

# ICONIC-444: A 3.1-Million-Image Dataset for OOD Detection Research

## Supplementary Material

In the following, we provide detailed insights into our dataset (Appendix A), covering its acquisition setup, process, and quality assurance protocols. Additionally, we outline the experimental setup (Appendix B) and provide detailed evaluations (Appendix C) that complement the results reported in the main manuscript. We extend our analysis to larger and more complex architectures (Appendix D), explore OOD detection performance using a vision language model (VLM) (Appendix E), and conduct ablation studies regarding the correlation between ID and OOD performance (Appendix F) and the impact of data corruption (Appendix G).

### A. Dataset and Quality Details

Ensuring the highest quality of recordings and maintaining class purity is critical to our dataset’s utility and, thus, a high priority during data acquisition. We follow a stringent acquisition protocol and implement comprehensive data cleaning and quality control measures to ensure the data meets rigorous standards for accuracy and consistency.

#### A.1. System Setup

Our data is acquired using a sorting machine prototype designed to closely mimic real-world applications. The setup (see Fig. 1) includes key components such as a vibrating conveyor, chute, camera, and lighting configuration, all optimized for capturing high-quality images of food objects, which serve as the in-distribution (ID) classes in our defined benchmark tasks.

The chute is adjusted to a 60-degree slope, and the camera is positioned at a 90-degree angle to the chute, ensuring it is perpendicular to the object’s motion for optimal image capture. The camera’s scan line is located just beyond the chute endpoint, capturing the objects as they enter free fall. The lighting system ensures uniform illumination and minimizes shadows. A blue LED backlight, aligned with the camera’s scan line, provides consistent contrast, while white LED lights at 45-degree angles above and below the camera, and focused on the scan line, illuminate the objects from multiple directions. The selection of a blue backlight is a deliberate, industry-driven choice. Uniform, narrow-band blue illumination provides high, stable contrast against the color spectrum of virtually all food items, supporting reliable and computationally efficient foreground–background segmentation in high-throughput bulk sorting. Camera settings, such as exposure, gains, aperture, and focus, are calibrated to capture high-resolution images with balanced brightness and con-

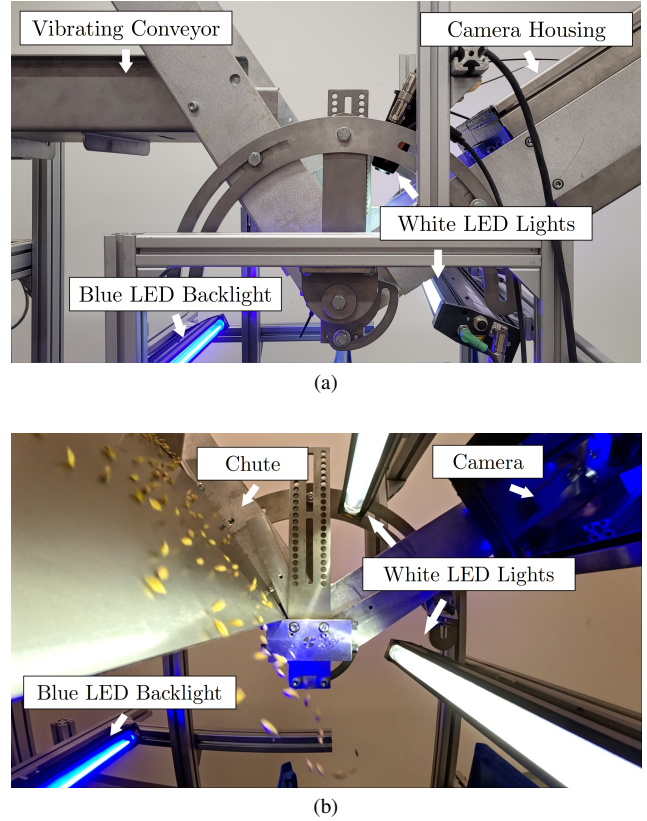


Figure 1. Views of the sorting machine prototype. (a) External overview, highlighting the vibrating conveyor, blue LED backlight, white LED lights, and camera housing. (b) Internal perspective, illustrating the blue LED backlight, white LED lights, chute, and camera within the machine’s main structure.

trast, tailored to the reflective properties and appearances of food items.

We capture our non-food classes using the same setup, which may result in overexposure, underexposure, reflections, or other artifacts, particularly for transparent objects like glass. However, we deem this typical for out-of-distribution data: since the system is optimized for food sorting, such effects are inevitable when OOD objects fall through the machine, making them perfectly normal and expected for OOD data. Including these classes in our ICONIC-444, we ensure the dataset reflects real-world constraints, where systems tailored to specific tasks naturally encounter such artifacts. Figs. 9 to 11 show sample images representing all classes included in ICONIC-444.

To clearly understand our setup and acquisition process, we also include a supplementary video showing the sorting

machine prototype and sample recordings.

## A.2. Acquisition Protocol

**Setup verification.** We use a white balance target as a reference to verify camera calibration and ensure consistent illumination across all recordings. The target’s intensity profiles are compared against initially saved references to detect deviations in lighting, camera settings, or hardware configuration. These checks ensure uniformity across acquisition sessions, preventing data quality shifts due to environmental factors or subtle hardware changes.

**Acquisition process.** We follow a sequential acquisition process to mitigate the risk of environmental contamination and ensure the integrity of all classes. Each acquisition session is organized into groups, with one class prepared and recorded at a time. This approach is essential for fine-grained classes within the same group, where cross-contamination between classes is nearly impossible to detect and rectify post-acquisition. Objects are placed on a vibrating conveyor and manually inspected to ensure the removal of rare foreign materials as an additional precaution to maintain the dataset’s high cleanliness standards. After completing the acquisition, we thoroughly clean the machine to prevent cross-contamination between different classes. During this procedure, we inspect the system for residual particles (*e.g.*, dust or debris) that could affect subsequent classes.

## A.3. Data Cleaning and Quality Control

We implement a comprehensive data cleaning and quality control process after acquisition to remove poorly acquired data and further improve the quality of our dataset. These procedures are designed to exclude contaminants and eliminate issues such as objects outside the illumination range or objects that are out of focus.

Certain classes, for example in the dried fruits, spices, or pulses groups, naturally contain contaminants like broken stems or pieces of skin that can separate from the main objects. These classes require more stringent cleaning procedures to ensure that any environmental or natural contaminants are systematically removed. The following describes the steps we implemented in the data cleaning and quality control process to achieve these goals.

Our semi-automatic data cleaning process is performed on a class-by-class basis and follows a structured, iterative approach. The goal is to ensure that each class in the dataset is free from contamination:

For each class, we first extract a set of handcrafted features, such as area, the mean and standard deviation of each color channel, bounding box dimensions, aspect ratio, and object brightness. Additionally, we compute feature vectors

from the penultimate layer of a ResNet50 model pre-trained on ImageNet to capture high-level semantic information.

The cleaning process begins by identifying extreme values (both minimum and maximum) for these handcrafted features. A class-specific threshold is set for selecting a fixed percentage of these extreme cases, which are then manually inspected. This process is iterative, continuing until no outliers or contaminants remain. Throughout this step, images are labeled as either *accepted* or *rejected*.

Next, we use the feature vectors to calculate the  $k$ -th nearest neighbor distance for all non-*rejected* images. Again, thresholds are determined on a class-by-class basis, with a percentage of images having the largest distances manually reviewed. This step is repeated until no further outliers are detected.

Once the initial cleaning is complete, we train a random forest classifier for each class using the manually labeled *accepted* and *rejected* images based on the extracted feature vectors. This classifier is then used to predict the status of images that were not manually inspected. We manually review all images classified as *rejected* to ensure no contaminations remain.

For classes where manual defects are introduced, all images are manually inspected to ensure the intended defects are accurately captured and visible in the image.

To further ensure the dataset is free from critical cross-contaminations and mislabels, we conduct a final verification step aimed at minimizing human error, especially in closely related classes. We organize classes into subgroups (*e.g.*, pasta, beans, nuts) and perform a cross-validation procedure by splitting the data 50/50 within each subgroup, training a ResNet18 on one half, and evaluating it on the other. This allows us to detect potential cross-contaminations, particularly in closely related classes such as *nuts acorn* and *nuts acorn damaged* or *kernel sunflower peeled* and *kernel sunflower white*. Incorrectly classified images are manually reviewed. Note that this is only possible for classes where manual inspection can reveal the true class of the object—for example, distinguishing between different breeds of wheat is not feasible for us. However, due to the quality protocols of the producers from which we sourced them, we are very confident in the high purity of these classes.

Finally, all data undergoes an independent review by individuals not involved in the acquisition or cleaning process. This additional verification step helps identify and quantify errors introduced during cleaning, ensuring the dataset is of high quality and reliability. Fig. 8 shows classes with natural contaminants, such as broken stems from dried fruits or shells from beans and lentils, which were flagged and rejected during the cleaning process.

Trainset	Testset		
	Almond	Almond w/o Bg	Almond w/o Fg
Almond	95.60	91.35	13.97
Almond w/o Bg	95.79	94.97	13.97
Almond w/o Fg	18.81	18.81	18.81

Table 1. ResNet18 model accuracy (in percentage) for the Almond task, trained and evaluated on three dataset versions: with background (Almond), with the background removed (w/o Bg), and with the foreground removed (w/o Fg). Each model was trained on a specific version and tested across all three to evaluate the influence of background noise on classification performance.

#### A.4. Environmental Noise

A further test we conduct focuses on detecting potential background noise and changes in the data acquisition setup between classes. Such changes could be caused by environmental factors, such as light sources or shadows, and might allow models to distinguish between objects based on non-object features.

For this test, we focus on the Almond task and create three variations of the dataset. The first version consists of the original images, serving as the control set. In the second version, we segment the objects and replace the background with a uniform blue color to eliminate any potential background patterns caused by environmental effects. In the third version, we mask the objects entirely, replacing them with a uniform blue color to test whether the model is still able to make correct predictions based solely on background patterns.

We train ResNet18 models on each version of the dataset using the same training parameters as described in the main manuscript and evaluate their performance across all three dataset variations. Tab. 1 shows that models trained on the original Almond dataset and the dataset with the background removed (w/o Bg) achieve similarly high accuracy when tested on both the original and w/o Bg test sets. This indicates that the model does not use background features to classify objects.

As expected, both models achieve only around 14% accuracy on the w/o Fg (foreground removed) dataset, confirming that the informative content is solely in the foreground, with no meaningful background patterns. The model trained on the w/o Fg dataset performs poorly across all test sets, further demonstrating that background information offers no useful guidance for predictions. The cross-entropy loss stays near  $\log(N)$  (where  $N$  is the number of ID classes), indicating the model is merely guessing and unable to learn meaningful representations.

Tests on other ID data, such as the wheat classes, show similar results, confirming that our data is free from information leakage in the background and that data quality is

consistent across the different classes in our dataset.

#### A.5. Limitations

Please note that it is virtually impossible to ensure correct labeling for every single image in our dataset. Manually checking over 3.1 million images is impractical and would not guarantee proper labeling due to human error and the difficulty of visually distinguishing closely related classes, such as different grain varieties.

To estimate the labeling accuracy, we conduct an additional manual inspection of 144 images per class across all 444 classes, totaling 63936 images (approximately 2% of the dataset). This inspection is performed by individuals not involved in the data acquisition or cleaning process to minimize bias. Across all classes, we identify a total of 25 images containing unexpected objects, which indicates that the fraction of mislabeled data within our dataset is well below 0.05%. These unexpected objects are not due to cross-contamination between different classes — thanks to our stringent sourcing and acquisition protocols, cross-contamination is extremely rare or nonexistent. Instead, the samples found are primarily natural contaminants (*e.g.*, dry fruit skin with high visual similarity to the actual dry fruit).

