Advancing Anomaly Detection: The IDW dataset and MC algorithm

Alexander D.J. Taylor[1]
adjt20@bath.ac.uk
Jon Morrison[2]
jonathan.morrison2@rolls-royce.com
Phillip Tregidgo[2]
phil.tregidgo@rolls-royce.com
Neill D.F. Campbell[1]
nc537@bath.ac.uk

[1] ART-AI
Department of Computer Science
University of Bath

[2] Defence Operations
Rolls-Royce

Abstract

In this work we present a novel anomaly detection dataset, Industrial Defects in the Wild (IDW). IDW contains images of various industrial and household inspection processes. It features real images with complex and varied perspectives from freely moving cameras. We show this is more challenging than the well-known MVTec dataset. We also present MultiCore (MC), a novel algorithm that achieves state-of-the-art results on the introduced IDW dataset and popular MVtec dataset. The MC algorithm trains multiple nearest neighbour predictors, each with different hyperparameters. We propose that an ensemble is more powerful than any individual model. Synthetic anomalies are created using a novel schema intended to systematically cover as many variations as possible. The ensemble output is fed into a heatmap fusion module, which is trained in a supervised fashion using the synthetic anomalies and a perimeter-based loss function. On the popular MVTec dataset, the MC algorithm achieves P-AUC score of 0.986. On the introduced and more challenging IDW dataset, the MC algorithm achieves a P-AUC of 0.935. We verify that these results are state-of-the-art by trialing the existing top fourteen anomaly detection algorithms which have code available. The IDW dataset can be found at: https://github.com/alext1995/IDW, and MultiCore code can be found at: https://github.com/alext1995/MultiCore.

1 Introduction

Many classification problems in the real world have vastly imbalanced data, with a lack of samples for certain classes. This makes training a supervised deep learning model challenging. This is particularly common in medicine where diseased examples are very rare [31] and in engineering where industrial defects are rare [6, 27]. Humans are able to recognise whether a given object or query appears out of place, or more formally, does not belong to the distribution of nominal objects. In light of this, the field of anomaly detection has received increased attention and growth [16, 33, 37, 38, 39, 42]. Anomaly detection is a

semi-supervised framework where a model only sees nominal examples during training, and must classify examples or pixels as nominal or anomalous during inference.

The current de-facto dataset, MVTec [3], is quasi-solved. Many algorithms reportedly score over 99% under a standard metric such as Imagewise-AUC (I-AUC). MVTec contains homographic (non-varied perspective) images in a standardised format, with clear lighting and orientation. This is unrealistic for many real-world problems.

In light of these issues, we make the following contributions:

1. We release a novel and challenging dataset, Industrial Defects in the Wild (IDW). This contains images of inspections of industrial components. This is more realistic than existing datasets as the images contain real anomalies, and the images are taken from a moving camera, offering varied viewpoints.

2. We release MultiCore (MC), an algorithm based on an ensemble of nearest neighbour predictors, a novel synthetic anomaly schema, and a supervised UNet. We also show that using a perimeter-based loss boosts the results. MC achieves state-of-the-art performance on the introduced IDW and the dominant MVTec datasets via the standard I-AUC and P-AUC metrics.

Throughout this work, we refer to the nearest neighbour algorithms in the ensemble as *predictors*, and the complete MultiCore instances as *models*.

2 Background

Initially researchers tackled visual anomaly detection using the well-known object classification datasets such as CIFAR [21], MNIST [13] and ImageNet [12]. In these studies, certain classes were designated as anomalous, while the remaining classes were used for training.

As the field has grown grew, specific datasets were developed to tackle the problem across different domains [3, 4, 5, 10, 22, 26, 27, 44]. These datasets have served the community well, but fail to provide the community with images which contain varied perspectives, and therefore do not represent all real-world challenges. Datasets such as DAGM and BTAD are too small and specific to represent real world processes. The Fishyscapes Web and Eyecandies datasets contain synthetic anomalies, and the Fishyscapes Lost and Found dataset is very small. The ShanghaiTech dataset consists of images from CCTV cameras in fixed positions, meaning the images are homographic. The MVTec dataset also contains homographic images, created in precisely controlled conditions such as constant lighting, orientation, angle, and distance.

Many families of anomaly detection algorithms exist. For a detailed review on AD algorithms, we point the reader to [30, 36]. Patchcore is the most related to our work [33]. Many existing anomaly detection algorithms use synthetic anomalies alongside supervision of some kind, such as [6, 24, 43].

