PaDiM：異常検出と局所化のためのパッチ分布モデリングフレームワーク

Thomas Defard, Aleksandr Setkov, Angelique Loesch, Romaric Audigier
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
thomas.defard@imt-atlantique.net, {aleksandr.setkov, angelique.loesch, romaric.audigier}@cea.fr

要約—私たちは、1クラス学習設定において画像内の異常を同時に検出・局所化する新たなフレームワーク「パッチ分布モデリング（PaDiM）」を提案します。PaDiMは、パッチ埋め込みに事前学習済みの畳み込み神経ネットワーク（CNN）を利用し、正常クラスの確率的表現を得るために多変量ガウス分布を採用しています。また、CNNの異なるセマンティックレベル間の相関を活かすことで、異常の局所化を向上させます。PaDiMは、MVTec ADおよびSTCデータセットにおいて、異常検出と局所化の両方で現在の最先端手法を凌駕します。現実の視覚的産業検査に合わせるため、評価プロトコルを拡張し、非一致データセットにおける異常局所化アルゴリズムの性能を評価します。PaDiMの最先端性能と低複雑度は、多くの産業応用における有望な候補となります。

I. Introduction

人間は、均一な自然画像の集合の中から異質なまたは予期しないパターンを検出することができます。このタスクは異常検出または新規性検出と呼ばれ、視覚的産業検査を含む多くの応用分野があります。しかし、製造ラインにおける異常は極めて稀なイベントであり、手動での検出は手間がかかります。したがって、異常検出の自動化は、注意力の低下を回避し、人間のオペレーター作業を容易にすることで、継続的な品質管理を可能にします。本論文では、異常検出に焦点を当て、特に産業検査の文脈における異常局所化に重点を置きます。コンピュータビジョンにおいて、異常検出は画像に異常スコアを付与する作業です。異常局在化はより複雑なタスクであり、各ピクセルまたはピクセルのパッチに異常スコアを付与し、異常マップを出力します。これにより、異常局在化はより正確で解釈可能な結果を生成します。当手法でMVTec異常検出（MVTec AD）データセット［1］の画像から異常を局在化した際の異常マップの例は図1に示されています。

異常検出は、正常クラスと異常クラス間の二値分類です。しかし、異常例が不足している場合が多く、さらに異常は予期しないパターンを示すため、このタスクで完全な監督学習モデルを訓練することはできません。したがって、異常検出モデルは通常、単一クラス学習設定で推定されます。つまり、訓練データセットには正常クラスの画像のみが含まれ、訓練中に異常例は利用できません。テスト時、正常な訓練データセットと異なる例は異常として分類されます。

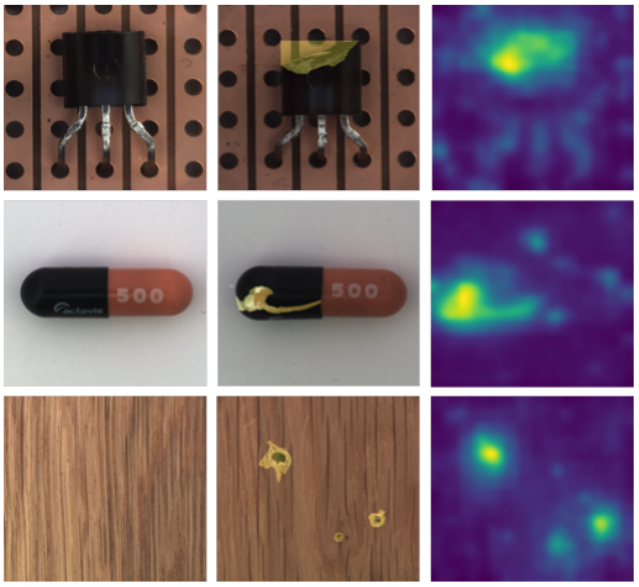


図1. MVTec AD ［1］からの画像サンプル。左列：トランジスタ、カプセル、ウッドクラスの正常画像。中央列：同じクラスの画像で、黄色でハイライトされた ground truth 異常。右列：当社の PaDiM モデルで得られた異常ヒートマップ。黄色の領域は検出された異常に対応し、青色の領域は正常領域を示します。カラー表示が推奨されます。

最近、異常局所化と異常検出タスクを1クラス学習設定で組み合わせる複数の方法が提案されています［2］-［5］。しかし、これらの方法は、深層神経ネットワークのトレーニング［3］、［6］を必要とし、煩雑な場合があるか、またはテスト時にトレーニングデータセット全体にK-近傍法（K-NN）アルゴリズム［7］を適用する［4］、［5］ものです。KNNアルゴリズムの線形複雑さは、トレーニングデータセットのサイズが増加するにつれ、時間と空間の複雑さを増加させます。これらの2つのスケーラビリティ問題は、異常局所化アルゴリズムの産業分野での展開を妨げる可能性があります。

上記の問題を緩和するため、私たちは新しい異常検出と局所化アプローチであるPaDiM（パッチ分布モデリング）を提案します。これは事前訓練された畳み込み神経ネットワーク（CNN）を埋め込み抽出に利用し、以下の2つの特性を持っています：

- 各パッチの位置は多変量ガウス分布で記述されます；
- PaDiM は、異なる領域間の相関関係を考慮します。

PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization

Thomas Defard, Aleksandr Setkov, Angelique Loesch, Romaric Audigier
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
thomas.defard@imt-atlantique.net, {aleksandr.setkov, angelique.loesch, romaric.audigier}@cea.fr

***Abstract*—We present a new framework for Patch Distribution Modeling, PaDiM, to concurrently detect and localize anomalies in images in a one-class learning setting. PaDiM makes use of a pretrained convolutional neural network (CNN) for patch embedding, and of multivariate Gaussian distributions to get a probabilistic representation of the normal class. It also exploits correlations between the different semantic levels of CNN to better localize anomalies. PaDiM outperforms current state-of-the-art approaches for both anomaly detection and localization on the MVTec AD and STC datasets. To match real-world visual industrial inspection, we extend the evaluation protocol to assess performance of anomaly localization algorithms on non-aligned dataset. The state-of-the-art performance and low complexity of PaDiM make it a good candidate for many industrial applications.**

I. Introduction

Humans are able to detect heterogeneous or unexpected patterns in a set of homogeneous natural images. This task is known as anomaly or novelty detection and has a large number of applications, among which visual industrial inspections. However, anomalies are very rare events on manufacturing lines and cumbersome to detect manually. Therefore, anomaly detection automation would enable a constant quality control by avoiding reduced attention span and facilitating human operator work. In this paper, we focus on anomaly detection and, in particular, on anomaly localization, mainly in an industrial inspection context. In computer vision, anomaly detection consists in giving an anomaly score to images. Anomaly localization is a more complex task which assigns each pixel, or each patch of pixels, an anomaly score to output an anomaly map. Thus, anomaly localization yields more precise and interpretable results. Examples of anomaly maps produced by our method to localize anomalies in images from the MVTec Anomaly Detection (MVTec AD) dataset [1] are displayed in Figure 1.

Anomaly detection is a binary classification between the normal and the anomalous classes. However, it is not possible to train a model with full supervision for this task because we frequently lack anomalous examples, and, what is more, anomalies can have unexpected patterns. Hence, anomaly detection models are often estimated in a one-class learning setting, *i.e.*, when the training dataset contains only images from the normal class and anomalous examples are not available during the training. At test time, examples that differ from the normal training dataset are classified as anomalous.

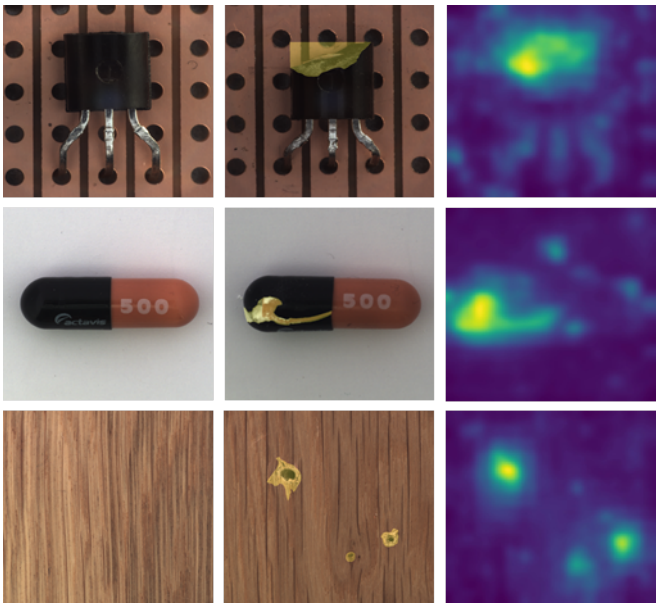


Fig. 1. Image samples from the MVTec AD [1]. *Left column*: normal images of Transistor, Capsule and Wood classes. *Middle column*: images of the same classes with the ground truth anomalies highlighted in yellow. *Right column*: anomaly heatmaps obtained by our PaDiM model. Yellow areas correspond to the detected anomalies, whereas the blue areas indicate the normality zones. Best viewed in color.

