Supplementary:
産業用異常検出における完全回収率の実現

A. 実装の詳細

モデルはPython 3.7 [51]とPyTorch [37]で実装しました。実験はNvidia Tesla V4 GPU上で実行されました。torchvisionのImageNet事前学習済みモデルとPyTorch Image Modelsリポジトリ [53]からモデルを使用しました。デフォルトでは、[10]と[14]に従い、PatchCoreは直接比較のためWideResNet50バックボーン [57]を使用しています。パッチレベルの特徴量は、ブロック2と3の最終出力のフィーチャーマップ集約から取得されます。すべての最近傍検索と距離計算には、faiss [27]を使用しています。

B. MVTec ADの完全比較

このセクションでは、MVTec ADに関するより詳細な比較を掲載しています。利用可能なすべてのMVTec ADサブデータセットにおいて、より多くのモデルとより詳細な性能比較を含めています。論文の本編では、この内容は§4.2で参照されています。対応する結果表はS1、S2、S3です。PatchCore-25%は15のMVTecデータセットのうち6つを解決し、ほとんどのデータセットおよび平均で最高のアウロコ性能を達成しています。
図S3は、PatchCoreのバリエーションおよび再実装された比較可能な手法SPADE [10]とPaDiM [14]（WideResNet50バックボーンを使用）の精度-再現率曲線とROC曲線を示しています。また、100%リコール時とF1最適閾値下での分類誤差をプロットし、比較可能な作業点を示しています。図から明らかなように、PatchCoreは定義された作業点においても一貫して低い分類誤差を達成し、データセット全体で近似最適なPrecision-Recall曲線とROC曲線を示しています。これに対し、SPADEとPaDiMは同様の性能を示していません。

最後に、表S4では、より大きな画像サイズ（280×280）とWideResNet-101バックボーンを使用したPatchCore-1%によるさらなる性能向上を、MVTec ADサブデータセット全体で詳細に示しています。これにより、より大きな画像でも推論時に効率的な異常検出が可能です。

C. 追加のアブレーションと詳細

C.1. 詳細な低ショット実験

このセクションでは、本論文の主要部分（§4.5）で提供された低ショット手法の研究に関する詳細な数値値を提供します。結果は表S5に示されており、検出と異常局在化メトリクスの数値が一貫して高いことが確認されました。

C.2. 事前学習済みネットワークへの依存性

PatchCoreを異なるバックボーンでテストしました。結果はS6に示されています。異なるバックボーンの選択に関わらず、結果は主に安定しています。WideResNet50の選択は、SPADEとPaDiMと比較可能にするためです。

C.3. 画像解像度の影響

次に、画像サイズが性能に与える影響を分析します。本論文では、過去の研究との比較のため224 × 224の画像サイズを使用しました。図S4では、画像サイズを288 × 288、360 × 360、448 × 448と変化させ、近傍サイズ（P）を3、5、7、9の範囲で変更しています。PatchCoreでは検出性能がやや向上し、性能が飽和する傾向が見られます。異常セグメンテーションでは一貫した向上が観察されるため、正確な局所化が重要であれば、この点は検証すべき要素です。

C.4. 残存する誤分類

高い画像レベル異常検出性能により、残りの誤分類を詳細に分析できます。異常とみなされるスコアの閾値（ワーキングポイント）は、F1最適点を使用して計算します。この閾値を使用すると、合計19件の偽陽性エラーと23件の偽陰性エラーが残ります。これらのエラーは、図S1とS2に可視化されています。各セグメンテーションマップは閾値値に正規化されているため、一部のケースでは背景スコアが不均衡に強調されています。

図S1を見ると、偽陽性エラーのほとんどは、a)（青色）ラベル付けの曖昧さ、つまり異常とラベル付けされる可能性のある画像の変化、およびb)（オレンジ色）非常に高い名目上の分散から生じていることがわかります。

Supplementary:
Towards Total Recall in Industrial Anomaly Detection

A. Implementation Details

We implemented our models in Python 3.7 [51] and PyTorch [37]. Experiments are run on Nvidia Tesla V4 GPUs. We used torchvision ImageNet-pretrained models from torchvision and the PyTorch Image Models repository [53]. By default, following [10] and [14], *PatchCore* uses a WideResNet50-backbone [57] for direct comparability. Patch-level features are taken from feature map aggregation of the final outputs in blocks 2 and 3. For all nearest neighbour retrieval and distance computations, we use `faiss` [27].

B. Full MVTec AD comparison

This section contains a more detailed comparison on MVTec AD. We include more models and a more finegrained performance comparison on all MVTec AD sub-datasets where available. In the main part of the paper this has been referenced in §4.2. The corresponding result tables are S1, S2 and S3. We observe that *PatchCore*-25% solves six of the 15 MVTec datasets and achieves highest AUROC performance on most datasets and in average.

Figure S3 show Precision-Recall and ROC curves for *PatchCore* variants as well as reimplemented, comparable methods SPADE [10] and PaDiM [14] using a WideResNet50 backbone. We also plot classification error both at 100% recall and under a F1-optimal threshold to give a comparable working point. As can be seen, *PatchCore* achieves consistently low classification errors with defined working points as well, with near-optimal Precision-Recall and ROC curves across datasets, in contrast to SPADE and PaDiM.

Finally, Table S4 showcases the detailed performance on all MVTec AD subdatasets for larger imagesizes (280×280) and a WideResNet-101 backbone for further performance boosts using PatchCore-1%, which allows for efficient anomaly detection at inference time even with larger images.

C. Additional Ablations & Details

C.1. Detailed Low-Shot experiments

This section offers detailed numerical values to the low-shot method study provided in the main part of this work (§4.5). The results are included in Table S5 and we find consistently higher numbers for detection and anomaly localization metrics.

C.2. Dependency on pretrained networks

We tested *PatchCore* with different backbones, the results are shown in S6. We find that results are mostly stable over the choice of different backbones. The choice of WideResNet50 was made to be comparable with SPADE and PaDiM.

C.3. Influence of image resolution

Next we study the influence of image size on performance. In the main paper we have used 224×224 to be comparable with prior work. In Figure S4 we vary the image size from 288×288, 360×360 to 448×448 and the neighborhood sizes (P) within 3, 5, 7, and 9. We observe slightly increased detection performance and the performance saturates for *PatchCore* . For anomaly segmentation we observe a consistent increase, so if good localization is of importance, this is an ingredient to validate over.

C.4. Remaining Misclassifications

The high image-level anomaly detection performance allows us to look into all remaining misclassifications in detail. We compute the working point (threshold above which scores are considered anomalous) using the F1-optimal point. With this threshold a total of 19 false-positive and 23 false-negative errors remain, all of which are visualized in Figures S1 and S2. Each segmentation map was normalized to the threshold value, so in some cases background scores are pronounced disproportionally.

Looking at Figure S1, we find that the majority of false-positive errors either stem from a) (in blue) ambiguity in labelling , i.e., image changes that could also be potentially labelled as anomalous, and b) (in orange) very high nominal variance,