VisionAD [35] intends to make the benchmarking of anomaly detection more fair. It provides a package with the most recent and performant anomaly detection algorithms. All algorithms are written through a standardised API, and shared data-loading, wrapper, and evaluation code. It also introduces a new metric, Proportion Localised (PL). PL reports the proportion of anomalies 'found' via classifying anomalies as found or missed through an IoU limit. Each anomaly mask is converted into its minimum area bounding box before comparison with the prediction. It is claimed that this provides a more interpretable method

of evaluation. Due to the use of bounding box labels, it is also claimed that the noise associated with pixel-by-pixel comparison of the prediction and target is mitigated. We use the VisionAD package for all of experiments to ensure fair comparison of algorithms. we also report the PL scores alongside the traditional metrics.

3 Industrial Defects in the Wild dataset

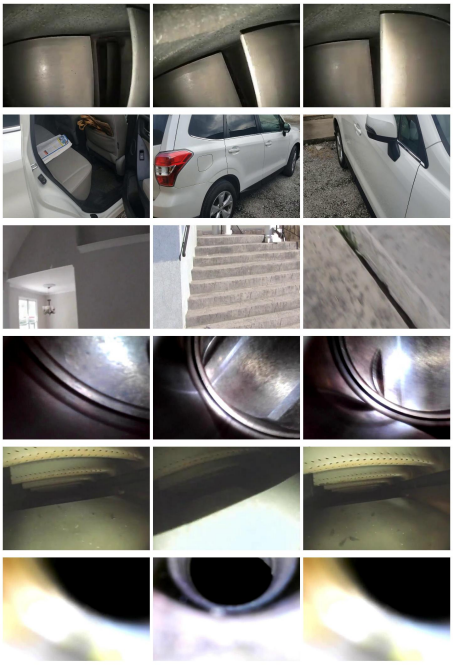
	Classes	Training	Regular Testing	Anomaly Testing	Varied Perspective	Real (non-synthetic)	Labels-Available	Year	Content
DAGM[22]	4	2,000	2,000	600			✓	2007	textures
Shanghai Tec[26]	13	317,398	42,883	17,090		✓	✓	2018	campus
MVtec-AD[3]	15	3,629	467	1,258		✓	✓	2019	industrial
Species[32]	1000	0**	0	700000	✓	✓	✓	2019	nature
StreetHazards[17]	1	5125	192	1500	✓		✓	2019	self-driving
BDD-Anomaly[17]	20	-	-	-	✓	✓	✓	2019	self-driving
Vistas-NP[15]	-	8,003	-	-	✓	✓	✓	2020	streets
Fishyscapes Web[4]	-	0*	-	-	✓			2021	self-driving
Fishyscapes LF[4]	-	0*	-	-	✓	✓		2021	self-driving
RoadAnomaly21[4]	-	0*	-	-	✓	✓		2021	self-driving
RoadObstacle21[4]	-	0*	-	-	✓	✓		2021	self-driving
BTAD[21]	3	1,799	451	290		✓	✓	2021	industrial
EyeCandies[8]	10	90,398	2000	2000	✓		✓	2022	candies
VisA[31]	12	9,621	962	1200		✓	✓	2022	industrial
IDW (ours)	7	8,280	2,510	1912	✓	✓	✓	2024	industrial

Table 1: Comparison of IDW and other popular anomaly detection datasets. Note the introduced IDW dataset is the first industrial dataset to contain non-synthetic and non-homographic images, i.e. real photos from varied angles and geometries. *Hidden evaluation sets published with the intention of training on the CityScapes dataset [10]. **Training is done on the ImageNet dataset.

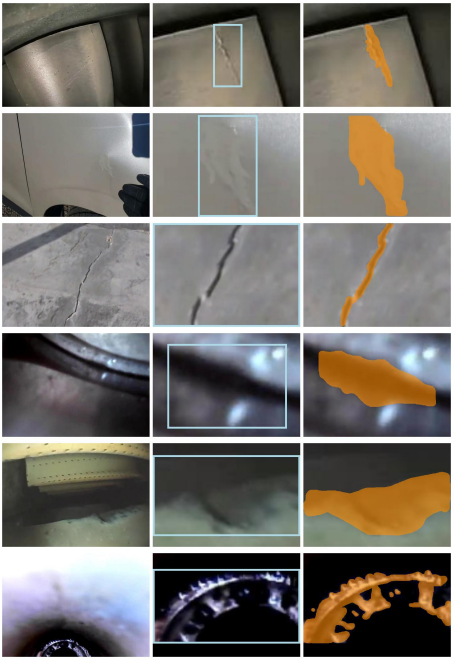
We release a new challenging anomaly detection dataset, Industrial Defects in the Wild (IDW). IDW contains high-resolution varied-perspective images with varying lighting and geometry. IDW contains pixel-wise labels of each anomalous pixel, and bounding box labels of each discrete anomaly. The dataset is intended to assess how state-of-the-art AD algorithms perform with less constrained data, such as that taken from a video feed of a moveable camera. We emphasize that this means some images are not as high quality as they would be if taken in controlled conditions. This is desirable as it is more representative of real-world problems. Table 1 shows a comparison between this and existing datasets. Figure 1 shows some example images, and Table 2 contains a breakdown of each class.