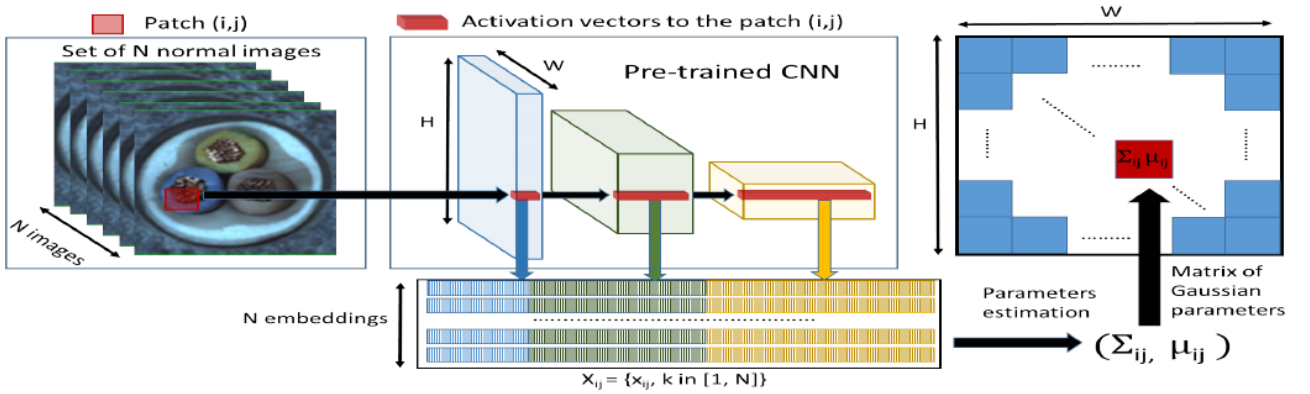
Recently, several methods have been proposed to combine anomaly localization and detection tasks in a one-class learning setting [2]–[5]. However, either they require deep neural network training [3], [6] which might be cumbersome, or they use a K-nearest-neighbor (K-NN) algorithm [7] on the entire training dataset at test time [4], [5]. The linear complexity of the KNN algorithm increases the time and space complexity as the size of the training dataset grows. These two scalability issues may hinder the deployment of anomaly localization algorithms in industrial context.

To mitigate the aforementioned issues, we propose a new anomaly detection and localization approach, named PaDiM for Patch Distribution Modeling. It makes use of a pretrained convolutional neural network (CNN) for embedding extraction and has the two following properties:

- Each patch position is described by a multivariate Gaussian distribution;
- PaDiM takes into account the correlations between dif-

図2. 最も大きなCNN特徴マップ内の位置（i, j）に対応する各画像パッチに対し、PaDiMはNつのトレーニング画像と3つの異なる

ameters (μ_{ij}, Σ_{ij}) from the pretrained CNN layers.



事前学習済みCNNの異なるセマンティックレベル。

この新しい効率的なアプローチにより、PaDiMはMVTec AD [1] とShanghaiTech Campus (STC) [8] データセットにおける異常検出と局所化において、既存の最先端手法を凌駕します。さらに、テスト時における時間と空間の複雑さはデータセットのトレーニングサイズに依存せず、産業応用における利点となります。私たちは評価プロトコルを拡張し、より現実的な条件下でのモデル性能を評価するため、非一致データセット上で評価を実施しました。

II. Related Work

異常検出と局所化手法は、再構築ベースまたは埋め込み類似性ベースの手法に分類されます。

再構築ベースの手法は、異常検出と局所化に広く使用されています。オートエンコーダー（AE）[1]、[9]-[11]、変分オートエンコーダー（VAE）[3]、[12]-[14]、または生成対抗ネットワーク（GAN）[15]-[17]などのニューラルネットワークアーキテクチャは、正常なトレーニング画像のみを再構築するように訓練されます。したがって、異常な画像は適切に再構築されないため検出可能です。画像レベルでは、再構築誤差を異常スコアとして使用する最も単純なアプローチ[10]がありますが、潜在空間[16]、[18]、中間活性化[19]、またはディｃкриミネーター[17]、[20]からの追加情報により、異常な画像をより正確に認識できます。異常を局所化するには、再構築ベースの手法はピクセル単位の再構築誤差を異常スコアとして使用できます[1]または構造的類似性[9]。または、異常マップは潜在空間から生成された視覚的注意マップである可能性があります[3]、[14]。再構築ベースの手法は直感的で解釈可能ですが、AEが異常画像に対しても良い再構築結果を生成する可能性があるため、その性能は制限されます[21]。

埋め込み類似性ベースの手法は、異常検出[6]、[22]-[24]では画像全体を記述する意味のあるベクトルを抽出するために深層神経ネットワークを使用し、異常局所化[2]、[4]、[5]、[25]では画像パッチを記述するために使用します。しかし、異常検出のみを行う埋め込み類似性に基づく手法は有望な結果を示すものの、異常画像のどの部分が異常スコアの高さに寄与しているかを特定できないため、解釈可能性に欠ける場合があります。この場合の異常スコアは、テスト画像の埋め込みベクトルとトレーニングデータセットから正常性を表す参照ベクトルとの距離です。正常な参照は、正常な画像の埋め込みを含むn次元の球の中心[4]，[22]、ガウス分布のパラメーター[23]，[26]、または正常な埋め込みベクトルの全体集合[5]，[24]などです。最後のオプションは、異常局所化において最も優れた結果を報告しているSPADE [5] で使用されています。しかし、テスト時に正常な埋め込みベクトルのセットに対してK-NNアルゴリズムを実行するため、推論の複雑さはトレーニングデータセットのサイズに線形にスケールします。これは、この手法の産業展開を妨げる可能性があります。

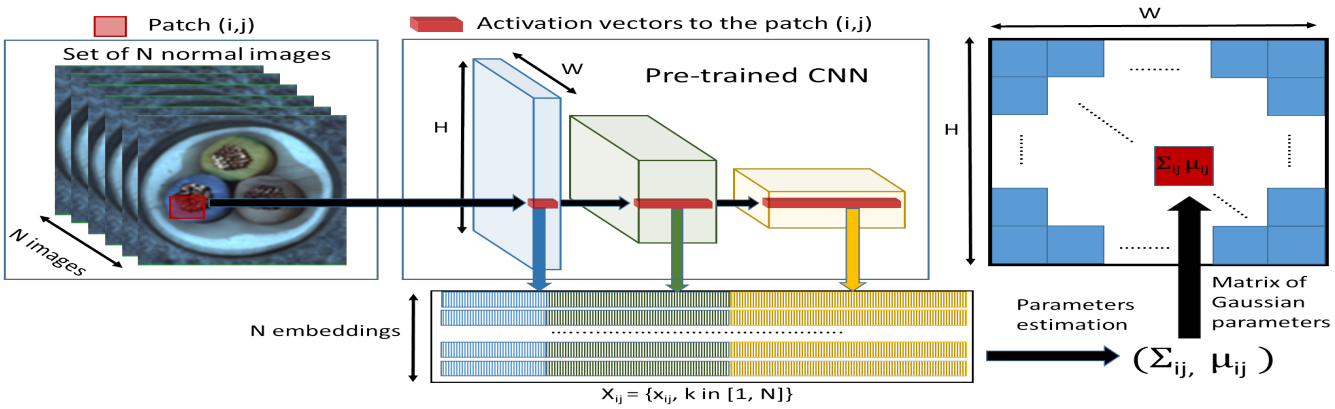
当社の手法であるPaDiMは、異常検出のためのパッチ埋め込みを生成する点で、前述の手法と類似しています。ただし、PaDiMにおける正常クラスは、使用される事前学習済みCNNモデルのセマンティックレベル間の相関関係をモデル化するガウス分布の集合を通じて記述されます。[5]、[23]にインスパイアされ、事前学習済みネットワークとしてResNet [27]、Wide-ResNet [28]、またはEfficientNet [29]を選択しています。このモデル化により、PaDiMは現在の最先端手法を凌駕しています。さらに、その時間複雑度は低く、予測段階ではトレーニングデータセットのサイズに依存しません。

III. パッチ分布モデリング

A. 埋め込み抽出

事前学習済みCNNは異常検出に適切な特徴を出力可能です[24]。そのため、事前学習済みCNNのみを使用してパッチ埋め込みベクトルを生成し、複雑なニューラルネットワーク最適化を回避しました。PaDiMのパッチ埋め込みプロセスはSPADE [5] のもの類似しており、図2に示されています。トレーニングフェーズにおいて、正常画像の各パッチは、事前学習済みCNNの活性化マップにおける空間的に対応する活性化ベクトルと関連付けられます。異なる層の活性化ベクトルを結合することで、異なるセマンティックレベルと解像度からの情報を保持する埋め込みベクトルを生成し、細粒度とグローバルな文脈をエンコードします。活性化マップはより低い

Fig. 2. For each image patch corresponding to position (i, j) in the largest CNN feature map, PaDiM learns the Gaussian parameters (μ_{ij}, Σ_{ij}) from the set of N training embedding vectors $X_{ij} = \{x_{ij}^k, k \in [\![1, N]\!]\}$, computed from N different training images and three different pretrained CNN layers.



ferent semantic levels of a pretrained CNN.

With this new and efficient approach, PaDiM outperforms the existing state-of-the-art methods for anomaly localization and detection on the MVTec AD [1] and the ShanghaiTech Campus (STC) [8] datasets. Besides, at test time, it has a low time and space complexity, independent of the dataset training size which is an asset for industrial applications. We also extend the evaluation protocol to assess model performance in more realistic conditions, *i.e.*, on a non-aligned dataset.

II. Related work

Anomaly detection and localization methods can be categorized as either reconstruction-based or embedding similarity-based methods.

Reconstruction-based methods are widely-used for anomaly detection and localization. Neural network architectures like autoencoders (AE) [1], [9]–[11], variational autoencoders (VAE) [3], [12]–[14] or generative adversarial networks (GAN) [15]–[17] are trained to reconstruct normal training images only. Therefore, anomalous images can be spotted as they are not well reconstructed. At the image level, the simplest approach is to take the reconstructed error as an anomaly score [10] but additional information from the latent space [16], [18], intermediate activations [19] or a discriminator [17], [20] can help to better recognize anomalous images. Yet to localize anomalies, reconstruction-based methods can take the pixel-wise reconstruction error as the anomaly score [1] or the structural similarity [9]. Alternatively, the anomaly map can be a visual attention map generated from the latent space [3], [14]. Although reconstruction-based methods are very intuitive and interpretable, their performance is limited by the fact that AE can sometimes yield good reconstruction results for anomalous images too [21].

