Evaluating State-of-the-art Methods for Industrial Anomaly Detection

Midterm Presentation

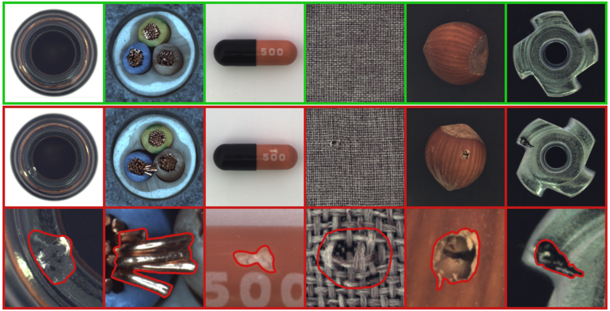
Table of Contents

Intro
- What is an anomaly
- Anomaly Detection (AD)
- Real-world application domains
- Motivation

Method
- SoTA Topics
- Dataset
- Models
- Architecture
- Sample codes

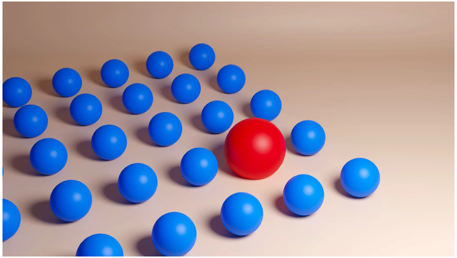
Evaluation
- Metrics
- Visualization
- Benchmark

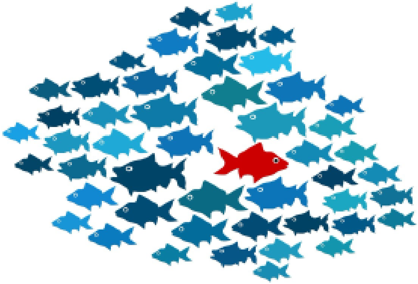
Conclusion
- Milestone Achievement
- Future Work



What is an anomaly

- something different, abnormal compared to the majority of data points
- deviate from the common patterns
- unbounded, unknown events
- ...





Anomaly Detection (AD)

TECHNISCHE UNIVERSITÄT DARMSTADT

Anomaly detection, or outlier detection, is the identification of observations, events, or data points that deviate from what is usual, standard, or expected, making them inconsistent with the rest of a data set [1].

Real-world application domains

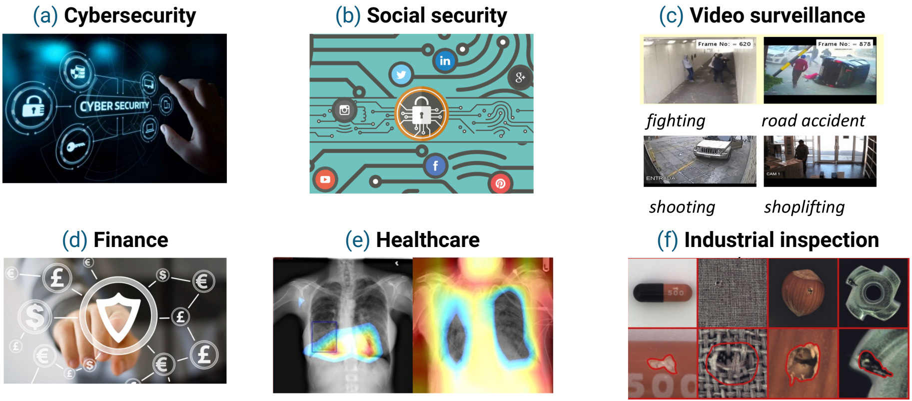


Figure: Real-world application domains[2]

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Lack of a unified benchmark for industrial anomaly detection
- Existing frameworks are difficult to configure
- Cross-platform support is usually lacking
- Limited methods for metrics evaluation
- Visualization is required for benchmarking

- **Visualize Data**: Shows data distribution in feature space.
- **Statistical Basis**: Uses PDFs or similar methods.
- **Likelihood**: Indicates likelihood of data values.
- **Identify Anomalies**: Compare new points to known distribution.
- **Set Thresholds**: Determine anomalies based on thresholds.