This is in contrast to the MVTec dataset that contains consistent perspective and orientation. These are controlled conditions which are not representative of some real-world problems. The classes of the MVTec dataset can all be considered low complexity, with training image sizes in the hundreds and static viewpoints. The introduced IDW dataset has fewer classes, but the classes are bigger, with thousands of training images and varying viewpoints. Whilst MVTec is able to test whether anomaly detection models work for low-complexity problems, the IDW dataset will test if algorithms can scale to more complex problems. The data consists of photo frames from videos of real inspection processes. The classes cover various industrial processes:

BirdStrike: Inspection of turbine blades of an undisclosed engine hit by a bird strike. Some defects are clear, while others are very subtle and require a zoomed-in view for an untrained



Training selection of images for each class.



From left to right: image, zoomed-in image on anomaly, zoomed-in image with pixel-wise label.

Figure 1: Example images from each class of the introduced IDW dataset.

human to notice.

CarBody: Inspection of surface dents and defects on a automobile. The colour and texture of the defects are very similar to the non-defected images. This tests the ability of algorithms to detect subtle changes in geometry.

HouseStructure: Inspection of structural components on the inside and outside of a house. The defect components are recorded at completely different angles to the equivalent non-defected components, forming a challenging problem.

HeadGasket: Inspection of the head gaskets inside an automobile engine. The train dataset contains images of good quality head gaskets, whilst the anomalous data contains cracks, warps, and oxidation to the surface. This is challenging because some of the anomalies are very small.

NozzleGuideVane: Inspection of the nozzle guide vanes of an aircraft engine. The defects are less subtle than some of those from the BirdStrike class. However there is less training data, and some of these defects occur in very low lighting.

InspectionVane: Inspection of the inspection vanes inside an automobile engine. Due to the fast movement of the camera, some of the images are blurred, up-close, and contain reflections.

Pooled: All training, testing and anomalous images from the previous classes are pooled into one class. A human expert is able to recognise anomalies across many domains. Therefore we should aspire that machine learning models do the same. The contributions from each

Class	Training	Regular Testing	Anomalous Testing	Discrete Anomalies
BirdStrike (BS)	1347	586	282	333
CarBody (CB)	944	156	182	273
HouseStructure (HS)	769	243	139	248
HeadGasket (HG)	499	102	59	79
NozzleGuideVane (NGV)	486	150	242	743
InspectionVane (IV)	95	18	51	260
Pooled (P)	4140	1255	956	1937

Table 2: Class breakdown of the IDW dataset. Discrete anomalies refer to discrete anomalous areas (not connected to other anomalous areas).

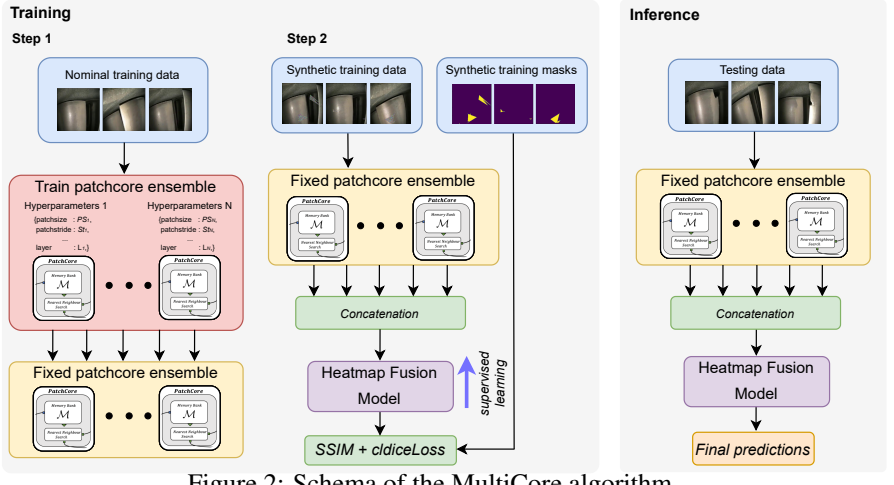


Figure 2: Schema of the MultiCore algorithm.

class are not balanced, as humans do not need to balance their time between certain domains. This class is intended to be as challenging and realistic as possible, forming a very difficult task for algorithms.