Embedding similarity-based methods use deep neural networks to extract meaningful vectors describing an entire image for anomaly detection [6], [22]–[24] or an image patch for anomaly localization [2], [4], [5], [25]. Still, embedding similarity-based methods that only perform anomaly detection give promising results but often lack interpretability as it is not possible to know which part of an anomalous images is responsible for a high anomaly score. The anomaly score is in this case the distance between embedding vectors of a test image and reference vectors representing normality from the training dataset. The normal reference can be the center of a n-sphere containing embeddings from normal images [4], [22], parameters of Gaussian distributions [23], [26] or the entire set of normal embedding vectors [5], [24]. The last option is used by SPADE [5] which has the best reported results for anomaly localization. However, it runs a K-NN algorithm on a set of normal embedding vectors at test time, so the inference complexity scales linearly to the dataset training size. This may hinder industrial deployment of the method.

Our method, PaDiM, generates patch embeddings for anomaly localization, similar to the aforementioned approaches. However, the normal class in PaDiM is described through a set of Gaussian distributions that also model correlations between semantic levels of the used pretrained CNN model. Inspired by [5], [23], we choose as pretrained networks a ResNet [27], a Wide-ResNet [28] or an EfficientNet [29]. Thanks to this modelisation, PaDiM outperforms the current state-of-the-art methods. Moreover, its time complexity is low and independent of the training dataset size at the prediction stage.

III. Patch Distribution Modeling

A. Embedding extraction

Pretrained CNNs are able to output relevant features for anomaly detection [24]. Therefore, we choose to avoid ponderous neural network optimization by only using a pretrained CNN to generate patch embedding vectors. The patch embedding process in PaDiM is similar to one from SPADE [5] and illustrated in Figure 2. During the training phase, each patch of the normal images is associated to its spatially corresponding activation vectors in the pretrained CNN activation maps. Activation vectors from different layers are then concatenated to get embedding vectors carrying information from different semantic levels and resolutions, in order to encode fine-grained and global contexts. As activation maps have a lower

入力画像の解像度よりも高い解像度では、多くのピクセルが同じ埋め込みベクトルを持ち、元の画像解像度では重なり合わないピクセルパッチを形成します。したがって、入力画像は （i, j） ∈ [1, W] × [1, H] のグリッドに分割され、ここで W × H は埋め込みを生成するために使用される最大の活性化マップの解像度です。最後に、このグリッド内の各パッチ位置 （i, j） は、上記で説明したように計算された埋め込みベクトル x_{ij} と関連付けられます。

生成されたパッチ埋め込みベクトルには冗長な情報が含まれている可能性があるため、そのサイズを削減する可能性を実験的に検討しました（セクションV-A）。ランダムにいくつかの次元を選択する方法は、古典的な主成分分析（PCA）アルゴリズム [30] よりも効率的であることが判明しました。この単純なランダム次元削減は、トレーニング時間とテスト時間の両方でモデルの複雑さを大幅に削減しつつ、最先端の性能を維持します。最後に、テスト画像のパッチ埋め込みベクトルは、次節で説明する正常クラスの学習済みパラメトリック表現の助けを借りて、異常マップを出力するために使用されます。

B. 正常性の学習

位置 （i, j） における正常画像の特徴を学習するため、まず図2に示すように、N枚の正常トレーニング画像から位置 （i, j） におけるパッチ埋め込みベクトルの集合 X_{ij} = {x_{kij} , k ∈ [[1, N]] } を計算します。この集合が持つ情報を要約するため、X_{ij} が多変量ガウス分布 N （μ_{ij} , Σ_{ij} ） によって生成されたと仮定します。ここで、μ_{ij} は X_{ij} のサンプル平均であり、サンプル共分散 Σ_{ij} は次のように推定されます：

$$\Sigma_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x_{ij}^k} - \mu_{\mathbf{ij}})(\mathbf{x_{ij}^k} - \mu_{\mathbf{ij}})^{\mathrm{T}} + \epsilon I \qquad (1)$$

正則化項Iは、サンプル共分散行列Σ_{ij} をフルランクかつ逆行列可能な状態にします。最後に、各可能なパッチ位置は、図2に示すガウスパラメータ行列によって表される多変量ガウス分布と関連付けられます。

当社のパッチ埋め込みベクトルは、異なるセマンティックレベルからの情報を保持しています。したがって、推定された多変量ガウス分布 N （μ_{ij} , Σ_{ij} ） も異なるレベルからの情報を捕捉し、Σ_{ij} にはレベル間の相関が含まれます。実験的に示したように（セクション V-A）、事前訓練されたCNNの異なるセマンティックレベル間の関係をモデル化することは、異常局所化性能の向上に役立ちます。

C. 推論：異常マップの計算

[23]、[26] に倣い、テスト画像の（i, j）位置のパッチに異常スコアを付与するために、マハラノビス距離 [31] M（x_{ij} ）を使用します。M （x_{ij} ） は、テストパッチの埋め込み x_{ij} と学習された分布 N （μ_{ij} , Σ_{ij} ） との間の距離と解釈できます。M （x_{ij} ） は次のように計算されます：

$$M(x_{ij}) = \sqrt{(x_{ij} - \mu_{ij})^T \Sigma_{ij}^{-1} (x_{ij} - \mu_{ij})} \qquad (2)$$

したがって、マハラノビス距離の行列 M = （M （x_{ij} ） ）_{1<i<W,1<j<H} は異常マップを形成し、計算可能です。このマップでの高スコアは異常領域を示します。画像全体の最終的な異常スコアは、異常マップ M の最大値です。最後に、テスト時において、当手法は K-NN ベースの手法 [4] - [6] , [25] のようなスケーラビリティの問題を抱えていません。なぜなら、パッチの異常スコアを計算するために大量の距離値を計算して並べ替える必要がないからです。

IV. Experiments

A. データセットとメトリクス

メトリクス。局在化性能を評価するため、2つの閾値に依存しないメトリクスを計算します。受信者動作特性曲線（ROC曲線）下の面積（AUROC）を使用し、真陽性率は異常と正しく分類されたピクセルの割合です。AUROCは大きな異常値に偏るため、地域重なりスコア（PRO-score） [2] も採用しています。これは、各接続成分に対して、偽陽性率0から0.3の範囲で正しく分類されたピクセル率の平均値をプロットした曲線から、正規化された積分値をPRO-scoreとします。高いPROスコアは、大規模な異常と小規模な異常の両方が適切に局所化されていることを意味します。

データセット。まず、産業用品質管理における異常検出アルゴリズムの評価を目的とした MVTec AD [1] データセットで、単一クラス学習設定においてモデルを評価します。このデータセットには、約 240 枚の画像からなる 15 のクラスが含まれています。元の画像解像度は700x700から1024x1024です。10のオブジェクトクラスと5のテクスチャクラスが存在します。オブジェクトはデータセット全体で常に中央に配置され、同じ方向に整列されています。これは図1のTransistorとCapsuleクラスで確認できます。元のデータセットに加え、異常検出モデルの性能をより現実的な文脈で評価するため、MVTec ADの改変版であるRdMVTec ADを作成しました。この改変版では、トレーニングセットとテストセットの両方にランダムな回転（-10~+10度）とランダムなクロップ（256×256から224×224まで）を適用しています。このMVTec ADの改変版は、品質管理における異常局在化の実際の使用ケースをより適切に表現する可能性があります。特に、関心対象のオブジェクトが画像内で常に中心に配置されず、整列していない場合です。

さらに評価するため、当社はPaDiMを静止カメラからのビデオ監視を模擬するShanghai Tech Campus （STC） データセット [8] でテストしました。このデータセットには、13のシーンに分割された274,515のトレーニングフレームと42,883のテストフレームが含まれています。元の画像解像度は856×480です。トレーニング動画は通常のシーケンスで構成され、テスト動画には歩行者区域での車両の出現や人々の喧嘩などの異常が含まれます。

B. 実験設定

PaDiMは、ImageNet [32] で事前学習された異なるバックボーン（ResNet18（R18） [27] 、Wide ResNet-50-2（WR50） [28] 、EfficientNet-B5 [29] ）で訓練されます。 [5] と同様に、バックボーンがResNetの場合、パッチ埋め込みベクトルは最初の3層から抽出され、

resolution than the input image, many pixels have the same embeddings and then form pixel patches with no overlap in the original image resolution. Hence, an input image can be divided in a grid of $(i, j) \in [1, W] \times [1, H]$ positions where WxH is the resolution of the largest activation map used to generate embeddings. Finally, each patch position (i, j) in this grid is associated to an embedding vector x_{ij} computed as described above.

The generated patch embedding vectors may carry redundant information, therefore we experimentally study the possibility to reduce their size (Section V-A). We noticed that randomly selecting few dimensions is more efficient that a classic Principal Component Analysis (PCA) algorithm [30]. This simple random dimensionality reduction significantly decreases the complexity of our model for both training and testing time while maintaining the state-of-the-art performance. Finally, patch embedding vectors from test images are used to output an anomaly map with the help of the learned parametric representation of the normal class described in the next subsection.

B. Learning of the normality

To learn the normal image characteristics at position (i, j), we first compute the set of patch embedding vectors at (i, j), $X_{ij} = \{x_{ij}^k, k \in [\![1, N]\!]\}$ from the N normal training images as shown on Figure 2. To sum up the information carried by this set we make the assumption that X_{ij} is generated by a multivariate Gaussian distribution $\mathcal{N}(\mu_{ij}, \boldsymbol{\Sigma_{ij}})$ where μ_{ij} is the sample mean of X_{ij} and the sample covariance Σ_{ij} is estimated as follows :

$$\Sigma_{ij} = \frac{1}{N-1} \sum_{k=1}^{N} (\mathbf{x_{ij}^k} - \mu_{\mathbf{ij}})(\mathbf{x_{ij}^k} - \mu_{\mathbf{ij}})^{\mathrm{T}} + \boldsymbol{\epsilon} I \qquad (1)$$

where the regularisation term $\boldsymbol{\epsilon} I$ makes the sample covariance matrix Σ_{ij} full rank and invertible. Finally, each possible patch position is associated with a multivariate Gaussian distribution as shown in Figure 2 by the matrix of Gaussian parameters.

