# Supplemental Materials
# Inter-Realization Channels: Unsupervised Anomaly Detection Beyond One-Class Classification

## A. Additional Hyperparamter Influences

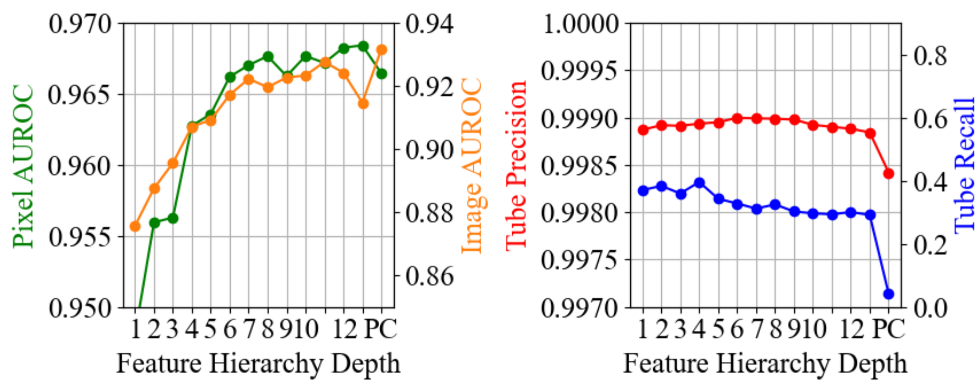


Figure 1. Segmentation AUROC, Image AUROC, InReaCh Precision, and InReaCh Recall vs. Feature Hierarchy depth. We also include PatchCore's feature hierarchy selection at the value of PC [9]. Testing is done with 10 corruptions in the dataset for each class.

**Feature Hierarchy Depth:** We evaluated feature hierarchy depths of Wide-ResNet-50 by considering the outputs of only the first 13 residual blocks, as higher depths increased training time [13], including the shallowest features cumulatively reduced bias toward the pre-training dataset (ImageNet [5]). Our search showed that, with diminishing returns, including additional feature hierarchies, performed better. We also evaluated our method using the deeper feature hierarchies of PatchCore and noted greater per-image performance but decreased localization performance, InReaCh precision, and recall [9]. In general, deeper feature hierarchies reduced the recall of InReaCh, with little relative effect on the precision. The significant decrease in precision and recall of InReaCh, in conjunction with the increased expected bias towards ImageNet classes of these deeper features, validates the selections of lower-level feature hierarchies.

## B. Full Image-Wise Detection Results

InReaCh solves multiple classes fully with 1.0 AUROC, such as Bottle and Leather, with full results in Table 1. Notably, the toothbrush class, which has the smallest dataset of only 60 training images, significantly improves from 0.642 image-wise AUROC with no data corruptions to 0.961 AUROC with 40 dataset corruptions. This improvement appears to come from the increase of training data by 67% by introducing the corrupt samples. Notably, performance for the screw class is much lower than all other classes for image-wise AUROC but similar for pixel-wise. In investigating the scoring, it was observed that the screw head slot was being scored highly as an anomaly in all images, nominal or not, but this was a very small region in the overall image. This seems to be because the screw is randomly rotated around its length, making the view of the screw head slot very heterogeneous. Because of this, these patches were filtered InReaCh and, therefore, never incorporated into the nominal model. We expect this issue will be resolved by incorporating more data to allow this to be better represented by the dataset, similar to the behaviour noted in the toothbrush class, or by tuning the $\sigma$ and span filter values for this specific class.

