (implementation details in supp. A) comparing to reimplementations of SPADE [10] and PaDiM [14] using WideResNet50 and operations on GPU where possible. These inference times include the forward pass through the backbone. As can be seen, inference time for joint image- and pixel-level anomaly detection of *PatchCore*−100% (without subsampling) are lower than SPADE [10] but with higher performance. With coreset subsampling, *Patchcore* can be made even faster, with lower inference times than even PaDiM while retaining state-of-the-art image-level anomaly detection and segmentation performance. Finally, we examine *PatchCore*−100% with approximate nearest neighbour search (IVFPQ [27]) as an orthogonal way of reducing inference time (which can also be applied to SPADE, however which already performs notably worse than even *PatchCore*−1%). We find a performance drop, especially for image-level anomaly detection, while inference times are still higher than *PatchCore*−1%. Though even with performance reduction, approximate nearest neighbour search on *PatchCore*−100% still outperforms other methods. A combination of coreset and approximate nearest neighbour would further reduce inference time, allowing scaling to much larger datasets.

4.4. Ablations Study

We report on ablations for the locally aware patch-features and the coreset reduction method. Supplementary experiments show consistency across different backbones (§C.2), scalability with increased image resolution (§C.3) and a qualitative analysis of remaining errors (§C.4).