Our patch embedding vectors carry information from different semantic levels. Hence, each estimated multivariate Gaussian distribution $\mathcal{N}(\mu_{ij}, \boldsymbol{\Sigma_{ij}})$ captures information from different levels too and $\boldsymbol{\Sigma_{ij}}$ contains the inter-level correlations. We experimentally show (Section V-A) that modeling these relationships between the different semantic levels of the pretrained CNN helps to increase anomaly localization performance.

C. Inference : computation of the anomaly map

Inspired by [23], [26], we use the Mahalanobis distance [31] $M(x_{ij})$ to give an anomaly score to the patch in position (i, j) of a test image. $M(x_{ij})$ can be interpreted as the distance between the test patch embedding x_{ij} and learned distribution $\mathcal{N}(\mu_{ij}, \boldsymbol{\Sigma_{ij}})$, where $M(x_{ij})$ is computed as follows:

$$M(x_{ij}) = \sqrt{(x_{ij} - \mu_{ij})^T \Sigma_{ij}^{-1} (x_{ij} - \mu_{ij})} \qquad (2)$$

Hence, the matrix of Mahalanobis distances $M = (M(x_{ij}))_{1<i<W, 1<j<H}$ that forms an anomaly map can be computed. High scores in this map indicate the anomalous areas. The final anomaly score of the entire image is the maximum of anomaly map M. Finally, at test time, our method does not have the scalability issue of the K-NN based methods [4]–[6], [25] as we do not have to compute and sort a large amount of distance values to get the anomaly score of a patch.

IV. Experiments

A. Datasets and metrics

Metrics. To assess the localization performance we compute two threshold independent metrics. We use the Area Under the Receiver Operating Characteristic curve (AUROC) where the true positive rate is the percentage of pixels correctly classified as anomalous. Since the AUROC is biased in favor of large anomalies we also employ the per-region-overlap score (PRO-score) [2]. It consists in plotting, for each connected component, a curve of the mean values of the correctly classified pixel rates as a function of the false positive rate between 0 and 0.3. The PRO-score is the normalized integral of this curve. A high PRO-score means that both large and small anomalies are well-localized.

Datasets. We first evaluate our models on the MVTec AD [1] designed to test anomaly localization algorithms for industrial quality control and in a one-class learning setting. It contains 15 classes of approximately 240 images. The original image resolution is between 700x700 and 1024x1024. There are 10 object and 5 texture classes. Objects are always well-centered and aligned in the same way across the dataset as we can see in Figure 1 for classes Transistor and Capsule. In addition to the original dataset, to assess performance of anomaly localization models in a more realistic context, we create a modified version of the MVTec AD, referred as Rd-MVTec AD, where we apply random rotation (-10, +10) and random crop (from 256x256 to 224x224) to both the train and test sets. This modified version of the MVTec AD may better describe real use cases of anomaly localization for quality control where objects of interest are not always centered and aligned in the image.

For further evaluation, we also test PaDiM on the Shanghai Tech Campus (STC) Dataset [8] that simulates video surveillance from a static camera. It contains 274 515 training and 42 883 testing frames divided in 13 scenes. The original image resolution is 856x480. The training videos are composed of normal sequences and test videos have anomalies like the presence of vehicles in pedestrian areas or people fighting.

B. Experimental setups

We train PaDiM with different backbones, a ResNet18 (R18) [27], a Wide ResNet-50-2 (WR50) [28] and an EfficientNet-B5 [29], all pretrained on ImageNet [32]. Like in [5], patch embedding vectors are extracted from the first three layers when the backbone is a ResNet, in order to combine

異なるセマンティックレベルからの情報を保持しつつ、局所化タスクに適した十分な解像度を維持します。このアイデアに基づき、EfficientNet-B5を使用する場合、層7（レベル2）、層20（レベル4）、層26（レベル5）からパッチ埋め込みベクトルを抽出します。また、ランダム次元削減（Rd）を適用します（セクションIII-AおよびV-A参照）。私たちのモデル名は、バックボーンと次元削減方法（使用する場合）を示しています。例えば、PaDiM-R18-Rd100は、ResNet18バックボーンを使用し、パッチ埋め込みベクトルに100のランダムに選択された次元を使用するPaDiMモデルです。デフォルトでは、式1から=0.01を使用します。

オリジナル論文で説明されているように、SPADE [5] モデルを再現し、バックボーンとしてWide ResNet-50-2 (WR50) [28] を使用します。SPADEとPaDiMには、[5] で使用されたのと同じ前処理を適用します。MVTec ADの画像を256x256にリサイズし、224x224に中央切り出しを行います。STCの画像には256x256のリサイズのみを適用します。画像と局所化マップはバイキュービック補間を使用してリサイズし、異常マップには [5] と同様にパラメーター $\sigma = 4$ のガウスフィルターを適用します。

私たちは、エンコーダーとしてResNet18を使用し、8×8の畳み込み潜在変数を持つ再構築ベースのベースラインとして独自のVAEを実装しました。各MVTec ADクラスに対して10,000枚の画像を使用し、以下のデータ拡張操作を実施してトレーニングを行います：ランダム回転（-2^{◦}、+2^{◦}）、292×292へのリサイズ、ランダムクロップで282×282に切り出し、最後にセンタークロップで256×256に切り出し。トレーニングは、Adamオプティマイザー [12] を使用し、初期学習率10^{-4}、バッチサイズ32画像で100エポック実施されました。局所化に対応する異常マップは、再構築のピクセル単位のL2誤差に対応します。

V. Results

A. アブレーション研究

まず、PaDiMにおけるセマンティックレベル間の相関をモデル化することの影響を評価し、次元削減を通じて手法を簡素化する可能性を探ります。

層間相関。ガウスモデルとマハラノビス距離の組み合わせは、以前の研究で敵対的攻撃の検出 [26] や画像レベルでの異常検出 [23] に既に採用されています。しかし、これらの方法はPaDiMで実施しているように、異なるCNNのセマンティックレベル間の相関をモデル化していません。表Iでは、ResNet18バックボーンを使用したPaDiMのMVTec ADにおける異常局所化性能を示しています。最初の3層（層1、層2、または層3）のいずれか1層のみを使用する場合と、これらの3モデルの出力合計をアンサンブル手法として用いた場合（最初の3層を考慮するが、層間の相関は考慮しない：層1+2+3）を比較しています。表Iの最終行（PaDiM-R18）は、提案するPaDiMのバージョンで、各パッチの位置は最初の3つのResNet18層とそれらの相関を考慮した1つのガウス分布で記述されます。3つの層のうち、Layer 3を使用した場合がAUROCにおいて最も良い結果を示すことが観察されます。これは、レイヤー3がより高いセマンティックレベルの情報を含み、正常性をより適切に記述するためです。

TABLE I
STUDY OF THE ANOMALY LOCALIZATION PERFORMANCE USING DIFFERENT SEMANTIC-LEVEL CNN LAYERS. RESULTS ARE DISPLAYED AS TUPLES (AUROC%, PRO-SCORE%) ON THE MVTEC AD.

Layer used	all texture classes	all object classes	all classes
Layer 1	(93.1, 87.1)	(95.6, 86.5)	(94.8, 86.8)
Layer 2	(95.0, 89.7)	(96.1, 87.9)	(95.7, 88.5)
Layer 3	(94.8, 89.6)	(97.1, 87.7)	(95.7, 88.3)
Layer 1+2+3	(95.4, 90.7)	(96.3, 88.1)	(96.0, 89.0)
PaDiM-R18	(**96.3**, **92.3**)	(**97.5**, **90.1**)	(**97.1**, **90.8**)

ただし、レイヤー3のPROスコアはレイヤー2よりもやや劣っており、これはレイヤー2の解像度が低く、異常局在化の精度に影響を与えるためです。表Iの最後の2行で示されるように、異なるレイヤーからの情報を集約することで、高いセマンティック情報と高い解像度とのトレードオフ問題を解決できます。モデルLayer 1+2+3が単純に出力を加算するのに対し、当社のモデルPaDiMR18は意味論的レベル間の相関を考慮します。その結果、AUROCで1.1ポイント、PROスコアで1.8ポイント、Layer 1+2+3を上回ります。これは、意味論的レベル間の相関をモデル化することの重要性を確認しています。

TABLE II
STUDY OF THE ANOMALY LOCALIZATION PERFORMANCE WITH A DIMENSIONALITY REDUCTION FROM 448 TO 100 AND 200 USING PCA OR RANDOM FEATURE SELECTION (RD). RESULTS ARE DISPLAYED AS TUPLES (AUROC%, PRO-SCORE%) ON THE MVTEC AD.

	all texture classes	all object classes	all classes
Rd 100	(95.7, 91.3)	(97.2, 89.4)	(96.7, 90.5)
PCA 100	(93.7, 88.9)	(93.5, 84.1)	(93.5, 85.7)
Rd 200	(96.1, 92.0)	(97.5, 89.8)	(97.0, 90.5)
PCA 200	(95.1, 91.8)	(96.0, 88.1)	(95.7, 89.3)
all (448)	(**96.3**, **92.3**)	(**97.5**, **90.1**)	(**97.1**, **90.8**)

次元削減。PaDiM-R18は、各448次元のパッチ埋め込みベクトルの集合から多変量ガウス分布を推定します。埋め込みベクトルのサイズを縮小することで、モデルの計算とメモリの複雑さを軽減できます。私たちは2つの異なる次元削減方法を検討しました。最初の方法は、主成分分析（PCA）アルゴリズムを適用してベクトルサイズを100または200次元へ削減するものです。2つ目の方法は、トレーニング前にランダムに特徴を選択するランダム特徴選択です。この場合、10つの異なるモデルをトレーニングし、平均スコアを算出します。ただし、ランダム性により異なるシード間で結果が変化することはありません。平均AUROCの標準誤差平均（SEM）は常に10^{-4}から10^{-7}の間です。

表IIから、同じ次元数において、ランダム次元削減（Rd）はMVTec ADのすべてのクラスでPCAをAUROCで少なくとも1.3ポイント、PROスコアで1.2ポイント上回ることがわかります。これは、PCAが最も分散の大きい次元を選択するため、正常クラスと異常クラスを区別するのに役立つ次元ではない可能性があるためです [23]。

information from different semantic levels, while keeping a high enough resolution for the localization task. Following this idea, we extract patch embedding vectors from layers 7 (level 2), 20 (level 4), and 26 (level 5), if an EfficientNet-B5 is used. We also apply a random dimensionality reduction (Rd) (see Sections III-A and V-A). Our model names indicate the backbone and the dimensionality reduction method used, if any. For example, PaDiM-R18-Rd100 is a PaDiM model with a ResNet18 backbone using 100 randomly selected dimensions for the patch embedding vectors. By default we use $\epsilon = 0.01$ for the ϵ from Equation 1.