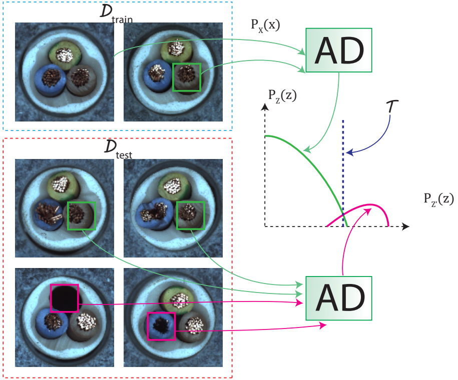


Figure: Distribution Map[3]

- **Assumption**: Normal data can be accurately reconstructed; anomalies cannot.
- **Training**: Use normal data to train a model (e.g., autoencoder).
- **Encoding**: Model learns to map input data to a low-dimensional representation.
- **Testing**: Model attempts to reconstruct new data points.
- **Anomaly Detection**: Large reconstruction error indicates a probable anomaly.

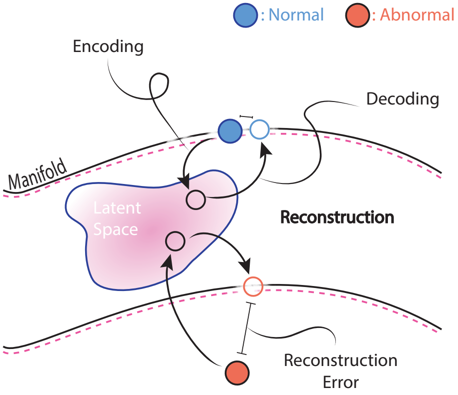


Figure: Reconstruction based[4]

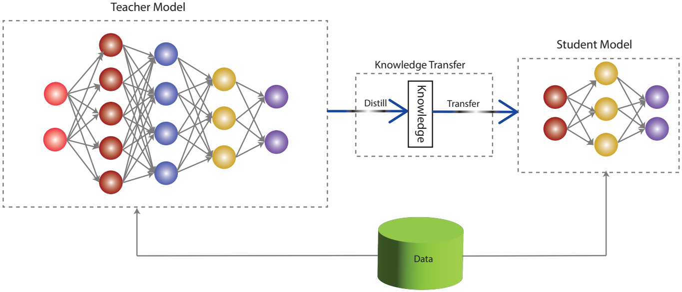


Figure: Teacher Student[5]

- **Definition**: Transferring knowledge from a large model to a smaller one.
- **Importance**: Addresses computational inefficiency of large models.
- **Process**: Small student model learns from larger teacher model.
- **Objective**: Student model achieves comparable or enhanced accuracy using teacher's knowledge.

- One-class classification (OCC) models a single class, often the normal or inlier class.
- Anomaly detection (AD) aims to identify normal instances accurately.
- Anomalies are outliers that significantly deviate from normal characteristics.
- OCC methods detect anomalies by comparing distribution or behavior to the normal class.

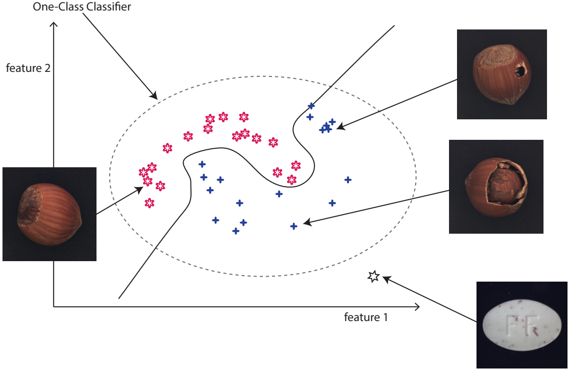


Figure: One Class Classification[6]

Dataset

Comparison of Datasets

Datasets	Sample Number	Categories
MVTec [3]	5354	15
MVTec3D [7]	>4000	10
Btech [8]	2830	3
Kolektor [9]	399	1
visa [10]	10821	12

Table: Comparison of Datasets

Models

Evaluated Models according to Topics

Topics	Models
Distribution Map	CFLOW [11], CSFlow [12], DFKDE [13], DFM [14], FastFlow [15], RKDE [16]
Reconstruction-based	DRAEM [17], DSR [18], GANomaly [19]
Memory Bank	PaDiM [20], PatchCore [21], CFA [22]
Teacher Student	Reverse Distillation [23], STFPM [24]
One Class Classification	UFLOW [25]

