

FRE: A Fast Method For Anomaly Detection And Segmentation

Ibrahima J. Ndiour
ibrahima.j.ndiour@intel.com

Nilesh Ahuja
nilesh.ahuja@intel.com

Utku Genc
utku.genc@intel.com

Omesh Tickoo
omesh.tickoo@intel.com

Intel Labs
Intel Corporation
USA

Abstract

This paper presents a fast and principled approach to address the challenge of visual anomaly detection and segmentation. Our method operates under the assumption of having access solely to anomaly-free training data while aiming to identify anomalies of an arbitrary nature on test data. We build upon prior research and present a generalized approach that utilizes a shallow linear autoencoder to perform out-of-distribution detection on the intermediate features generated by a pre-trained deep neural network. More specifically, we compute the *feature reconstruction error* (FRE) and establish it as a principled measure of uncertainty. We rigorously connect our technique to the theory of linear auto-associative networks to provide a solid theoretical foundation and to offer multiple practical implementation strategies. Furthermore, extending the FRE concept to convolutional layers, we derive FRE maps that provide precise pixel-level spatial localization of the anomalies within images, effectively achieving segmentation. Extensive experimentation demonstrates that our method outperforms the current state of the art. It excels in terms of speed, robustness, and remarkable insensitivity to parameterization. We make our code available at: <https://intellabs.github.io/dfm>

1 Introduction

AI and deep-learning based approaches to industrial anomaly detection and segmentation have gathered a lot of interest in recent times (see [23] for an excellent survey review of anomaly detection research). Anomaly detection is typically framed as an unsupervised one-class classification problem in which a distribution of the defect-free data is learnt and anomalies are then detected as deviations from this model. Supervised approaches are largely unsuitable as they require access to a sufficient number of labeled training samples. In industrial settings, this poses a challenge due to the rare appearance of anomalies and the inherent unpredictability of the specific types of anomalies that may arise during operations.

Within this one-class paradigm, various types of models have been proposed in literature. Deep autoencoder (AE) models [24] are trained on the normal data and then use the test-time

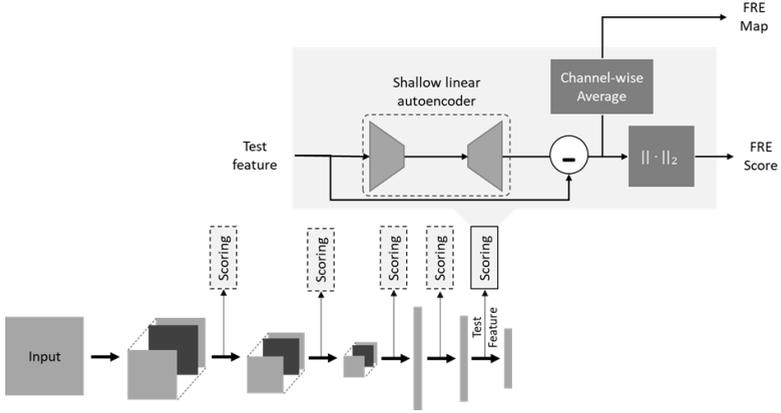


Figure 1: Block diagram of our proposed approach.

reconstruction error in the pixel space as an indicator of anomalies. As pointed out in [14], however, the AE can sometimes “generalize” so well that it can also reconstruct the abnormal inputs well. On the other hand, AEs are often sensitive to typical image degradations (such as blur and noise) and might end up identifying normal samples as defective. Methods that operate in the feature space of deep neural networks (DNN) rather than directly in the pixel space have been found to be more effective and robust. Earlier works in this space involved training deep one-class neural networks [9, 24], or deep generative models [11, 22].

Another class of methods use DNNs pre-trained on large datasets such as ImageNet and then build anomaly models on the induced feature spaces. Not only do they avoid DNN training (often a cumbersome process), but they also are able to leverage the rich space of feature representations learnt from a larger, more diverse dataset. These include methods such as [10, 11, 21, 28] that model clusters or distributions on a multi-level pyramid of deep-features. The results can be impressive, but come at the cost of significant computational and storage complexity.

