

# SUBSPACE MODELING FOR FAST OUT-OF-DISTRIBUTION AND ANOMALY DETECTION

Ibrahima J. Ndiour, Nilesh A. Ahuja, Omesh Tickoo

Intel Labs

## ABSTRACT

This paper presents a fast, principled approach for detecting anomalous and out-of-distribution (OOD) samples in deep neural networks (DNN). We propose the application of linear statistical dimensionality reduction techniques on the semantic features produced by a DNN, in order to capture the low-dimensional subspace truly spanned by said features. We show that the *feature reconstruction error* (FRE), which is the  $\ell_2$ -norm of the difference between the original feature in the high-dimensional space and the pre-image of its low-dimensional reduced embedding, is highly effective for OOD and anomaly detection. To generalize to intermediate features produced at any given layer, we extend the methodology by applying nonlinear kernel-based methods. Experiments using standard image datasets and DNN architectures demonstrate that our method meets or exceeds best-in-class quality performance, but at a fraction of the computational and memory cost required by the state of the art. It can be trained and run very efficiently, even on a traditional CPU.

**Index Terms**— Anomaly detection, out-of-distribution detection, uncertainty estimation, subspace modeling.

## 1. INTRODUCTION

Deep networks deployed in real-world conditions will inevitably encounter out-of-distribution data, which leads to outputs that are unpredictable, unexplainable and sometimes catastrophic. The ability to detect OOD data is therefore critical for the deployment of safe and transparent systems.

OOD detection is typically performed by estimating an uncertainty score (along with the output) for each input. Early methods include the softmax score [1] and its temperature-scaled variants [2]. Bayesian neural networks [3] and ensembles of discriminative classifiers [4] are more recent and can generate high quality uncertainty, but at the cost of complex model representations, and substantial compute and memory. Deep generative models learn distributions over the input data, and then evaluate the likelihood of new inputs with respect to the learnt distributions [5, 6, 7]. Gradient-based characterization of abnormality in autoencoders is highlighted in

[8]. Finally, methods such as [9, 10] learn parametric class-conditional probability distributions over the features and evaluate likelihoods over these as uncertainty scores.

Within the more general problem of OOD detection, anomaly detection has become particularly important in industrial applications. Its goal is to identify rare and abnormal events from the observation of data. Anomaly detection algorithms rely on good, defect-free samples during the training stage, and identify anomalous samples by comparing against the learned distribution of good data. Since this is really a specific type of OOD detection, state-of-the-art methods in anomaly detection are based on the same principles as the general OOD problem. For instance, [11, 12] use deep generative models to model the distribution of good samples. Alternately, methods such as [13, 14] model clusters or distributions on a multi-level pyramid of deep-features. [15] uses greedy coresets subsampling (which is NP-hard) on the feature-banks to reduce memory requirements. These methods show impressive results to varying degrees, but at the cost of significant computational and storage complexity during training or inference.

We present here a method for anomaly and OOD detection based on appropriately modeling the subspace of the intermediate features produced by a DNN. The high dimensionality of this feature space makes it very challenging, both computationally and algebraically, to perform a variety of otherwise routine tasks on the features, a phenomenon known as the *curse of dimensionality*. For instance, it leads to rank-deficiency in the data-matrix of the features. The *manifold hypothesis* states, however, that real-world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional space. [16] prescribes that these high-dimensional spaces should be modeled by appropriate low-dimensional manifolds and sub-spaces. For the input image space, the problems highlighted by the *manifold hypothesis* are well understood: most points in the high-dimensional image space do not correspond to realistic natural images. In the context of the intermediate features of a DNN, this implies that the features sparsely occupy the high-dimensional space they live in. Hence, the true subspace spanned by the features can be accurately captured by appropriately mapping the original high-dimensional feature space to a reduced lower-dimensional subspace.

In this work, using a single-layer of a pretrained DNN, we show that the *feature reconstruction error* (FRE), the  $\ell_2$ -norm

---

We would like to thank Ergin Utku Genc at Intel Labs for insightful discussions that have helped improve this research, and for help with the open-sourcing of the code.