Table: Evaluated Models

Models

Evaluated Models according to Tasktype

Tasktype	Models
Segmentation	CFLOW [11], CSFlow [12], DFM [14], FastFlow [15], DRAEM [17], DSR [18], GANomaly [19], PaDiM [20], PatchCore [21], cfa [22], Reverse Distillation [23], STFPM [24], UFLOW [25]
Detection	RKDE [16]
Classification	DFKDE [13]

Table: Task Types

Architecture
Workflow

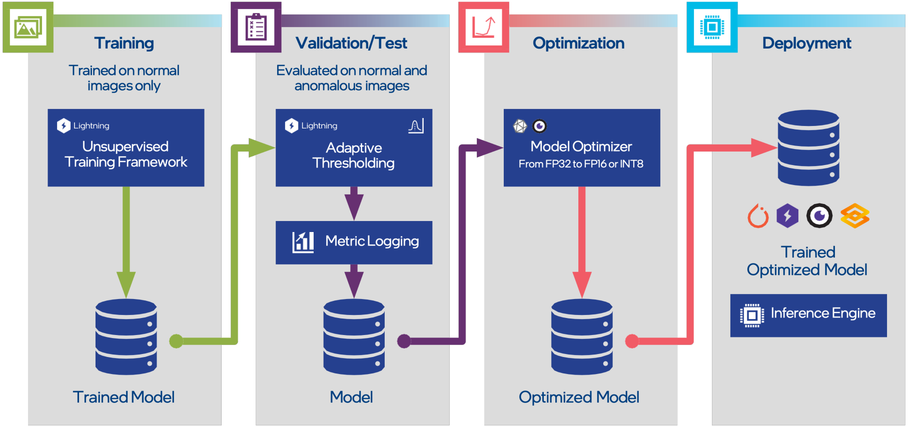


Figure: Workflow[26]

Architecture
Overview of Architecture

TECHNISCHE UNIVERSITÄT DARMSTADT

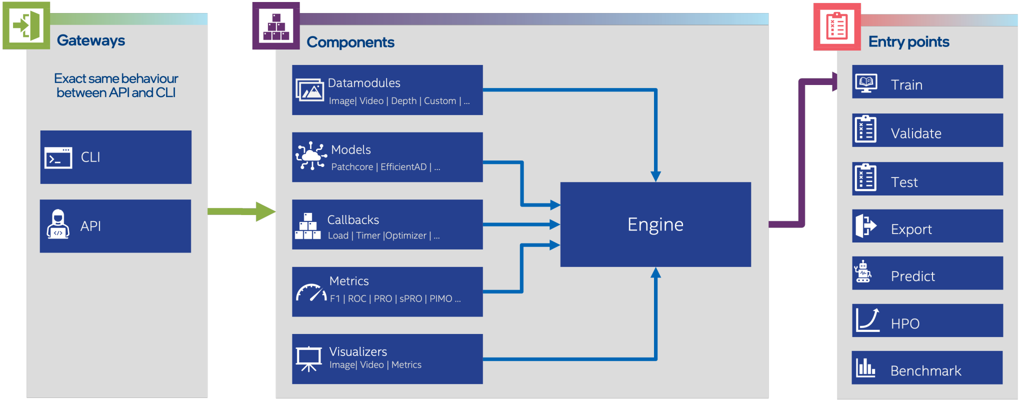


Figure: Overview of Architecture[26]

Sample Code

Training and Testing using API

```
datasets = ['screw', 'pill', 'capsule', 'carpet', 'grid', 'tile', 'wood', 'zipper',
            'cable', 'toothbrush', 'transistor', 'metal_nut', 'bottle', 'hazelnut',
            'leather']

for dataset in datasets:
    model = Padim()
    datamodule = MVTec(category=dataset, num_workers=0, train_batch_size=256,
                       eval_batch_size=256)
    engine = Engine(pixel_metrics=["AUROC", "PRO"], image_metrics=["AUROC", "PRO"],
    task=TaskType.SEGMENTATION)

    engine.fit(model=model, datamodule=datamodule)

    test_results = engine.test(
        model=model,
        datamodule=datamodule,
        ckpt_path=engine.trainer.checkpoint_callback.best_model_path,
    )
```

