

産業環境で必要な抽象的な特徴量。さらに、これらの手法がテスト時に使用可能な名目上のコンテキストは、抽出可能な高レベルの特徴表現の数が少ないため、実質的に制限されています。

本論文では、PatchCoreを有効な解決策として提案する。その理由は、(1) テスト時に利用可能な名目上の情報を最大化すること、(2) ImageNetクラスへのバイアスを軽減すること、(3) 高い推論速度を維持することの3点にある。画像が単一のパッチが異常であるだけで異常と分類できるという事実[14, 56]に基づき、PatchCoreは局所的に集約された中間レベルの特徴パッチを活用してこれを実現します。中間レベルネットワークの特徴パッチの使用により、PatchCoreは高解像度においてImageNetクラスへのバイアスを最小限に抑えつつ、局所的な近傍における特徴の集約により十分な空間的文脈を維持します。これにより、PatchCoreはテスト時に利用可能な名目上の文脈を最適に活用できる広範なメモリバンクを実現します。最後に、実用性を考慮し、PatchCoreは名目上の特徴バンクに対して貪欲なコアセットサブサンプリング[1]を導入し、抽出されたパッチレベルメモリバンクの冗長性を削減するとともに、ストレージメモリと推論時間を大幅に削減する重要な要素として機能し、現実的な産業応用において非常に魅力的なソリューションとなっています。

MVTec AD[5]や専門的な磁気タイル欠陥 (MTD) [26]の産業用異常検出ベンチマークにおける多様な実験は、PatchCoreの産業用異常検出における力を示しています。PatchCoreは、MVTec ADとMTDにおいて最先端の画像レベル検出スコアを達成し、MVTec ADではほぼ完璧なスコア (AUROC 99.6%まで) を実現し、従来の方法の検出誤りを半分以上に削減しています。また、産業用異常局所化性能においても最先端の性能を示しています。PatchCoreは、手元のデータセットでのトレーニングを必要とせずに、高速な推論時間を維持しながらこれらの成果を達成しています。これにより、PatchCoreは産業用異常検出の実用的な用途において非常に魅力的です。さらに、追加の実験では、PatchCoreの高いサンプル効率が示されており、既存の異常検出手法と同等の性能を達成しつつ、名目上のトレーニングデータのほんの一部分のみを使用しています。

## 2. Related Works

ほとんどの異常検出モデルは、名義データ固有の表現を学習する能力に依存しています。これは例えばオートエンコーダーモデル[44]の使用を通じて実現可能です。正規特徴分布のより良い推定を促進するため、ガウス混合モデル[60]、生成対抗トレーニング目標[2, 39, 43]、事前定義された物理的拡張に対する不変性[25]、再構築の再導入に対する隠れた特徴の頑健性[29]、プロトタイプ記憶バンク[21]、アテンションガイド[52]、構造的目標[7, 59]、または制約付き表現空間[38]に基づく拡張が提案されています。他の無監督表現学習手法も同様に利用可能です。

例えば、GANs[13]、事前定義された幾何変換の予測学習[20]、または正規化フロー[42]を通じてです。それぞれの名義表現と新規テスト表現が与えられた場合、異常検出は再構築誤差[44]、k近傍までの距離[18]、またはこれらの特徴量上にOC-SVMs[46]やSVDD[50, 56]のような1クラス分類モデルの微調整を行うことで、単純な問題となります。これらのアプローチのほとんどでは、ピクセル単位の再構築誤差に基づいて異常局所化が自然に実現されます。また、GradCAM [47] や XRAI [28] のような注目度に基づくアプローチは、異常セグメンテーション [42, 45, 52] にも適用可能です。

産業用異常検出。学習された標準的な表現を用いた一般的な異常検出に関する文献は豊富ですが、産業用画像データには独自の課題が存在します[5]。これに対し、[4]以降の研究では、ImageNet[16]などの大規模な外部自然画像データセットで事前訓練されたモデルを使用し、手元のデータへの適応なしに最先端の検出性能を示しています。これにより、事前学習済み特徴のより良い再利用に依存する他の産業用異常検出手法が提案されました。例えばSPADE [10]は、多様な特徴階層からなるメモリバンクを活用し、kNNベースの[18]異常セグメンテーションと画像レベル異常検出を実現しています。同様に、[14]は最近PaDiMを提案し、局所的に制約されたbag-of-featuresアプローチ[8]を採用し、パッチレベルの特徴分布のモーメント (平均と共分散) を推定し、パッチレベルのマハラノビス距離測定[33]に適用しています。このアプローチは、[40]で全画像を対象に研究されたものと類似しています。自然事前学習データと産業用画像データ間の分布シフトをより適切に考慮するため、後続の適応処理を実施できます。例えば、[6, 45]で採用された学生-教師知識蒸留[24]や、事前学習済みネットワークの特徴量[42]上に訓練された正規化フロー[17, 30]などが挙げられます。

PatchCoreで使用される具体的なコンポーネントは、主にSPADEとPaDiMに関連しています。SPADEは、事前訓練されたバックボーンネットワークから抽出された名目上の特徴量のメモリバンクを活用し、画像レベルとピクセルレベルの異常検出にそれぞれ異なるアプローチを採用しています。PatchCoreも同様にメモリバンクを使用しますが、より高い性能を実現するために、近隣意識型のパッチレベルの特徴量が重要です。これは、より多くの名目上の文脈が保持され、より適切な誘導バイアスが組み込まれるためです。さらに、メモリバンクはコアセットサブサンプリングにより、高い性能を維持しつつ推論コストを低減するように設計されています。コアセットは、利用可能なセットの構造を最もよく近似する部分集合を特定し、コストを大幅に削減しながら近似解を見つけるための手法として、基本的なkNNやkMeansアプローチ[22]、混合モデル[19]などで長年活用されてきました。最近では、コアセットベースの手法はディープラーニング手法にも応用され、例えばネットワークの剪定[34]、アクティブラーニング[48]、有効なデータの増加などに活用されています。

the abstract features required in an industrial environment. In addition, nominal context usable by these methods at test time is effectively limited by the small number of extractable high-level feature representations.