Other notable alternative approaches, which have demonstrated superb performance, include [16, 20, 23, 26]. More recently, a method was proposed for out-of-distribution (OOD) detection that involved projecting the intermediate features of a DNN into a lower-dimensional subspace with principal component analysis (PCA) and using the projection error (named as *feature reconstruction error*) as an anomaly score [19]. Limited but impressive results were reported on general OOD detection and the auxiliary task of anomaly detection.

Contributions: This paper extends and generalizes the work in [19], which provided an intuition for the method and limited empirical validation only for anomaly detection. Here, we lay out the theoretical background and a more general and comprehensive framework. The principal contributions are as follows:

1. We propose a generalized approach to solve the problem of anomaly detection with the utilization of a shallow linear autoencoder as a principled out-of-distribution detection method operating in the feature space produced by a pre-trained DNN. We show that [19] is a particular instance of the more general solution presented here.

2. We make a rigorous connection to linear auto-associative networks in order to provide a theoretical justification for the method. In particular, we leverage this rigorous connection to establish that the *feature reconstruction error* is a precise quantitative out-of-distribution measure of anomalous input patterns.
3. In addition to image-level detection, we simultaneously solve the more challenging task of pixel-level anomaly segmentation.
4. We offer multiple implementation strategies for efficient training and inference of the method that address concerns related to memory, computational complexity and various use cases, such as underlying computing resources, processing speed and dataset size. As a result, our approach can seamlessly and efficiently run on CPU, GPU and heterogeneous computing platforms.
5. We conduct extensive experimentation, benchmarking and ablation studies to quantify the quality and performance of our method. Our findings show that our approach yields state-of-the-art results on both detection and segmentation, and is robust and largely insensitive to parametrization. Moreover, it is highly computationally efficient, and can train and run inference at very high-frame rates on both CPU and GPU.

2 Preliminaries: Linear auto-associative networks

Auto-associative multi-layer perceptrons are neural networks that attempt to map each input vector onto itself. Here, we consider a linear auto-associative network with an N -dimensional input layer, an M -dimensional hidden layer ($M < N$), and an output layer of the same size as the input. We assume linear activation functions. In this form, this type of network is nowadays called a shallow undercomplete autoencoder. It learns to approximate the identity mapping by optimizing weights in order to minimize the so-called reconstruction error E , the quadratic error between inputs and outputs on the training data:

$$E(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^n \|x_i - \hat{x}_i\|^2 = \sum_{i=1}^n \|x_i - \mathbf{B}\mathbf{A}x_i\|^2 \quad (1)$$

where $\{x_i\}_{i=1}^n$ represents the training data, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the mapping function from the input to the hidden layer, and $\mathbf{B} \in \mathbb{R}^{N \times M}$ is the demapping from hidden to output layer. Next, we briefly recapitulate a few interesting results about these linear auto-associative networks. For an in-depth proof and derivation of these results, please refer to Baldi and Hornik [3], Bourlard and Kamp [4], Denceux and Masson [5] and Bishop [6].

Proposition 1: *The error function E has, up to an equivalence, a unique local and global minimum corresponding to an orthogonal projection onto the principal subspace (spanned by the first eigenvectors of the covariance matrix associated with the training data). All other critical points of E are saddle points [3, 4, 5].* In other words, the optimal map \mathbf{A} performs the same linear dimensionality reduction as principal component analysis (PCA). This holds true even when a nonlinear activation function is used.

Proposition 2: *A precise quantitative measure of out-of-distribution for a new input is exactly given by its distance to the principal subspace [3].*

Proposition 3: *If \mathbf{C} is an invertible matrix, the global mapping can be expressed as $\mathbf{AB} = \mathbf{ACC}^{-1}\mathbf{B} = (\mathbf{AC})(\mathbf{C}^{-1}\mathbf{B})$ and therefore the optimal functions for mapping (encoding) and demapping (decoding) are not unique. PCA corresponds to the optimum when \mathbf{C} is the identity matrix [14].* However, this particular solution is generally not obtained when the error function is minimized iteratively, such as with backpropagation. The solution it converges to corresponds to an arbitrary \mathbf{C} . This lack of determinism does not pose an issue for "end-to-end" methods that are driven only by the reconstruction error and never process information in the reduced space at the bottleneck of the network (e.g. compression). However, for many computer vision methods that perform data analysis in the reduced space (e.g. clustering, classification, visualization), deriving meaningful results directly from the reduced embeddings is problematic because they are scaled by this arbitrary matrix \mathbf{C} . To alleviate this, one effective approach is to tie the demapping (decoder) to the mapping (encoder) with the constraint $\mathbf{B} = \mathbf{A}^T$ because of the following corollary [14].