Sample Code

Training and Testing using bash

```
datasets=('screw' 'pill' 'capsule' 'carpet' 'grid' 'tile' 'wood' 'zipper' 'cable'
'toothbrush' 'transistor' 'metal_nut' 'bottle' 'hazelnut' 'leather')
config_file="./configs/models/padim.yaml"

for dataset in "${datasets[@]}"
do
    command = anomalib train --data anomalib.data.MVTec --data.category $dataset --
        config $config_file
    # Execute command
    $command
done
```

Metrics

Metrics	Formular	Usage
Precision (P)	$P = TP/(TP + FP)$	True Positive (TP), False Positive (FP)
Recall (R)	$R = TP/(TP + FN)$	False Negative (FN)
True Positive Rate (TPR)	$TPR = TP/(TP + FN)$	Classification
False Positive Rate (FPR)	$FPR = FP/(TN + FP)$	True Negative (TN)
AUROC	$\int_0^1 (TPR) d(FPR)$	Classification
PRO	$\frac{1}{N} \sum_i \sum_k \frac{P_i \cap C_{i,k}}{C_{i,k}}$	Total ground-truth number (N), Predicted abnormal pixels (P), Defect ground-truth regions (C), Segmentation
AUPR	$\int_0^1 (P) d(R)$	Localization, Segmentation
Forgetting Measure (FM) [27]	$FM_j^k = \max_{l \in \{1, \ldots k-1\}} T_{l,j} - T_{k,j}$	Task (T), Number of tasks (k), Task to be evaluated (j)

Table: Used Metrics

Visualization

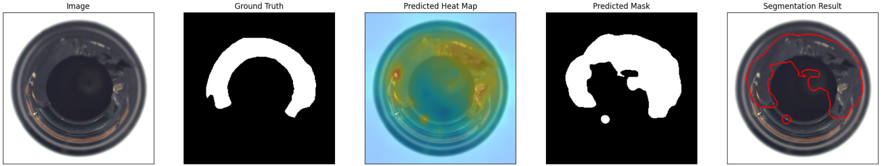


Figure: MVTec bottle broken large

Benchmark

Image Level AUROC

Method	Screw	Pill	Capsule	Carpet	Grid	Tile	Wood	Zipper	Cable	Toothbrush	Transistor	Metal Nut	Bottle	Hazelnut	Leather
CFA	87.68	93.10	48.17	50.00	84.04	37.50	62.50	40.34	61.96	**98.06**	45.00	39.78	50.00	60.00	50.00
CFLOW	81.57	92.91	95.45	97.27	88.47	**100.00**	**99.39**	97.64	94.15	94.17	94.88	**100.00**	**100.00**	**99.68**	**100.00**
CSFLOW	41.58	38.09	60.77	**99.16**	78.36	90.44	95.75	88.93	67.31	44.72	50.54	74.46	89.92	71.02	**100.00**
DFKDE	69.17	64.18	72.04	68.42	46.69	92.50	81.32	88.26	68.67	78.89	81.12	76.20	92.94	77.57	77.79
DFM	77.79	96.89	92.98	90.93	65.41	98.67	97.81	97.51	94.27	96.39	94.79	91.72	**100.00**	96.82	**100.00**
DRAEM	30.01	74.55	77.02	72.47	80.45	86.80	95.96	78.90	63.98	70.42	90.21	93.55	**98.10**	77.50	86.65
DSR	56.67	70.16	72.72	43.82	**97.08**	80.70	90.75	78.97	76.96	**96.94**	91.04	81.04	86.59	81.64	83.97
FASTFLOW	65.85	76.60	69.64	93.38	**96.32**	93.36	98.16	72.69	67.62	72.50	89.62	81.33	**99.68**	79.86	**99.90**
GANomaly	32.79	59.96	26.57	21.71	57.23	54.42	60.88	41.05	52.47	49.17	33.46	26.30	47.78	53.86	36.68
PaDiM	78.95	79.95	86.48	**97.99**	87.47	94.55	97.46	77.46	85.96	82.50	94.54	**98.34**	**99.52**	88.32	**100.00**
PatchCore	**98.11**	94.76	97.85	**99.12**	**98.08**	98.81	98.77	**99.21**	**99.10**	**100.00**	**100.00**	**99.80**	**100.00**	**100.00**	**100.00**
RD	**98.03**	97.63	97.93	**99.36**	95.49	**100.00**	**99.39**	97.16	95.45	91.39	97.87	**100.00**	**100.00**	**100.00**	**100.00**
RKDE	50.58	68.77	51.93	-	75.36	67.72	62.54	75.37	85.83	77.17	65.00	90.63	85.07	**100.00**	**100.00**
STFPM	77.62	40.78	60.59	**98.88**	59.23	**97.08**	**98.95**	75.39	**91.34**	47.50	61.88	40.22	43.97	96.50	**100.00**
UFLOW	49.19	94.84	56.72	**100.00**	**99.33**	**99.39**	95.09	89.73	62.67	64.17	81.46	55.77	**99.21**	90.39	**100.00**

