表1. MVTec AD [5]における異常検出性能（AUROC）。PaDiM*は[14]における問題特異的なバックボーン選択を用いた結果を示す。誤分類の総数は、F1最適閾値に基づく偽陽性および偽陰性予測の合計として算出された。競合手法の個々の異常スコアがなかったため、この数値はPatchCoreのみに対して計算された。

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

表2. MVTec AD [5]における異常セグメンテーション性能（ピクセル単位のAUROC）

Method	AE_{SSIM} [5]	γ-VAE + grad. [15]	CAVGA-R_u [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

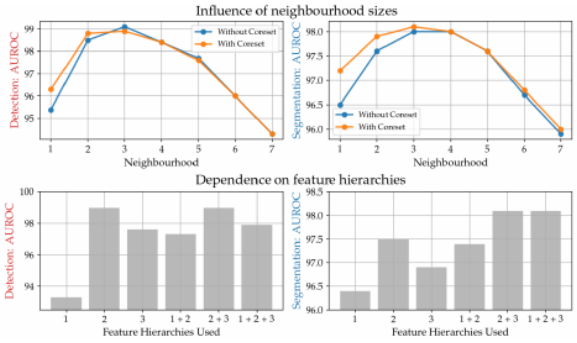

図4. 局所意識とネットワーク特徴の深さ対検出性能。PROスコアの結果は補足資料に記載されています。

4.2. MVTec ADにおける異常検出

MVTecにおける画像レベル異常検出の結果は表1に示されています。PatchCoreでは、メモリバンクサブサンプリングのさまざまなレベル（25%、10%、1%）について報告しています。すべてのケースにおいて、PatchCoreはすべてのサブデータセットで一貫して高い性能を維持しつつ、平均画像異常検出性能で著しく高い性能を達成しています（詳細な比較は補足資料Bを参照）。注意：PaDiMの2.1%の誤差からPatchCore-25%の0.9%への減少は、誤差を57%削減したことを意味します。産業検査の現場では、これは関連性があり重要な削減です。MVTecの最適F1閾値において、1725枚の画像のうち42枚が誤分類され、全クラスの3分の1が完全に解決されています。補足資料Bでは、F1最適動作点および完全なリコール時においても、SPADEとPaDiMと比較して分類誤りが低いことを示しています。PatchCoreでは、誤分類される画像は50枚未満です。さらに、PatchCoreはピクセル単位のAUROC（表2、98.1対PaDiMの97.5）とPROメトリクス（表3、93.5対92.1）の両方で最先端の異常セグメンテーションを実現しています。図1のサンプルセグメンテーションは、正確な異常局在化の定性的印象を提供しています。

さらに、コアセットメモリのサブサンプリングの有効性により、デフォルト解像度での推論時間をPatchCore-10%未満に維持しつつ、高解像度画像（例：280／320ではなく224）やアンサンブルシステムにPatchCore-1%を適用できます。これにより、表4で強調されているように、画像レベルとピクセルレベルの異常検出をさらに推進できます（詳細な結果は補足参照）。4（詳細な結果は補足資料参照）で示されたように、画像レベルAUROCで1%から0.4%へなど、エラーをさらに半分以下に削減できます。

4.3. Inference Time

もう1つの関心事の次元は推論時間です。表5（実装の詳細は補足Aを参照）に、WideResNet50を使用し、可能な限りGPU上で実行したSPADE [10]とPaDiM [14]の再実装との比較結果を報告します。これらの推論時間には、バックボーンを通るフォワードパスが含まれています。図からわかるように、PatchCore-100%（サブサンプリングなし）の共同画像レベルとピクセルレベル異常検出の推論時間はSPADE [10]よりも短く、かつ性能が向上しています。コアセットサブサンプリングを使用すると、PatchCoreはさらに高速化され、PaDiMよりも短い推論時間を実現しつつ、最先端の画像レベル異常検出とセグメンテーション性能を維持できます。最後に、PatchCore-100%に近似最近傍検索（IVFPQ [27]）を適用し、推論時間を削減する別の方法を検討しました（これはSPADEにも適用可能ですが、SPADEは既にPatchCore-1%よりも著しく劣る性能を示しています）。画像レベル異常検出において性能低下が見られ、推論時間は依然としてPatchCore-1%よりも長くなっています。性能低下にもかかわらず、PatchCore-100%における近似最近傍検索は他の手法よりも優れています。コアセットと近似最近傍検索の組み合わせはさらに推論時間を短縮し、はるかに大規模なデータセットへのスケーリングを可能にします。

4.4. Ablations Study

局所的に意識したパッチ特徴量とコアセット削減手法に関するアブレーション実験の結果を報告します。補足実験では、異なるバックボーン間での一貫性（§C.2）、画像解像度向上時のスケーラビリティ（§C.3）、残存エラーの定性的分析（§C.4）が示されています。

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

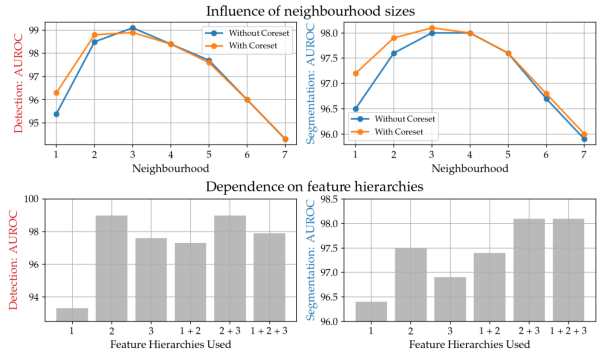

Figure 4. Local awareness and network feature depths vs. detection performance. PRO score results in the supplementary.

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).