#### A.6. Societal and Ethical Impact

Our dataset has been constructed with careful attention to ethical considerations. No personal data were collected, processed, or included in the construction of this dataset. While comprehensive, the dataset cannot consider all possible real-world scenarios, and it tends to focus on foods commonly used in specific regions of the world. OOD detection methods tested on our dataset may not account for every potential OOD instance encountered in practice. However, we consider the impact of these limitations relatively minor because we took great care of sourcing a wide variety of different products for ICONIC-444.

### B. Experimental Details

#### B.1. Implementation Details

**Software setup.** Our implementation builds on the OpenOOD framework [45, 47], extended by [18] (modified GRAM [35] implementation and added support for CompactTransformer [9]). OpenOOD currently serves as the most comprehensive framework for evaluating state-of-the-art OOD detection methods. Our main modifications include: (i) integrating our four benchmark tasks into the framework, (ii) adding support for evaluating custom-defined OOD groups, (iii) enabling the evaluation of custom metrics, (iv) incorporating ATS [20] as an additional post-hoc OOD detection method, (v) adding the option to generate synthetic OOD samples for in-distribution (ID) datasets (building on the codebase of Bitterwolf *et al.* [3]), and pro-

Model	Pretrained	Params	Inference Time	timm name
CCT-7.7x2	—	4.5M	2.65ms	—
ResNet18	—	11.6M	2.41ms	—
ConvNeXt-P	ImageNet-1k	9.0M	2.41ms	convnext_pico.d1.in1k
ConvNeXt-T	ImageNet-12k	28.6M	3.69ms	convnext_tiny.in12k.ft.in1k
ViT-S/16	ImageNet-21k	22.1M	3.68ms	vit_small_patch16_224.augreg.in21k.ft.in1k
ViT-B/16	ImageNet-21k	86.6M	4.44ms	vit_base_patch16_224.augreg2.in21k.ft.in1k
CLIP-B/16	openai	150M	12.71ms	—

Table 2. Overview of the evaluated models.

viding support for models from the timm-repository [43] (e.g. ConvNeXt, ViTs). For CLIP-based OOD detection methods (MCM [29], GL-MCM [30], GalLop [21]), we extend their original repositories.

**Computational environment.** We run our experiments on a server equipped with an Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz, paired with three NVIDIA GeForce RTX A4000 GPUs. The computational setup operates on Ubuntu 22.04, incorporating PyTorch 2.0.1, and leverages CUDA 11.8 and cuDNN 8.7.0.

## B.2. Training

Our main evaluations from the main manuscript focus on two architectures: a Convolutional Neural Network (CNN) using ResNet18 [10] and a Transformer using the Compact Transformer (CCT-7.7x2.224) [9]. These models were selected for their efficiency, effectiveness, and practicality while balancing performance, runtime efficiency, and computational cost. This makes them particularly suitable for real-world applications where cloud computation is not feasible due to latency constraints, such as real-time quality control, automated sorting systems, and embedded AI applications in industrial environments. By ensuring both research accessibility and deployment readiness, these models provide a strong foundation for OOD detection across diverse settings.

To further broaden architectural diversity, we additionally evaluate two larger and more complex architectures: ConvNeXt [26] and Vision Transformers [7] (ViT). For ConvNeXt, we include both a compact variant (ConvNeXt-P, or "pico"), which has a parameter count comparable to ResNet18, as well as the larger ConvNeXt-T. Additionally, we evaluate two ViT models, ViT-S and ViT-B. Unlike ResNet18 and CCT, these models leverage pre-trained weights for fine-tuning—to mitigate overfitting—except for ConvNeXt-P, which is trained from scratch to match the training paradigm of our baseline models. Tab. 2 gives an overview of the evaluated models. All pre-trained model weights were taken from the public available timm-repository [43], except for the zero-shot CLIP model, which is available via github<sup>1</sup>.

<sup>1</sup><https://github.com/openai/CLIP>

Method	Hyperparameter Search Space
GRAM [35]	
ViM [42]	
KNN [39]	$K \in \{50, 100, 200, 500, 1000\}$
OpenMax [1]	
ATS [20]	
MDS [22]	
ODIN [23]	temperature $T \in \{1, 10, 100, 1000\}$ perturbation mag. $\sigma \in \{0.0, 0.0007, 0.0014, 0.0028\}$
RMDS [34]	
ReAct [38]	percentile $\in \{70, 80, 85, 90, 95, 99\}$ gamma $\in \{0.01, 0.1, 0.5, 1, 2, 5, 10\}$ top-M classes $\in \{2, 4, 5, 10, 15, 20, 25, 50, 100, 200, 300\}$
GEN [25]	
TempScale [8]	
MSP [13]	
KLM [15]	
SCALE [44]	percentile $\in \{65, 70, 75, 80, 85, 90, 95\}$
ASH-s [6]	percentile $\in \{65, 70, 75, 80, 85, 90, 95\}$
DICE [37]	percentile $\in \{10, 30, 50, 70, 90\}$
MLS [15]	
EBO [24]	
SHE [46]	
GradNorm [17]	
ASH-b [6]	percentile $\in \{65, 70, 75, 80, 85, 90, 95\}$
RankFeat [36]	

Table 3. Hyperparameter search space for each OOD detection method.

Unless stated otherwise, all models are trained from scratch. ResNet18 models are trained for 100 epochs using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of  $1 \times 10^{-1}$  and cosine annealing with a weight decay of  $5 \times 10^{-3}$ . Compact Transformer models are trained for 150 epochs using the AdamW [27] optimizer with a learning rate of  $5 \times 10^{-4}$  and a weight decay of  $6 \times 10^{-2}$ . ConvNeXt-P follows the same setup and is trained from scratch for 100 epochs, while ConvNeXt-T, ViT-S, and ViT-B are fine-tuned for 50 epochs using AdamW with the same learning rate and weight decay as CCT. To mitigate class imbalance, we use per-class weights in the cross-entropy loss, with  $w_c = 1/f_c$  where  $f_c$  is the class frequency on the training split; weights are normalized to have mean 1. The same scheme is applied for all training recipes and fine-tuning runs.

**OOD detection methods.** Since our dataset differs significantly from commonly used benchmarks like CIFAR10/100 [19] and ImageNet [5], on which most OOD detection methods are developed, and given that some of the evaluated methods are sensitive to hyperparameters, we optimized these methods using both the ID and our specifically designed OOD validation set. Tab. 3 shows the 22 methods along with their respective hyperparameter search spaces, where applicable.

## B.3. Extreme- and Synthetic-OOD Details

To provide a comprehensive and diverse OOD test set, we add extreme and synthetic OOD samples in addition to the near- and far-OOD samples sourced from ICONIC-444:



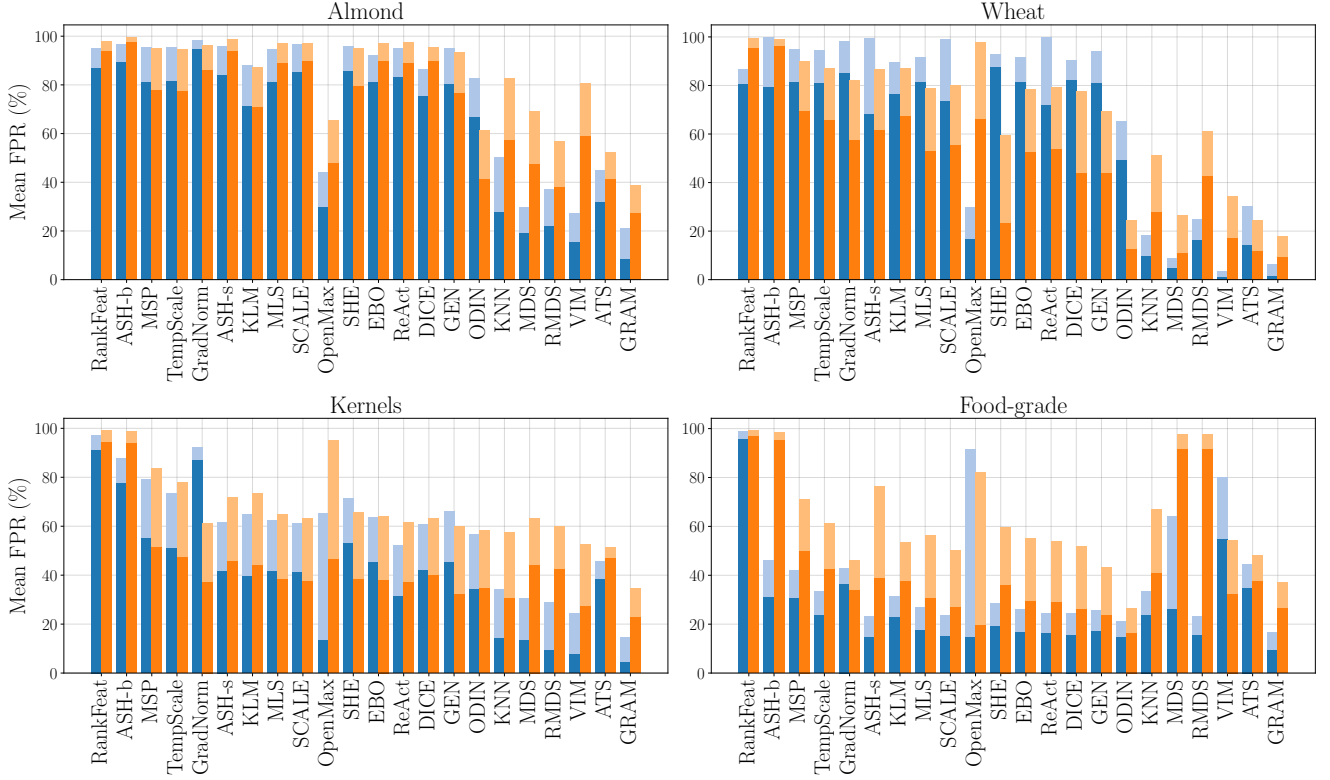


Figure 2. Comparison of mean FPR at 95% and 99% TPR across all methods for each individual task (Almond, Wheat, Kernels, and Food-grade) and architectures (ResNet18 and CCT), averaged over three runs and our four OOD categories (near, far, extreme and synthetic). The blue and orange bars represent the ResNet18 and CCT architectures, respectively. For each method, the dark-shaded portion of the bar indicates the mean FPR95, and the light-shaded segment on top represents the additional increase up to FPR99. Methods are ordered by their overall mean FPR99 score across tasks, reflecting their relative performance ranking.

**Extreme-OOD.** We add samples from four commonly known datasets—ImageNet [32], iNaturalist [40], Places365 [48], and Textures [4]—to our OOD test set and categorize them as extreme-OOD. These samples are coarse-grained OOD examples compared to the ICONIC-444 data (and thus to all considered ID tasks), allowing our benchmark to test OOD detection performance on samples with no semantic correlation to the ID data. Fig. 12 shows sample images from all four datasets.

**Synthetic-OOD.** We generate 500 samples each for 25 distinct synthetic OOD types. Out of these, 16 types are taken from literature, *i.e.* uniform noise [13], Gaussian noise [13], Rademacher noise [14], IN pixel permutations [11], black [3], white [3], monochrome [3], tri-colour [3], horizontal stripes [3], vertical stripes [3], smooth noise [2, 11, 28], smooth noise+ [3], smooth color [3], smooth IN pixel permutations [11], and blobs [14]. For these, we adhere to the configuration settings detailed by Bitterwolf *et al.* [3] to ensure consistency and reproducibility in our comparisons. Additionally, we extend the syn-

thetic OOD dataset with variants more closely aligned with the ICONIC-444 dataset. Below, we describe these newly introduced types in further detail:

- **IN channel permutations:** We choose a random image from the ID data and randomly shuffle its color channels.
- **red:** The red channel is set to 1.0 the others to 0.
- **green:** The green channel is set to 1.0 the others to 0.
- **blue:** The blue channel is set to 1.0 the others to 0.
- **random shape:** The background of the image is set to blue (with the blue channel at 1.0 and the others at 0.0). A random RGB color is uniformly sampled from the range  $[0.4, 0.78]^3$ , and the red and blue channels are shifted by 0.15 and  $-0.3$ , respectively. A random shape (triangle, quadrangle, pentagon, octagon, or ellipse) is then drawn using the pre-generated color.