Table: Image Level AUROC Scores for MVTec[3] dataset categories

Benchmark
Pixel Level AUROC

Method	Screw	Pill	Capsule	Carpet	Grid	Tile	Wood	Zipper	Cable	Toothbrush	Transistor	Metal Nut	Bottle	Hazelnut	Leather
CFA	**98.54**	**98.21**	14.96	22.27	**97.30**	31.83	35.07	23.30	24.63	**98.58**	26.06	35.98	24.52	17.44	29.79
CFLOW	**97.40**	**97.48**	**98.84**	**99.04**	**97.38**	96.09	95.02	**97.45**	96.06	**98.30**	87.90	**96.65**	**98.58**	**98.72**	**99.54**
CSFLOW	67.61	63.83	79.20	70.34	59.42	67.07	64.05	59.02	64.62	70.36	52.93	67.12	69.03	73.10	66.89
DFKDE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DFM	90.41	**96.91**	94.24	95.51	83.43	91.75	84.55	89.11	96.06	95.95	**97.79**	94.73	93.88	97.14	95.43
DRAEM	65.04	62.63	72.93	55.56	42.77	77.31	70.12	65.49	44.90	69.65	57.17	73.83	74.22	67.57	52.11
DSR	62.75	43.83	56.51	54.86	68.48	71.69	72.80	60.69	58.96	79.58	57.58	55.36	65.17	73.15	65.55
FASTFLOW	89.65	92.15	93.43	**97.65**	**97.27**	88.51	94.53	88.90	70.57	90.58	93.54	84.14	**97.05**	93.78	**99.56**
GANomaly	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PaDiM	**97.85**	95.14	**98.46**	**98.99**	94.18	92.74	92.95	**98.20**	96.98	**98.67**	97.56	**96.57**	**98.42**	98.02	**99.02**
PatchCore	**98.91**	97.67	**98.76**	**98.87**	**97.96**	94.75	92.94	**97.94**	**98.01**	**98.64**	97.16	**98.22**	**98.16**	**98.45**	**98.99**
RD	**99.52**	97.21	**98.87**	**99.05**	**99.06**	95.08	94.82	**98.44**	96.72	**98.85**	90.44	**96.85**	**98.59**	**98.78**	**99.28**
RKDE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
STFPM	22.10	55.29	61.46	**98.61**	78.55	86.76	91.93	88.39	94.38	64.33	62.27	68.92	66.13	97.09	**97.90**
UFLOW	85.49	**97.79**	82.39	**99.40**	**97.86**	**96.04**	92.04	**96.69**	78.14	70.14	86.52	77.19	**97.98**	**98.71**	**99.37**