In this paper, we present *PatchCore* as an effective remedy by (1) maximizing nominal information available at test time, (2) reducing biases towards ImageNet classes and (3) retaining high inference speeds. Relying on the fact that an image can be already classified as anomalous as soon as a single patch is anomalous [14, 56], *PatchCore* achieves this by utilizing locally aggregated, mid-level features patches. The usage of mid-level network patch features allows *PatchCore* to operate with minimal bias towards ImageNet classes on a high resolution, while a feature aggregation over a local neighbourhood ensures retention of sufficient spatial context. This results in an extensive memory bank allowing *PatchCore* to optimally leverage available nominal context at test time. Finally, for practical applicability, *PatchCore* additionally introduces greedy coreset subsampling [1] for nominal feature banks as a key element to both reduce redundancy in the extracted, patch-level memory bank as well as significantly bringing down storage memory and inference time, making *PatchCore* very attractive for realistic industrial use cases.

Thorough experiments on the diverse MVTec AD [5] as well as the specialized Magnetic Tile Defects (MTD) [26] industrial anomaly detection benchmarks showcase the power of *PatchCore* for industrial anomaly detection. It achieves state-of-the-art image-level detection scores on MVTec AD and MTD, with nearly perfect scores on MVTec AD (up to AUROC 99.6%), reducing detection error of previous methods by **more than half**, as well as state-of-the-art industrial anomaly localization performance. *PatchCore* achieves this while retaining fast inference times without requiring training on the dataset at hand. This makes *PatchCore* very attractive for practical use in industrial anomaly detection. In addition, further experiments showcase the high sample efficiency of *PatchCore*, matching existing anomaly detection methods in performance while using only a fraction of the nominal training data.

## 2. Related Works

Most anomaly detection models rely on the ability to learn representations inherent to the nominal data. This can be achieved for example through the usage of autoencoding models [44]. To encourage better estimation of the nominal feature distribution, extensions based on Gaussian mixture models [60], generative adversarial training objectives [2, 39, 43], invariance towards predefined physical augmentations [25], robustness of hidden features to reintroduction of reconstructions [29], prototypical memory banks [21], attention-guidance [52], structural objectives [7, 59] or constrained representation spaces [38] have been pro-

posed. Other unsupervised representation learning methods can similarly be utilised, such as via GANs [13], learning to predict predefined geometric transformations [20] or via normalizing flows [42]. Given respective nominal representations and novel test representations, anomaly detection can then be a simple matter of reconstruction errors [44], distances to  $k$  nearest neighbours [18] or finetuning of a one-class classification model such as OC-SVMs [46] or SVDD [50, 56] on top of these features. For the majority of these approaches, anomaly localization comes naturally based on pixel-wise reconstruction errors, saliency-based approaches such as GradCAM [47] or XRAI [28] can be used for anomaly segmentation [42, 45, 52] as well.

**Industrial Anomaly Detection.** While literature on general anomaly detection through learned nominal representations is vast, industrial image data comes with its own challenges [5], for which recent works starting with [4] have shown state-of-the-art detection performance using models pretrained on large external natural image datasets such as ImageNet [16] without any adaptation to the data at hand. This has given rise to other industrial anomaly detection methods reliant on better reuse of pretrained features such as SPADE [10], which utilizes memory banks comprising various feature hierarchies for finegrained, kNN-based [18] anomaly segmentation and image-level anomaly detection. Similarly, [14] recently proposed PaDiM, which utilizes a locally constrained bag-of-features approach [8], estimating patch-level feature distribution moments (mean and covariance) for patch-level Mahalanobis distance measures [33]. This approach is similar to [40] studied on full images. To better account for the distribution shift between natural pre-training and industrial image data, subsequent adaptation can be done, e.g. via student-teacher knowledge distillation [24] such as in [6, 45] or normalizing flows [17, 30] trained on top of pretrained network features [42].

The specific components used in *PatchCore* are most related to SPADE and PaDiM. SPADE makes use of a memory-bank of nominal features extracted from a pretrained backbone network with separate approaches for image- and pixel-level anomaly detection. *PatchCore* similarly uses a memory bank, however with neighbourhood-aware patch-level features critical to achieve higher performance, as more nominal context is retained and a better fitting inductive bias is incorporated. In addition, the memory bank is coreset-subsampled to ensure low inference cost at higher performance. Coresets have seen longstanding usage in fundamental kNN and kMeans approaches [22] or mixture models [19] by finding subsets that best approximate the structure of some available set and allow for approximate solution finding with notably reduced cost [1, 9]. More recently, coreset-based methods have also found their way into Deep Learning approaches, e.g for network pruning [34], active learning [48] and increasing effective data