Fig. 13 provides examples of each of the 25 synthetic OOD types used in this study.

## C. Detailed Results

In the following, we (i) summarize per-task trends on the baseline backbones (ResNet-18, CCT; Appendix C.1),

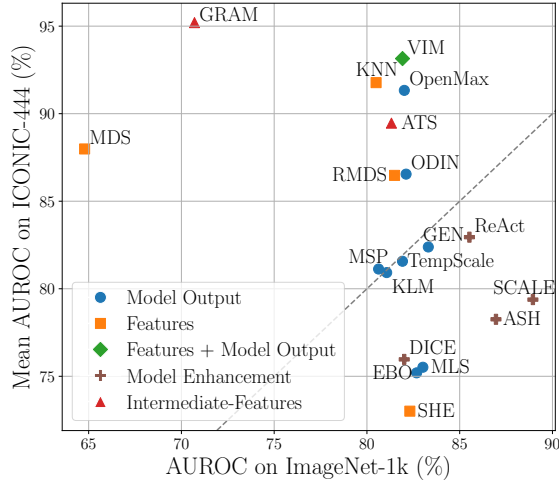


Figure 3. Comparison of the OOD detection methods based on the AUROC across the ICONIC-444 and the ImageNet benchmark.

(ii) assess robustness under stricter operating points (FPR95→FPR99; Appendix C.2), and (iii) analyze feature spaces and contrast ICONIC-444 with ImageNet-1k (Appendix C.3).

### C.1. Details on Baseline Architectures

A detailed breakdown of results on individual tasks is presented in Fig. 2, where the mean FPR at 95% and 99% TPR for each method and architecture is shown. The general outperformance of ResNet18 over the more recent CCT architecture aligns with findings from other benchmarks [47], where this is attributed to many post-hoc OOD methods being overfit to the representational style of traditional CNNs after being implicitly developed and tuned on ResNet backbones. Methods that leverage information from layers beyond the model output, such as GRAM, ViM, KNN, and MDS, consistently achieve low FPRs across all tasks, with this effect being more pronounced for the ResNet architecture. However, the substantial performance gap between these methods and model enhancement methods (*e.g.*, ReAct, DICE, ASH), as well as baseline methods (*e.g.*, MSP, MLS, and EBO), progressively narrows from the Almond task to the more complex (in terms of ID classification) Food-grade task.

Interestingly, model enhancement methods begin to outperform methods like MDS, KNN, and ViM on the Food-grade task, which involves a larger number of ID classes. This shift suggests that an increasing number of ID classes may favor model enhancement approaches, possibly because methods like ReAct, DICE, and ASH were developed on diverse datasets like ImageNet, which aligns more closely with the Food-grade task’s 324 ID classes. GRAM remains a high-performing method across all tasks and re-

tains its advantage even as the number of classes and class granularity increase.

All methods demonstrate task-dependent performance variations, with a relatively consistent degradation from FPR95 to FPR99 within each task. OpenMax, however, deviates from this trend, showing a notably larger increase in mean FPR on the Kernels and Food-grade tasks, where its FPR more than doubles.

Furthermore, the results suggest that as the number of ID classes increases, the overall OOD detection performance across all methods improves. This observation contrasts with findings by Huang *et al.* [16], who show that the FPR of baseline methods increases as the number of classes grows, particularly on ImageNet where classes are randomly sampled. In our dataset, however, the controlled environment, with a uniform background and many fine-grained classes that transition smoothly to more coarse-grained ones, appears to positively affect OOD detection performance. Nevertheless, even for state-of-the-art approaches, ICONIC-444 is still far from being saturated, as demonstrated by the high FPR rates.

In summary, our results reveal that all OOD methods are highly sensitive to the complexity and granularity of ID tasks, as well as the architecture used, underscoring the importance of diverse OOD detection benchmarks. Such benchmarks are crucial for accurately evaluating each method’s strengths and weaknesses, enabling more informed selection for real-world applications, and ultimately enhancing the reliability and safety of OOD detection methods in practical settings.

### C.2. Robustness under Stricter Evaluation

Tab. 6 shows the average OOD detection performance along with the standard deviation for each method, evaluated over three random seeds and two architectures (ResNet18 and CCT) across the four OOD categories. As expected, performance degrades significantly for all methods across all OOD categories when moving from 95% to 99% TPR. This trend is particularly pronounced in the near- and far-OOD categories and remains evident in the extreme- and synthetic-OOD samples.

Among all methods, GRAM and ATS stand out as the only two that effectively handle extreme- and synthetic-OOD samples, at least in the FPR95 setting. GRAM shows significantly better performance in the FPR99 setting in these categories, likely due to its use of higher-order moments for modeling ID data, whereas ATS relies solely on the mean. This more comprehensive statistical modeling allows GRAM to capture finer distinctions between ID and OOD samples, making it the more robust approach also for the more strict evaluation with 99% TPR.

These results emphasize the importance of considering both AUROC and FPR when evaluating OOD detection

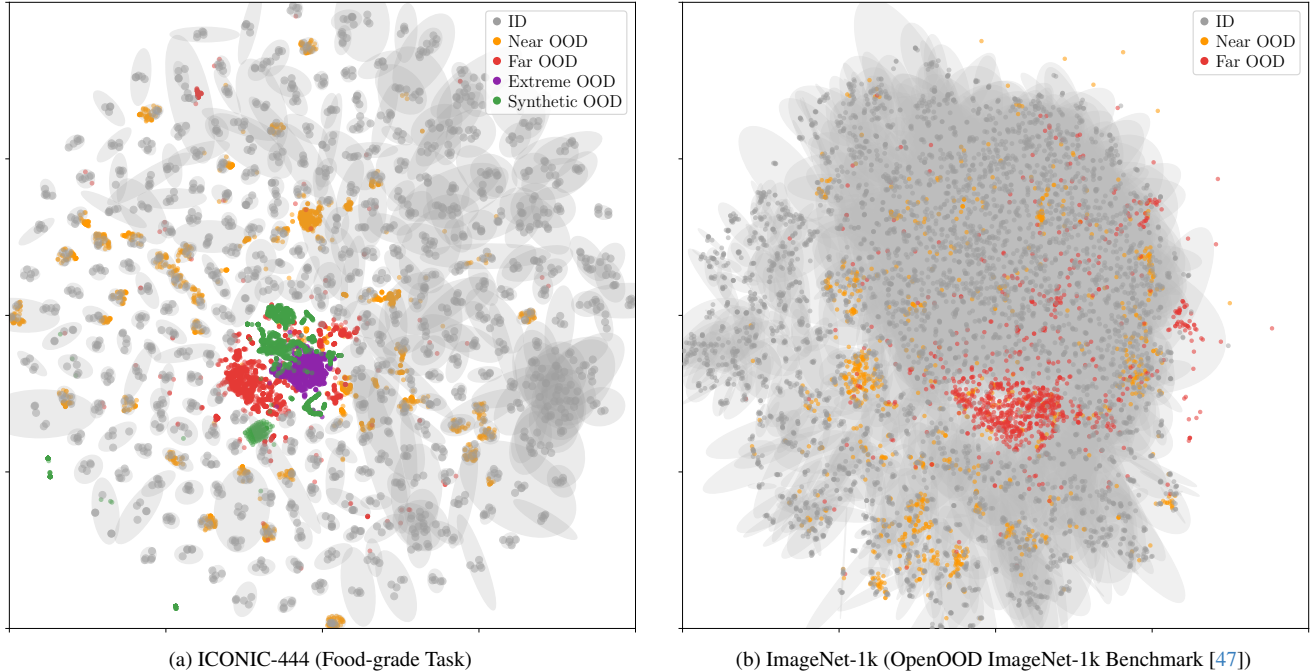


Figure 4. t-SNE visualization of penultimate layer feature embeddings from (a) the ICONIC-444 Food-grade task and (b) the ImageNet-1k benchmark. Features for ICONIC-444 were extracted from a ResNet18 model, while ImageNet-1k features were extracted from a ResNet50. The t-SNE visualization includes only a subset of samples to enhance readability. For ID data, 5 samples per class are shown; for OOD data, only a small subset of classes is depicted.

methods. For example, MSP achieves an average AUROC ranking, yet at the critical FPR99 operating point, it performs worse than all methods except ASH-b and RankFeat. This highlights the need for stricter evaluation metrics beyond AUROC alone, especially for real-world applications where minimizing false positives is crucial.

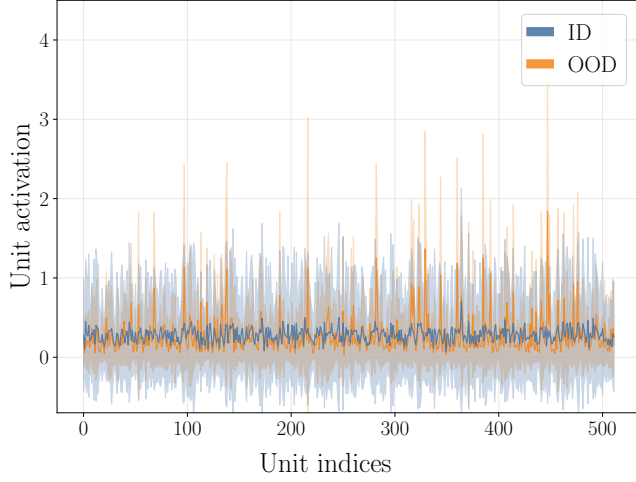
### C.3. Need for Complementary Benchmarks

Fig. 3 compares the OOD detection performance (mean AUROC) of evaluated methods on ICONIC-444 against their reported performance on the ImageNet-1k benchmark from OpenOOD [47]. The comparison reveals that no single class of methods is universally superior; instead, the optimal approach is highly dependent on the dataset’s intrinsic properties. Notably, feature-based methods (*e.g.*, GRAM, ViM, KNN, ATS) significantly outperform other approaches on ICONIC-444, while model enhancement techniques (*e.g.*, ASH, SCALE, ReAct) are the top performers on the ImageNet benchmark.

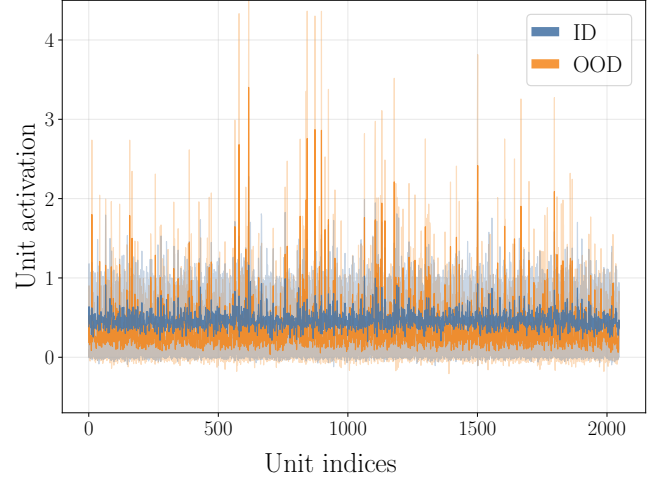
We hypothesize that this divergence is caused by fundamental structural differences between the datasets’ feature spaces, an idea supported by both visual and quantitative analysis. Unless stated otherwise, all feature space analyses use penultimate layer activations: ResNet18 for ICONIC-444 (Food-grade task) and ResNet50 for ImageNet-1k. The t-SNE visualizations in Fig. 4 provide a clear visual in-

tuition, showing that ICONIC-444’s features form more compact and distinct clusters compared to the diffuse distribution typical of ImageNet. This observation is confirmed quantitatively. ICONIC-444’s controlled acquisition process (object-centric) yields a lower mean variance of pixel intensity (0.154 vs. 0.226 for ImageNet), a consistent background, and finer-grained class distinctions compared to ImageNet’s broader semantic and textural variability (scene-centric). At the feature level, we observe substantially higher activation sparsity, which is the fraction of near-zero feature values (0.081 vs. 0.005 for ImageNet); and stronger class separation (inter/intra  $\approx 3.67$  vs. 1.31 on ImageNet). These indicators are consistent with a more statistically coherent representation where distance/statistics-based detectors like GRAM and KNN are highly effective at modeling feature distributions.