The dataset has been created through manually labelling and filtering publicly available images sourced under fair use [8, 14, 18, 28, 29]. This means the dataset is released with an academic and non-commercial licence, and therefore cannot be used for commercial applications.

4 MultiCore

For the nearest neighbour estimator, we make some changes to the Patchcore algorithm as published in [33]. In the original publication, the coreset size was calculated as a fraction of the training size. We change this such that the coreset size is preset, and unrelated to the dataset size. This allows the coreset size to be set to the maximum to fill the GPU memory constraint, and is not dependent on the dataset size. We find a coreset size of 25,000 makes sufficient use of a 24GB GPU. The original publication used Euclidean distance for the coreset sub-sampling and inference. We create an option for other distance metrics to be used. We also allow the patch size and patch stride to be adjusted. We refer to this modified

algorithm as Patchcore+.

A schema of the MultiCore (MC) algorithm is shown in Figure 2. The algorithm comprises an ensemble of Patchcore+ predictors, where each predictor has different hyperparameters. The selection of these hyperparameter sets is done using synthetic training anomalies to avoid test set leakage. The synthetic anomaly algorithm is discussed in Section 4.1. The hyperparameter set selection is discussed in Section 4.2. For each hyperparameter set, a Patchcore+ predictor is individually trained on the training data. This gives a set of predictors, which during inference each produce individual heatmaps. To create a final prediction, these heatmaps need to be fused into a single heatmap. To achieve this, a Heatmap Fusion Module (HFM) is used, which is discussed in Section 4.4.

During inference, a test image is put through each estimator to form a set of heatmaps, and these heatmaps are concatenated and put through the HFM, which produces a final prediction heatmap.

4.1 Stratified Synthetic Anomalies

We introduce a novel method of creating synthetic anomalies, Stratified Synthetic Anomalies (SSA). Our goal is to create a set of anomalies that systematically cover as many variations as possible. To avoid leakage, no statistics or information from any test datasets are used. To make the anomalies as general as possible, we employ a stratified schema. The synthetic anomalies are created by applying masked noise over a randomly chosen image from the training set, where the mask serves as the ground truth for the resulting image.

Firstly, we discuss the mask generation process. In order for the masks to be representative of as many defect sizes as possible, 29 buckets are created corresponding to indices n from 1 to 29, with K images per bucket. For an image width d and bucket n, anomaly shapes are created with a scale of roughly d/n. To create the shapes, a random polygon generation algorithm [1] is employed, which uniformly samples a shape between 3 and 7 points. The shape is then chosen to either remain unstretched, stretched in the y direction, or stretched in the x direction, with probabilities of 0.5, 0.25, and 0.25 respectively. In the case of stretching, a value was uniformly sampled between 0.05 and 1, and the aspect ratio was set as the reciprocal of this number.

The anomalies are systematically positioned. For each bucket n, each anomaly is centered on one point of an equally spaced nxn grid. If K<n*n, then these grid points are uniformly sampled with replacement. Finally a small perturbation of +/-0.5*d/n is added to each anomaly, to ensure each anomaly does not lie in the center of its designated area. This results in a stratified set of masks which systematically cover scale, aspect ratio, and position. We require that some masks have more than one anomaly. We use a to represent the number of anomalies in a mask. For each a value between 2-7, we require m masks. We select a maximum value of 7 as we believe it is very unlikely that a test image has than 10 anomalies. To create test samples with more than one anomaly, non-overlapping

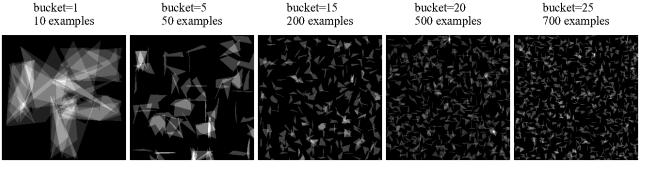


Figure 3: Demonstration of various masks for different buckets n. These are overlaid for demonstration purposes, note each image contains 1-10 anomalies.

masks from the masks already created were randomly selected and combined until the value of m is reached. This means a proportion of the test images have 2-7 anomalies. Figure 3 shows many of these masks overlaid for different buckets, to give the reader an idea of the types of shapes created. Note that for smaller anomalies, the stratified sampling of placement means they cover the whole image. Note this image shows many of these masks overlaid, when in reality, each mask only has 1-7 anomalies.