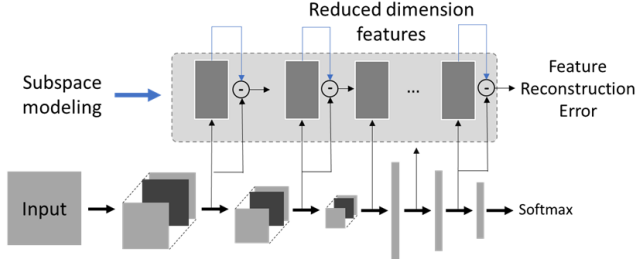


Fig. 1. Proposed system.

of the difference between the original feature in the high-dimensional space and the pre-image of its low-dimensional reduced embedding, is a highly effective uncertainty score for anomaly and OOD detection. This circumvents the need to perform any subsequent processing in the reduced feature space, thereby greatly simplifying the procedure both during modeling and inference. We show that this approach achieves state-of-the-art anomaly and OOD detection performance (section 3.1), but with significantly lower complexity compared to other methods. This makes it very attractive for deployment in real-world industrial usages on low-cost edge platforms without requiring investment into expensive discrete GPUs. Furthermore, the approach does not modify the network’s parameters, which is a significant advantage for trained networks already deployed.

## 2. APPROACH

Consider a deep neural network (DNN) trained on an  $N$ -class classification problem. For an input  $\mathbf{x}$ , let  $f(\mathbf{x})$  denote the output at an 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  $\mathbf{D} = [f(\mathbf{x}_1) | \dots | f(\mathbf{x}_M)]$  constructed from the features is rank deficient. Table 1 shows the severity of the rank deficiencies in the higher-dimensional inner layers of a ResNet18 model used in our experiments. Hence, we learn a transformation  $\mathcal{T}$  that maps the high-dimensional features onto an appropriate subspace,  $\mathcal{T} : \mathcal{H} \rightarrow \mathcal{L}$  with  $\dim(\mathcal{L}) \ll \dim(\mathcal{H})$ , along with the inverse transformation  $\mathcal{T}^\dagger$ . During inference, this transformation is applied to a test feature sample to obtain its reduced-dimension embedding. This reduced embedding is inverse-transformed into the original space and a *feature reconstruction error* (FRE) score is calculated as the  $\ell_2$  norm of the difference between the original and reconstructed vectors:

$$FRE(\mathbf{x}) = \|f(\mathbf{x}) - (\mathcal{T}^\dagger \circ \mathcal{T})(f(\mathbf{x}))\|_2 \quad (1)$$

This score can be used as an uncertainty score for OOD detection. In what follows, we explain the various aspects of the subspace modeling process. A complete flow-diagram of our approach is shown in Figure 1.

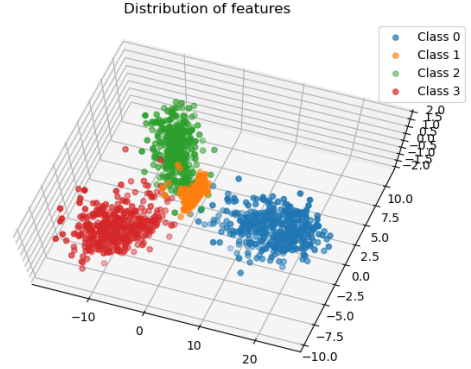


Fig. 2. Distribution of features in 3D space. The features for Class 1 have spread mainly in the Z-dimension.