| Corrupt Images* | 0 | 0 | 0 | 0 | 0 | 0 | **10** | **40** |
|---|---|---|---|---|---|---|---|---|
| **Class↓ Method →** | SPADE [2] | Cut-Paste [6] | PatchSVDD [12] | PaDiM [3] | PatchCore [9] | **InReaCh** | **InReaCh** | **InReaCh** |
| Bottle | - | 0.992 | 0.986 | - | 1.000 | 0.998 | 1.000 | 1.000 |
| Cable | - | 0.871 | 0.903 | - | 0.995 | 0.922 | 0.939 | 0.922 |
| Capsule | - | 0.879 | 0.767 | - | 0.981 | 0.865 | 0.835 | 0.822 |
| Carpet | - | 0.679 | 0.929 | - | 0.987 | 0.992 | 0.995 | 0.992 |
| Grid | - | 0.999 | 0.946 | - | 0.982 | 0.954 | 0.971 | 0.963 |
| Hazelnut | - | 0.913 | 0.920 | - | 1.000 | 0.996 | 0.988 | 0.990 |
| Leather | - | 0.997 | 0.909 | - | 1.000 | 1.000 | 1.000 | 1.000 |
| Metal Nut | - | 0.968 | 0.940 | - | 1.000 | 0.948 | 0.942 | 0.911 |
| Pill | - | 0.934 | 0.861 | - | 0.966 | 0.827 | 0.844 | 0.807 |
| Screw | - | 0.544 | 0.813 | - | 0.981 | 0.610 | 0.634 | 0.621 |
| Tile | - | 0.959 | 0.978 | - | 0.987 | 0.989 | 0.986 | 0.990 |
| Toothbrush | - | 0.948 | 1.000 | - | 1.000 | 0.642 | 0.917 | 0.961 |
| Transistor | - | 0.964 | 0.915 | - | 1.000 | 0.882 | 0.871 | 0.865 |
| Wood | - | 0.949 | 0.965 | - | 0.992 | 0.942 | 0.959 | 0.918 |
| Zipper | - | 0.994 | 0.979 | - | 0.994 | 0.975 | 0.970 | 0.951 |
| **Average** | 0.855 | 0.909 | 0.921 | 0.953 | 0.991 | 0.903 | 0.923 | 0.914 |
| **Supervision*** | OCC | OCC | OCC | OCC | OCC | **None** | **None** | **None** |

Table 1. MvTec AD Anomaly Detection performance in image-wise AUROC [1]. *Supervision is given as OCC (one-class classification) or None (Unsupervised). *Corrupt Images are the number of anomalous images randomly added to the training set from the test set to show the unsupervised filtering abilities of InReaCh.

# C. Alternative Anomaly Scoring Methods

| | Ablation Results | | | | Proposed InReaCh | Alternative Anomaly Scoring | |
|---|---|---|---|---|---|---|---|
| **Class↓ Configuration* →** | Base InReaCh | Local Agg. | Span Filte | $\sigma$ Filter | Pos. Embedding | Mahalanobis [7, 3] | PatchCore [9] |
| Bottle | 0.977 | 0.977 | 0.977 | 0.981 | 0.981 | 0.636 | 0.981 |
| Cable | 0.905 | 0.905 | 0.905 | 0.934 | 0.962 | 0.615 | 0.961 |
| Capsule | 0.977 | 0.977 | 0.977 | 0.981 | 0.977 | 0.270 | 0.976 |
| Carpet | 0.992 | 0.992 | 0.992 | 0.993 | 0.993 | 0.252 | 0.993 |
| Grid | 0.984 | 0.984 | 0.984 | 0.982 | 0.982 | 0.883 | 0.981 |
| Hazelnut | 0.972 | 0.972 | 0.972 | 0.973 | 0.973 | 0.613 | 0.972 |
| Leather | 0.992 | 0.992 | 0.992 | 0.992 | 0.992 | 0.231 | 0.992 |
| Metal Nut | 0.969 | 0.969 | 0.969 | 0.977 | 0.977 | 0.544 | 0.976 |
| Pill | 0.908 | 0.908 | 0.908 | 0.942 | 0.937 | 0.430 | 0.935 |
| Screw | 0.958 | 0.958 | 0.958 | 0.959 | 0.957 | 0.552 | 0.955 |
| Tile | 0.953 | 0.953 | 0.953 | 0.960 | 0.960 | 0.396 | 0.959 |
| Toothbrush | 0.974 | 0.974 | 0.974 | 0.986 | 0.984 | 0.651 | 0.983 |
| Transistor | 0.898 | 0.898 | 0.898 | 0.902 | 0.947 | 0.787 | 0.946 |
| Wood | 0.923 | 0.923 | 0.923 | 0.933 | 0.933 | 0.575 | 0.931 |
| Zipper | 0.930 | 0.930 | 0.930 | 0.971 | 0.960 | 0.607 | 0.959 |
| **Average** | 0.954 | 0.954 | 0.954 | 0.964 | 0.968 | 0.536 | 0.967 |

Table 2. InReach ablation and alternative anomaly scoring function results on MvTec AD Anomaly Segmentation task in pixel-wise AUROC [1]. All testing was done with 10 corruptions in each class's training set. The nearly identical results between the first three configurations are correct. *The configuration for the ablation results is cumulative from left to right.