The above schema will give t=29*k + 6*m masks. Note that we also use r regular images (no anomalies added) with zeroed masks for training. We use values k, m, and r values of 60, 20, and 200 respectively. This results in 2060 masks.

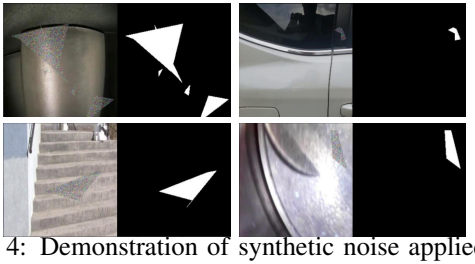


Figure 4: Demonstration of synthetic noise applied to images from the introduced IDW.

We find no difference when using Perlin noise or Gaussian noise. We trial fixed values of Gaussian noise, and Gaussian noise uniformly sampled in a given range. We find our algorithm to perform better with a point value of Gaussian noise as opposed to uniformly sampled noise. However, we find the model to be relatively nonsensitive to the selected value of noise. We believe that this is because the supervision allows the model to calibrate to the weight of synthetic noise used. We find 0 centered noise with a standard deviation of 0.5 works. Figure 4 shows a demo of some synthetic anomalies.

4.2 Hyperparameter selection

The underlying assumption of MultiCore is that the combination of many different nearest neighbour predictors is better than any individual predictor. Here the hyperparameters values are systematically chosen for the combination of predictors. We vary patchsize, patchstride, and distance metric. Patchsize can take any value from [3, 1, 5, 7][1], patchstride can take any value from [1, 3, 5], and the distance metric can be *euclidean*, *cosine*, *minkowski3*, or *manhatten*. We make a Cartesian product of each hyperparameter which gives a list of unique Hyperparameter Sets (H), covering every possible combination. We refer to a unique set of hyperparameters as h.

For each combination, we train a Patchcore+ predictor on the bottle (MVTec) and BirdStrike (IDW) classes, We use synthetic anomalies to test this. These are produced as described above, and are identical for each predictor. *Note we do not use any testing data from the bottle or BirdStrike classes, we only use the training images and synthetic test data.* Each predictor gives an array of predictions, which correspond to a hyperparameter set. We intend to find the group of predictors, which can be combined to produce the best score on a given metric. We combine the output from a group of predictors by taking the mean of their normalised predictions, then calculating the metric as normal. We refer to this score as the 'coverage' of a group of predictors. This coverage shows how well a combination of predictors alleviates the errors of any individual predictors when combined. We propose that predictor combinations with better coverage will perform better overall in the MC algorithm,

[1]The unusual order is deliberate to ensure the original hyperparameter set from [53] comes first.

where the HFM will combine the outputs from each predictor.

We use P-AUC as the metric to evaluate coverage. We wish to final the list of hyperparameters sets h, whih produces the best coverage. The computation cost of brute force testing the coverage of each combination of predictors is infeasible. We assume the best predictor uses the original hyperparameters from [33], and therefore, we start with this predictor. We combine this predictor with every other, and calculate the scores. The predictor corresponding to the best score is added to the list of final predictors. This process is then repeated, adding more predictors to the final list. When the desired metric does not increase, the process is finished.

Table 3 shows the hyperparameters of the final predictor selections, and Table 4 shows the coverage values for various metrics. We notice that as we include more predictors, the increase in scores decrease.

predictor #	Patch size	Patch stride	Distance
1	3	1	euclidean
2	1	1	minkowski3
3	7	3	euclidean
4	3	3	euclidean
5	7	1	euclidean
6	1	3	euclidean
7	1	1	euclidean
8	3	5	euclidean

Table 3: Hyperparameters of the final predictors.

Predictor coverage	I-AUC	P-AUC	P-AUPRO	PL
1	0.992	0.976	0.901	0.833
1-2	0.995	0.981	0.923	0.862
1-3	0.996	0.983	0.944	0.884
1-4	0.998	0.984	0.956	0.889
1-5	1	0.984	0.963	0.901
1-6	1	0.985	0.969	0.909
1-7	1	0.985	0.973	0.913
1-8	1	0.985	0.973	0.914

Table 4: Coverage values across the synthetic anomalies.