We reproduce the model SPADE [5] as described in the original publication with a Wide ResNet-50-2 (WR50) [28] as backbone. For SPADE and PaDiM we apply the same prepocessing as in [5]. We resize the images from the MVTec AD to 256x256 and center crop them to 224x224. For the images from the STC we use a 256x256 resize only. We resize the images and the localization maps using bicubic interpolation and we use a Gaussian filter on the anomaly maps with parameter $\sigma = 4$ like in [5].

We also implement our own VAE as a reconstruction-based baseline implemented with a ResNet18 as encoder and a 8x8 convolutional latent variable. It is trained on each MVTec AD class with 10 000 images using the following data augmentations operations: random rotation ($-2°$, $+2°$), 292x292 resize, random crop to 282x282, and finally center crop to 256x256. The training is performed during 100 epochs with the Adam optimizer [12] with an initial learning rate of 10^{-4} and a batch size of 32 images. The anomaly map for the localization corresponds to the pixel-wise L2 error for reconstruction.

V. Results

A. Ablative studies

First, we evaluate the impact of modeling correlations between semantic levels in PaDiM and explore the possibility to simplify our method through dimensionality reduction.

Inter-layer correlation. The combination of Gaussian modeling and the Mahalanobis distance has already been employed in previous works to detect adversarial attacks [26] and for anomaly detection [23] at the image level. However those methods do not model correlations between different CNN's semantic levels as we do in PaDiM. In Table I we show the anomaly localization performance on the MVTec AD of PaDiM with a ResNet18 backbone when using only one of the first three layers (Layer 1, Layer 2, or Layer 3) and when summing the outputs of these 3 models to form an ensemble method that takes into account the first three layers but not the correlations between them (Layers 1+2+3). The last row of Table I (PaDiM-R18) is our proposed version of PaDiM where each patch location is described by one Gaussian distribution taking into account the first three ResNet18 layers and correlations between them. It can be observed that using Layer 3 produces the best results in terms of AUROC among the three layers. It is due to the fact that Layer 3 carries higher semantic level information which helps to better describe normality. However, Layer 3 has a slightly worse PRO-score than Layer 2 that can be explained by the lower resolution of Layer 2 which affects the accuracy of anomaly localization. As we see in the two last rows of Table I, aggregating information from different layers can solve the trade-off issue between high semantic information and high resolution. Unlike model Layer 1+2+3 that simply sums the outputs, our model PaDiM-R18 takes into account correlations between semantic levels. As a result, it outperforms Layer 1+2+3 by 1.1p.p (percent point) for AUROC and 1.8p.p for PRO-score. It confirms the relevance of modeling correlation between semantic levels.

TABLE I
Study of the anomaly localization performance using different semantic-level CNN layers. Results are displayed as tuples (AUROC%, PRO-score%) on the MVTec AD.

Layer used	all texture classes	all object classes	all classes
Layer 1	(93.1, 87.1)	(95.6, 86.5)	(94.8, 86.8)
Layer 2	(95.0, 89.7)	(96.1, 87.9)	(95.7, 88.5)
Layer 3	(94.8, 89.6)	(97.1, 87.7)	(95.7, 88.3)
Layer 1+2+3	(95.4, 90.7)	(96.3, 88.1)	(96.0, 89.0)
PaDiM-R18	(**96.3**, **92.3**)	(**97.5**, **90.1**)	(**97.1**, **90.8**)

TABLE II
Study of the anomaly localization performance with a dimensionality reduction from 448 to 100 and 200 using PCA or random feature selection (Rd). Results are displayed as tuples (AUROC%, PRO-score%) on the MVTec AD.

	all texture classes	all object classes	all classes
Rd 100	(95.7, 91.3)	(97.2, 89.4)	(96.7, 90.5)
PCA 100	(93.7, 88.9)	(93.5, 84.1)	(93.5, 85.7)
Rd 200	(96.1, 92.0)	(97.5, 89.8)	(97.0, 90.5)
PCA 200	(95.1, 91.8)	(96.0, 88.1)	(95.7, 89.3)
all (448)	(**96.3**, **92.3**)	(**97.5**, **90.1**)	(**97.1**, **90.8**)

Dimensionality reduction. PaDiM-R18 estimates multivariate Gaussian distributions from sets of patch embeddings vectors of 448 dimensions each. Decreasing the embedding vector size would reduce the computational and memory complexity of our model. We study two different dimensionality reduction methods. The first one consists in applying a Principal Component Analysis (PCA) algorithm to reduce the vector size to 100 or 200 dimensions. The second method is a random feature selection where we randomly select features before the training. In this case, we train 10 different models and take the average scores. Still the randomness does not change the results between different seeds as the standard error mean (SEM) for the average AUROC is always between 10^{-4} and 10^{-7}.

From Table II we can notice that for the same number of dimensions, the random dimensionality reduction (Rd) outperforms the PCA on all the MVTec AD classes by at least 1.3p.p in the AUROC and 1.2p.p in the PRO-score. It can be explained by the fact that PCA selects the dimensions with the highest variance which may not be the ones that help to discriminate the normal class from the anomalous one [23].

TABLE III
COMPARISON OF OUR PADIM MODELS WITH THE STATE-OF-THE-ART FOR THE ANOMALY LOCALIZATION ON THE MVTEC AD. RESULTS ARE DISPLAYED AS TUPLES (AUROC%, PRO-SCORE%)

Type	再構築ベースの方法			埋め込み類似性に基づく		methods	Our methods	
Model	AE simm [1], [2], [9]	AE L2 [1], [2]	VAE	Student [2]	Patch SVDD [4]	SPADE [5]	PaDiM-R18-Rd100	PaDiM-WR50-Rd550
Carpet	(87, 64.7)	(59, 45.6)	(59.7, 61.9)	(-, 69.5)	(92.6, -)	(97.5, 94.7)	(98.9, 96.0)	(**99.1**, **96.2**)
Grid	(94, 84.9)	(90, 58.2)	(61.2, 40.8)	(-, 81.9)	(96.2, -)	(93.7, 86.7)	(94.9, 90.9)	(**97.3**, **94.6**)
Leather	(78, 56.1)	(75, 81.9)	(67.1, 64.9)	(-, 81.9)	(97.4, -)	(97.6, 97.2)	(99.1, 97.9)	(**99.2**, **97.8**)
Tile	(59, 17.5)	(51, 89.7)	(51.3, 24.2)	(-, 91.2)	(91.4, -)	(87.4, 75.9)	(91.2, 81.6)	(**94.1**, **86.0**)
Wood	(73, 60.5)	(73, 72.7)	(66.6, 57.8)	(-, 72.5)	(90.8, -)	(88.5, 87.4)	(93.6, 90.3)	(**94.9**, **91.1**)
All texture classes	(78, 56.7)	(70, 69.6)	(61.2, 49.9)	(-, 79.4)	(93.7, -)	(92.9, 88.4)	(95.6, 91.3)	(**96.9**, **93.2**)
Bottle	(93, 83.4)	(86, 91.0)	(83.1, 70.5)	(-, 91.8)	(98.1, -)	(**98.4**, **95.5**)	(98.1, 93.9)	(98.3, 94.8)
Cable	(82, 47.8)	(86, 82.5)	(83.1, 77.9)	(-, 86.5)	(96.8, -)	(**97.2**, **90.9**)	(95.8, 86.2)	(96.7, 88.8)
Capsule	(94, 86.0)	(88, 86.2)	(81.7, 77.9)	(-, 91.6)	(95.8, -)	(**99.0** ,**93.7**)	(98.3, 91.9)	(98.5, 93.5)
Hazelnut	(97, 91.6)	(95, 91.7)	(87.7, 77.0)	(-, 93.7)	(97.5, -)	(**99.1**, **95.4**)	(97.7, 91.4)	(98.2, 92.6)
Metal Nut	(89, 60.3)	(86, 83.0)	(78.7, 57.6)	(-, 89.5)	(98.0, -)	(**98.1**, **94.4**)	(96.7, 81.9)	(97.2, 85.6)
Pill	(91, 83.0)	(85, 89.3)	(81.3, 79.3)	(-, 93.5)	(95.1, -)	(**96.5**, **94.6**)	(94.7, 90.6)	(95.7, 92.7)
Screw	(96, 88.7)	(96, 75.4)	(75.3, 66.4)	(-, 92.8)	(95.7, -)	(**98.9**, **96.0**)	(97.4, 91.3)	(98.5, 94.4)
Toothbrush	(92, 78.4)	(93, 82.2)	(91.9, 85.4)	(-, 86.3)	(98.1, -)	(97.9, **93.5**)	(98.7, 92.3)	(**98.8**, 93.1)
Transistor	(90, 72.5)	(86, 72.8)	(75.4, 61.0)	(-, 70.1)	(97.0, -)	(94.1, **87.4**)	(97.2, 80.2)	(**97.5**, 84.5)
Zipper	(88, 66.5)	(77, 83.9)	(71.6, 60.8)	(-, 93.3)	(95.1, -)	(96.5, 92.6)	(98.2, 94.7)	(**98.5**, **95.9**)
All object classes	(91, 75.8)	(88, 83.8)	(81.0, 71.4)	(-, 88.9)	(96.7, -)	(97.6, **93.4**)	(97.3, 89.4)	(**97.8**, 91.6)
All classes	(87, 69.4)	(82, 79.0)	(74.4, 64.2)	(-, 85.7)	(95.7, -)	(96.5, 91.7)	(96.7, 90.1)	(**97.5**, **92.1**)