Along with our naive proposed minimal distance scoring method for anomaly detection, we also evaluate the use of the Mahalanobis distance [7] as used in [3] and the anomaly scoring method in PatchCore [9]. We note extremely poor performance in using the Mahalanobis distance. This is caused by the poor values generated for the standard deviations used. We used the calculated standard deviation of the specific patch's channel for the Mahalanobis calculation, but since these channels often contain less than 10 patches, the standard deviation can vary wildly. We see very similar, if slightly lower, performance using PatchCore's scoring method to our proposed method. The increased complexity of PatchCore, finding multiple nearest neighbours and calculating exponentials, is not justified by the performance.

| Class | Bottle | Cable | Capsule | Carpet | Grid | Haz. | Leather | M. Nut | Pill | Screw | Tile | Tooth. | Trans. | Wood | Zipper |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Transpose Test | 123 | 1030 | 7840 | 8 | 12 | 71 | 96 | 82 | 5064 | 26 | 78 | 5916 | 1096 | 6 | 6990 |

Table 3. InReaCh transpose test scores for each data class for determining if positions are consistent across images in the training data and if positional embedding should be incorporated.

## D. Additional Positional Embedding Details

Each class in the MVTec AD Dataset is clearly separated in two groups based on the transpose test, a group where the transpose test values are less than 150 and those above 1000. The group above 1000 all are classes that have positional structure and could benefit from positional embedding, and the opposite for the classes with scores below 150. Notably, some object classes are in the group with low transpose test scores. This is because of the random positioning of the object in the frame, such as for hazelnut and screw. From the clear separation of these classes, we set the threshold at a reasonable 600 for the transpose test, determining if each class should utilize positional embeddings or not. Overall positional embedding does not make a significant impact on our final overall. When testing classes that fail the positional embedding transpose test, we rotate each image by 1-degree increments and compare the center quarter of the image to the first image in the dataset choosing the rotation angle associated with the smallest L2 distance. Once all images are aligned to the first image in the dataset, we re-run the transpose test to determine if a rotation only disrupts positional information. Classes that pass the transpose test after rotational alignment maintain their new alignment, and positional embeddings are appended to the patch features. Otherwise, they are returned to their original state and processed without positional embeddings.

## E. Implementation Details

We implemented InReaCh in Python 3.8 [10]. We did our distance calculations and feature extraction using PyTorch [8]. Following previous methods for direct comparison [9, 2, 4], we use the official Pytorch torchvision WideResNet50 weights. We use Scipy's Gaussian filter to smooth our pixel-wise predictions with a standard deviation of 5, and we did not tune this parameter [11].

## References


[1] 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*, pages 9592–9600, 2019.

[2] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *CoRR, abs/2005.02357*, 2020.

[3] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. January 10–15, 2021*, pages 475–489. Springer, 2021.

[4] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. *arXiv preprint arXiv:2002.03734*, 2020.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021.

[7] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[9] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022.