Corollary: *Constraining $\mathbf{B} = \mathbf{A}^T$ is equivalent to constraining \mathbf{C} to be an orthogonal matrix: the optimum obtained by backpropagation is then equivalent to PCA, up to an arbitrary isometric transformation [14].* Reduced embeddings obtained are then the PCA embeddings subjected to a rigid transformation (e.g. rotation, translation, reflection). A remarkable by-product of the "tied" variant is that it converges much faster than the standard network, thanks to a reduction in the number of parameters by half.

3 Proposed Approach

Our approach consists in performing out-of-distribution detection on the features generated by a pre-trained DNN. To achieve this, we employ a shallow linear autoencoder with a single-layer encoder and decoder and compute the feature reconstruction error (FRE) to serve as an uncertainty score.

3.1 Feature reconstruction error (FRE)

We consider a deep neural network pre-trained on an N -class classification problem. For an input \mathbf{I} , let $\mathbf{u} \triangleq f_k(\mathbf{I}) \in \mathbb{R}^{C_k \times H_k \times W_k}$ denote the output feature at the k^{th} intermediate layer of the network. The features induced by the training dataset do not fully span the high-dimensional space in which they reside. Hence, for a training dataset of size M , the data-matrix $\mathbf{D} = [f_k(\mathbf{I}_1) | \dots | f_k(\mathbf{I}_M)]$ constructed from the features is rank deficient. Consequently, we learn a transformation $\mathcal{T} : \mathcal{H} \rightarrow \mathcal{L}$ (with $\dim(\mathcal{L}) \ll \dim(\mathcal{H})$) that maps the high-dimensional features onto an appropriate subspace, along with the corresponding inverse transformation \mathcal{T}^{inv} .

In [14], PCA was used as the transform of choice and its Moore-Penrose pseudo-inverse was the corresponding inverse transform [14]. Here, we generalize this notion to a shallow linear auto-encoder as explained in Section 2. The encoder comprises a single linear layer that projects the input feature to a lower-dimensional subspace. The decoder, which re-projects the lower-dimensional representation to the original size naturally constitutes the inverse transform. During inference, the transformation \mathcal{T} is applied to a test feature, $\mathbf{u} \triangleq f_k(\mathbf{I}) \in \mathbb{R}^{C_k \times H_k \times W_k}$ to obtain its reduced-dimension embedding. In practice, the output feature \mathbf{u} is vectorized prior to dimensionality reduction. This reduced embedding is

inverse-transformed into the original space by applying \mathcal{T}^{inv} and a *feature reconstruction error* (FRE) is calculated as the difference between the original and reconstructed vectors,

$$FRE(\mathbf{u}, \mathcal{T}) = \mathbf{e} \triangleq \mathbf{u} - \hat{\mathbf{u}} = \mathbf{u} - (\mathcal{T}^{inv} \circ \mathcal{T})\mathbf{u}. \quad (2)$$

Note that \mathbf{e} has the same dimension as the vectorized feature tensor \mathbf{u} , owing to the inverse transformation \mathcal{T}^{inv} . From the FRE, we derive both a score, S_{AD} , for image-level anomaly detection, and a segmentation map, \mathbf{M} , for pixel-level anomaly localization.

The detection score, defined as $S_{AD} = \|\mathbf{e}\|_2$ is highly effective at discriminating between normal and anomalous samples because FRE is simply the distance from a given test feature to the principal subspace, which we established as a precise quantitative measure of out-of-distribution in Section 2. A simple visual interpretation is that anomalous samples will typically lie far away from the principal subspace, while normal samples will lie on it or very close to it.

To derive the localization map, \mathbf{M} , we rearrange the vector \mathbf{e} into a $C_k \times H_k \times W_k$ shaped tensor (i.e. the shape of the feature $f_k(\mathbf{I})$). We then perform channel-wise averaging at all pixel locations in order to accumulate the out-of-distribution (FRE) errors along the channel dimension. This produces a single-channel FRE anomaly map of size $H_k \times W_k$, i.e.

$$\mathbf{M}_k(i, j) = \frac{1}{C_k} \sum_{c=1}^{C_k} \mathbf{e}(c, i, j)$$

Finally, we resize this map to the dimension of the input image $H_{im} \times W_{im}$ in order to produce the final anomaly map \mathbf{M} . This anomaly map is a heatmap of the same resolution as the input image, with higher intensity regions corresponding to anomalies.