Table: Pixel Level AUROC Scores for MVTec [3] dataset categories

Benchmark

Pixel Level PRO

Method	Screw	Pill	Capsule	Carpet	Grid	Tile	Wood	Zipper	Cable	Toothbrush	Transistor	Metal Nut	Bottle	Hazelnut	Leather
CFA	40.24	75.15	**100.00**	**100.00**	49.28	**99.99**	**100.00**	**100.00**	99.01	33.04	**100.00**	**100.00**	**100.00**	**100.00**	**99.99**
CFLOW	26.35	71.89	36.84	82.32	47.11	86.31	81.82	67.94	53.86	32.35	57.84	90.76	84.26	82.56	77.02
CSFLOW	3.40	73.62	19.70	32.76	43.65	60.71	47.55	69.04	34.64	19.96	23.17	61.62	28.38	56.45	77.36
DFKDE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DFM	10.06	7.48	16.21	24.63	12.65	72.37	44.84	50.03	30.50	13.54	36.63	38.99	67.90	50.28	25.14
DRAEM	11.76	43.98	36.83	13.35	4.37	37.52	23.73	18.33	15.26	19.60	32.27	62.74	42.19	56.31	10.19
DSR	15.21	31.27	18.63	8.89	29.96	22.24	37.75	18.75	20.65	21.06	25.73	54.51	32.74	43.73	20.00
FASTFLOW	22.78	65.64	43.42	68.76	52.59	60.29	83.90	48.77	45.20	26.01	55.02	74.10	79.29	72.87	83.67
GANomaly	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PaDiM	30.00	80.57	33.36	75.64	41.35	80.18	83.89	70.17	74.36	55.86	72.39	91.40	83.55	69.49	90.14
PatchCore	39.05	73.99	42.25	66.39	52.92	75.70	61.65	70.79	66.17	21.35	90.73	79.43	81.62	80.82	59.28
RD	48.35	82.31	49.69	70.49	56.88	87.72	70.25	73.66	60.30	27.36	66.67	91.35	82.73	82.93	68.25
RKDE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
STFPM	1.57	57.32	23.66	73.19	22.65	64.49	74.15	36.58	60.27	24.96	12.37	92.29	84.00	66.46	69.99
UFLOW	2.03	69.63	11.97	79.76	41.02	86.69	73.60	60.55	16.37	22.36	50.78	66.00	71.60	67.67	70.85

Table: Pixel Level PRO Scores for MVTec[3] dataset categories

Milestone Achievement

- Metrics: Image Level/Pixel Level PRO, AUROC ✅
- Evaluation of Performance of 15 popular models on MVTec dataset ✅
- Training using API/Bash ✅
- Testing using API/Bash ✅
- Docker and Conda Configuration ✅
- Cross Plattform: Linux (Ubuntu22/20), Windows (Win11/10) ✅

Future Work

- Implemetation of evaluation metrics (AUPR, FM)
- Inference of target data
- Support of video dataset and custom dataset
- Visualization and statistics
- Employment of multiple Docker container
- Integration to huggingface

[1] IBM. *Anomaly Detection*. https://www.ibm.com/topics/anomaly-detection. Accessed: 2024-06-14.

[2] CVPR 2023 Tutorial Organizers. *CVPR 2023 Tutorial on Anomaly Detection*. https://sites.google.com/view/cvpr2023-tutorial-on-ad/. Accessed: 2024-06-14. 2023.

[3] Paul Bergmann et al. "MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019. URL: https://openaccess.thecvf.com/content_CVPR_2019/html/Bergmann_MVTec_AD_--_A_Comprehensive_Real-World_Dataset_for_Unsupervised_Anomaly_CVPR_2019_paper.html.

[4] Woosang Shin et al. "Anomaly Detection using Score-based Perturbation Resilience". en. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 23315–23325. ISBN: 9798350307184. DOI: 10.1109/ICCV51070.2023.02136. URL: https://ieeexplore.ieee.org/document/10376577/ (visited on 03/01/2024).

[5] S. Amit. *Everything You Need to Know About Knowledge Distillation (aka Teacher-Student Model)*. https://amit-s.medium.com/everything-you-need-to-know-about-knowledge-distillation-aka-teacher-student-model-d6ee10fe7276. Accessed: 2024-06-18. 2023.

[6] David Tax. *One-Class Classification*. https://www.tudelft.nl/ewi/over-de-faculteit/afdelingen/intelligent-systems/pattern-recognition-bioinformatics/pattern-recognition-bioinformatics/people/david-tax/one-class-classification. Accessed: 2024-06-18.

[7] Paul Bergmann et al. "The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization". In: *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2022. DOI: 10.5220/0010865000003124. URL: http://dx.doi.org/10.5220/0010865000003124.