Table 1. Feature dimensions and data-matrix ranks for Resnet18 trained on CIFAR10

Layer	Layer 0	Layer 1	Layer 2
Dimension	512	256	512
Rank	85	255	478
With 99.5% PCA	29	239	463

### 2.1. Global vs Per-Class Subspace Modeling

Subspace modeling can be performed either on the features from all classes of the training set at once, or on a per-class basis. In what follows, we refer to the former as global and the latter as per-class. Global modeling is appealing when the number of samples per class is small relative to the feature dimension, such as when there are a large number of classes but limited training data per class. It is also an option when class labels are not available (semi-supervised OOD). However, it may not adequately model the feature space’s underlying structure as there may be multiple well-separated clusters, each corresponding to a separate class. Modeling subspaces separately for each class can then often result in much better performance. This situation is clearly demonstrated in Figure 2 which shows the distribution of features from the penultimate layer of a simple CNN model trained with four classes.

### 2.2. Linear vs Non-Linear Subspace Modeling

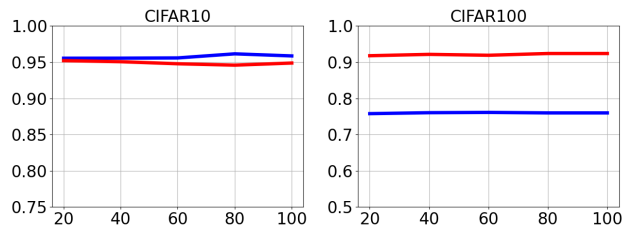
**Linear Subspace Modeling:** One popular choice for dimensionality reduction is principal component analysis (PCA) [17]. In this framework,  $\mathcal{H}$  and  $\mathcal{L}$  are, respectively, Euclidean spaces  $\mathbb{R}^d$  and  $\mathbb{R}^m$ , with  $m \ll d$ .  $\mathcal{T}$  can then be calculated from the singular value decomposition (SVD) [18] of the data matrix  $\mathbf{D}$ . Table 1 provides an intuition for the extent of dimensionality reduction achieved when applying PCA. For instance, we see that for Layer 0, the subspace dimension after applying PCA with 99.5% variability retention drops from 512 to 29, indicat-

**Table 2.** AUROC results for OOD detection performance.

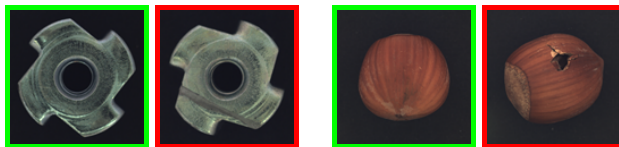
	Mahal	LL	FRE	kFRE	Mahal	LL	FRE	kFRE	
<b>CIFAR100</b>	<b>SVHN (OOD)</b>				<b>LSUN (OOD)</b>				
	Layer2	91.5	92.8	<b>93.1</b>	<b>93.4</b>	<b>98.5</b>	<b>98.8</b>	<b>98.3</b>	98.2
	Layer1	91.2	90.0	93.2	<b>95.8</b>	<b>98.7</b>	<b>99.1</b>	98.4	98.3
	Layer0	75.0	<b>84.8</b>	79.3	75.9	<b>97.3</b>	95.1	<b>97.1</b>	91.5
Softmax	74.3				84.7				
<b>CIFAR10</b>	<b>SVHN (OOD)</b>				<b>LSUN (OOD)</b>				
	Layer2	94.6	94.5	77.2	<b>98.5</b>	98.8	<b>99.4</b>	95.3	<b>99.0</b>
	Layer1	86.4	88.8	48.5	<b>92.4</b>	72.5	86.0	65.2	<b>87.0</b>
	Layer0	95.2	95.0	<b>96.7</b>	93.9	<b>95.1</b>	<b>95.5</b>	<b>95.3</b>	<b>95.1</b>
Softmax	93.4				94.0				
<b>SVHN</b>	<b>CIFAR10 (OOD)</b>				<b>LSUN (OOD)</b>				
	Layer2	<b>94.2</b>	<b>93.8</b>	85.2	<b>93.7</b>	<b>94.3</b>	<b>93.9</b>	90.1	93.5
	Layer1	90.4	<b>94.9</b>	94.2	<b>94.7</b>	90.6	<b>95.2</b>	94.5	<b>94.9</b>
	Layer0	92.3	<b>96.8</b>	96.0	95.6	92.5	<b>97.1</b>	96.0	95.8
Softmax	93.0				92.5				