However, this controlled environment does not imply a simpler task; it creates a different one. The challenge shifts from handling nuisance variables common in *in-the-wild* data—such as cluttered backgrounds—to the more focused task of detecting subtle semantic differences between visually similar classes. Our results, both in the main paper and this supplementary material, show that **even top methods yield enormously high false positive rates** on our near- and far-OOD categories, proving the benchmark’s difficulty.



(a) ICONIC-444 (Food-grade Task, ResNet18)



(b) ImageNet-1k (OpenOOD ImageNet-1k Benchmark [47], ResNet50)

Figure 5. Per-unit penultimate layer activation profiles for ID and the far-OOD category on a) ICONIC-444 Food-grade with ResNet-18 and (b) ImageNet-1k (OpenOOD benchmark) with ResNet-50. The x-axis indexes individual units in the penultimate layer (each position corresponds to one unit), and the y-axis is activation magnitude; the solid line denotes the mean and the shaded region the standard deviation.

			OOD-Dataset									
Method	Pretrained on ImageNet		Near		Far		Extreme		Synthetic		Average	
			FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
ConvNeXt-P	GRAM [35]	✗	34.99± 2.74	91.42± 1.07	22.44± 5.58	94.75± 1.47	<b>0.00</b> ± 0.00	<b>99.99</b> ± 0.00	<b>0.00</b> ± 0.00	<b>99.95</b> ± 0.03	14.36±17.36	96.53± 4.20
	ATS [20]	✗	61.39±10.83	81.29± 3.98	32.74±12.62	91.50± 3.60	<u>0.00</u> ± 0.00	99.83± 0.25	11.29±10.68	95.58± 4.82	26.35±27.02	92.05± 7.94
	VIM [42]	✗	84.09± 2.76	62.21± 2.98	86.74± 6.26	61.77± 5.76	7.23± 7.80	97.83± 3.10	27.11±13.24	88.08± 5.08	51.29±40.24	77.47±18.32
	KNN [39]	✗	61.83± 1.20	83.77± 0.89	60.78±10.10	88.29± 2.81	34.07±10.48	91.76± 3.73	42.14±10.34	89.08± 4.29	49.71±13.80	88.22± 3.32
	SCALE [44]	✗	81.75± 2.69	69.35± 0.19	82.93± 5.51	70.88± 2.64	81.53±12.97	50.24±12.92	71.45±18.25	59.97±10.62	79.41± 5.35	62.61± 9.55
	MSP [13]	✗	83.54± 0.20	73.14± 0.48	84.08± 1.86	74.64± 2.70	87.91± 4.77	57.50± 7.50	85.07± 4.17	61.05± 6.95	85.15± 1.94	66.58± 8.58
ConvNeXt-T	GRAM [35]	✓	36.73± 2.06	93.57± 0.33	40.16± 2.03	94.08± 0.35	<b>0.00</b> ± 0.00	99.90± 0.11	<u>0.77</u> ± 0.28	99.57± 0.12	19.41±22.02	96.78± 3.42
	ATS [20]	✓	43.63± 5.79	86.92± 3.34	27.30± 6.01	92.94± 0.98	<b>0.00</b> ± 0.00	99.99± 0.01	13.67± 0.75	95.83± 0.70	21.15±18.68	93.92± 5.49
	VIM [42]	✓	<u>30.70</u> ± 4.19	93.17± 1.13	31.12± 4.89	94.20± 0.90	0.19± 0.26	99.74± 0.25	7.92± 0.95	98.50± 0.38	17.48±15.82	96.40± 3.21
	KNN [39]	✓	47.05± 1.08	90.60± 0.41	45.21± 2.08	92.93± 0.36	19.53±15.60	97.42± 1.34	25.18± 3.57	96.36± 0.77	34.24±13.94	94.33± 3.14
	SCALE [44]	✓	59.22±10.60	82.53± 7.40	55.46± 2.20	84.24± 6.15	38.69±19.02	93.87± 2.79	32.12± 8.05	91.22± 3.72	46.37±13.03	87.97± 5.44
	MSP [13]	✓	59.48± 0.73	87.71± 1.23	56.42± 4.48	89.42± 0.31	46.01±15.93	93.92± 2.14	35.64± 1.86	93.87± 0.85	49.39±10.83	91.23± 3.15
ConvNeXt-T	GRAM [35]	✓	36.62± 2.04	<u>93.91</u> ± 0.51	41.68± 1.52	93.62± 0.53	<b>0.00</b> ± 0.00	99.93± 0.02	5.30± 1.57	99.27± 0.06	20.90±21.29	96.68± 3.38
	ATS [20]	✓	47.68±13.12	86.85± 4.72	32.84± 5.35	91.94± 0.86	<b>0.00</b> ± 0.00	99.96± 0.03	11.62± 3.55	97.43± 0.69	23.03±21.33	94.04± 5.85
	VIM [42]	✓	<b>23.40</b> ± 1.70	<b>94.65</b> ± 0.43	<u>21.64</u> ± 5.32	<u>95.89</u> ± 1.29	0.00± 0.01	99.98± 0.01	5.41± 0.87	99.12± 0.17	<b>12.61</b> ±11.67	<b>97.41</b> ± 2.55
	KNN [39]	✓	39.70± 2.72	92.34± 0.25	39.07± 1.80	93.91± 0.43	9.89± 2.91	97.90± 0.48	22.43± 6.00	96.14± 1.27	27.77±14.36	95.07± 2.45
	SCALE [44]	✓	65.91± 7.32	80.84± 1.95	64.61± 4.13	82.35± 1.76	66.50±22.32	80.88±11.43	46.69± 5.77	83.25± 1.27	60.93± 9.52	81.83± 1.18
	MSP [13]	✓	55.88± 5.41	87.16± 2.13	56.09± 3.50	88.95± 1.09	50.05±16.83	88.29± 5.49	35.46± 3.82	88.35± 3.40	49.37± 9.69	88.19± 0.75
ViT-S/16	GRAM [35]	✓	33.06± 4.72	93.41± 0.72	<b>19.66</b> ± 4.00	<b>96.29</b> ± 0.52	<b>0.00</b> ± 0.00	99.99± 0.00	1.59± 0.17	99.69± 0.03	13.57±15.76	97.35± 3.11
	ATS [20]	✓	57.84± 3.46	85.15± 0.78	29.27± 3.13	93.10± 0.73	0.04± 0.05	99.98± 0.01	11.85± 0.34	96.80± 0.53	24.75±25.11	93.76± 6.39
	VIM [42]	✓	57.27± 5.09	84.73± 1.94	47.55± 5.86	90.82± 1.84	4.04± 3.12	99.06± 0.56	23.64± 4.25	94.24± 1.17	33.13±23.99	92.21± 6.03
	KNN [39]	✓	62.27± 3.50	80.60± 0.37	58.61± 4.30	86.89± 0.48	40.20± 8.41	91.88± 2.38	57.80± 2.39	82.37± 3.33	54.72± 9.87	85.43± 5.05
	SCALE [44]	✓	55.76±10.88	84.62± 5.43	53.08±11.27	87.89± 4.46	16.10± 8.48	97.18± 1.59	40.82±10.20	88.79± 3.98	41.44±18.10	89.62± 5.35
	MSP [13]	✓	71.50± 3.57	83.40± 1.05	70.68± 3.53	85.94± 0.78	52.30± 7.75	92.83± 1.55	70.68± 3.16	84.05± 1.51	66.29± 9.33	86.56± 4.32
ViT-B/16	GRAM [35]	✓	35.01± 2.57	92.77± 0.72	24.29± 3.68	95.58± 0.54	<b>0.00</b> ± 0.00	99.97± 0.02	2.63± 0.37	99.54± 0.07	15.48±16.97	96.97± 3.42
	ATS [20]	✓	68.38± 4.66	83.35± 0.61	49.41± 8.85	88.02± 2.03	1.03± 1.07	99.82± 0.18	16.86± 0.61	93.39± 0.20	33.92±30.55	91.14± 7.09
	VIM [42]	✓	52.99± 8.99	86.82± 1.84	44.87±15.37	91.73± 1.92	<b>0.00</b> ± 0.01	99.91± 0.06	15.84± 1.40	95.61± 0.72	28.43±24.77	93.52± 5.57
	KNN [39]	✓	49.44± 1.49	88.71± 0.16	51.65± 6.21	90.31± 0.46	51.56±10.62	91.90± 2.34	37.92± 1.55	90.34± 1.05	47.64± 6.56	90.31± 1.30
	SCALE [44]	✓	57.86± 9.06	86.10± 1.99	53.71±11.75	88.33± 2.79	78.39± 9.34	80.83± 4.50	70.33± 8.80	71.90± 4.12	65.07±11.34	81.79± 7.31
	MSP [13]	✓	80.41± 0.46	78.87± 0.43	80.77± 0.30	80.02± 0.19	85.91± 2.34	76.90± 2.89	86.03± 1.66	69.02± 3.13	83.28± 3.11	76.20± 4.96

Table 4. Average OOD detection performance for the Almond task, evaluated across four architectures (ConvNeXt-P, ConvNeXt-T, ViT-S/16, and ViT-B/16) and reported per OOD category (near, far, extreme, and synthetic). Results are reported as mean ± standard deviation over three random seeds. Arrows (↑/↓) indicate whether higher or lower values are better. Cells are color-coded from blue (high performance) to white (low performance). Additionally, the **best** and second-best results in each column are highlighted in bold and underlined, respectively. All values are reported as percentages.



As further evidence, the penultimate layer activation profiles in Fig. 5 show that the ReAct [38] signature observed on ImageNet—near-constant ID means with lower, more variable, positively skewed OOD activations—attenuates on ICONIC-444, where the ID and OOD profiles largely overlap across units. Consequently, activation-shaping heuristics (*e.g.*, ReAct) offer limited gains on ICONIC-444, whereas statistic/distance-based detectors (GRAM, ViM, KNN) remain effective by exploiting the structure of intermediate features.

This analysis empirically demonstrates ICONIC-444’s value as a diagnostic tool that proves **the optimal OOD detection strategy is highly task dependent**. It complements large-scale OOD benchmarks like ImageNet by providing a challenging benchmark for fine-grained OOD, which is essential for progress in safety- and sustainability-critical applications like industrial sorting and recycling.

## D. Evaluation on Complex Architectures

In this section, we analyze the benchmark results on more complex architectures, specifically ConvNeXt and Vision Transformers (ViTs). For OOD detection, we evaluate six post-hoc methods: the three top-performing methods from our main benchmark (GRAM, ATS, and ViM, selected based on average FPR99), KNN (which utilizes penultimate layer features), SCALE (one of the most recent model-enhancement methods), and MSP (the initial OOD detection baseline). Tab. 4 reports the average OOD detection performance for each method over three random seeds on the Almond task, broken down by near, far, extreme, and synthetic OOD categories.

Despite higher capacity and potentially richer feature representations (see model details in Tab. 2), ConvNeXt and ViT do not consistently outperform ResNet in OOD detection, aligning with observations from Zhang *et al.* [47]. A possible explanation is that most OOD detection methods have been primarily designed and tuned for ResNet-based backbones, suggesting they may be implicitly optimized for the representational style of traditional CNNs.