4.3 Heatmap Fusion Module

We use the Unet from [32] for the HFM. The accepts the concatenated predictions from the ensemble. We find that dice, L1 and MSE loss functions are not performant. SSIM loss is commonly used for segmentation tasks. We find this loss works well. However, we see a boost in performance by adding a perimeter based loss. Work from the medical imaging community [19] has found that perimeter loss functions are able to increase the performance of segmentation problems where organs tend to have awkward and small shapes. We propose that irregular organ shapes share features with industrial anomalies. We find the cldiceloss from [19], the perimeter version of the dice loss, works best when added to the SSIM loss.

4.4 Training procedure

The prediction from each Patchcore+ predictor is normalised against the mean and standard deviation of the train predictions from the respective Patchcore+ predictor. For the supervision of the HFM, a learning rate of 1e-5 is used, with weight decay of 1-e5, gradient clipping of 1, and root mean square prop optimisation with momentum of 0.999. Each predictor is trained for 300 epochs. We use a synthetic anomaly set of size of 4000 (s=2000). We run three variants of the MC algorithm, MC2, MC4, and MC8, which use 2, 4, and 8 Patchcore+ predictors respectively. We use a Nvidia RTX3090 GPU (24Gb) for our experiments.

	I-AUC	P-AUC	P-AUPRO	PL [35]
MC8 (ours)	**0.9992**	0.983	0.908	0.887
MC4 (ours)	0.9994	**0.986**	0.912	0.888
MC2 (ours)	0.997	0.985	0.927	**0.891**
CFA[23]	0.981	0.985	0.922	0.87
FastFlow[49]	0.991	0.984	0.920	0.883
MSFlow[1]	0.990	0.983	0.896	0.828
PEFM[33]	0.979	0.981	0.924	0.849
Reverse Distillation[10]	0.992	0.981	**0.929**	0.89
Patchcore[33]	0.983	0.980	0.911	0.885
CFlow[18]	0.973	0.980	0.899	0.829
FastFlow+AltUB[49, 48]	0.988	0.976	0.884	0.881
EfficientAD[1]	0.991	0.975	0.861	0.77
PFM[33]	0.980	0.975	0.913	0.836
SimpleNet[25]	0.963	0.975	0.874	0.807
CDO[1]	0.942	0.974	0.915	0.842
AST[30]	0.934	0.959	0.844	0.723
MemSeg[40]	0.963	0.881	0.539	0.328

MVTec: Ranked using P-AUC; I-AUC, AUPRO, and PL[35] also shown.

	I-AUC	P-AUC	P-AUPRO	PL [35]
MC8 (ours)	**0.974**	0.915	0.566	0.473
MC4 (ours)	0.955	0.921	0.572	0.481
MC2 (ours)	0.908	**0.935**	0.602	**0.515**
Patchcore[33]	0.821	0.933	0.589	0.478
CDO[1]	0.82	0.932	**0.62**	0.433
Rev. Distillation[10]	0.739	0.932	0.557	0.372
PFM[33]	0.78	0.931	0.574	0.423
PEFM[33]	0.805	0.93	0.596	0.431
MSFlow[1]	0.777	0.898	0.467	0.323
FastFlow[49]	0.786	0.878	0.498	0.289
EfficientAD[1]	0.764	0.868	0.429	0.311
SimpleNet[25]	0.752	0.863	0.413	0.354
FF+AltUB[49, 48]	0.742	0.821	0.442	0.225
CFlow[18]	0.664	0.78	0.387	0.115
CFA[23]	0.66	0.764	0.392	0.248
AST[30]	0.608	0.713	0.348	0.244
MemSeg[40]	0.789	0.667	0.120	0.203

IDW: Ranked using P-AUC; I-AUC, AUPRO, and PL[35] also shown.

Table 5: Results on MVTec and IDW datasets.

5 Results

We run the MC algorithm on MVTec and the introduced IDW. To compare the results of MC with existing algorithms, we aggregate the top 14 anomaly detection algorithms with code available, as selected by the recent benchmarking work, VisionAD [35]. These algorithms are shown in Table 5. To ensure fair benchmarking, we use the code from VisionAD for all experiments. In addition, we present the results of the PL metric, also introduced in the same work. PL reports the proportion of anomalies which have an IoU agreement between the prediction and label greater than 0.3. It is intended to be more interpretable than the other metrics.