Taken together, the FRE score and anomaly map derived from the intermediate features of a DNN suffice to perform both anomaly detection and pixel-level anomaly localization (segmentation). A block diagram of the complete end-to-end pipeline is shown in Fig. 1.

Combining layers: Our technique is versatile and can be employed with the features generated from any intermediate layer of a DNN. Moreover, we can enhance and broaden the applicability of our method by not confining feature modeling to a single layer. Instead, we generate FRE maps at multiple layers and combine them into a unified map to achieve improved performance for detection and segmentation. One such combination consists in taking the (pixel-wise) geometric average of the maps:

$$\mathbf{M}_{comb} = \prod_{k=1}^n (\mathbf{M}_k)^{1/n} \quad (3)$$

In Section 4, we study the segmentation improvement obtained with the combination on a variety of pre-trained DNN models (backbones).

3.2 Implementation Strategies

Our approach can accommodate several implementation strategies. Direct implementation of the deterministic PCA is done by eigen-decomposition of the training data covariance matrix, or singular value decomposition of the training data matrix. Alternatively, the iterative approach consists in training the standard linear autoencoder or its tied variant with back-propagation or some other optimization technique such as a Newton’s method. We will

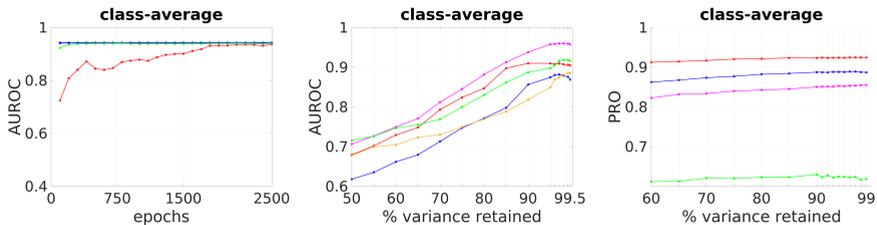


Figure 2: *Left:* Detection results from layer3 after every 100 epochs for FRE-PCA (blue) , FRE-AE (red) and FRE-AEt (green), *Middle and Right:* For each layer, we show the class-average detection and segmentation results while varying the only parameter of our method. Blue, red, magenta, green and gold resp. denote the following layers: layer1, layer2, layer3, layer4, avgpool. ResNet50 backbone used.

denote the former as FRE-PCA, and the latter as FRE-AE and FRE-AEt respectively. We showed in Section 2 that they are rigorously equivalent, and will validate it empirically in Section 4. We also envision hybrid strategies where training can be performed with PCA to learn weights that are later loaded onto a neural network for inference, and vice-versa.

From a numerical analysis perspective, the PCA approach based on eigen-decomposition or SVD is superior to iterative methods for several reasons. It offers determinism, consistently yielding the same solution, ensures convergence within a finite number of steps, and provides an ordered set of eigenvalues with corresponding orthonormal eigenvectors [6]. This unique feature means that, unlike iterative methods, PCA calculations only need to be performed once, even when changing the dimension of the reduced feature subspace. However, there exists a compelling case for employing iterative solvers. When dealing with large datasets, computing PCA can become increasingly challenging, often rendered impossible due to limitations like insufficient memory, algebraic singularities, or numerical instability. While randomized methods for PCA do exist, the iterative implementations proposed here are naturally better suited for handling large datasets thanks to their batch-based training paradigm. Moreover, iterative approaches can accommodate lower precision, such as 16 bits, whereas SVD typically demands 32 or 64-bit precision [4]. Additionally, iterative implementations allow for the introduction of supplementary constraints (e.g., regularization) through straightforward additions to the quadratic loss function. The choice between these implementations hinges on several factors, including dataset size, available computing resources (e.g., CPU vs. GPU), and other pertinent considerations.