ing that 99.5% of the information in the 512-dimensional features is actually contained within a 29-dimensional subspace! The results for other layers are less dramatic but nonetheless point to the need for appropriate subspace modeling. As the mapping from the high-dimensional feature space to the lower-dimensional reduced subspace is non-injective, there isn't a uniquely defined inverse image (pre-image) for a reduced feature i.e.  $\mathcal{T}^\dagger$  is not uniquely defined. In the linear case, a common practice is to use the Moore-Penrose pseudo-inverse of the forward transformation [18].

**Non-Linear Subspace Modeling:** Linear methods like PCA are most effective at subspace modeling if the underlying data is Gaussian, since PCA can only remove second-order dependencies [17]. However, the assumption of normality for the intermediate features of a DNN was justified only at the penultimate layer [9]. For other layers, the distribution of features could significantly depart from the normal distribution. In general, it will depend on the dataset used to induce the feature set, as well as the network topology. In such situations, modeling the data as living in a lower-dimensional sub-manifold can yield vastly improved outcomes. Here, we use kernel PCA (kPCA) [19] to model the underlying non-linear structure of the data. The choice of kPCA is motivated by the fact that it is a nonlinear extension of PCA, which allows for a direct comparison of performance between the linear and nonlinear OOD schemes. It is also computationally cheaper than other manifold learning methods such as Isomap [16] and locally linear embedding (LLE) [20], which can be described as kPCA on specially constructed Gram matrices [19]. In general, though, our approach can employ any nonlinear manifold learning technique that provides an explicit mapping function for new data points. For computation of the inverse transformation, we



**Fig. 3.** AUROC (Y-axis) for OOD detection with CIFAR10 and CIFAR100 as in-distribution datasets, and SVHN (red) and LSUN (blue) as OOD sets as we decrease the percentage of training data used (X-axis) for our OOD method all the way down to 20%. There is virtually no degradation in performance.



**Fig. 4.** Good (green) and defective (red) samples from the MVTEC dataset.

refer to [21] for a seminal paper on the topic.

### 3. EXPERIMENTS AND RESULTS

#### 3.1. Out-of-Distribution Detection

##### Experimental setup and evaluation metrics:

For the problem of OOD detection, we use CIFAR10, CIFAR100, and SVHN as the in-distribution datasets. To test across networks of various depths and complexities, we train SVHN on Resnet20 (0.27M parameters), CIFAR10 on Resnet18 (11.2M parameters), and CIFAR100 on Wide-Resnet (36.5M parameters). For models trained on CIFAR10 and CIFAR100, we use SVHN and a resized version of LSUN as OOD datasets; for those trained on SVHN, CIFAR10 and LSUN are used as OOD datasets. In all experiments, the subspace transformations are estimated from the training split of the in-distribution dataset, while performance metrics are calculated on the test splits. We apply both the linear (PCA) and nonlinear (kPCA with RBF kernel) subspace techniques to model the feature subspace on a per-class basis, with results reported separately in Table 2. We tested our method on three layers of each of the networks, with layers chosen to be located uniformly along the network path. The layers are labelled as 0, 1, and 2, with 0 being the (semantic) outermost layer, and 1, 2 being progressively deeper within the network.

During testing, the FRE (Eq. (1)) is used to distinguish between in-distribution and out-of-distribution data. This effectively creates a binary classifier, whose performance is characterized by the receiver operating characteristics (ROC) curve and we report the area under the ROC curve (AUROC)

**Table 3.** Anomaly Detection AUROC MVTec dataset

\* Methods not reporting class-itemized results. For PaDim, results were recreated using *anomalib*[22]