表IIから、埋め込みベクトルの次元をランダムに100次元まで削減しても、異常局所化性能にほとんど影響を与えないことがわかります。AUROCは0.4ポイント、PROスコアは0.3ポイントそれぞれ低下するだけです。この単純ながら効果的な次元削減方法は、第V-D節で示すように、PaDiMの時間と空間の複雑さを大幅に削減します。

B. 最先端技術との比較

局所化手法を訓練し、性能を評価しました。表IIIでは、MVTec ADにおける異常局所化のAUROCとPROスコアの結果を示しています。公平な比較のため、SPADE [5] で使用されているバックボーンとしてWide ResNet-50-2 (WR50) を採用しました。他のベースラインがより小さなバックボーンを使用しているため、ResNet18 (R18) も試しました。PaDiMの埋め込みサイズをWR50とR18それぞれに対してランダムに550と100に削減しました。

まず、PaDiM-WR50-Rd550がすべてのクラスにおいてPROスコアとAUROCの平均で他のすべての手法を上回っていることがわかります。非常に軽量なモデルであるPaDiM-R18-Rd100も、MVTec ADのクラスにおいて平均AUROCで他のモデルを少なくとも0.2ポイント上回っています。PaDiMの性能をさらに分析すると、オブジェクトクラスにおける差は小さく、PaDiM-WR50-Rd550はAUROC (+0.2p.p) で最も優れていますが、SPADE [5] はPROスコア (+1.8p.p) で最も優れています。しかし、当社のモデルはテクスチャクラスにおいて特に正確です。PaDiM-WR50-Rd550は、テクスチャクラスにおいて平均でSPADE [5] をPROスコアで4.8p.p、AUROCで4.0p.p上回っています。実際、PaDiMはSPADE [5] やPatch-SVDD [4] とは異なり、通常のクラスに対して明示的な確率モデルを学習します。テクスチャ画像において特に効率的である理由は、オブジェクト画像のように整列や中心合わせがされていなくても、PaDiMが正常なトレーニングデータセット全体での統計的類似性を効果的に捕捉するためです。

さらに、当モデルをSTCデータセットで評価しました。当手法を、時系列情報を用いない異常検出で最も優れた2つの報告モデル、CAVGA-RU [3] とSPADE [5] と比較しました。表IVに示すように、STCデータセットにおける最良の結果 (AUROC) は、当モデルの最もシンプルなバージョンであるPaDiM-R18-Rd100が2.1ポイントの差で達成されています。実際、このデータセットでは画像内の歩行者の位置が非常に変動するため、セクションV-Cで示されるように、当社の方法は非一致データセットでも良好な性能を発揮します。

検出。当社のモデルが発行する異常マップの最大スコアを採用し (セクションIII-C参照)、画像全体に異常スコアを付与することで、画像レベルでの異常検出を実施します。PaDiMの異常検出性能を、SPADEで用いられるWide ResNet-50-2 (WR50) [28] とEfficientNet-B5 [29] で評価しました。表Vに示すように、当社のモデルPaDiM-WR50-Rd550は、最良のバックボーンとして報告されたEfficientNet-B4を用いるMahalanobisAD [23] を除くすべての方法よりも優れています。さらに、当社のPaDiM-EfficientNet-B5は、すべてのクラスにおいてAUROCで平均2.6ポイント以上、すべてのモデルを上回っています。また、異常検出における第2位の最良手法であるMahalanobisAD [23] とは対照的に、当社のモデルは画像内の異常領域をより正確に特徴付ける異常セグメンテーションも実行します。

C. 非一致データセットにおける異常局所化

異常局所化手法の頑健性を評価するため、PaDiMと複数の

TABLE IV
C異常局在化におけるSTCでのAUROC%における当社のPaDiMモデルと最先端手法の比較。

Model	CAVGA-RU [3]	SPADE [5]	PaDiM-R18-Rd100
AUROC score%	85	89.9	**91.2**

TABLE III
COMPARISON OF OUR PADIM MODELS WITH THE STATE-OF-THE-ART FOR THE ANOMALY LOCALIZATION ON THE MVTEC AD. RESULTS ARE DISPLAYED AS TUPLES (AUROC%, PRO-SCORE%)

Type	Reconstruction-based methods			Embedding similarity based methods			Our methods	
Model	AE simm [1], [2], [9]	AE L2 [1], [2]	VAE	Student [2]	Patch SVDD [4]	SPADE [5]	PaDiM-R18-Rd100	PaDiM-WR50-Rd550
Carpet	(87, 64.7)	(59, 45.6)	(59.7, 61.9)	(-, 69.5)	(92.6, -)	(97.5, 94.7)	(98.9, 96.0)	(**99.1**, **96.2**)
Grid	(94, 84.9)	(90, 58.2)	(61.2, 40.8)	(-, 81.9)	(96.2, -)	(93.7, 86.7)	(94.9, 90.9)	(**97.3**, **94.6**)
Leather	(78, 56.1)	(75, 81.9)	(67.1, 64.9)	(-, 81.9)	(97.4, -)	(97.6, 97.2)	(99.1, 97.9)	(**99.2**, **97.8**)
Tile	(59, 17.5)	(51, 89.7)	(51.3, 24.2)	(-, 91.2)	(91.4, -)	(87.4, 75.9)	(91.2, 81.6)	(**94.1**, **86.0**)
Wood	(73, 60.5)	(73, 72.7)	(66.6, 57.8)	(-, 72.5)	(90.8, -)	(88.5, 87.4)	(93.6, 90.3)	(**94.9**, **91.1**)
All texture classes	(78, 56.7)	(70, 69.6)	(61.2, 49.9)	(-, 79.4)	(93.7, -)	(92.9, 88.4)	(95.6, 91.3)	(**96.9**, **93.2**)
Bottle	(93, 83.4)	(86, 91.0)	(83.1, 70.5)	(-, 91.8)	(98.1, -)	(**98.4**, **95.5**)	(98.1, 93.9)	(98.3, 94.8)
Cable	(82, 47.8)	(86, 82.5)	(83.1, 77.9)	(-, 86.5)	(96.8, -)	(**97.2**, **90.9**)	(95.8, 86.2)	(96.7, 88.8)
Capsule	(94, 86.0)	(88, 86.2)	(81.7, 77.9)	(-, 91.6)	(95.8, -)	(**99.0** ,**93.7**)	(98.3, 91.9)	(98.5, 93.5)
Hazelnut	(97, 91.6)	(95, 91.7)	(87.7, 77.0)	(-, 93.7)	(97.5, -)	(**99.1**, **95.4**)	(97.7, 91.4)	(98.2, 92.6)
Metal Nut	(89, 60.3)	(86, 83.0)	(78.7, 57.6)	(-, 89.5)	(98.0, -)	(**98.1**, **94.4**)	(96.7, 81.9)	(97.2, 85.6)
Pill	(91, 83.0)	(85, 89.3)	(81.3, 79.3)	(-, 93.5)	(95.1, -)	(**96.5**, **94.6**)	(94.7, 90.6)	(95.7, 92.7)
Screw	(96, 88.7)	(96, 75.4)	(75.3, 66.4)	(-, 92.8)	(95.7, -)	(**98.9**, **96.0**)	(97.4, 91.3)	(98.5, 94.4)
Toothbrush	(92, 78.4)	(93, 82.2)	(91.9, 85.4)	(-, 86.3)	(98.1, -)	(97.9, **93.5**)	(98.7, 92.3)	(**98.8**, 93.1)
Transistor	(90, 72.5)	(86, 72.8)	(75.4, 61.0)	(-, 70.1)	(97.0, -)	(94.1, **87.4**)	(97.2, 80.2)	(**97.5**, 84.5)
Zipper	(88, 66.5)	(77, 83.9)	(71.6, 60.8)	(-, 93.3)	(95.1, -)	(96.5, 92.6)	(98.2, 94.7)	(**98.5**, **95.9**)
All object classes	(91, 75.8)	(88, 83.8)	(81.0, 71.4)	(-, 88.9)	(96.7, -)	(97.6, **93.4**)	(97.3, 89.4)	(**97.8**, 91.6)
All classes	(87, 69.4)	(82, 79.0)	(74.4, 64.2)	(-, 85.7)	(95.7, -)	(96.5, 91.7)	(96.7, 90.1)	(**97.5**, **92.1**)

It can also be noted from Table II that randomly reducing the embedding vector size to only 100 dimensions has a very little impact on the anomaly localization performance. The results drop only by 0.4p.p in the AUROC and 0.3p.p in the PRO-score. This simple yet effective dimensionality reduction method significantly reduces PaDiM time and space complexity as it will be shown in Section V-D.

B. Comparison with the state-of-the-art

Localization. In Table III, we show the AUROC and the PRO-score results for anomaly localization on the MVTec AD. For a fair comparison, we used a Wide ResNet-50-2 (WR50) as this backbone is used in SPADE [5]. Since the other baselines have smaller backbones, we also try a ResNet18 (R18). We randomly reduce the embedding size to 550 and 100 for PaDiM with WR50 and R18 respectively.