4 Experimental Results

We experiment with several pretrained DNN models, and compare our results against multiple benchmarks. We rely on two popular datasets extensively used and cited: the Magnetic Tile Defect (MTD) dataset [15] and the MVTec-AD dataset [1] which is an aggregation of 15 different datasets across a variety of objects, textures and defect artifacts. Similar to other methods, we resize the input images to 256×256 prior to applying our method. For detection, the binary classifier is assessed by measuring the area under the ROC curve (AUROC). For segmentation, a similar per-pixel AUROC metric is used. However, as most objects occupy relatively small, even minuscule, areas within the image, this metric can be skewed high. To address this, the PRO metric was devised as a more precise segmentation evaluation

Table 1: Anomaly Detection AUROC on MVTec dataset

* Methods not reporting class-itemized results. For PaDiM, results were recreated using *anomalib* [1]

Category	GANomaly [1]	DifferNet [2]	SPADE* [3]	PaDiM* [4]	PatchCore [5]	AST [6]	FRE (Ours)
Carpet	69.9	92.9	-	99.5	98.7	97.5	100
Grid	70.8	84.0	-	94.2	98.2	99.1	99.5
Leather	84.2	97.1	-	100	100	100	99.5
Tile	79.4	99.4	-	97.4	98.7	100	100
Wood	83.4	99.8	-	99.3	99.2	100	97.6
Bottle	89.2	99.0	-	99.9	100	100	100
Cable	75.7	95.9	-	87.8	99.5	98.5	100
Capsule	73.2	86.9	-	92.7	98.1	99.7	98.5
Hazelnut	78.5	99.3	-	96.4	100	100	94.8
Metal Nut	70.0	96.1	-	98.9	100	98.5	96.1
Pill	74.3	88.8	-	93.9	96.6	99.1	98.9
Screw	74.6	96.3	-	84.5	98.1	99.7	99.4
Toothbrush	65.3	98.6	-	94.2	100	96.6	99.1
Transistor	79.2	91.1	-	97.6	100	99.3	99.3
Zipper	74.5	95.1	-	88.2	99.4	99.1	97.0
Average	76.2	94.9	85.5	97.9	99.1	99.2	98.6

Table 2: Anomaly Detection AUROC on Magnetic Tile dataset

GANomaly [1]	1-NN [7]	DifferNet [2]	PatchCore [5]	FRE (Ours)
76.6	80.0	97.7	97.9	99.2

measure. It offers a threshold-independent evaluation metric based on per-region overlap, ensuring equal weighting of ground-truth regions regardless of size differences. We refer to [8] for more details on these metrics. While we sometimes provide only class-average results, detailed class-itemized results are included in the supplementary material. Bold font is used to highlight the best results when comparing against benchmarks. Finally, it is worth mentioning that quality is measured by using both the (image-level) detection performance for the classification task *and* the pixel-level localization performance for the challenging segmentation task.

Validation of implementation strategies: Here, we compare the different implementations strategies FRE-PCA, FRE-AE and FRE-AE_t on MVTec-AD using a ResNet50 backbone. For the iterative implementations, we took snapshots of the models being learned after every 100 epochs and performed detection on the test set in order to show the speed of convergence. Fig. 2 displays the incremental class-average AUROC obtained. They show that all implementations produce the same result and that the tied variant FRE-AE_t converges much faster than FRE-AE. Because of this equivalence, unless needed, we will simply refer to FRE for the rest of this text, and remain agnostic to the specific implementation used.

Sensitivity to parameterization: The only parameter for our method is the dimension of the reduced subspace at the network’s bottleneck. For the PCA implementation, this is usually expressed as a percentage of the variability in the input data (i.e. training features) that should be retained. We study the impact of this parameter for both detection and localization by measuring both AUROC and PRO metrics across a range of retained-variance values. The results for ResNet50 are shown in Fig. 2. Although the performance does degrade gracefully with retained-variance value, it is largely insensitive to it within reasonable bounds [0.9, 1.0].