Focusing on ConvNeXt-P, we compare models trained from scratch versus those fine-tuned from ImageNet-pretrained weights. Pretraining generally improves extreme-OOD detection, likely because the semantics learned from ImageNet (*e.g.*, generic object shapes or textures) transfer well to OOD examples that share partial visual characteristics with those seen during pretraining. Also, this improvement is partially explained by dataset overlap: our extreme-OOD category includes ImageNet images for consistency across benchmarks, meaning that pre-trained models have already been exposed to similar data. Although we strictly use the ImageNet test set for OOD evaluation, the semantic structures of these samples remain familiar to the model, potentially inflating performance.

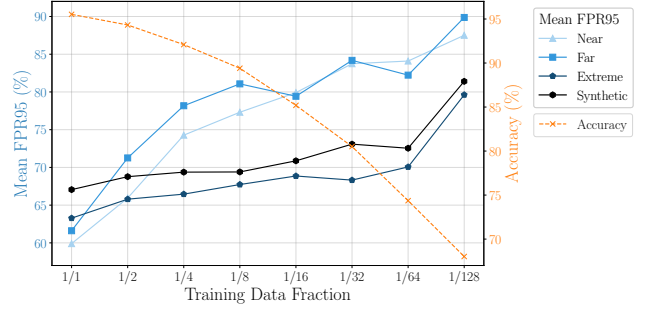


Figure 6. Mean FPR95 for the ResNet18 model across four OOD categories and accuracy for the Almond task, averaged over three seeds and all OOD detection methods, with progressively reduced training set size while keeping validation and test sets fixed.

This underscores the importance of either training models from scratch or ensuring that test-time OOD samples are entirely absent from pretraining data, as this is crucial for strict and robust OOD evaluation. Additionally, this benefit is not uniform across all methods; for instance, GRAM exhibits a notable performance drop on the FPR95 metric when applied to pre-trained models.

In summary, **scaling up model capacity or adopting newer architectures does not automatically yield better OOD detection**. Most importantly, our results confirm that ICONIC-444 remains a challenging benchmark **even for larger and more complex architectures**, underscoring its value in advancing OOD detection research; nevertheless, FPR remains prohibitively high, limiting real-world applicability.

## E. VLM-based OOD Detection Methods

To link ICONIC-444 to recent developments in Image Classification and OOD research, we evaluate how foundation models perform on our benchmark. Foundation models are large, pre-trained models that can be adapted to a wide range of downstream tasks, often outperforming more specialized supervised approaches [33]. In particular, vision language models (VLMs) such as CLIP [33] have demonstrated impressive zero-shot classification capabilities, inspiring several CLIP-based OOD detection methods [31].

We evaluate three CLIP-based OOD detection methods on the Almond task, selected for its simplicity in terms of in-distribution and near-OOD granularity. We consider MCM [29] and GL-MCM [30] (both zero-shot and training-free), as well as GalLop [21], which employs a few-shot approach using 16 ID training images. The standard parametrization of zero-shot CLIP is insufficient, achieving only 20.34% ID accuracy. Thus, we fine-tuned the text prompts and class descriptions (on the ID validation set) and could increase the accuracy to 34.70%. Nonetheless, even with few-shot fine-tuning, GalLop achieves only

Method	ID Image Availability	Training Type	OOD-Dataset									
			Near		Far		Extreme		Synthetic		Average	
			FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
MCM [29]	Zero-shot	Training-free	96.10	43.72	98.65	37.53	99.01	22.94	99.41	9.60	98.29	27.70
GL-MCM [30]	Zero-shot	Training-free	96.31	43.44	98.91	35.96	99.77	18.39	99.14	13.82	98.53	27.90
GalLop [21]	Few-shot	ID Training	<b>89.91</b>	<b>73.67</b>	<b>90.65</b>	<b>72.77</b>	<b>97.88</b>	<b>68.79</b>	<b>66.36</b>	<b>87.25</b>	<b>86.20</b>	<b>75.62</b>

Table 5. CLIP-based (CLIP-B/16) OOD detection performance on the Almond task, measured in FPR95 and AUROC. Arrows (↑/↓) indicate whether higher or lower values are better. All values are reported as percentages.

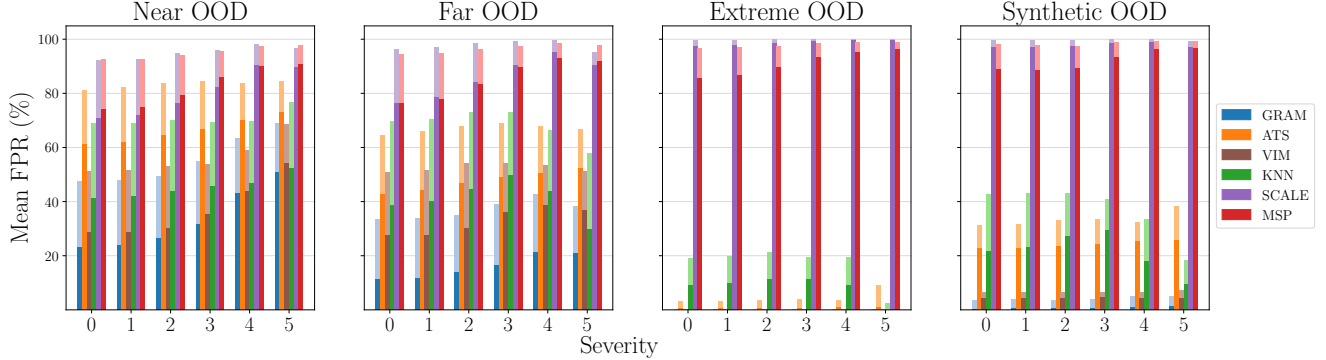


Figure 7. Average OOD detection performance on the Almond task using ResNet18 over three random seeds, measured as FPR at 95% (FPR95) and 99% (FPR99) TPR. Results are reported for each method, OOD category (near, far, extreme, and synthetic), and Gaussian noise corruption with five different severity levels. The dark-shaded portion of each bar represents the mean FPR95, while the light-shaded extension indicates the additional increase up to FPR99.

78.45% accuracy—noticeably lower than our ResNet baselines’ 95.53%.

Tab. 5 shows each method’s OOD detection performance across near-, far-, extreme-, and synthetic-OOD categories. As expected, these CLIP-based approaches generally underperform specialized OOD detection methods. As demonstrated by Radford *et al.* [33] (on the MNIST dataset), CLIP does not directly address the fundamental deep learning’s generalization challenge; instead, it aims to cover a broad diversity of data such that most real-world samples become effectively ID. However, ICONIC-444 contains high-resolution, industrial-domain images that are rarely—if at all—present in publicly available web collections, placing them firmly OOD for CLIP.

Our findings show that, within our benchmark, CLIP underperforms in both ID classification and OOD detection. Moreover, most existing CLIP-based OOD detection methods are evaluated on benchmarks where ID performance is comparable to supervised approaches and where only coarse-grained OOD samples are analyzed [31]. Therefore, our results further underscore the value of ICONIC-444 as a challenging benchmark for VLMs in both classification and OOD detection contexts.

More broadly, our results suggest that CLIP and similar foundation models cannot serve as a universal solution for OOD detection, as they inevitably fail on specialized

datasets outside their training distribution. This aligns with the limitations identified by Radford *et al.* [33], particularly in the context of zero-shot classification. ICONIC-444 demonstrates such domain-specific data, highlighting the limitations of current foundation models in handling fine-grained, high-resolution, and industrial-domain OOD detection tasks, where their pre-training coverage is insufficient.

## F. ID-OOD Performance Correlation Ablation

Fig. 6 shows how progressively reducing training data impacts ID accuracy and OOD detection performance (FPR95) across our four OOD categories. As the training data is reduced, the accuracy steadily declines, and a corresponding decrease in OOD detection performance is observed. This trend is more pronounced for the near- and far-OOD samples, which show a sharper degradation, likely due to their higher semantic correlation with the ID data. In contrast, the extreme- and synthetic-OOD categories, which are less semantically related, exhibit a smaller drop in performance, requiring a more drastic reduction in training data to show a significant decline.

This finding aligns with recent discussions in our field: Vaze *et al.* [41] show a positive correlation between classifier accuracy and OOD detection performance. Recent

work [18] further analyzes this correlation, showing that OOD detection performance relies largely on the method’s ability to separate misclassified ID samples from true OOD samples. Our results extend this understanding by highlighting that the complexity and semantic similarity of OOD samples play a significant role in this correlation.

## G. Ablation on Corruptions

A key objective in the acquisition of ICONIC-444 was to ensure high-resolution, high-quality images, providing a strong benchmark for OOD detection research. While the dataset maintains a high level of visual clarity, minor imperfections naturally occur due to the free-fall acquisition setup. Variations in object speed and positioning during scanning introduce slight illumination inconsistencies, motion blur, or occasional dust accumulation, reflecting real-world environmental challenges. Importantly, the intrinsically high quality of ICONIC-444 allows for controlled degradation, enabling systematic robustness testing for OOD detection models.

Fig. 7 shows a detailed analysis of OOD detection performance on the Almond task using ResNet18 under inference-time image corruption. Specifically, we introduce Gaussian noise with five severity levels  $\sigma = \{0.01, 0.02, 0.03, 0.05, 0.1\}$ , following the corruption methodology proposed by Hendrycks *et al.* [12]. The clean, uncorrupted images correspond to severity level zero.

As expected, increased corruption generally degrades OOD detection performance. However, an exception is observed for KNN on far-, extreme-, and synthetic-OOD categories, where at severity levels four and five, OOD detection performance improves. This suggests that noise may introduce additional cues that help distinguish OOD samples in the case of KNN, similar to the effect exploited by ODIN [23], where input perturbations enhance OOD separability.