[10] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[11] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[12] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[13] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

| Bottle | Cable | Capsule | Carpet | Grid |
|---|---|---|---|---|
| good/018.png | good/032.png | crack/007.png | color/006.png | good/013.png |
| contamination/010.png | combined/010.png | squeeze/000.png | color/012.png | metal_contamination/003.png |
| broken_large/009.png | missing_cable/000.png | squeeze/006.png | cut/001.png | bent/009.png |
| broken_small/010.png | missing_wire/009.png | good/010.png | cut/015.png | metal_contamination/006.png |
| broken_large/003.png | good/027.png | crack/021.png | metal_contamination/007.png | metal_contamination/000.png |
| good/014.png | good/002.png | good/003.png | hole/000.png | good/015.png |
| good/016.png | bent_wire/002.png | squeeze/007.png | hole/002.png | metal_contamination/007.png |
| broken_large/014.png | missing_cable/011.png | scratch/008.png | thread/013.png | bent/005.png |
| good/011.png | cable_swap/003.png | faulty_imprint/015.png | thread/001.png | good/019.png |
| broken_small/007.png | missing_cable/008.png | good/007.png | good/012.png | good/006.png |
| contamination/018.png | good/041.png | good/014.png | cut/013.png | thread/001.png |
| good/003.png | good/024.png | faulty_imprint/004.png | metal_contamination/004.png | broken/007.png |
| broken_large/002.png | combined/000.png | scratch/021.png | metal_contamination/015.png | good/012.png |
| contamination/011.png | good/057.png | scratch/011.png | hole/010.png | good/002.png |
| good/002.png | cut_outer_insulation/008.png | squeeze/003.png | cut/007.png | good/004.png |
| good/010.png | cut_outer_insulation/005.png | poke/012.png | good/022.png | glue/010.png |
| broken_large/016.png | good/047.png | squeeze/009.png | color/009.png | broken/001.png |
| broken_large/017.png | cable_swap/005.png | poke/018.png | cut/012.png | thread/002.png |
| broken_small/001.png | good/033.png | faulty_imprint/016.png | color/014.png | good/020.png |
| good/001.png | bent_wire/007.png | poke/006.png | cut/008.png | good/018.png |
| contamination/014.png | good/051.png | scratch/001.png | metal_contamination/001.png | broken/006.png |
| good/019.png | bent_wire/000.png | crack/009.png | hole/015.png | broken/003.png |
| good/005.png | cut_inner_insulation/013.png | squeeze/013.png | thread/009.png | glue/002.png |
| contamination/016.png | good/054.png | faulty_imprint/005.png | color/003.png | good/003.png |
| broken_small/006.png | combined/008.png | scratch/014.png | color/005.png | bent/006.png |
| broken_small/003.png | missing_wire/007.png | faulty_imprint/008.png | metal_contamination/008.png | metal_contamination/010.png |
| broken_small/014.png | cable_swap/000.png | faulty_imprint/006.png | metal_contamination/011.png | glue/000.png |
| broken_large/018.png | good/009.png | crack/014.png | good/002.png | broken/010.png |
| contamination/015.png | poke_insulation/007.png | scratch/015.png | color/001.png | good/005.png |
| contamination/008.png | good/030.png | poke/019.png | good/015.png | metal_contamination/008.png |
| broken_small/012.png | good/029.png | scratch/010.png | thread/014.png | bent/008.png |
| contamination/020.png | good/021.png | crack/012.png | hole/009.png | good/011.png |
| good/012.png | good/001.png | squeeze/016.png | good/020.png | thread/003.png |
| broken_large/000.png | cut_inner_insulation/007.png | squeeze/015.png | good/009.png | good/008.png |
| good/008.png | good/012.png | squeeze/018.png | metal_contamination/010.png | thread/004.png |
| broken_large/004.png | missing_wire/000.png | scratch/005.png | metal_contamination/002.png | thread/010.png |
| broken_large/015.png | good/050.png | good/004.png | thread/018.png | good/001.png |
| broken_large/005.png | good/005.png | crack/001.png | metal_contamination/013.png | metal_contamination/009.png |
| broken_large/011.png | good/055.png | faulty_imprint/002.png | good/025.png | good/016.png |
| broken_large/013.png | good/048.png | crack/003.png | good/006.png | metal_contamination/005.png |

Table 4. InReaCh training data corruptions for 40 corruptions splits for the MVTec AD Dataset. For 10 corruption splits, only the first 10 test image labels are incorporated into the training set.