We first notice that PaDiM-WR50-Rd550 outperforms all the other methods in both the PRO-score and the AUROC on average for all the classes. PaDiM-R18-Rd100 which is a very light model also outperforms all models in the average AUROC on the MVTec AD classes by at least 0.2p.p. When we further analyze the PaDiM performances, we see that the gap for the object classes is small as PaDiM-WR50-Rd550 is the best only in the AUROC (+0.2p.p) but SPADE [5] is the best in the PRO-score (+1.8p.p). However, our models are particularly accurate on texture classes. PaDiM-WR50-Rd550 outperforms the second best model SPADE [5] by 4.8p.p and 4.0p.p in the PRO-score and the AUROC respectively on average on texture classes. Indeed, PaDiM learns an explicit probabilistic model of the normal classes contrary to SPADE [5] or Patch-SVDD [4]. It is particularly efficient on texture images because even if they are not aligned and centered like object images, PaDiM effectively captures their statistical similarity accross the normal train dataset.

Additionally, we evaluate our model on the STC dataset. We compare our method to the two best reported models performing anomaly localization without temporal information, CAVGA-RU [3] and SPADE [5]. As shown in Table IV, the best result (AUROC) on the STC dataset is achieved with our simplest model PaDiM-R18-Rd100 by a 2.1p.p. margin. In fact, pedestrian positions in images are highly variable in this dataset and, as shown in Section V-C, our method performs well on non-aligned datasets.

Detection. By taking the maximum score of the anomaly maps issued by our models (see Section III-C) we give anomaly scores to entire images to perform anomaly detection at the image level. We test PaDiM for anomaly detection with a Wide ResNet-50-2 (WR50) [28] used in SPADE and an EfficientNet-B5 [29]. The Table V shows that our model PaDiM-WR50-Rd550 outperforms every method except MahalanobisAD [23] with their best reported backbone, an EfficientNet-B4. Still our PaDiM-EfficientNet-B5 outperforms every model by at least 2.6p.p on average on all the classes in the AUROC. Besides, contrary to the second best method for anomaly detection, MahalanobisAD [23], our model also performs anomaly segmentation which characterizes more precisely the anomalous areas in the images.

C. Anomaly localization on a non-aligned dataset

To estimate the robustness of anomaly localization methods, we train and evaluate the performance of PaDiM and several

TABLE IV
COMPARISON OF OUR PADIM MODEL WITH THE STATE-OF-THE-ART FOR THE ANOMALY LOCALIZATION ON THE STC IN THE AUROC%.

Model	CAVGA-RU [3]	SPADE [5]	PaDiM-R18-Rd100
AUROC score%	85	89.9	**91.2**

TABLE V
AMVTEC ADにおける画像レベルでの異常検出結果（AUROC%を使用）。 AUROC%.

Model	GANomaly [20]	ITAE [11]	Patch SVDD [4]	SPADE [5] (WR50)	MahalanobisAD [23] (EfficientNet-B4)	PaDiM-WR50-Rd550	PaDiM EfficientNet-B5
all textures classes	-	-	94.6	-	97.2	98.8	**99.0**
all objects classes	-	-	90.9	-	94.8	93.6	**97.2**
all classes	76.2	83.9	92.1	85.5	95.8	95.3	**97.9**

最先端の手法（SPADE ［5］、VAE）を、第IV-A節で説明されるMVTec ADの改変版であるRd-MVTec ADに適用しました。この実験の結果は表VIに示されています。各テスト構成において、MVTec ADに対してランダムシードを使用してデータ前処理を5回実行し、5つの異なるデータセットバージョン（Rd-MVTec AD）を取得します。その後、得られた結果を平均化し、表VIに報告します。提示された結果によると、PaDiM-WR50Rd550は、PROスコアとAUROCの両方で、テクスチャとオブジェクトクラスにおいて他のモデルよりも優れた性能を示しています。さらに、SPADE ［5］とVAEのRd-MVTec ADにおける性能は、通常のMVTec ADでの結果と比較して、PaDiM-WR50-Rd550の性能よりも大幅に低下しています（表IIIを参照）。AUROC結果は、PaDiM-WR50-Rd550で5.3ポイント減少したのに対し、VAEとSPADEではそれぞれ12.2ポイントと8.8ポイントの減少でした。したがって、当手法は既存のテスト済み手法に比べて非一致画像に対してより頑健であると考えられます。

TABLE VI
非整列R-MVTEC ADにおける異常局所化結果。結果はタプル（AUROC%、PRスコア%）として表示されます。

Model	VAE (R18)	SPADE (WR50)	PaDiM-WR50-Rd550
all texture classes	(54.7, 23.1)	(84.6, 75.6)	(**92.4**, **77.9**)
all object classes	(65.8, 30.2)	(88.2, 65.8)	(**92.1**, **70.8**)
all classes	(62.1, 27.8)	(87.2, 69.0)	(**92.2**, **73.1**)

D. Scalability gain

時間複雑度。PaDiMでは、ガウスパラメータを全トレーニングデータセットを使用して推定するため、トレーニングの時間複雑度はデータセットのサイズに比例して増加します。しかし、深層神経ネットワークのトレーニングを必要とする方法とは異なり、PaDiMは事前学習済みのCNNを使用するため、複雑な深層学習トレーニングが不要です。したがって、MVTec ADのような小規模データセットでのトレーニングは非常に高速かつ簡単です。最も複雑なモデルであるPaDiM-WR50-Rd550の場合、CPU（Intel CPU 6154 3GHz 72th）でのシリアル実装によるトレーニングは、MVTec ADクラスでは平均150秒、STC動画シーンでは平均1500秒かかります。

TABLE VII
MVT EC AD上でCPU Intel I 7-4710HQ @ 2.50GHzを使用した場合の異常検出の平均推論時間（秒単位）。

Model	SPADE (WR50)	VAE (R18)	PaDiM R18-Rd100	PaDiM-WR50-Rd550
Inference time (sec.)	7.10	0.21	0.23	0.95

これらのトレーニング手順は、GPUハードウェアを使用してフォワードパスと共分散推定を実行することでさらに高速化可能です。一方、セクションIV-Bで説明された手順に従い、MVTec ADで各クラスごとに10,000枚のイメージを使用してVAEをトレーニングする場合、1つのGPU NVIDIA P5000を使用すると、各クラスあたり2時間40分かかります。一方、SPADE ［5］は学習パラメーターが存在しないため、トレーニングが不要です。それでも、テスト前に通常のトレーニング画像の埋め込みベクトルをすべて計算してメモリに格納します。これらのベクトルはK-NNアルゴリズムの入力となり、表VIIに示すようにSPADEの推論が非常に遅くなります。表VIIでは、主流のCPU（Intel i7-4710HQ CPU @ 2.50GHz）を使用したシリアル実装でモデル推論時間を測定しています。MVTec ADにおいて、SPADEの推論時間は、同様のバックボーンを持つ当社のPaDiMモデルに比べて約7倍遅いです。これは計算コストの高いNN検索が原因です。当社のVAE実装（再構築ベースのモデルと類似）は最も高速なモデルですが、シンプルなモデルPaDiM-R18-Rd100の推論時間は同じオーダーです。同様の複雑さを持つにもかかわらず、PaDiMはVAE手法を大幅に上回っています（セクションV-B参照）。

メモリ複雑度。SPADE ［5］やPatch SVDD ［4］とは異なり、当モデルの空間複雑度はデータセットのトレーニングサイズに依存せず、画像解像度のみに依存します。PaDiMはメモリに事前学習済みのCNNと各パッチに関連するガウスパラメータのみを保持します。表VIIIでは、パラメータをfloat32でエンコードした場合のSPADE、当VAE実装、PaDiMのメモリ要件を示しています。同等のバックボーンを使用した場合、SPADEはMVTec ADにおいてPaDiMよりもメモリ消費量が少ない。しかし、STCのような大規模なデータセットでSPADEを使用すると、そのメモリ消費量は扱いにくくなるのに対し、PaDiM-WR50-Rd550は7倍少ないメモリを必要とする。PaDiMの空間複雑度は、セクションIV-Bで説明したように、後者のデータセットで入力画像の解像度がより高いことから、MVTec ADからSTCへ移行する際に増加します。最後に、当社のフレームワークPaDiMの利点の一つは、ユーザーが推論時間要件、リソース制限、または期待される性能に合わせて、バックボーンと埋め込みサイズを選択することで、方法を容易に適応できる点です。

VI. Conclusion

私たちは、分布モデリングに基づく異常検出と局所化のためのフレームワークPaDiMを、1クラス学習設定において提案しました。これはMVTec ADとSTCデータセットで最先端の性能を達成しています。さらに、評価プロトコルを非一致データに拡張し、初めて

TABLE V
ANOMALY DETECTION RESULTS (AT THE IMAGE LEVEL) ON THE MVTEC AD USING AUROC%.

Model	GANomaly [20]	ITAE [11]	Patch SVDD [4]	SPADE [5] (WR50)	MahalanobisAD [23] (EfficientNet-B4)	PaDiM-WR50-Rd550	PaDiM EfficientNet-B5
all textures classes	-	-	94.6	-	97.2	98.8	**99.0**
all objects classes	-	-	90.9	-	94.8	93.6	**97.2**
all classes	76.2	83.9	92.1	85.5	95.8	95.3	**97.9**

state-of-the-art methods (SPADE [5], VAE) on a modified version of the MVTec AD, Rd-MVTec AD, described in Section IV-A. Results of this experiment are displayed in Table VI. For each test configuration we run 5 times data preprocessing on the MVTec AD with random seeds to obtain 5 different versions of the dataset, denoted as Rd-MVTec AD. Then, we average the obtained results and report them in Table VI. According to the presented results, PaDiM-WR50-Rd550 outperforms the other models on both texture and object classes in the PRO-score and the AUROC. Besides, the SPADE [5] and VAE performances on the Rd-MVTec AD decrease more than the performance of PaDiM-WR50-Rd550 when comparing to the results obtained on the normal MVTec AD (refer to Table III). The AUROC results decrease by 5.3p.p for PaDiM-WR50-Rd550 against 12.2p.p and 8.8p.p decline for VAE and SPADE respectively. Thus, we can conclude that our method seems to be more robust to non-aligned images than the other existing and tested works.