Category	GANomaly	DifferNet	SPADE*	PaDim*	PatchCore	FRE (Ours)
Carpet	69.9	92.9	-	99.5	98.7	100
Grid	70.8	84.0	-	94.2	98.2	95.8
Leather	84.2	97.1	-	100	100	100
Tile	79.4	99.4	-	97.4	98.7	97.8
Wood	83.4	99.8	-	99.3	99.2	99.4
Bottle	89.2	99.0	-	99.9	100	100
Cable	75.7	95.9	-	87.8	99.5	99.3
Capsule	73.2	86.9	-	92.7	98.1	99.4
Hazelnut	78.5	99.3	-	96.4	100	99.8
Metal Nut	70.0	96.1	-	98.9	100	96.9
Pill	74.3	88.8	-	93.9	96.6	95.7
Screw	74.6	96.3	-	84.5	98.1	97.5
Toothbrush	65.3	98.6	-	94.2	100	99.4
Transistor	79.2	91.1	-	97.6	100	98.6
Zipper	74.5	95.1	-	88.2	99.4	96.4
<b>Average</b>	<b>76.2</b>	<b>94.9</b>	<b>85.5</b>	<b>97.9</b>	<b>99.1</b>	<b>98.4</b>

**Table 4.** Anomaly Detection AUROC on Magnetic Tile

GANomaly	I-NN	DifferNet	PatchCore	FRE (Ours)
76.6	80.0	97.7	97.9	99.2

in Table 2. For baseline comparison, we use a binary OOD classifier based on Softmax scores (applicable only at the Softmax layer). At each layer tested, we also benchmark against a binary classifier based on Mahalanobis scores (denoted as Mahal). The Mahalanobis scores are the likelihoods of test samples w.r.t feature probability distributions fitted with multivariate Gaussian distributions sharing the same covariance across classes. In Table 2, FRE (resp. kFRE) refers to the feature reconstruction error with PCA (resp. kPCA) and LL corresponds to results obtained with [10].

**OOD detection results:** Table 2 shows that the proposed method is very competitive, often outperforming benchmarks that are much more computationally demanding and memory hungry, and typically within a half-percentage point of those. In particular, both benchmarks require density estimation and likelihood evaluation in high-dimensional spaces, while the proposed method relies on a few simple dot-product operations in the linear case. Of note, is the trend of the nonlinear scheme providing better results than its linear counterpart for layers deeper into the network. This is consistent with the earlier hypothesis that the outer-most layers produce features with a distribution close to Gaussian while deeper layers have feature spaces that may exhibit complex nonlinearity in their structure. **Robustness to reduced training data:** In practical situations,

the entire training dataset might not be available. We show that our method remains very effective at OOD detection even when its training (subspace modeling) is performed with a fraction of the training data. The plots in Figure 3 show the variations in AUROC for the CIFAR100 dataset, using both linear and non-linear subspace modeling, as the percentage for training data is gradually reduced to 20%. We see that the performance remains very stable, showing a decrease of less than 1% in AUROC scores in the majority of cases. We observe this trend for the other datasets as well.

### 3.2. Application to Anomaly Detection

Finally, we apply our method to the problem of anomaly detection in images. We learn a linear PCA transform,  $\mathcal{T}$ , on the features of an EfficientNet-B5 pretrained on Imagenet from only the defect-free images. We then use the feature-reconstruction score to distinguish between good and defective samples. We test our approach on two datasets: the MVTec anomaly detection dataset [23] (sample images shown in Figure 4), and the Magnetic Tile defect (MTD) dataset [24]. The results, comparing against various available benchmarks, are shown in Tables 3 and 4 respectively. Our method attains the best performance on MTD and a close second on MVTec, despite its simplicity. Our code for anomaly detection can be found at: <https://github.com/intellabs/dfm>

**Complexity:** Existing state-of-the-art methods involve significant complexity during training or inference. [11, 12] use deep generative models (GANs or normalizing flows) to model the distribution of normal samples, which require expensive training. While [14, 13] mitigate training complexity by using pretrained models, they both involve modeling clusters or probability distributions on a multi-level pyramid of deep-features. [15] reduces the memory storage requirements of feature-pyramid based methods but uses greedy coresets sampling on the stored feature banks to accomplish this, which is known to be NP-hard. By contrast, our method doesn't involve training a new model (discriminative or generative), operates on features from a single layer of the deep-network (instead of a feature pyramid), and doesn't involve any complex probabilistic modeling. It achieves state-of-the-art performance with remarkably low computational overhead, making it very attractive to deploy in real-world industrial usages.