Method	OOD-Dataset														
	Near			Far			Extreme			Synthetic			Average		
	FPR95 ↓	FPR99 ↓	AUROC ↑	FPR95 ↓	FPR99 ↓	AUROC ↑	FPR95 ↓	FPR99 ↓	AUROC ↑	FPR95 ↓	FPR99 ↓	AUROC ↑	FPR95 ↓	FPR99 ↓	AUROC ↑
GRAM [35]	<b>35.74</b> ±23.66	<b>54.59</b> ±24.59	<b>88.13</b> ± 8.84	<b>19.75</b> ±16.47	<b>36.89</b> ±21.11	<b>94.03</b> ± 5.42	<b>0.00</b> ± 0.00	<b>0.00</b> ± 0.00	<b>99.99</b> ± 0.01	<b>0.44</b> ± 0.61	<b>2.40</b> ± 2.43	<b>99.88</b> ± 0.12	<b>13.98</b> ±17.18	<b>23.47</b> ±26.73	<b>95.51</b> ± 5.65
ATS [20]	66.27±26.96	80.76±19.56	76.35±13.17	48.90±21.86	67.99±17.83	86.04±10.09	<u>0.81</u> ± 2.76	<u>2.25</u> ± 6.07	<u>99.84</u> ± 0.38	<u>13.07</u> ± 8.09	<u>20.02</u> ±10.52	95.59± 3.51	32.26±30.50	<u>42.76</u> ±37.59	89.45±10.47
VIM [42]	45.37±28.34	66.79±29.74	85.53±11.91	32.42±21.74	56.64±28.32	<u>93.57</u> ± 5.45	15.92±24.39	31.44±35.47	<u>97.12</u> ± 4.48	<u>14.07</u> ±17.73	<u>23.87</u> ±24.75	<u>96.35</u> ± 5.52	<u>26.94</u> ±14.79	44.69±20.33	<u>93.14</u> ± 5.30
RMDS [34]	45.27±24.47	<u>64.29</u> ±18.18	82.96±15.25	42.53±28.08	60.64±24.79	84.61±15.03	24.66±35.25	33.01±36.45	89.60±17.48	27.07±31.30	36.65±32.67	88.71±17.54	34.88±10.52	48.65±16.09	86.47± 3.19
MDS [22]	48.23±28.66	68.30±27.44	82.33±16.47	36.92±31.46	58.31±29.59	87.75±16.03	20.35±31.57	35.52±37.63	92.19±16.18	23.74±33.01	32.54±35.37	89.65±17.93	32.31±12.80	48.67±17.43	87.98± 4.18
KNN [39]	<u>43.75</u> ±18.84	65.54±18.16	84.32±11.10	33.64±17.93	57.85±21.02	<u>92.47</u> ± 4.01	18.93±20.73	38.97±30.67	<u>96.32</u> ± 4.96	<u>20.26</u> ±17.05	35.14±22.69	<u>93.96</u> ± 6.82	29.14±11.79	49.37±14.65	<u>91.77</u> ± 5.21
ODIN [23]	47.70±10.58	66.19±13.16	83.84± 3.85	<u>31.44</u> ±17.51	<u>51.61</u> ±22.26	91.14± 5.25	27.29±29.87	40.63±32.22	86.73±21.94	29.01±26.07	39.51±29.16	84.45±18.82	33.86± 9.38	49.49±12.40	86.54± 3.31
GEN [25]	57.27±14.18	77.53±12.38	81.66± 6.05	46.29±27.56	68.15±25.93	87.36± 9.29	48.21±36.55	63.07±37.99	81.57±22.06	48.92±30.05	64.82±27.21	78.94±20.11	50.17± 4.86	68.39± 6.44	82.38± 3.55
DICE [37]	60.00±16.80	77.82±13.28	77.87± 9.36	47.64±27.05	69.66±24.90	85.13±10.76	51.73±41.07	65.41±37.60	70.42±32.97	48.95±32.60	62.30±27.89	70.41±28.31	52.08± 5.55	68.80± 6.73	75.96± 7.05
ReAct [38]	60.46±16.93	80.07±14.68	81.04± 6.51	47.69±25.93	71.59±26.45	87.41± 8.52	48.98±39.01	64.46±38.05	<u>82.04</u> ±20.73	49.52±31.56	66.02±29.53	<u>81.27</u> ±19.19	51.66± 5.92	70.54± 7.05	<u>82.94</u> ± 3.01
EBO [24]	59.77±14.43	78.72±11.67	77.59± 7.27	48.57±27.13	69.76±24.72	84.65± 9.99	55.54±40.76	66.70±37.37	69.26±30.90	53.80±30.44	68.90±25.29	69.28±26.14	54.42± 4.63	71.02± 5.29	75.19± 7.43
SHE [46]	60.57±16.33	80.02±11.15	73.77±13.33	51.46±27.72	72.99±22.69	81.28±17.64	49.99±39.29	67.52±34.44	69.09±36.60	49.78±32.98	63.97±27.75	67.88±31.08	52.95± 5.13	71.13± 7.00	73.00± 6.07
OpenMax [1]	44.06±11.86	80.23±14.83	<u>86.50</u> ± 4.15	36.38±22.15	80.28±13.43	90.45± 5.56	<u>20.10</u> ±30.63	60.63±40.17	<u>95.18</u> ± 6.35	<u>27.29</u> ±23.36	64.33±31.81	<u>93.17</u> ± 5.32	31.96±10.46	71.37±10.37	<u>91.33</u> ± 3.76
SCALE [44]	60.90±16.80	80.74±14.60	79.46± 5.92	46.63±28.00	70.20±28.34	86.74± 9.87	52.66±39.18	65.37±38.95	76.83±25.80	52.99±31.89	69.43±27.39	74.47±22.94	53.29± 5.85	71.44± 6.55	79.38± 5.32
MLS [15]	59.07±14.49	79.12±11.78	78.32± 7.10	48.56±26.94	71.14±24.65	85.14± 9.66	55.00±40.41	66.89±36.94	69.44±31.02	54.18±30.27	69.32±24.98	69.14±26.45	54.20± 4.33	71.62± 5.30	75.51± 7.71
KLM [15]	61.36± 9.13	80.25± 8.45	76.51± 7.33	51.45±21.07	72.20±20.86	84.90± 8.74	48.52±32.51	64.91±32.16	<u>83.59</u> ±15.78	54.58±21.71	70.51±22.12	<u>78.68</u> ±15.51	53.98± 5.51	71.97± 6.34	80.92± 3.98
ASH-s [6]	62.47±17.03	83.09±14.80	78.92± 6.08	49.56±27.30	74.63±28.06	<u>85.70</u> ±10.10	55.66±38.40	72.99±33.88	75.31±25.83	56.93±28.24	76.21±26.17	73.08±21.99	56.15± 5.30	76.73± 4.44	78.25± 5.52
GradNorm [17]	73.72±13.76	88.56± 8.46	69.71±12.64	69.32±18.42	84.27±15.39	73.69±12.54	56.18±42.18	65.91±40.84	63.75±38.58	60.16±31.81	69.87±29.69	61.34±32.68	64.84± 8.08	77.15±10.96	67.12± 5.62
TempScale [8]	66.08± 9.07	84.48± 9.03	<u>80.44</u> ± 4.90	56.63±21.82	77.54±20.43	<u>86.50</u> ± 7.35	54.44±33.51	71.28±32.94	<u>81.24</u> ±20.07	58.52±22.72	75.53±23.07	<u>78.08</u> ±17.67	58.92± 5.06	77.21± 5.51	<u>81.57</u> ± 3.55
MSP [13]	68.81± 7.46	87.78± 6.59	79.90± 4.96	60.65±17.82	82.50±15.60	<u>85.90</u> ± 6.86	57.20±31.21	75.84±29.75	<u>81.02</u> ±19.61	62.36±20.67	79.69±20.80	<u>77.65</u> ±17.13	62.26± 4.87	81.45± 5.02	<u>81.12</u> ± 3.48
ASH-b[6]	86.50±12.01	95.60± 6.19	64.45±12.66	80.47±22.16	92.57±13.26	66.32±17.46	82.30±31.94	86.63±31.73	56.08±24.90	81.82±25.97	88.67±23.04	53.50±20.24	82.77± 2.60	90.87± 4.01	60.09± 6.25
RankFeat [36]	92.35± 6.82	97.69± 3.73	50.49± 9.07	89.03±13.13	95.33±10.32	53.74±14.28	92.55±16.73	96.19±11.37	35.92±22.19	93.83± 7.29	97.21± 5.52	36.05±11.24	91.94± 2.05	96.60± 1.06	44.05± 9.41

Table 6. Average OOD detection performance measured as FPR at 95% and 99% TPR, reported per method and OOD category (near, far, extreme, and synthetic). Results are shown as mean ± standard deviation over three random seeds, two architectures (ResNet18 and CCT), and four tasks (Almond, Wheat, Kernels, and Food-grade). The arrow (↓) indicates that lower values are better. Cells are color-coded from **blue** (high performance) to white (low performance). Additionally, **best** and **second-best** results in each column are highlighted in bold and underlined, respectively. All values are reported as percentages, and methods are sorted based on their average FPR99 score.



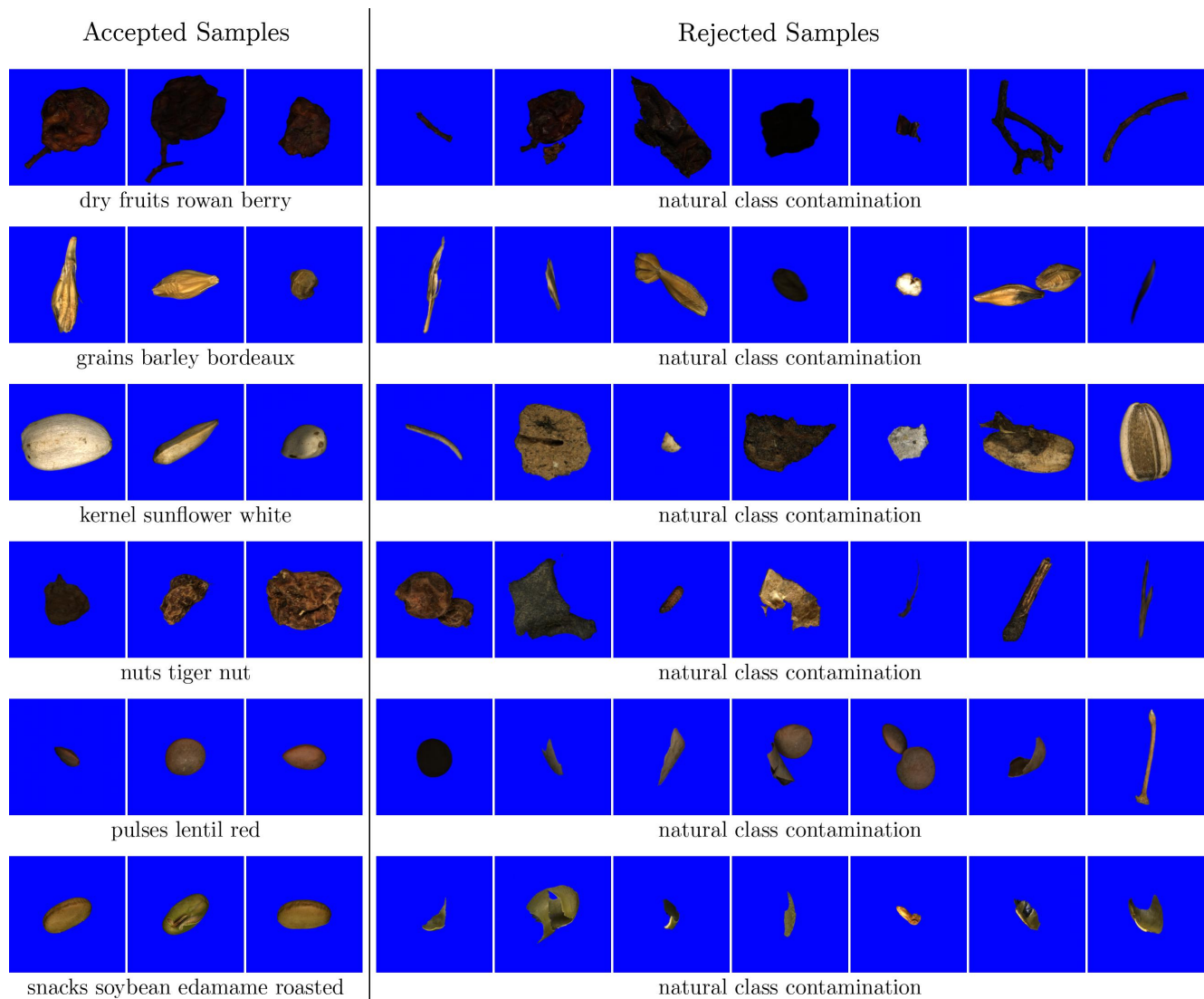


Figure 8. Illustrative examples of the dataset’s quality control process across six representative classes. Each row displays images from a single class, beginning with three accepted reference samples (left), followed by seven examples of rejected items (right) that were flagged during the cleaning process.