TABLE VI
ANOMALY LOCALIZATION RESULTS ON THE NON-ALIGNED RD-MVTEC AD. RESULTS ARE DISPLAYED AS TUPLES (AUROC%, PRO-SCORE%)

Model	VAE (R18)	SPADE (WR50)	PaDiM-WR50-Rd550
all texture classes	(54.7, 23.1)	(84.6, 75.6)	(**92.4**, **77.9**)
all object classes	(65.8, 30.2)	(88.2, 65.8)	(**92.1**, **70.8**)
all classes	(62.1, 27.8)	(87.2, 69.0)	(**92.2**, **73.1**)

D. Scalability gain

Time complexity. In PaDiM, the training time complexity scales linearly with the dataset size because the Gaussian parameters are estimated using the entire training dataset. However, contrary to the methods that require to train deep neural networks, PaDiM uses a pretrained CNN, and, thus, no deep learning training is required which is often a complex procedure. Hence, it is very fast and easy to train it on small datasets like MVTec AD. For our most complex model PaDiM-WR50-Rd550, the training on a CPU (Intel CPU 6154 3GHz 72th) with a serial implementation takes on average 150 seconds on the MVTec AD classes and 1500 seconds on average on the STC video scenes. These training procedures could be further accelerated using GPU hardware for the forward pass and the covariance estimation. In contrast, training the VAE with 10 000 images per class on the MVTec AD following the procedure described in Section IV-B takes 2h40 per class using one GPU NVIDIA P5000. Conversely, SPADE [5] requires no training as there are no parameters to learn. Still, it computes and stores in the memory before testing all the embedding vectors of the normal training images. Those vectors are the inputs of a K-NN algorithm which makes SPADE's inference very slow as shown in Table VII.

TABLE VII
AVERAGE INFERENCE TIME OF ANOMALY LOCALIZATION IN SECONDS ON THE MVTEC AD WITH A CPU INTEL I7-4710HQ @ 2.50GHZ.

Model	SPADE (WR50)	VAE (R18)	PaDiM R18-Rd100	PaDiM-WR50-Rd550
Inference time (sec.)	7.10	0.21	0.23	0.95

In Table VII, we measure the model inference time using a mainstream CPU (Intel i7-4710HQ CPU @ 2.50GHz) with a serial implementation. On the MVTec AD, the inference time of SPADE is around seven times slower than our PaDiM model with equivalent backbone because of the computationally expensive NN search. Our VAE implementation, which is similar to most reconstruction-based models, is the fastest model but our simple model PaDiM-R18-Rd100 has the same order of magnitude for the inference time. While having similar complexity, PaDiM largely outperfoms the VAE methods (see Section V-B).

Memory complexity. Unlike SPADE [5] and Patch SVDD [4], the space complexity of our model is independent of the dataset training size and depends only on the image resolution. PaDiM keeps in the memory only the pretrained CNN and the Gaussian parameters associated with each patch. In Table VIII we show the memory requirement of SPADE, our VAE implementation, and PaDiM, assuming that parameters are encoded in float32. Using equivalent backbone, SPADE has a lower memory consumption than PaDiM on the MVTec AD. However, when using SPADE on a larger dataset like the STC, its memory consumption becomes intractable, whereas PaDiM-WR50-Rd550 requires seven times less memory. The PaDiM space complexity increases from the MVTec AD to the STC only because the input image resolution is higher in the latter dataset as described in Section IV-B. Finally, one of the advantages of our framework PaDiM is that the user can easily adapt the method by choosing the backbone and the embedding size to fit its inference time requirements, resource limits, or expected performance.

VI. CONCLUSION

We have presented a framework called PaDiM for anomaly detection and localization in one-class learning setting which is based on distribution modeling. It achieves state-of-the-art performance on MVTec AD and STC datasets. Moreover, we extend the evaluation protocol to non-aligned data and the first

TABLE VIII
M メモリ要件（GB単位）MVT EC ADおよびSTCデータセットで訓練された異常局所化手法。

model	SPADE (WR50)	VAE (R18)	PaDiM R18-Rd100	PaDiM-WR50-Rd550
MVTec AD	1.4	0.09	0.17	3.8
STC	37.0	-	0.21	5.2

結果から、PaDiMはこれらのより現実的なデータに対して頑健であることが示されています。PaDiMの低メモリ消費量と低計算時間、およびその使いやすさは、視覚的産業制御など、多様な応用分野に適しています。

REFERENCES

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *CVPR*, 2019.

[2] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *CVPR*, 2020.

[3] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *arXiv, 1911.08616*, 2019.

[4] J. Yi and S. Yoon, "Patch svdd: Patch-level svdd for anomaly detection and segmentation," in *arXiv, 2006.16067*, 2020.

[5] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," in *arXiv, 2005.02357*, 2020.

[6] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *ICLR*, 2020.

[7] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[8] W. Liu, D. L. W. Luo, and S. Gao, "Future frame prediction for anomaly detection – a new baseline," in *CVPR*, 2018.

[9] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *VISIGRAPP*, 2019.

[10] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *ICCV*, 2019.

[11] C. Huang, F. Ye, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," in *arXiv, 1911.10676*, 2019.

[12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[13] K. Sato, K. Hama, T. Matsubara, and K. Uehara, "Predictable uncertainty-aware unsupervised deep anomaly segmentation," in *IJCNN*, 2019.

[14] W. Liu, R. Li, M. Zheng, S. Karanam, Z.Wu, B. Bhanu, R. J. R., and O. Camps, "Towards visually explaining variational autoencoders," in *CVPR*, 2020.

[15] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *CVPR*, 2018.

[16] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *NIPS*, 2018.

[17] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," *ACCV*, 2018.

[18] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *CVPR*, 2019.

[19] K. H. Kim, S. Shim, Y. Lim, J. Jeon, J. Choi, B. Kim, and A. S. Yoon, "Rapp: Novelty detection with reconstruction along projection pathway," in *ICLR*, 2020.

[20] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," in *IJCNN*, 2019.

[21] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: one-class novelty detection using gans with constrained latent representations," in *CVPR*, 2019.

[22] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *ICLM*, 2018.

[23] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *arXiv, 2005.14140*, 2020.

[24] L. Bergman, N. Cohen, and Y. Hoshen, "Deep nearest neighbor anomaly detection," in *arXiv, 2002.10445*, 2020.

[25] S. R. Napoletano P, Piccoli F, "Anomaly detection in nanofibrous materials by cnn-based self-similarity," in *Sensors.*, vol. 18, no. 1, 2018, p. 209.

[26] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NIPS*, 2018.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *ICML*, 2016.

[28] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

[29] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019.

[30] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[31] P. Mahalanobis, "On the generalized distance in statistics," in *National Institute of Science of India*, 1936.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

TABLE VIII
MEMORY REQUIREMENT IN GB OF THE ANOMALY LOCALIZATION METHODS TRAINED ON THE MVTEC AD AND THE STC DATASET.

model	SPADE (WR50)	VAE (R18)	PaDiM R18-Rd100	PaDiM-WR50-Rd550
MVTec AD	1.4	0.09	0.17	3.8
STC	37.0	-	0.21	5.2

results show that PaDiM can be robust on these more realistic data. PaDiM low memory and time consumption and its ease of use make it suitable for various applications, such as visual industrial control.

REFERENCES

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *CVPR*, 2019.

[2] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *CVPR*, 2020.

[3] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *arXiv, 1911.08616*, 2019.

[4] J. Yi and S. Yoon, "Patch svdd: Patch-level svdd for anomaly detection and segmentation," in *arXiv, 2006.16067*, 2020.

[5] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," in *arXiv, 2005.02357*, 2020.

[6] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *ICLR*, 2020.

[7] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[8] W. Liu, D. L. W. Luo, and S. Gao, "Future frame prediction for anomaly detection – a new baseline," in *CVPR*, 2018.

[9] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *VISIGRAPP*, 2019.

[10] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *ICCV*, 2019.

[11] C. Huang, F. Ye, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," in *arXiv, 1911.10676*, 2019.

[12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[13] K. Sato, K. Hama, T. Matsubara, and K. Uehara, "Predictable uncertainty-aware unsupervised deep anomaly segmentation," in *IJCNN*, 2019.

[14] W. Liu, R. Li, M. Zheng, S. Karanam, Z.Wu, B. Bhanu, R. J. R., and O. Camps, "Towards visually explaining variational autoencoders," in *CVPR*, 2020.

[15] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *CVPR*, 2018.

[16] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *NIPS*, 2018.

[17] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," *ACCV*, 2018.

[18] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *CVPR*, 2019.

[19] K. H. Kim, S. Shim, Y. Lim, J. Jeon, J. Choi, B. Kim, and A. S. Yoon, "Rapp: Novelty detection with reconstruction along projection pathway," in *ICLR*, 2020.

[20] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection," in *IJCNN*, 2019.

[21] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: one-class novelty detection using gans with constrained latent representations," in *CVPR*, 2019.

[22] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *ICLM*, 2018.

[23] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *arXiv, 2005.14140*, 2020.

[24] L. Bergman, N. Cohen, and Y. Hoshen, "Deep nearest neighbor anomaly detection," in *arXiv, 2002.10445*, 2020.

[25] S. R. Napoletano P, Piccoli F, "Anomaly detection in nanofibrous materials by cnn-based self-similarity," in *Sensors.*, vol. 18, no. 1, 2018, p. 209.

[26] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NIPS*, 2018.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *ICML*, 2016.

[28] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.

[29] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019.

[30] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[31] P. Mahalanobis, "On the generalized distance in statistics," in *National Institute of Science of India*, 1936.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.