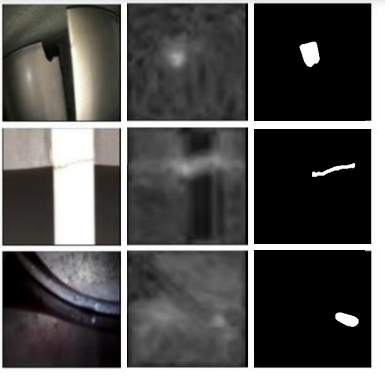


Figure 5: Demonstration of MC predictions on randomly chosen images from the IDW dataset. Left to right, input image, prediction, ground truth.

We see that the MC algorithm achieves state-of-the-art values across I-AUC, P-AUC, and PL on the MVTec dataset. Similarly on the IDW dataset, state-of-the-art values are achieved across the same metrics. We note that increasing the number of predictors does not always increase the results. Infact, on the IDW dataset, for P-AUC and PL, the best values are achieved by MC2. Figure 5 shows some MC2 demonstration predictions from the IDW dataset.

Notable in the results is the significant increase in I-AUC achieved by the MC variants, in comparison to the best existing methods. We believe this occurs due to the ensemble nature of MC. The approach of using multiple predictors reduces the chance of overfitting,

and therefore increases the likelihood of a correct overall prediction of each image. On the MVTec dataset, MC4 achieves 0.9994 I-AUC, meaning it almost perfectly separates the images.

The MC2, MC4, and MC8 takes roughly 20, 40 and 80 minutes to train on a dataset with 1000 train images. The MC2, MC4, and MC8 models have an single image inference time of 0.036, 0.06, and 0.25 seconds respectively. A 24GB GPU is required to achieve these values.

6 Conclusion

We introduce a new anomaly detection dataset, Industrial Defects in the Wild (IDW). This dataset overcomes the shortfalls of previous datasets by providing varied-perspective images of the real industrial inspections. Previous datasets contain images from the same perspective taken in controlled conditions, which are not representative of the real world. We present a novel algorithm, MultiCore (MC). This algorithm trains an ensemble of nearest neighbours predictors with varying hyperparameters. Through a novel synthetic anomaly schema and supervised learning, we achieve state-of-the-art I-AUC, P-AUC, and PL values on the MVTec and IDW datasets.

A limitation of the IDW is its specialism to industrial problems. A limitation of MC algorithm is the complexity of training multiple Patchcore+ models, and the memory requirement of storing multiple memory banks. A further limitation is the choice of the number of Patchcore+ models to choose, as different values perform better on different metrics.

Funded by Rolls-Royce Defence; supported by UK Research and Innovation (UKRI), grant reference number EP/S023437/1.

References

[1] Bast. polygenerator. https://github.com/bast/polygenerator, 2021. Accessed on April 20, 2023.

[2] K. Batzner, L. Heckler, and R. Konig. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 127–137, Los Alamitos, CA, USA, jan 2024. IEEE Computer Society. doi: 10.1109/WACV57701.2024.00020. URL https://doi.ieeecomputersociety.org/10.1109/WACV57701.2024.00020.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad - a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2019. doi: 10.1109/CVPR.2019.00982.

[4] Hermann Blum, Paul-Edouard Sarlin, Juan I. Nieto, Roland Siegwart, and Cesar Cadena. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *CoRR*, abs/1904.03215, 2019. URL http://arxiv.org/abs/1904.03215.

[5] Luca Bonfiglioli, Marco Toschi, Davide Silvestri, Nicola Fioraio, and Daniele De Gregorio. The eyecandies dataset for unsupervised multimodal anomaly detection and localization, 2022. URL https://arxiv.org/abs/2210.04570.

[6] Yunkang Cao, Xiaohao Xu, Zhaoge Liu, and Weiming Shen. Collaborative discrepancy optimization for reliable image anomaly localization. *IEEE Transactions on Industrial Informatics*, pages 1–10, 2023. doi: 10.1109/TII.2023.3241579.

[7] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/d67d8ab4f4c10bf22aa353e27879133c-Paper-round2.pdf.

[8] ChrisFix. What does the inside of an engine with a bad head gasket look like?, youtube, https://www.youtube.com/watch?v=jpozuzakeoe. URL https://www.youtube.com/watch?v=JpozuZakEoE.

[9] Cool-Xuan. Cool-xuan/msflow: The official code for "msflow: Multi-scale normalizing flows for unsupervised anomaly detection". URL https://github.com/cool-xuan/msflow.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9727–9736, 2022. doi: 10.1109/CVPR52688.2022.00951.

[12] 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.

[13] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[14] Aircraft Maintenance Engineer. Engine borescope inspection after bird strike - airbus a330, youtube, https://www.youtube.com/watch?v=zv29vsuzwg4. URL https://www.youtube.com/watch?v=zv29VSUZWg4.