## 4. CONCLUSION

This work sketched initial progress on anomaly and OOD detection with the application of linear and nonlinear dimensionality reduction on the semantic features of a DNN, prior to leveraging the *feature reconstruction error* as an uncertainty score. The method is simple, principled and very fast. Experimentations show qualitative performance at par or better than state-of-the-art methods that are significantly more complex.

## 5. REFERENCES

- [1] Dan Hendrycks and Kevin Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” in *International Conference on Learning Representations*, 2017.
- [2] Shiyu Liang, Yixuan Li, and R Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” in *International Conference on Learning Representations*, 2018.
- [3] Yarin Gal and Zoubin Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *international conference on machine learning*, 2016, pp. 1050–1059.
- [4] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in neural information processing systems*, 2017, pp. 6402–6413.
- [5] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al., “Conditional image generation with pixelcnn decoders,” in *Advances in neural information processing systems*, 2016, pp. 4790–4798.
- [6] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich, “Deep anomaly detection with outlier exposure,” *Proceedings of the International Conference on Learning Representations*, 2019.
- [7] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan, “Likelihood ratios for out-of-distribution detection,” in *Advances in Neural Information Processing Systems*, 2019, pp. 14707–14718.
- [8] G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, “Novelty detection through model-based characterization of neural networks,” in *IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3179–3183.
- [9] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *Advances in Neural Information Processing Systems*, 2018.
- [10] Nilesh A. Ahuja, Ibrahima J. Ndiour, Trushant Kalyanpur, and Omesh Tickoo, “Probabilistic modeling of deep features for out-of-distribution and adversarial detection,” in *Bayesian Deep Learning workshop, NeurIPS*, 2019.
- [11] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn, “Same same but different: Semi-supervised defect detection with normalizing flows,” in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1906–1915.
- [12] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon, “Ganomaly: Semi-supervised anomaly detection via adversarial training,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 622–637.
- [13] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier, “Padim: a patch distribution modeling framework for anomaly detection and localization,” in *International Conference on Pattern Recognition*. Springer, 2021, pp. 475–489.
- [14] Niv Cohen and Yedid Hoshen, “Sub-image anomaly detection with deep pyramid correspondences,” *arXiv preprint arXiv:2005.02357*, 2020.
- [15] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler, “Towards total recall in industrial anomaly detection,” *arXiv preprint arXiv:2106.08265*, 2021.
- [16] Joshua B Tenenbaum, Vin De Silva, and John C Langford, “A global geometric framework for nonlinear dimensionality reduction,” *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [17] Jonathon Shlens, “A tutorial on principal component analysis,” *arXiv preprint arXiv:1404.1100*, 2014.
- [18] Gene H Golub and Charles F Van Loan, *Matrix computations*, JHU press, 2013.
- [19] Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf, “A kernel view of the dimensionality reduction of manifolds,” in *Proceedings of the 21st international conference on Machine learning*, 2004.
- [20] Sam T Roweis and Lawrence K Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [21] S. Mika, B. Schölkopf, AJ. Smola, K-R. Müller, M. Scholz, and G. Rätsch, “Kernel pca and de-noising in feature spaces,” in *Advances in Neural Information Processing Systems 11*, Cambridge, MA, USA, June 1999, Max-Planck-Gesellschaft, pp. 536–542, MIT Press.
- [22] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc, “Anomalib: A deep learning library for anomaly detection,” 2022.
- [23] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger, “Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9592–9600.
- [24] Yibin Huang, Congying Qiu, and Kui Yuan, “Surface defect saliency of magnetic tile,” *The Visual Computer*, vol. 36, no. 1, pp. 85–96, 2020.