[8] Pankaj Mishra et al. "VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization". In: *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. 2021, pp. 01–06. DOI: 10.1109/ISIE45552.2021.9576231.

[9] Domen Tabernik et al. "Segmentation-Based Deep-Learning Approach for Surface-Defect Detection". In: *Journal of Intelligent Manufacturing* (May 2019). ISSN: 1572-8145. DOI: 10.1007/s10845-019-01476-x. URL: https://link.springer.com/article/10.1007/s10845-019-01476-x.

[10] Yang Zou et al. “SPot-the-Difference Self-supervised Pre-training for Anomaly Detection and Segmentation”. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 392–408. ISBN: 978-3-031-20056-4. URL: https://link.springer.com/chapter/10.1007/978-3-031-20056-4_23.

[11] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. “CFLOW-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2022, pp. 98–107. URL: https://openaccess.thecvf.com/content/WACV2022/html/Gudovskiy_CFLOW-AD_Real-Time_Unsupervised_Anomaly_Detection_With_Localization_via_Conditional_Normalizing_WACV_2022_paper.html.

[12] Marco Rudolph et al. “Fully Convolutional Cross-Scale-Flows for Image-Based Defect Detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2022, pp. 1088–1097. URL: https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=Fully+Convolutional+Cross-Scale-Flows+for+Image-Based+Defect+Detection&btnG=.

[13] Samet Akcay et al. “Anomalib: A deep learning library for anomaly detection”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 1706–1710. URL: https://ieeexplore.ieee.org/abstract/document/9897283.

[14] Nilesh A. Ahuja et al. *Probabilistic Modeling of Deep Features for Out-of-Distribution and Adversarial Detection*. 2019. arXiv: 1909.11786 [stat.ML].

[15] Jiawei Yu et al. *FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows*. 2021. arXiv: 2111.07677 [cs.CV].

[16] Philip Adey et al. “Region Based Anomaly Detection with Real-Time Training and Analysis”. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pp. 495–499. DOI: 10.1109/ICMLA.2019.00092.

[17] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. “DRAEM - A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 8330–8339. URL: https://openaccess.thecvf.com/content/ICCV2021/html/Zavrtanik_DRAEM_-_A_Discriminatively_Trained_Reconstruction_Embedding_for_Surface_Anomaly_ICCV_2021_paper.html.

[18] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. "DSR – A Dual Subspace Re-Projection Network for Surface Anomaly Detection". In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 539–554. ISBN: 978-3-031-19821-2. URL: https://link.springer.com/chapter/10.1007/978-3-031-19821-2_31.

[19] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training". In: *Computer Vision – ACCV 2018*. Ed. by C. V. Jawahar et al. Cham: Springer International Publishing, 2019, pp. 622–637. ISBN: 978-3-030-20893-6. URL: https://link.springer.com/chapter/10.1007/978-3-030-20893-6_39.

[20] Thomas Defard et al. "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization". In: *Pattern Recognition. ICPR International Workshops and Challenges*. Ed. by Alberto Del Bimbo et al. Cham: Springer International Publishing, 2021, pp. 475–489. ISBN: 978-3-030-68799-1. URL: https://link.springer.com/chapter/10.1007/978-3-030-68799-1_35.

[21] Karsten Roth et al. *Towards Total Recall in Industrial Anomaly Detection*. 2021. arXiv: 2106.08265 [cs.CV].

[22] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. "CFA: Coupled-Hypersphere-Based Feature Adaptation for Target-Oriented Anomaly Localization". In: *IEEE Access* 10 (2022), pp. 78446–78454. DOI: 10.1109/ACCESS.2022.3193699.

[23] Hanqiu Deng and Xingyu Li. "Anomaly Detection via Reverse Distillation From One-Class Embedding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 9737–9746. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Deng_Anomaly_Detection_via_Reverse_Distillation_From_One-Class_Embedding_CVPR_2022_paper.html.

[24] Guodong Wang et al. *Student-Teacher Feature Pyramid Matching for Anomaly Detection*. 2021. arXiv: 2103.04257 [cs.CV].