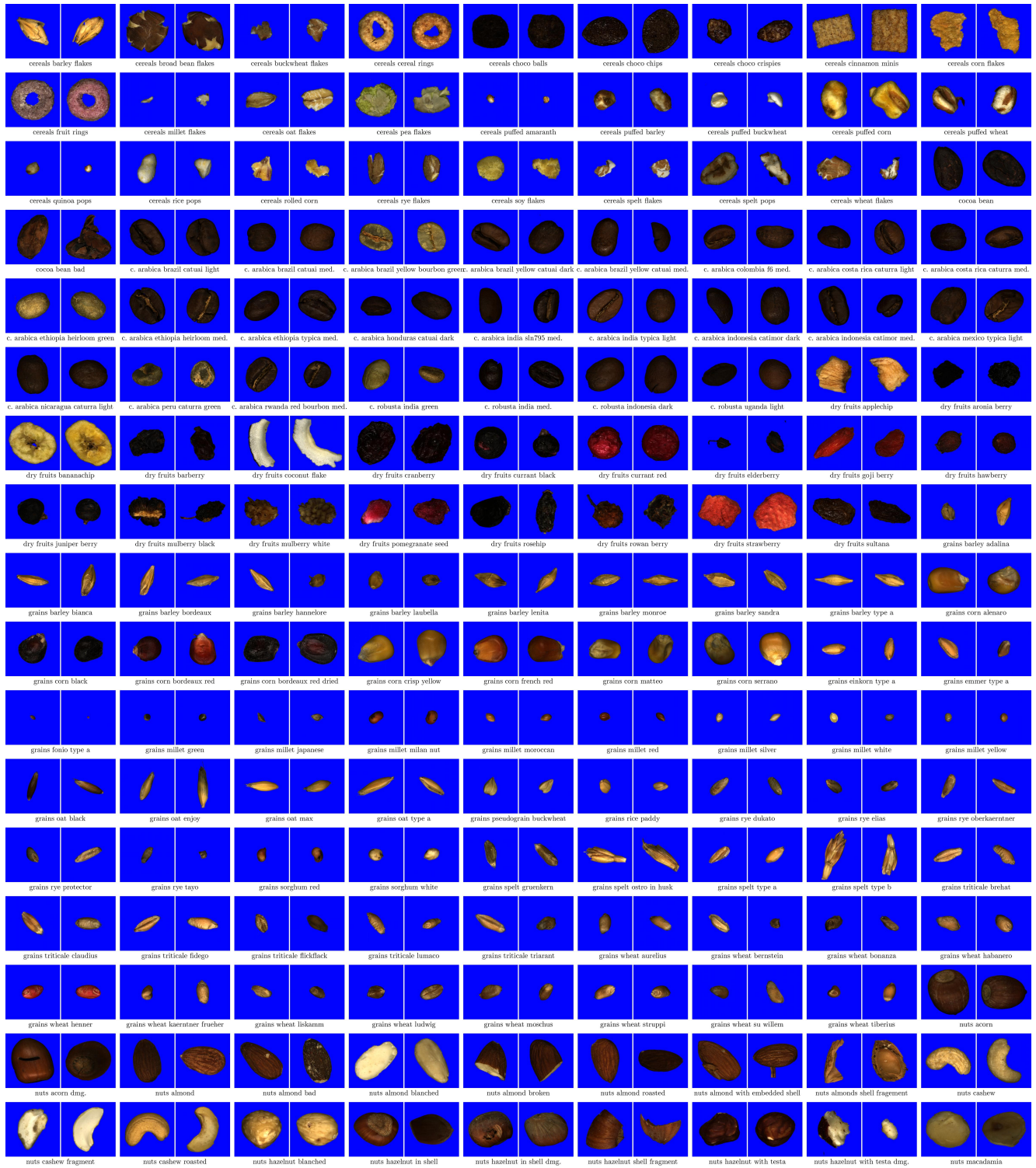


Figure 9. Samples for each class of the ICONIC-444 dataset (1/3), best viewed on screen.

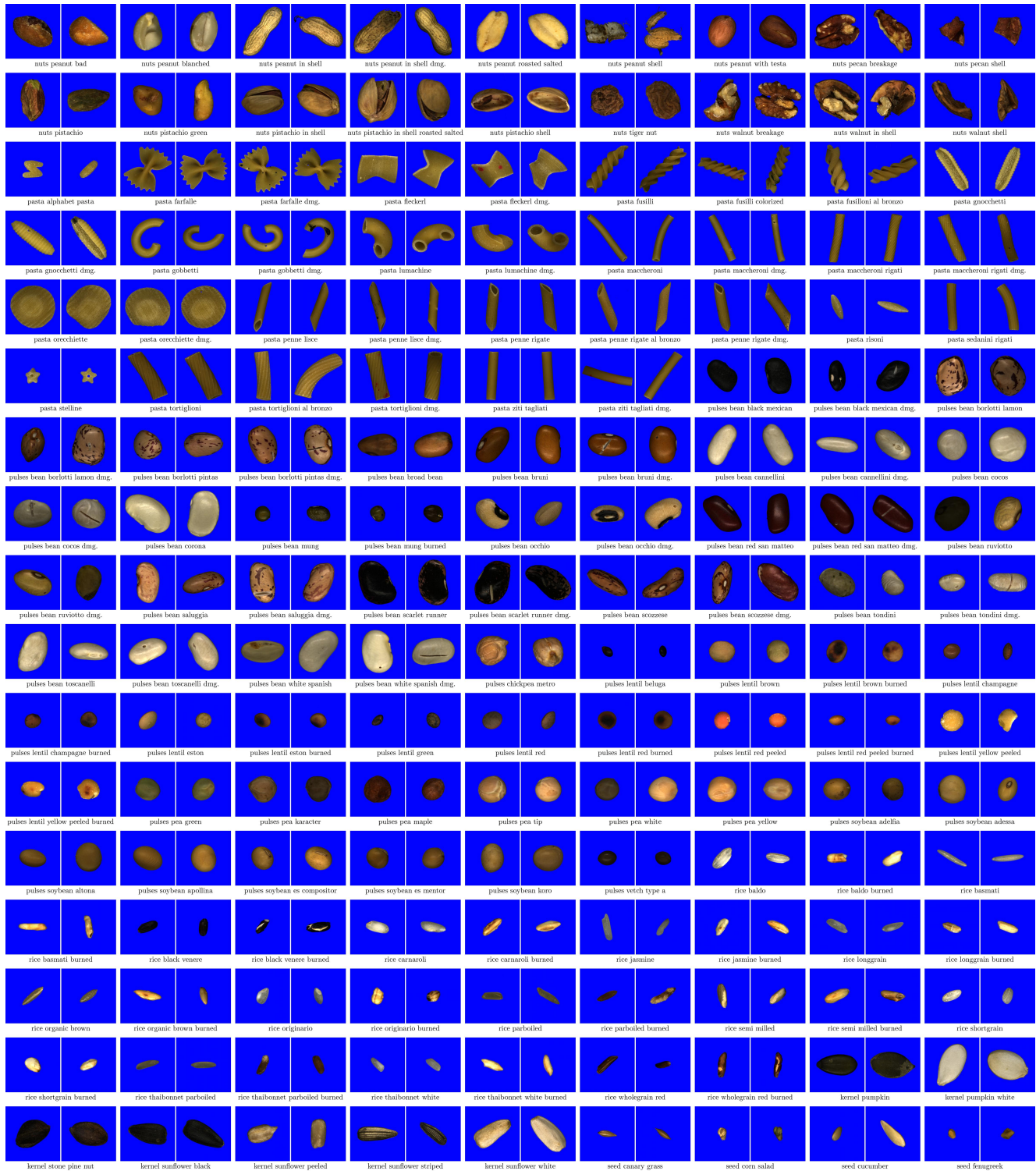


Figure 10. Samples for each class of the ICONIC-444 dataset (2/3), best viewed on screen.

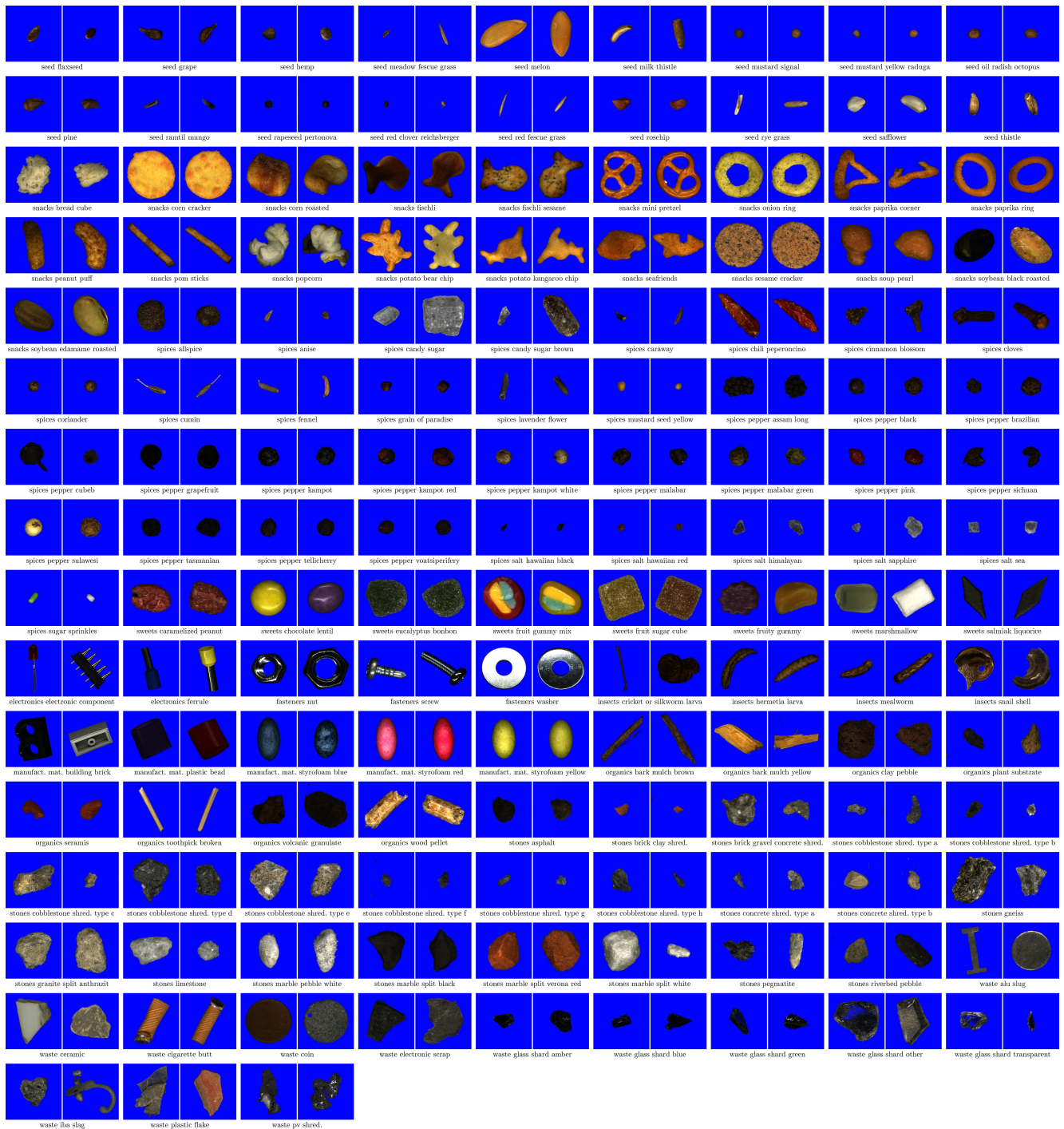
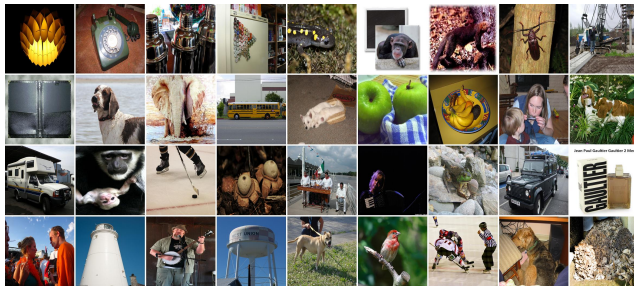
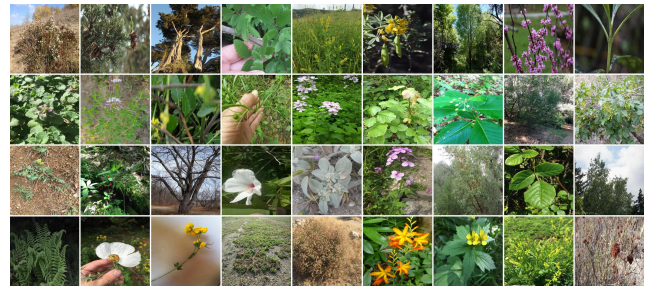


Figure 11. Samples for each class of the ICONIC-444 dataset (3/3), best viewed on screen.