| Hazelnut | Leather | Metal Nut | Pill | Screw |
|---|---|---|---|---|
| crack/012.png | fold/007.png | color/006.png | color/018.png | manipulated_front/011.png |
| good/026.png | glue/013.png | color/012.png | pill_type/006.png | manipulated_front/010.png |
| crack/006.png | good/020.png | scratch/022.png | crack/022.png | thread_side/005.png |
| good/032.png | poke/009.png | good/013.png | scratch/001.png | thread_side/018.png |
| print/016.png | poke/016.png | bent/000.png | good/006.png | manipulated_front/014.png |
| print/004.png | color/012.png | bent/010.png | crack/023.png | thread_top/019.png |
| hole/002.png | poke/014.png | bent/012.png | pill_type/005.png | manipulated_front/020.png |
| good/002.png | cut/014.png | flip/010.png | crack/021.png | good/035.png |
| hole/000.png | good/018.png | color/020.png | good/020.png | good/037.png |
| cut/002.png | glue/009.png | scratch/005.png | good/002.png | scratch_neck/001.png |
| good/010.png | glue/002.png | good/011.png | contamination/013.png | scratch_neck/013.png |
| crack/009.png | cut/010.png | good/019.png | crack/025.png | thread_top/020.png |
| cut/000.png | glue/018.png | bent/008.png | color/002.png | thread_top/014.png |
| good/027.png | poke/003.png | bent/020.png | crack/007.png | manipulated_front/016.png |
| good/029.png | cut/004.png | good/005.png | crack/013.png | manipulated_front/019.png |
| cut/009.png | good/002.png | scratch/015.png | contamination/015.png | good/002.png |
| crack/014.png | glue/014.png | color/009.png | crack/012.png | manipulated_front/004.png |
| hole/016.png | good/006.png | good/010.png | pill_type/000.png | scratch_head/019.png |
| hole/004.png | fold/014.png | color/014.png | scratch/015.png | scratch_head/011.png |
| cut/005.png | glue/005.png | good/006.png | color/014.png | thread_side/019.png |
| cut/012.png | poke/002.png | good/016.png | crack/014.png | thread_side/006.png |
| crack/003.png | color/006.png | bent/023.png | good/009.png | thread_side/016.png |
| crack/005.png | color/005.png | flip/006.png | combined/000.png | thread_side/013.png |
| good/022.png | color/009.png | color/003.png | color/011.png | scratch_neck/004.png |
| crack/001.png | cut/002.png | color/005.png | combined/008.png | good/014.png |
| good/015.png | color/014.png | bent/001.png | contamination/007.png | scratch_neck/009.png |
| good/038.png | fold/010.png | bent/004.png | faulty_imprint/004.png | manipulated_front/021.png |
| print/000.png | fold/000.png | flip/018.png | contamination/000.png | good/034.png |
| cut/016.png | good/022.png | color/001.png | good/008.png | scratch_head/014.png |
| good/009.png | color/001.png | scratch/008.png | color/017.png | thread_top/013.png |
| good/014.png | color/003.png | flip/011.png | color/023.png | good/026.png |
| cut/013.png | cut/015.png | bent/019.png | combined/004.png | manipulated_front/022.png |
| hole/015.png | glue/001.png | scratch/013.png | crack/009.png | thread_top/008.png |
| hole/013.png | good/015.png | scratch/002.png | scratch/021.png | good/031.png |
| hole/005.png | glue/011.png | bent/003.png | good/024.png | thread_side/011.png |
| hole/006.png | fold/013.png | good/017.png | combined/003.png | scratch_head/006.png |
| good/019.png | fold/008.png | flip/015.png | good/022.png | thread_top/010.png |
| print/011.png | fold/009.png | bent/006.png | color/019.png | manipulated_front/008.png |
| good/007.png | fold/001.png | scratch/018.png | faulty_imprint/003.png | good/015.png |
| good/031.png | good/029.png | flip/022.png | combined/001.png | manipulated_front/013.png |

Table 5. InReaCh training data corruptions for 40 corruptions split for the MVTec AD Dataset. For 10 corruption splits, only the first 10 test image labels are incorporated into the training set.