Comparison to existing methods: For both image-level detection and segmentation, we benchmark against state-of-the-art anomaly detection techniques [10, 11, 12, 13]. The results

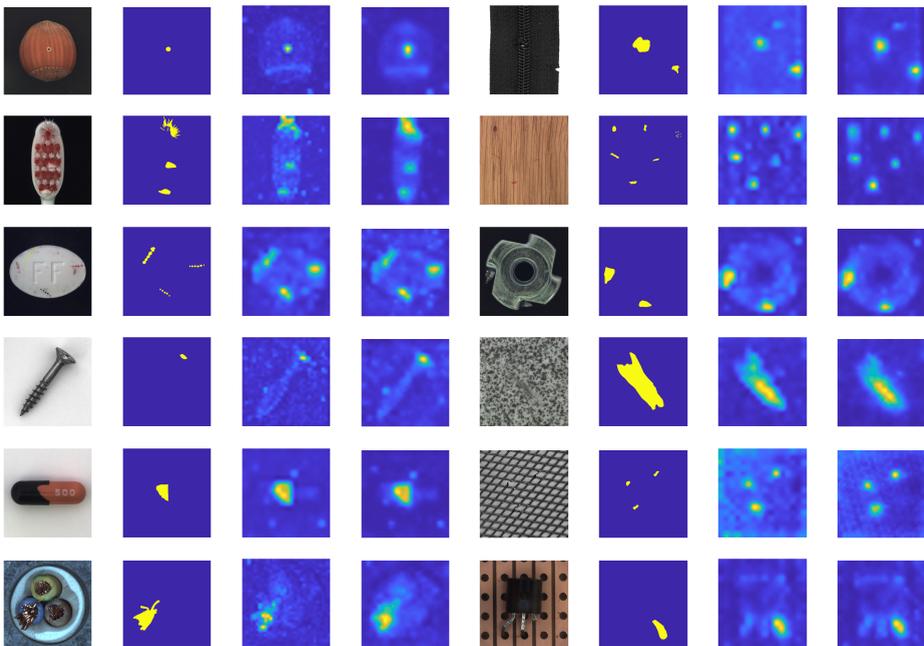


Figure 3: From left to right, each set of four images comprises: Original image, ground truth mask, anomaly heatmap (single layer FRE), anomaly heatmap (three-layer FRE).

Table 3: MVTec Anomaly Detection: AUROC for FRE across backbones

ResNet18	ResNet50	VGG16	EfficientNet-B5
95.4	96.0	91.6	98.6

are presented in Tables 1, 2, 4 and 6. We see that our method attains excellent results, competitive or exceeding the state-of-the-art.

Choice of backbones: We apply our method to a variety of commonly used pretrained backbones, including VGG16, Resnet18, Resnet50, and EfficientNet-B5. The AUROC and PRO results for all MVTEC categories are presented in Tables 3, 5 and 7 respectively. Our method attains excellent results on all backbones, showing that it is not sensitive to the architecture used. 1L and 3L indicate results obtained by either using FRE maps from a single layer or by fusing FRE maps from three layers as explained in Section 3.1.

Single vs Multiple Layers: We report results for our method both with a single layer and by combining multiple layers as described in Sec. 3.1. For single layer, we choose a layer in the middle of the network – not too close to either the input or the output – as features from such layers have a good balance between spatial and semantic information. See for example, the results for Resnet50 in Table 8; other backbones show similar trends.

Table 4: MVTec Anomaly Segmentation benchmark with pixel-wise AUROC metric.

	CNN Dict[]	AE L2[]	SPADE[]	PaDiM[]	PatchCore[]	AST[]	FRE (1L & 3L)
Average	78	82	96	97.5	98.1	95.0	97.8 98.2

Table 5: MVTec Anomaly Segmentation: pixel-wise AUROC for FRE across backbones

	Efficientnet B5		VGG16		Resnet18		Resnet50	
	1L	3L	1L	3L	1L	3L	1L	3L
Average	96.4	97.2	97.4	97.4	96.8	97.8	97.8	98.2

Table 6: MVTec Anomaly Segmentation benchmark with PRO metric.