[15] Matej Grcic, Petra Bevandic, and Sinisa Segvic. Dense open-set recognition with synthetic outliers generated by real NVP. *CoRR*, abs/2011.11094, 2020. URL https://arxiv.org/abs/2011.11094.

[16] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1819–1828, 2022. doi: 10.1109/WACV51458.2022.00188.

[17] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *ICML*, 2022.

[18] RVI Ltd Remote Visual Inspections. Cf6 80 hpt ngv, youtube, https://www.youtube.com/watch?v=yzzlqbpc9iy.

[19] Rosana El Jurdi, Caroline Petitjean, V. Cheplygina, and Fahed Abdallah. A surprisingly effective perimeter-based loss for medical image segmentation. In *International Conference on Medical Imaging with Deep Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:233323389.

[20] Yeongmin Kim, Huiwon Jang, DongKeon Lee, and Ho-Jin Choi. Altub: Alternating training method to update base distribution of normalizing flow for anomaly detection, 2022. URL https://arxiv.org/abs/2210.14913.

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[22] L1. Dagm dataset. https://universe.roboflow.com/l1-9rj1y/dagm, feb 2023. URL https://universe.roboflow.com/l1-9rj1y/dagm. visited on 2023-03-05.

[23] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 10: 78446–78454, 2022. doi: 10.1109/ACCESS.2022.3193699.

[24] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. *CoRR*, abs/2104.04015, 2021. URL https://arxiv.org/abs/2104.04015.

[25] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20402–20411, June 2023.

[26] Weixin Luo, Wen Liu, Dongze Lian, Jinhui Tang, Lixin Duan, Xi Peng, and Shenghua Gao. Video anomaly detection with sparse coding inspired deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[27] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06, 2021. doi: 10.1109/ISIE45552.2021.9576231.

[28] Garage Noise. How to repair body damage on a car. auto body tech diy, youtube, https://www.youtube.com/watch?v=xbxrldbja3c. URL https://www.youtube.com/watch?v=XBXRLdbJA3c.

[29] International Association of Certified Home Inspectors (InterNACHI). Structural inspection of a house, youtube, https://www.youtube.com/watch?v=6rcxihivs_g. URL https://www.youtube.com/watch?v=6RcXIHivs_g.

[30] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *CoRR*, abs/2007.02500, 2020. URL https://arxiv.org/abs/2007.02500.

[31] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L. Ball, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Mura: Large dataset for abnormality detection in musculoskeletal radiographs, 2017. URL https://arxiv.org/abs/1712.06957.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[33] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14298–14308, 2022. doi: 10.1109/CVPR52688.2022.01392.

[34] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. *arXiv preprint 10.48550/ARXIV.2210.07829*, 2022. doi: 10.48550/ARXIV.2210.07829. URL https://arxiv.org/abs/2210.07829.

[35] Alexander D. J. Taylor, Phillip Tregidgo, Jonathan James Morrison, and Neill D. F. Campbell. VisionAD, a software package of performant anomaly detection algorithms, and proportion localised, an interpretable metric. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=o5kYH7bNe3.

[36] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7, 07 2020. doi: 10.1186/s40537-020-00320-x.

[37] Chin-Chia Tsai, Tsung-Hsuan Wu, and Shang-Hong Lai. Multi-scale patch-based representation learning for image anomaly detection and segmentation. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3065–3073, 2022. doi: 10.1109/WACV51458.2022.00312.

[38] Qian Wan, Liang Gao, Xinyu Li, and Long Wen. Unsupervised image anomaly detection and segmentation based on pre-trained feature mapping. *IEEE Transactions on Industrial Informatics*, 2022. doi: 10.1109/TII.2022.3182385.

[39] Qian Wan, Cao YunKang, Liang Gao, Shen Weiming, and Xinyu Li. Position encoding enhanced feature mapping for image anomaly detection. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022.

[40] Minghui Yang, Peng Wu, Jing Liu, and Hui Feng. Memseg: A semi-supervised method for image surface defect detection using differences and commonalities, 2022. URL https://arxiv.org/abs/2205.00908.

[41] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2020.

[42] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *CoRR*, abs/2111.07677, 2021. URL https://arxiv.org/abs/2111.07677.

[43] V. Zavrtanik, M. Kristan, and D. Skocaj. DrÆm – a discriminatively trained reconstruction embedding for surface anomaly detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8310–8319, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00822. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00822.

[44] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 392–408, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20056-4.