[25] Matías Tailanian, Álvaro Pardo, and Pablo Musé. *U-Flow: A U-shaped Normalizing Flow for Anomaly Detection with Unsupervised Threshold*. 2023. arXiv: 2211.12353 [cs.CV].

[26] Anomalib Developers. *SDD: Developer Guide*. https://anomalib.readthedocs.io/en/latest/markdown/guides/developer/sdd.html. Accessed: 2024-06-19. 2024.

[27] Guoyang Xie et al. *IM-IAD: Industrial Image Anomaly Detection Benchmark in Manufacturing*. 2024. arXiv: 2301.13359 [cs.CV]. URL: https://ieeexplore.ieee.org/abstract/document/10443076.

Appendix
Explanation of Abbreviation

shortcut	explanation
AUROC	Area under ROC Curve
PRO	Per Region Overlap
AUPR	Aarea under the PR curve
CFA	Coupled Hypersphere-based Feature Adaptation
CFLOW	Conditional Normalizing Flows
CSFLOW	Cross-Scale-Flows
DFKDE	Deep Feature Kernel Density Estimation
DFM	Probabilistic Modeling of Deep Features for Out-of-Distribution and Adversarial Detection
DRAEM	A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection
DSR	Dual Subspace Re-Projection
FastFlow	Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows
GANomaly	Semi-Supervised Anomaly Detection via Adversarial Training
PaDiM	A Patch Distribution Modeling
RD	Reverse Distillation
RKDE	Region-Based Kernel Density Estimation
STFPM	Student-Teacher Feature Pyramid Matching
UFLOW	U-shaped Normalizing Flow

Appendix

CFA Parameters Settings

Configuration Parameter	Value	Description
backbone	wide_resnet50_2	Backbone CNN network
gamma_c	1	gamma_c value from the paper
gamma_d	1	gamma_d value from the paper
num_nearest_neighbors	3	Number of nearest neighbors
num_hard_negative_features	3	Number of hard negative features
max_epochs	30	Maximum number of training epochs
radius	1.0e-05	Radius of the hypersphere to search the soft boundary
patience	5	Number of checks with no improvement
mode	max	One of 'min', 'max'. In 'min' mode, training will stop when the quantity monitored has stopped decreasing and in 'max' mode it will stop when the quantity monitored has stopped increasing.

Table: CFA Parameters Settings

Appendix
PaDiM Configuration Parameters

Configuration Parameter	Value	Description
layers	layer1, layer2, layer3	Layers used for feature extraction
backbone	resnet18	Pre-trained model backbone
pre_trained	true	Boolean to check whether to use a pre_trained backbone
n_features	null	Number of features to retain in the dimension reduction step. Default values from the paper are available for: resnet18 (100), wide_resnet50_2 (550). Defaults to “None“.

Table: PaDiM Configuration Parameters

Appendix
CSFLOW Parameters Settings

Configuration Parameter	Value	Description
cross_conv_hidden_channels	1024	Number of hidden channels in the cross convolution
n_coupling_blocks	4	Number of coupling blocks in the model
clamp	3	Clamp value for glow layer
num_channels	3	Number of channels in the model
max_epochs	240	Maximum number of training epochs
patience	3	Number of checks with no improvement

Table: CSFLOW Parameters Settings

Appendix
PatchCore Parameters Settings

Configuration Parameter	Value	Description
backbone	wide_resnet50_2	Backbone CNN network
layers	layer2, layer3	Layers to extract features from the backbone CNN
pre_trained	true	Boolean to check whether to use a pre_trained backbone
coreset_sampling_ratio	0.1	Coreset sampling ratio to subsample embedding
num_neighbors	9	Number of nearest neighbors

Table: PatchCore Parameters Settings

Appendix

Reverse Distillation Parameters Settings

Configuration Parameter	Value	Description
backbone	wide_resnet50_2	Backbone of CNN network
layers	layer1, layer2, layer3	Layers to extract features from the backbone CNN
anomaly_map_mode	ADD	Mode to generate anomaly map
pre_trained	true	Boolean to check whether to use a pre_trained backbone
patience	3	Number of checks with no improvement
check_val_every_n_epoch	200	Frequency of validation checking (in epochs)
max_epochs	200	Maximum number of training epochs

Table: Reverse Distillation Parameters Settings