Category	CNN Dict[□]	AE L2[□]	SPADE[□]	PaDiM[□]	PatchCore[□]	FRE (Ours)	
						1L	3L
Carpet	46.9	45.6	94.7	96.2	96.6	93.1	96
Grid	18.3	58.2	86.7	94.6	96	87.2	89.9
Leather	64.1	81.9	97.2	97.8	98.9	98.3	98.7
Tile	79.7	89.7	75.6	86	87.3	82.5	86
Wood	62.1	72.7	87.4	91.1	89.4	92.7	94.2
Bottle	74.2	91	95.5	94.8	96.2	95.5	96
Cable	55.8	82.5	90.9	88.8	92.5	90.2	89.1
Capsule	30.6	86.2	93.7	93.5	95.5	94.3	96
Hazelnut	84.4	91.7	95.4	92.6	93.8	94.1	96
Metal nut	35.8	83	94.4	85.6	91.4	92	93.8
Pill	46	89.3	94.6	92.7	93.2	93	93.5
Screw	27.7	75.4	96	94.4	97.9	95.4	96.5
Toothbrush	15.1	82.2	93.5	93.1	91.5	91.6	93.7
Transistor	62.8	72.8	87.4	84.5	83.7	92.9	94
Zipper	70.3	83.9	92.6	95.9	97.1	94.6	95.4
Average	51.5	79.0	91.7	92.1	93.4	92.5	93.9

For multiple layers, we combine maps from three layers: the chosen single layer and the two layers immediately adjacent to it. These results are reported in Tables 5 and 7 for various backbones. We observe that single-layer results far exceed most other methods and are at par with the current state of the art, thereby validating the effectiveness of our approach. However, combining scores using FRE maps from three layers allows us to further improve the single-layer results. It is possible to include even more layers, but the quality increase needs to be balanced against the additional computations incurred.

Finally, we show in Fig. 3 the segmentation maps for several sample input images. We see that these FRE heatmaps from the single-layer already correspond very well to the ground truth. A small but noticeable improvement is further observed when fusing FRE maps from three layers, consistent with the improvement in the pixelwise AUROC and PRO metrics.

4.1 Complexity and Speed

We conducted a comprehensive evaluation of computational performance metrics, including training time and inference run-time, while using our method with a ResNet50 backbone on the MVTec dataset as the workload. For training, we present the averaged training time across all fifteen MVTec categories. In terms of inference, we gauge performance by measuring throughput in frames-per-second (fps). To facilitate comparison, we use the same workload to profile the performance of PatchCore [[21](#)] with a ResNet50 backbone and the AST method [[23](#)], using the reference implementations provided by the respective authors on their GitHub pages. All methods are implemented on three systems respectively featuring an Intel Core-i7 1270P CPU (laptop class), an Intel Xeon Platinum 8280 CPU (server class), and an NVidia 2080-Ti GPU. In our approach, we report performance using both the FRE-AEt and FRE-PCA implementations, as elaborated earlier. It’s worth noting that during inference, the speed of FRE-AEt and FRE-PCA is identical, as they entail the same computations.

The results, presented in Table 9 and more comprehensively in the supplemental material, unveil that both FRE implementations exhibit significantly faster training times than

Table 7: MVTec Anomaly Segmentation: PRO for FRE across backbones

	Efficientnet B5		VGG16		Resnet18		Resnet50	
	1L	3L	1L	3L	1L	3L	1L	3L
carpet	79.5	90.8	90.7	91.6	93.3	95.4	93.1	96.0
grid	80.8	89.1	86.6	86.5	85.1	87.5	87.2	89.9
leather	93.7	96.7	94.4	95.3	98.0	98.6	98.3	98.7
tile	79.7	85.4	74.6	71.5	66.6	73.9	82.5	86.0
wood	81.6	85.7	90.5	91.2	89.6	92.6	92.7	94.2
bottle	94.4	94.5	93.4	93.6	94.6	95.8	95.5	96.0
cable	87.1	89.1	83.5	84.4	87.2	88.5	90.2	89.1
capsule	93.7	93.7	95.2	96.7	93.6	95.7	94.3	96.0
hazelnut	91.4	92.1	94.8	94.7	93.3	96.0	94.1	96.0
metal nut	89.6	89.9	91.0	90.9	90.6	93.7	92.0	93.8
pill	91.1	92.3	93.5	93.9	91.1	92.9	93.0	93.5
screw	93.7	93.8	97.8	98.1	96.0	97.2	95.4	96.5
toothbrush	91.4	90.6	93.0	93.5	92.6	94.5	91.6	93.7
transistor	92.4	95.1	91.4	91.9	91.8	92.9	92.9	94.0
zipper	90.1	92.8	93.3	91.7	93.6	95.4	94.6	95.4
Average	88.7	91.4	90.9	91.0	90.5	92.7	92.5	93.9