| Tile | Toothbrush | Transistor | Wood | Zipper |
|---|---|---|---|---|
| gray_stroke/006.png | defective/024.png | good/024.png | scratch/018.png | rough/010.png |
| gray_stroke/012.png | good/005.png | damaged_case/006.png | good/008.png | fabric_interior/008.png |
| good/019.png | defective/018.png | good/030.png | hole/007.png | split_teeth/006.png |
| oil/000.png | defective/007.png | good/035.png | good/011.png | combined/014.png |
| oil/009.png | defective/006.png | damaged_case/005.png | good/005.png | fabric_interior/009.png |
| glue_strip/001.png | defective/019.png | misplaced/002.png | scratch/020.png | fabric_border/003.png |
| glue_strip/003.png | defective/013.png | damaged_case/009.png | good/012.png | broken_teeth/010.png |
| crack/016.png | defective/011.png | good/027.png | hole/003.png | split_teeth/009.png |
| crack/004.png | good/000.png | good/051.png | good/003.png | rough/004.png |
| good/002.png | defective/026.png | damaged_case/003.png | scratch/011.png | broken_teeth/001.png |
| good/031.png | defective/005.png | good/053.png | liquid/005.png | broken_teeth/013.png |
| oil/006.png | defective/002.png | bent_lead/007.png | color/007.png | good/018.png |
| oil/017.png | good/002.png | good/036.png | scratch/017.png | good/007.png |
| glue_strip/011.png | defective/020.png | cut_lead/004.png | scratch/007.png | split_teeth/002.png |
| good/025.png | good/003.png | good/043.png | scratch/009.png | squeezed_teeth/002.png |
| good/012.png | good/009.png | good/040.png | liquid/003.png | good/002.png |
| gray_stroke/009.png | defective/021.png | damaged_case/001.png | color/001.png | squeezed_teeth/008.png |
| good/030.png | good/008.png | good/056.png | liquid/006.png | fabric_border/001.png |
| gray_stroke/014.png | good/011.png | cut_lead/006.png | good/004.png | squeezed_teeth/012.png |
| good/026.png | defective/017.png | good/017.png | good/002.png | good/015.png |
| oil/003.png | defective/022.png | cut_lead/009.png | color/006.png | broken_teeth/017.png |
| glue_strip/016.png | defective/010.png | good/050.png | color/003.png | split_teeth/005.png |
| crack/012.png | good/001.png | cut_lead/001.png | combined/006.png | fabric_border/004.png |
| gray_stroke/003.png | defective/015.png | good/012.png | scratch/008.png | combined/013.png |
| gray_stroke/005.png | defective/012.png | good/032.png | hole/004.png | broken_teeth/004.png |
| oil/010.png | defective/016.png | good/000.png | good/016.png | fabric_interior/006.png |
| oil/013.png | good/006.png | good/008.png | combined/004.png | rough/001.png |
| rough/007.png | good/007.png | good/038.png | combined/002.png | good/000.png |
| gray_stroke/001.png | defective/003.png | misplaced/004.png | scratch/010.png | split_teeth/014.png |
| good/005.png | defective/023.png | bent_lead/000.png | good/013.png | fabric_border/014.png |
| rough/000.png | defective/027.png | good/058.png | hole/006.png | rough/008.png |
| glue_strip/010.png | good/010.png | good/005.png | scratch/016.png | fabric_border/007.png |
| good/010.png | defective/025.png | good/055.png | liquid/007.png | good/004.png |
| rough/014.png | defective/000.png | misplaced/000.png | scratch/013.png | rough/005.png |
| oil/012.png | defective/009.png | good/020.png | liquid/008.png | combined/007.png |
| oil/004.png | defective/014.png | good/033.png | scratch/004.png | split_teeth/013.png |
| rough/004.png | defective/001.png | good/025.png | scratch/006.png | good/025.png |
| oil/015.png | defective/028.png | bent_lead/004.png | good/015.png | good/022.png |
| good/015.png | defective/029.png | good/018.png | good/000.png | broken_teeth/014.png |
| rough/011.png | defective/008.png | good/011.png | good/010.png | broken_teeth/015.png |

Table 6. InReaCh training data corruptions for 40 corruptions split for the MVTec AD Dataset. For 10 corruption splits, only the first 10 test image labels are incorporated into the training set.