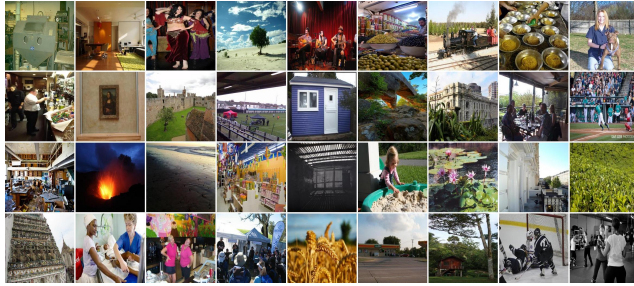




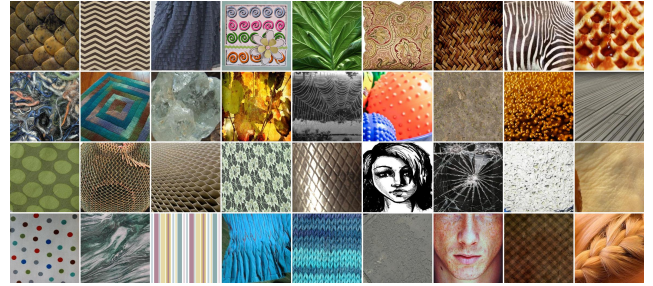
(a) ImageNet [5]



(b) iNaturalist [40]



(c) Places365 [48]



(d) Textures [4]

Figure 12. Randomly selected samples (36 per dataset) from the Extreme OOD set.

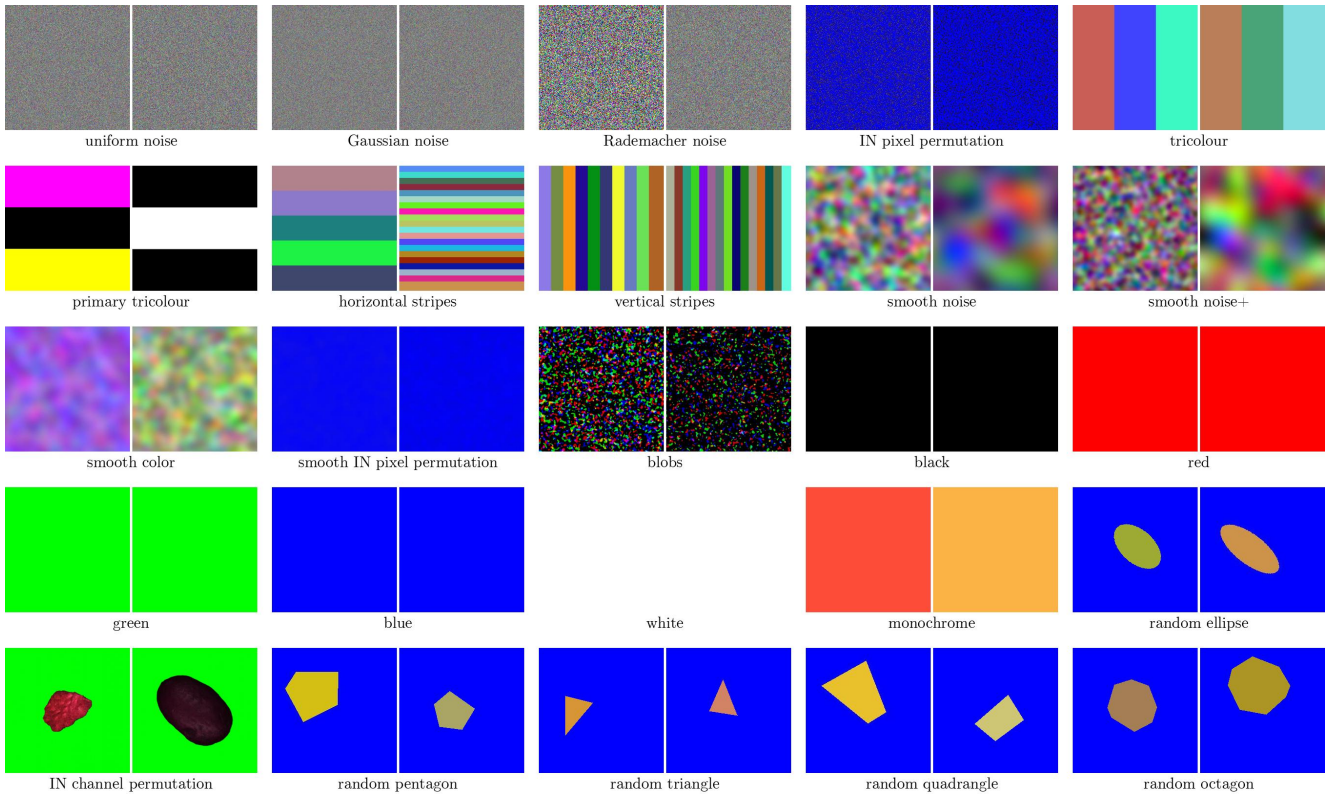


Figure 13. Examples of two randomly selected samples from each of the 25 synthetic OOD classes.

## References

- [1] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Proc. CVPR*, 2016. 4, 12
- [2] Julian Bitterwolf, Alexander Meinke, and Matthias Hein. Certifiably Adversarially Robust Detection of Out-of-Distribution Data. In *NeurIPS*, 2020. 5
- [3] Julian Bitterwolf, Maximilian Mueller, and Matthias Hein. In or Out? Fixing ImageNet Out-of-Distribution Detection Evaluation. In *Proc. ICLR Workshops*, 2023. 3, 5
- [4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. In *Proc. CVPR*, 2014. 5, 17
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 4, 17
- [6] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely Simple Activation Shaping for Out-of-Distribution Detection. In *Proc. ICLR*, 2023. 4, 12
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. ICLR*, 2021. 4
- [8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proc. ICML*, 2017. 4, 12
- [9] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the Big Data Paradigm with Compact Transformers. *arXiv preprint arXiv:2104.05704*, 2022. 3, 4
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016. 4
- [11] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *Proc. CVPR*, 2019. 5
- [12] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proc. ICLR*, 2019. 11
- [13] Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Proc. ICLR*, 2017. 4, 5, 8, 12
- [14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep Anomaly Detection with Outlier Exposure. In *Proc. ICLR*, 2019. 5
- [15] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling Out-of-Distribution Detection for Real-World Settings. In *Proc. ICML*, 2022. 4, 12
- [16] Rui Huang and Yixuan Li. MOS: Towards Scaling Out-of-distribution Detection for Large Semantic Space. In *Proc. CVPR*, 2021. 6
- [17] Rui Huang, Andrew Geng, and Yixuan Li. On the Importance of Gradients for Detecting Distributional Shifts in the Wild. In *NeurIPS*, 2021. 4, 12
- [18] Galadrielle Humblot-Renaux, Sergio Escalera, and Thomas B. Moeslund. A noisy elephant in the room: Is your out-of-distribution detector robust to label noise? In *Proc. CVPR*, 2024. 3, 11
- [19] Alex Krizhevsky and Geoffrey E. Hinton. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 4
- [20] Gerhard Krumpl, Henning Avenhaus, Horst Possegger, and Horst Bischof. ATS: Adaptive Temperature Scaling for Enhancing Out-of-Distribution Detection Methods. In *Proc. WACV*, 2024. 3, 4, 8, 12
- [21] Marc Lafon, Elias Ramzi, Clément Rambour, Nicolas Audebert, and Nicolas Thome. Gallop: Learning global and local prompts for vision-language models. In *Proc. ECCV*, 2024. 4, 9, 10
- [22] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *NeurIPS*, 2018. 4, 12
- [23] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *Proc. ICLR*, 2018. 4, 11, 12
- [24] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based Out-of-distribution Detection. In *NeurIPS*, 2020. 4, 12
- [25] Xixi Liu, Yaroslava Lochman, and Zach Christopher. Gen: Pushing the limits of softmax-based out-of-distribution detection. In *Proc. CVPR*, 2023. 4, 12
- [26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proc. CVPR*, 2022. 4
- [27] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proc. ICLR*, 2019. 4
- [28] Alexander Meinke, Julian Bitterwolf, and Matthias Hein. Provably Adversarially Robust Detection of Out-of-Distribution Data (Almost) for Free. In *NeurIPS*, 2022. 5
- [29] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyu Sun, Wei Li, and Yixuan Li. Delving into Out-of-Distribution Detection with Vision-Language Representations. In *NeurIPS*, 2022. 4, 9, 10
- [30] Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. GL-MCM: Global and Local Maximum Concept Matching for Zero-Shot Out-of-Distribution Detection. *IJCV*, 2023. 4, 9, 10
- [31] Atsuyuki Miyai, Jingkan Yang, Jingyang Zhang, Yifei Ming, Yueqian Lin, Qing Yu, Go Irie, Shafiq Joty, Yixuan Li, Hai Li, Ziwei Liu, Toshihiko Yamasaki, and Kiyoharu Aizawa. Generalized Out-of-Distribution Detection and Beyond in Vision Language Model Era: A Survey. *arXiv preprint arXiv:2407.21794*, 2024. 9, 10
- [32] Anh M Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *Proc. CVPR*, 2015. 5
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

- Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. ICML*, 2021. 9, 10
- [34] Jie Jessie Ren, Stanislav Fort, Jeremiah Zhe Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A Simple Fix to Mahalanobis Distance for Improving Near-OD Detection. *arXiv preprint arXiv:2106.09022*, 2021. 4, 12
- [35] Chandramouli Shama Sastry and Sageev Oore. Detecting Out-of-Distribution Examples with Gram Matrices. In *Proc. ICML*, 2020. 3, 4, 8, 12
- [36] Yue Song, Nicu Sebe, and Wei Wang. RankFeat: Rank-1 Feature Removal for Out-of-distribution Detection. In *NeurIPS*, 2022. 4, 12
- [37] Yiyu Sun and Yixuan Li. DICE: Leveraging Sparsification for Out-of-Distribution Detection. In *Proc. ECCV*, 2022. 4, 12
- [38] Yiyu Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution Detection With Rectified Activations. In *NeurIPS*, 2021. 4, 9, 12
- [39] Yiyu Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution Detection with Deep Nearest Neighbors. In *Proc. ICML*, 2022. 4, 8, 12
- [40] Grant Van Horn, Oisín Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The INaturalist Species Classification and Detection Dataset. In *Proc. CVPR*, 2018. 5, 17
- [41] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-Set Recognition: a Good Closed-Set Classifier is All You Need? In *Proc. ICLR*, 2022. 10
- [42] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-Of-Distribution with Virtual-logit Matching. In *Proc. CVPR*, 2022. 4, 8, 12
- [43] Ross Wightman. Pytorch image models. <https://github.com/huggingface/pytorch-image-models>, 2019. 4
- [44] Kai Xu, Rongyu Chen, Gianni Franchi, and Angela Yao. Scaling for Training Time and Post-hoc Out-of-distribution Detection Enhancement. In *Proc. ICLR*, 2024. 4, 8, 12
- [45] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, WENXUAN PENG, Haoqi Wang, Guangyao Chen, Bo Li, Yiyu Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. OpenOOD: Benchmarking Generalized Out-of-Distribution Detection. In *NeurIPS Datasets and Benchmarks Track*, 2022. 3
- [46] Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, Xiaoguang Liu, Shi Han, and Dongmei Zhang. Out-of-Distribution Detection based on In-Distribution Data Patterns Memorization with Modern Hopfield Energy. In *Proc. ICLR*, 2023. 4, 12
- [47] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyu Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. *J. of DMLR*, 2024. 3, 6, 7, 8, 9
- [48] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million Image Database for Scene Recognition. *TPAMI*, 2017. 5, 17