Table 8: MVTec Anomaly Segmentation: FRE ResNet50 Performance across layers

	Layer1	Layer2	Layer3	Layer4
Avg PRO	85.5	92.5	91.3	71.3
Avg Pixel-AUROC	94.7	97.8	97.4	93.2

PatchCore [21] and AST [23], while achieving notably higher throughput during inference. Importantly, FRE maintains its high performance even when executed on a laptop-class CPU. In particular, we emphasize that FRE is able to train within a matter of seconds, even on the laptop-class CPU, which is order of magnitude faster than the comparison methods. This is a remarkable property of the method, especially for usages that demand low-cost budget at the edge and other use cases that require frequent re-training of the anomaly detection models.

5 Conclusion and Future Work

This work presented a fast, generalized approach for visual anomaly detection and segmentation. We propose applying a shallow, linear autoencoder on the intermediate features produced by a pretrained DNN and computing the *feature reconstruction error* (FRE) for use as uncertainty score. When applied at convolutional layers, we further derive FRE maps that provide pixel-level segmentation of the anomalies in the image. Detailed experimentation shows qualitative performance that meets or exceeds the state of the art. Moreover, our method is fast and requires no tedious manual tuning of parameters.

Future work will seek to extend this approach to uncertainty estimation in deep neural networks, with applications to several downstream tasks, such as general out-of-distribution detection, active learning, and continual learning. In an open world setting, we will need to learn more complex surfaces to embed the features. In practice, this will require the utilization of (deeper) nonlinear autoencoders in order to achieve nonlinear manifold learning.

Table 9: Training times and inference framerates evaluated on different computing platforms.

	Training Time (↓)				Inference frames-per-sec (↑)		
	PatchCore[21]	AST[23]	FRE-AEt	FRE-PCA	PatchCore[21]	AST[23]	FRE
Intel Xeon 8280	3mn 38s	25min 24s	1min 18s	6.3s	7.11	11.8	52.6
Intel i7-1270P	4mns 5s	5hr 8min	2min 52s	22.8s	1.28	2.8	16.4

References

- [1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conference on Computer Vision (ACCV)*, pages 622–637. Springer, 2018.
- [2] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. AnomaLib: A deep learning library for anomaly detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 1706–1710, 2022.
- [3] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.
- [4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9592–9600, 2019.
- [5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4183–4192, 2020.
- [6] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [7] Hervé Bourlard and Selen Kabil. Autoencoders reloaded. *Biological Cybernetics*, 2022. doi: 10.1007/s00422-022-00937-6.
- [8] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.
- [9] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.
- [10] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *CoRR*, abs/2005.02357, 2020. URL <https://arxiv.org/abs/2005.02357>.
- [11] Thomas Defard, Aleksandr Setkov, Angélique Loesch, and Romaric Audigier. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 475–489, 2021.
- [12] Thierry Denœux and M.-H Masson. Principal component analysis of fuzzy data using autoassociative neural networks. *IEEE Transactions on Fuzzy Systems*, pages 336 – 349, 2004.
- [13] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- [14] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 1705–1714, 2019.

- [15] Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36(1):85–96, 2020.
- [16] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9659–9669, 2021.
- [17] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, 18(1), 2018. doi: 10.3390/s18010209. URL <https://www.mdpi.com/1424-8220/18/1/209>.
- [18] Tiago S. Nazaré, Rodrigo Fernandes de Mello, and Moacir A. Ponti. Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos? *CoRR*, abs/1811.08495, 2018. URL <http://arxiv.org/abs/1811.08495>.
- [19] Ibrahima J Ndiour, Nilesh A Ahuja, and Omesh Tickoo. Subspace modeling for fast out-of-distribution and anomaly detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 3041–3045, 2022.
- [20] Nicolae-Catalin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13576–13586, 2022.
- [21] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14298–14308, 2022.
- [22] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but different: Semi-supervised defect detection with normalizing flows. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1906–1915, 2021.
- [23] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2591–2601, 2023.
- [24] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning (ICML)*, pages 4393–4402, 2018.
- [25] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5): 756–795, 2021.
- [26] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. DRAEM - a discriminatively trained reconstruction embedding for surface anomaly detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8330–8339, 2021.

-
- [27] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.
- [28] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision (ECCV)*, pages 392–408, 2022.